

# NHẬP MÔN DB2 Express-C

Một cuốn sách của cộng đồng dành cho cộng đồng

Lời nói đầu: DR. ARVIND KRISHNA



**KHÔNG CHỈ LÀ...DỮ LIỆU**

Xuất bản lần đầu (tháng 11 năm 2007)

Tài liệu này sử dụng cho IBM DB2 Express-C phiên bản 9.1 với hệ điều hành Linux, UNIX và Windows

Bản quyền thuộc công ty IBM

## Mục Lục

Thông cáo và Nhãn hiệu hàng hóa.....	7
Ai nên đọc cuốn sách này?.....	8
Cấu trúc của quyển sách này như thế nào?.....	8
Một quyển sách do cộng đồng làm và giành cho cộng đồng.....	8
Tác giả và những người đóng góp.....	8
Lời cảm ơn.....	9
Lời cảm ơn bản tiếng Việt.....	10
Lời nói đầu.....	12
<b>Phần I – Tổng quan và Cài đặt.....</b>	<b>13</b>
<b>Chương 1 - DB2 Express-C là gì?.....</b>	<b>13</b>
1.1 Tự do phát triển, triển khai và phân phối... không có giới hạn!.....	13
1.2 Hỗ trợ kỹ thuật.....	14
1.3 Các máy chủ DB2.....	14
1.4 Máy khách DB2.....	15
1.5 Tự do phát triển phần mềm ứng dụng.....	16
1.6 Phiên bản và ấn bản DB2 (version and edition).....	16
1.7 Chuyển sang ấn bản DB2 khác.....	17
1.8 Bảo trì DB2 Express-C.....	17
1.9 Một số phần mềm miễn phí liên quan.....	17
1.9.1 DB2 Developer Workbench (công cụ phát triển).....	17
1.9.2 DB2 9 Embedded Application Server (máy chủ nhúng).....	18
1.9.3 DB2 9 Net Search Extender (mở rộng tìm kiếm trên mạng).....	18
1.9.4 Bộ công cụ bắt đầu Rails cho DB2.....	18
1.9.5 Bộ công cụ bắt đầu Web 2.0 dành cho DB2.....	18
<b>Chương 2 – Các tính năng liên quan và các sản phẩm.....</b>	<b>19</b>
2.1 Các tính năng bao hàm trong đăng ký sử dụng DB2 Express-C.....	20
2.1.1 Các chương trình sửa lỗi.....	20
2.1.2 Tính sẵn sàng cao và khôi phục sau sự cố (HADR).....	20
2.1.3 Khả năng tái tạo dữ liệu (di trú dữ liệu).....	20
2.2 Các tính năng không có trong bản DB2 Express-C.....	21
2.2.1 Phân vùng cơ sở dữ liệu.....	21
2.2.2 Bộ tập trung kết nối.....	21
2.2.3 Bộ mở rộng không gian địa lý.....	22
2.2.4 Bộ kiểm soát truy vấn.....	22
2.3 Các sản phẩm có trả phí liên quan đến DB2.....	22
2.3.1 DB2 Connect.....	22
2.3.2 WebSphere Federation Server.....	22
2.3.3 WebSphere Replication Server.....	23
<b>Chương 3 - Cài đặt DB2.....</b>	<b>24</b>
3.1 Các điều kiện trước khi cài đặt.....	24
3.2 Quyền cài đặt hệ điều hành.....	24
3.3 Cài đặt theo hướng dẫn từng bước.....	25
3.4 Cài đặt hàng loạt (Silent Install).....	28
Bài tập nhanh #1: Cài đặt DB2 Express-C và tạo ra cơ sở dữ liệu mẫu.....	30
<b>Chương 4 - Các công cụ DB2.....</b>	<b>32</b>
4.1 Trung tâm điều khiển (Control Center).....	33
4.2 Nhập lệnh trực tiếp.....	36
4.3. Trình trợ giúp SQL.....	38
4.4 Trình bày nút Show SQL.....	40

Bài tập nhanh #2 - Tạo một CSDL mới với Trung tâm điều khiển .....	40
4.5 Tạo kịch bản.....	42
4.5.1 Các kịch bản SQL .....	42
4.5.2 Các kịch bản với hệ điều hành .....	44
Bài tập nhanh #3 – Tạo một kịch bản cài đặt cho cơ sở dữ liệu EXPRESS .....	45
4.6 Trung tâm tác vụ (Task Center) .....	48
4.6.1 Cơ sở dữ liệu Tools Catalog .....	48
4.7 Nhật ký.....	49
4.8 Giám sát tình trạng (health) .....	51
4.8.1 Trung tâm tình trạng .....	51
<b>Chương 5 - Môi trường của DB2.....</b>	<b>54</b>
5.1 Cấu hình DB2 .....	62
5.1.1 Các biến môi trường .....	63
5.1.2 Tập cấu hình quản trị cơ sở dữ liệu (dbm cfg).....	63
5.1.3 Tập cấu hình cơ sở dữ liệu (db cfg) .....	65
5.1.4 Đăng ký lý lịch DB2 (profile).....	66
5.2 Máy chủ quản trị DB2.....	67
<b>Phần II – Học DB2: Quản trị Cơ sở dữ liệu .....</b>	<b>68</b>
<b>Chương 6 - Kiến trúc của DB2 .....</b>	<b>68</b>
6.1 Mô hình xử lý của DB2 .....	68
6.2 Mô hình bộ nhớ của DB2.....	69
6.3 Mô hình lưu trữ của DB2.....	70
6.3.1 Trang và phạm vi .....	70
6.3.2 Vùng nhớ đệm.....	70
6.3.3 Các không gian bảng.....	73
<b>Chương 7 – Kết nối máy khách DB2.....</b>	<b>77</b>
7.1 Trợ giúp cấu hình.....	77
7.1.1 Những cài đặt bắt buộc trên máy chủ .....	77
7.1.2 Những cài đặt cần thiết trên máy khách.....	79
7.1.3 Tạo lý lịch cho máy khách và máy chủ.....	82
Bài tập nhanh #4 – Sử dụng Trợ giúp thiết lập cấu hình .....	86
<b>Chương 8 - Làm việc với các đối tượng dữ liệu .....</b>	<b>88</b>
8.1 Sơ đồ .....	88
8.2 Bảng .....	88
8.2.1 Kiểu dữ liệu.....	89
8.2.2 Cột khoá chính (Identity columns) .....	91
8.2.3 Các đối tượng tuần tự (Sequence object).....	92
8.2.4. Các bảng danh sách hệ thống.....	92
8.2.5. Khai báo các bảng tạm thời.....	93
Bài tập nhanh #5 - Khởi tạo một bảng mới.....	94
8.3 Các khung nhìn (Views) .....	96
8.4 Các chỉ mục (Index).....	96
8.4.1 Trợ giúp thiết kế (Design Advisor).....	97
8.5 Tính toàn vẹn tham chiếu.....	98
<b>Chương 9 – Công cụ di chuyển dữ liệu .....</b>	<b>100</b>
9.1 Công cụ EXPORT (xuất) .....	101
9.2 Công cụ IMPORT (nhập).....	102
9.3. LOAD ( nạp) .....	102
9.4 Công cụ db2move .....	104
9.5 Công cụ db2look .....	104



Bài tập nhanh #6 - Trích xuất DDL từ cơ sở dữ liệu EXPRESS .....	107
<b>Chương 10 – Bảo mật cơ sở dữ liệu.....</b>	<b>110</b>
10.1 Định danh.....	111
10.2 Quyền hạn .....	112
10.3 Quyền quản trị cơ sở dữ liệu DBADM.....	114
10.4 Nhóm PUBLIC .....	115
10.5 Lệnh GRANT và REVOKE.....	115
10.6 Kiểm tra việc cấp quyền và quyền.....	115
10.7 Xem xét quyền trên nhóm.....	117
Bài tập nhanh # 7: Cho phép và hủy bỏ (phân quyền) quyền người sử dụng.....	118
<b>Chương 11 – Sao lưu và khôi phục.....</b>	<b>121</b>
11.1 Nhật ký cơ sở dữ liệu .....	121
11.2 Các kiểu nhật ký.....	122
11.3 Các kiểu ghi nhật ký.....	122
11.3.1 Ghi tuần tự quay vòng.....	122
11.3.2 Nhật ký lưu trữ.....	123
11.4 Nhật ký cơ sở dữ liệu từ Trung tâm điều khiển.....	123
11.5 Các tham số của nhật ký .....	124
11.6 Sao lưu cơ sở dữ liệu.....	125
Bài tập nhanh #8 - Lập lịch sao lưu .....	127
11.7 Khôi phục cơ sở dữ liệu .....	129
11.7.1 Các kiểu khôi phục dữ liệu .....	129
11.7.2 Khôi phục cơ sở dữ liệu .....	129
11.8 Các hoạt động khác với BACKUP và RESTORE.....	130
<b>Chương 12 - Nhiệm vụ bảo trì .....</b>	<b>131</b>
12.1 REORG, RUNSTATS, REBIND .....	131
12.1.1 Lệnh REORG.....	131
12.1.2 Lệnh RUNSTATS.....	132
12.1.3 BIND/REBIND.....	132
12.1.4 Các công việc duy trì từ Trung tâm điều khiển.....	133
12.2 Các lựa chọn bảo trì .....	135
Bài tập nhanh #9 – Cấu hình việc bảo trì tự động .....	137
<b>Chương 13 – Truy cập đồng thời và khóa .....</b>	<b>139</b>
13.1 Các giao dịch.....	139
13.2 Truy cập đồng thời.....	140
13.3 Các vấn đề khi không kiểm tra truy cập đồng thời .....	141
13.3.1 Mất dữ liệu cập nhật .....	141
13.3.2 Đọc với không cam kết .....	142
13.3.3 Đọc không lặp lại .....	142
13.3.4 Đọc thừa.....	143
13.4 Các mức cô lập.....	144
13.4.1 Đọc với không cam kết .....	144
13.4.2 Ổn định con trỏ .....	144
13.4.3 Đọc ổn định.....	145
13.4.4 Đọc lặp lại .....	145
13.4.5 So sánh các mức độ cô lập dữ liệu.....	145
13.4.6 Đặt mức cô lập.....	146
13.5 Khóa leo thang .....	147
13.6 Kiểm soát khóa .....	148
13.7 Chờ khóa .....	148

13.8 Vấn đề khóa phụ thuộc và phát hiện.....	149
13.9 Truy cập đồng thời và thực tiễn khóa tốt nhất .....	150
<b>PHẦN 3: HỌC DB2 – PHÁT TRIỂN ỨNG DỤNG .....</b>	<b>152</b>
<b>Chương 14 – Các thủ tục SQL PL.....</b>	<b>153</b>
14.1 DB2 Developer Workbench.....	154
14.1.2 Tạo một thủ tục lưu trong DWB .....	154
14.2 Một số khái niệm về thủ tục SQL PL .....	158
14.2.1 Cấu trúc của thủ tục .....	158
14.2.2 Những thuộc tính tùy chọn của thủ tục .....	158
14.2.3 Các loại tham số.....	159
14.2.4 Chú thích trong thủ tục SQL PL .....	159
14.2.5 Câu lệnh ghép .....	160
14.2.6 Khai báo biến.....	160
14.2.7 Câu lệnh gán .....	160
14.3 Các con trỏ (cursor) .....	161
14.4 Điều khiển luồng.....	161
14.5 Gọi thủ tục.....	162
14.6 Lỗi và xử lý điều kiện .....	163
14.7 SQL động.....	165
<b>Chương 15 – Hàm nội tuyến, hàm người dùng định nghĩa, các hàm xử lý các sự kiện bẫy (Inline SQL PL, UDF, Triggers).....</b>	<b>166</b>
15.1 Hàm nội tuyến SQL .....	166
15.2 Bẫy sự kiện (Triggers) .....	167
15.2.1 Các loại bẫy sự kiện.....	167
Bài tập nhanh #10 - Khởi tạo một bẫy bằng Trung tâm điều khiển.....	170
15.3 Hàm người dùng định nghĩa (UDF).....	172
15.3.1 Hàm vô hướng .....	172
15.3.2 Hàm bảng.....	173
Bài tập nhanh #11 - Khởi tạo UDF sử dụng DB2 Developer Wordbench .....	175
<b>Chương 16 – SQL/XML và XQuery .....</b>	<b>176</b>
16.1 Sử dụng XML với các cơ sở dữ liệu .....	176
16.2 Cơ sở dữ liệu XML.....	177
16.2.1 cơ sở dữ liệu hỗ trợ XML .....	177
16.2.2 Cơ sở dữ liệu thuần XML .....	178
16.3 XML trong DB2.....	178
16.3.1 Ưu điểm của công nghệ pureXML trong DB2 9 .....	179
16.3.2 Khái niệm cơ bản về XPath .....	181
16.3.3 Định nghĩa XQuery.....	184
16.3.4 Chèn các tài liệu XML.....	185
16.3.5 Truy vấn dữ liệu XML.....	188
16.3.6 Nối (join) với SQL/XML.....	193
16.3.7 Nối (Join) với XQuery .....	194
16.3.8 Các lệnh cập nhật và xóa .....	194
16.3.9 Chỉ mục XML.....	195
Bài tập nhanh #12 – SQL/XML và XQuery .....	197
<b>Chương 17 – Phát triển với Java, PHP và Ruby.....</b>	<b>198</b>
17.1 Phát triển ứng dụng bằng Java.....	198
17.1.1 Trình điều khiển JDBC kiểu 2 (type 2).....	198
17.1.2 Trình điều khiển JDBC kiểu 4 .....	199
17.2 Phát triển ứng dụng bằng PHP.....	200

17.2.1	Lựa chọn kết nối DB2 cho PHP.....	200
17.2.2	Ứng dụng công nghệ Zend Core dành cho IBM.....	201
17.3	Phát triển ứng dụng Ruby trên nền Rails.....	203
17.3.1	Bộ công cụ phát triển DB2 trên nền Rails.....	203
A.1	Thêm thông tin về mã các lỗi.....	204
A.2	SQLCODE and SQLSTATE.....	205
A.3	Nhật ký khai báo quản trị DB2.....	205
A.4	db2diag.log.....	206
A.5	Theo vết CLI.....	206
A.6	Khắc phục và sửa lỗi trong DB2.....	206
	Các trang web.....	207
	Sách.....	208

## VỀ CUỐN SÁCH NÀY

### Thông cáo và Nhãn hiệu hàng hóa

Copyright IBM Corporation 2007  
All right Reserved.  
IBM Canada  
8200 Warden Avenue  
Markham, ON  
L6G 1C7  
Canada

Không được sao chép hoặc sử dụng dưới bất kỳ dạng thức nào hoặc dịch sang ngôn ngữ khác toàn bộ hoặc bất cứ phần nào của tài liệu này mà không được sự cho phép trước của người giữ bản quyền ở trên.

IBM không đảm bảo hay đại diện cho những nội dung được nêu ra ở đây và không chịu trách nhiệm về tính thương mại hoặc tính hợp lý cho bất kỳ mục đích nào. IBM cho rằng mình không có trách nhiệm với bất kỳ lỗi nào trong tài liệu này. Thông tin trong tài liệu này có thể thay đổi mà không cần báo trước. IBM giành quyền được thay đổi và không bắt buộc phải thông báo cho bất cứ ai về sự thay đổi này. IBM không cam kết những thông tin trong tài liệu này là mới nhất.

Các thông tin trong tài liệu này liên quan đến sản phẩm không phải của IBM đã có được qua các nhà cung cấp sản phẩm đó. IBM không kiểm tra các sản phẩm đó và không khẳng định về tính chính xác khi thực hiện, tính tương thích hoặc bất kỳ thiếu sót nào liên quan đến sản phẩm không phải của IBM. Các câu hỏi về khả năng của các sản phẩm không phải của IBM cần được gửi cho các nhà cung cấp sản phẩm đó.

IBM, biểu tượng IBM, DB2, DB2 Connect, DB2 Universal Database, i5/OS, pureXML, WebSphere, và z/OS là nhãn hiệu hàng hóa đã đăng ký của tập đoàn IBM tại Mỹ, các nước khác, hoặc cả hai.

Java và tất cả các nhãn hiệu Java là nhãn hiệu hàng hóa đăng ký của công ty Sun Microsystem, Inc. tại Mỹ, các nước khác, hoặc cả hai.

Microsoft và Windows là nhãn hiệu hàng hóa của tập đoàn Microsoft tại Mỹ, các nước khác, hoặc cả hai.

Linux là nhãn hiệu đăng ký của Linus Torvalds tại Mỹ, các nước khác, hoặc cả hai.

Tên của công ty khác, sản phẩm, hoặc dịch vụ có thể là nhãn hiệu đăng ký của họ.

Trích dẫn các sản phẩm và dịch vụ của IBM trong tài liệu này không có nghĩa là IBM đưa chúng đến tất cả các quốc gia mà IBM có hoạt động.

### Ai nên đọc cuốn sách này?

Quyển sách này giành cho bất kỳ ai đang làm việc hoặc muốn làm việc với cơ sở dữ liệu, như quản trị hệ cơ sở dữ liệu (DBAs), người phát triển ứng dụng, nhà tư vấn, kiến trúc phần mềm, giám đốc sản phẩm, giáo viên và sinh viên.

### Cấu trúc của quyển sách này như thế nào?

Phần I, Tổng quan và Cài đặt, giải thích ấn bản DB2 Express-C là gì, giới thiệu họ sản phẩm DB2 và các chức năng, giúp đỡ cài đặt và tạo mới cơ sở dữ liệu, và khám phá các công cụ sẵn có của DB2.

Phần II, Học DB2: Quản trị cơ sở dữ liệu, được thiết kế để bạn làm quen với môi trường, kiến trúc, kết nối từ xa, các đối tượng dữ liệu, chuyển dữ liệu (nhập và xuất dữ liệu), an toàn, sao lưu và khôi phục, truy cập đồng thời và khóa của DB2, và các công việc bảo trì thông thường.

Phần III, Học DB2: Phát triển ứng dụng, bao gồm thủ tục, hàm do người dùng định nghĩa, các bất sự kiện, SQL/XML, XQuery, phát triển ứng dụng bằng Java, PHP và Ruby.

Phụ lục có các thông tin cần thiết về sửa lỗi.

Bài tập ở đây là bài tập nhanh, có ở hầu hết các chương, tất cả các tệp cần thiết để thực hiện bài tập này có thể tìm thấy ở tệp nén **expressc\_book\_quicklabs.zip** đi kèm theo sách này, hoặc sẽ tìm thấy ở trang web của IBM DB2 Express-C, địa chỉ [www.ibm.com/db2/express](http://www.ibm.com/db2/express)

Tài liệu của cuốn sách cũng được sử dụng làm bài giảng của chương trình “DB2 với Đại học” (DB2 on Campus). Cuốn sách này cũng giúp bạn chuẩn bị cho bài kiểm tra của chương trình “DB2 với Đại học”, chương trình đầy đủ của nó bài học với 16 giờ trên lớp. Bạn có thể xem thêm thông tin về chương trình này ở trang DB2 Express-C, địa chỉ [www.ibm.com/db2/express/students.html](http://www.ibm.com/db2/express/students.html).

### Một quyển sách do cộng đồng làm và giành cho cộng đồng

Đội ngũ làm DB2 Express-C đã có sáng kiến làm quyển sách này để cung cấp miễn phí cho cộng đồng. Viết và chỉnh lý một cuốn sách là việc làm khó khăn. Mục đích của chúng tôi là nội dung cuốn sách sẽ được duy trì và mở rộng bằng việc tham gia của cộng đồng DB2 Express-C. Nếu bạn muốn đóng góp ý kiến, đóng góp những nội dung mới để nâng cao nội dung hiện tại hoặc dịch cuốn sách này sang ngôn ngữ khác, làm ơn gửi kế hoạch thực hiện về địa chỉ [db2x@ca.ibm.com](mailto:db2x@ca.ibm.com) với tiêu đề “DB2 Express-C book changes.”

### Tác giả và những người đóng góp

Những người sau đây đã cung cấp nội dung và có các đóng góp có ý nghĩa cho cuốn sách này

Tên	Đơn vị	Chức danh	Tài liệu đóng góp	Ngày
Raul F. Chong	IBM	GĐ chương trình “DB2 với Đại học”	Bản thảo tất cả các chương	Tháng 5 – 2007
Ian Hakes	IBM	Cộng đồng DB2 Express-C	Xem lại và sửa chữa	Tháng 5, 6 – 2007
Rav Ahuja	IBM	GĐ sản phẩm DB2	Xem lại, cập nhật và trình bày cuốn sách	Tháng 7-10 - 2007

## Lời cảm ơn

Chúng tôi chân thành cảm ơn các cá nhân sau đây với sự trợ giúp

- Ted Wasserman, Clara Liu và Paul Yip ở phòng thí nghiệm IBM Toronto, người đã phát triển ứng dụng trong khuôn khổ cuốn sách này
- Don Chamberlin và Cindy Saracco với các bài báo của họ về XQuery đăng trên developerWorks, và Matthía Nicola với bài trình bày về pureXML.
- Kevin Czap và Grant Hutchinsonson với các tài liệu kỹ thuật về phát triển DB2
- Katherine Boyachok với thiết kế bìa
- Susan Visser giúp đỡ xuất bản cuốn sách này

## Lời cảm ơn bản tiếng Việt

Để có được bản dịch tiếng Việt này, rất nhiều các bạn sinh viên đã tham gia dịch và hiệu đính. Chúng ta tự hào rằng đây đúng là một quyển sách do cộng đồng và vì cộng đồng. Vì trình độ và thời gian hạn chế, dù có thể còn nhiều khiếm khuyết, nhưng rất mong các bạn đón nhận nó nhiệt tình và đóng góp ý kiến cho bản dịch hoàn thiện hơn. Hy vọng rằng các bạn sinh viên sẽ tiếp tục chuyển đến cộng đồng những ấn phẩm mới, với chất lượng chắc chắn tốt hơn.

Danh sách sinh viên tham gia dịch và hiệu đính:

	Họ và tên	Trường
1	Bùi Thanh Sơn	ĐH Bách Khoa Hà Nội
2	Châu Uyên Minh	Học viện Công nghệ Bưu chính Viễn thông TPHCM
3	Đặng Chiến Công	Coltech Hà Nội
4	Đỗ Bá Lâm	ĐH Bách khoa Hà Nội
5	Đỗ Tuấn Anh	
6	Đỗ Tuấn Anh	ĐH Khoa học Tự nhiên, Tp HCM
7	Hà Thúy Hằng	ĐH Khoa học Tự nhiên, Tp HCM
8	Hà Tuấn Trung	ĐH Bách Khoa Hà Nội
9	Hà Tuấn Trung	
10	Hoàng Minh Hiền	Trường Đại Học Công Nghệ, VNU Hà Nội
11	Huỳnh Hữu Hiệp	Học Viện công nghệ BCVT
12	Huỳnh Minh Tân	ĐH Khoa Học Tu Nhien, Tp HCM
13	Lê Anh Đức	ĐH Bách Khoa Hà Nội
14	Lê Bá Trọng	ĐH Khoa học Tự nhiên, Tp HCM
15	Lê Huỳnh Công Thảo	ĐH Công nghệ Thông tin, Tp HCM
16	Lê Nguyễn Thúy An	ĐH Công nghệ Thông tin, Tp HCM
17	Lê Thành Huy	ĐH Cần Thơ
18	Ngô Phi Thành	Đại học Duy Tân, Đà Nẵng
19	Nguyễn Anh Khoa	ĐH Khoa học Tự nhiên, Tp HCM
20	Nguyễn Chiến Thắng	ĐHBK Đà Nẵng
21	Nguyễn Hải Phong	ĐH Bách Khoa Hà Nội
22	Nguyễn Hoàng Minh Quốc	
23	Nguyễn Hùng Thông	ĐH Sư phạm Kỹ thuật, Tp. HCM
24	Nguyễn Hữu Thọ	ĐH Khoa học Tự nhiên, Tp HCM
25	Nguyễn Lương Sơn	Đại học Duy Tân, Đà Nẵng
26	Nguyễn Minh Châu	ĐH Khoa học Tự nhiên, Tp HCM
27	Nguyễn Minh Tuấn	ĐH Bách khoa Hà Nội
28	Nguyễn Minh Tuấn	
29	Nguyễn Ngọc Duy	
30	Nguyễn Ngọc Trí	ĐH Sư phạm Kỹ thuật, Tp. HCM
31	Nguyễn Thành Trung	ĐH Bách khoa Hà Nội
32	Nguyễn Thế Phúc	Đại học Duy Tân, Đà Nẵng
33	Nguyễn Thị Diệu Hằng	ĐH Khoa học Huế
34	Nguyễn Thị Hồng Hà	ĐH Bách Khoa Hà Nội
35	Nguyễn Thị Kim Phụng	ĐH Công nghệ Thông tin, Tp HCM
36	Vũ Thị Lan Chi	ĐH Công nghệ Thông tin, Tp HCM
37	Nguyễn Trang	
38	Nguyễn Trung Hiếu	ĐH Khoa học Tự nhiên, Tp HCM
39	Nguyễn Văn Nam	ĐH Khoa học Huế
40	Nguyễn Văn Nam	ĐH Khoa học Huế

41	Nguyễn Xuân Trường	ĐH Khoa học Tự nhiên, Tp HCM
42	Phạm Ngọc Huy	Đại học Công nghệ-Đại học quốc gia Hà nội
43	Phạm Thị Minh Hiền	ĐH Bách khoa Đà Nẵng
44	Phạm Vũ Tuấn	
45	Quách Tiêu Thuận	Trung tâm Công nghệ Phần mềm - ĐH Cần Thơ
46	Thái Quang Hòa	ĐH Khoa học Huế
47	Thái Tuyền	Đại học Nông Lâm tp. HCM
48	Trần Anh Huy	Đại học Duy Tân, Đà Nẵng
49	Trần Thị Thúy Trinh	ĐH Bách Khoa Đà Nẵng
50	Trần Văn Tấn	ĐH Bách khoa Hà Nội
51	Trần Văn Tấn	
52	Trần Vương Trung	
53	Tuấn Việt	



## Lời nói đầu

Sáng tạo là sự đột phá của các tiên bộ công nghệ. Tại IBM, sáng tạo đã trở thành cội nguồn của sự phát triển cơ sở dữ liệu. Là người đi tiên phong trong kỹ thuật quản trị dữ liệu trong những năm 60 và 70, chúng tôi tiếp tục đưa ra những sáng tạo công nghệ về quản trị thông tin, biểu hiện bằng hàng nghìn đăng ký phát minh về quản trị dữ liệu của những nhà công nghệ của IBM. Kết quả là, một số các tổ chức lớn nhất hành tinh ngày nay đã tin tưởng vào các sản phẩm của IBM như DB2 để vận hành các giải pháp có nhu cầu cao nhất, quản trị những dữ liệu tối quan trọng của họ.

Tuy nhiên DB2 không chỉ dành cho các doanh nghiệp lớn. Với việc đưa ra bản DB2 Express-C, công nghệ nổi tiếng DB2 đã sẵn sàng cho các doanh nghiệp vừa và nhỏ - và không bắt buộc phải mất chi phí! Mặc dù còn có các sản phẩm miễn phí hay các phần mềm nguồn mở khác, DB2 Express-C trao cho bạn sức mạnh duy nhất vượt trên tất cả các lựa chọn cùng loại.

Có rất nhiều sáng tạo công nghệ hiện diện trong DB2 Express-C. Một số trong chúng nhằm vào khả năng tiên tiến mới, một số khác lại hướng tới giảm thiểu gánh nặng quản trị, một số khác nữa lại cải thiện hiệu năng, giảm chi phí hạ tầng. Chúng tôi sẽ không trình bày chúng ở đây, hy vọng các bạn sẽ bị cuốn hút khi đọc quyển sách này – chúng tôi chỉ xin mô tả một vấn đề hóc búa nhất.

DB2 Express-C được xây dựng trên nền công nghệ “Viper”, nó là cơ sở dữ liệu lai đầu tiên quản lý cả dữ liệu quan hệ và dữ liệu XML dưới dạng bản sinh. Điều này làm cho DB2 trở nên rất lý tưởng cho các xu hướng mới của ứng dụng SOA và Web 2.0 trong đó các luồng dữ liệu XML là rất nhiều. Không giống với hệ quản trị cơ sở dữ liệu của các hãng thương mại khác, DB2 Express-C không giới hạn dung lượng dữ liệu lưu trữ trong cơ sở dữ liệu hay số lượng cơ sở dữ liệu bạn có thể tạo ra trong hệ thống. Và tất nhiên, nếu bạn cần trợ giúp hỗ trợ từ IBM, bạn chỉ cần nhấn chuột vào nút Help.

Cuốn sách này dùng để giúp những người mới bắt đầu sử dụng DB2 Express-C, nó sẽ giúp bạn hiểu các khái niệm của DB2 và cho phép bạn phát triển kỹ năng quản trị cũng như phát triển ứng dụng DB2. Kỹ năng và sự hiểu biết nhận được từ cuốn sách này có liên quan rất nhiều đến các bản DB2 tiên tiến hơn trên Linux, UNIX và Windows.

Mặc dù DB2 Express-C không phải là sản phẩm mã nguồn mới, IBM chúng tôi vẫn rất tin tưởng vào sự hỗ trợ và cổ vũ từ cộng đồng. Tôi vui mừng được thấy cuốn sách này do các thành viên của cộng đồng DB2 Express-C phát triển và phát hành miễn phí cho cộng đồng. Tôi mong các bạn cập nhật, làm giàu thêm kiến thức của cuốn sách với hiểu biết, kinh nghiệm của các bạn, và giúp dịch cuốn sách này sang các ngôn ngữ khác, như vậy những người khác sẽ có lợi từ sự hiểu biết của bạn.

Arvind Krishna  
Phó chủ tịch, Cơ sở dữ liệu  
Ban Quản trị thông tin, Nhóm phần mềm IBM

# 1

## Phần I – Tổng quan và Cài đặt

### Chương 1 - DB2 Express-C là gì?

DB2 Express-C là một sản phẩm thuộc dòng IBM DB2 – phần mềm máy chủ dữ liệu quản lý cả dữ liệu quan hệ và dữ liệu XML. DB2 Express-C là ấn bản DB2 được dùng miễn phí, không hạn chế và dễ sử dụng. Chữ C trong DB2 Express-C là viết tắt của Community (Cộng đồng). Nghĩa là cộng đồng những người sử dụng DB2 Express-C kết hợp lại để hỗ trợ lẫn nhau cả trực tuyến và không trực tuyến. Cộng đồng DB2 Express-C bao gồm các cá nhân và các công ty thiết kế, phát triển, triển khai, hay sử dụng các giải pháp cơ sở dữ liệu như:

- Các nhà phát triển ứng dụng có nhu cầu về một phần mềm cơ sở dữ liệu chuẩn mở để xây dựng các ứng dụng độc lập, dạng khách-chủ, dạng nền web hoặc các ứng dụng lớn.
- ISVs – các nhà cung cấp phần mềm độc lập, các nhà cho thuê phần cứng, cơ sở hạ tầng hay những người cung cấp giải pháp, muốn đóng gói một máy chủ dữ liệu với đầy đủ tính năng như là một phần trong các giải pháp của họ.
- Người cố vấn, quản trị dữ liệu, và kiến trúc sư công nghệ thông tin, những người cần một máy chủ dữ liệu mạnh để tự học tập, phát triển kỹ năng, đánh giá hoặc thử nghiệm.
- Các công ty mới thành lập, vừa và nhỏ cần một máy chủ dữ liệu tin cậy cho công việc và ứng dụng của họ.
- Những người yêu thích cơ sở dữ liệu, say mê tìm hiểu công nghệ muốn có một máy chủ dữ liệu để sử dụng để xây dựng ứng dụng Web 2.0 hay các ứng dụng thế hệ sau.
- Sinh viên, giáo viên hay những học giả muốn có một máy chủ dữ liệu linh hoạt để giảng dạy, học tập, nghiên cứu.

DB2 Express-C có đầy đủ các tính năng cơ bản như các ấn bản DB2 thương mại trên nền Linux, UNIX và Windows. DB2 Express-C có thể chạy trên cả hệ thống 32-bit hoặc 64-bit của Linux hay Windows. Nó được tối ưu cho các hệ thống tối đa 2 bộ xử lý, bộ nhớ 4GB, và không yêu cầu bất kỳ một bộ nhớ chuyên dụng hay các cài đặt hệ thống nào khác. DB2 Express-C cũng bao gồm pureXML, pureXML là một công nghệ đặc trưng của DB2 dùng để lưu trữ và xử lý các văn bản XML bẩm sinh.

#### 1.1 Tự do phát triển, triển khai và phân phối...không có giới hạn!

Các tư tưởng cốt lõi của DB2 Express-C gồm:

- **Tự do phát triển:** Nếu bạn là người phát triển ứng dụng và cần một cơ sở dữ liệu cho ứng dụng của mình, bạn có thể sử dụng DB2 Express-C.

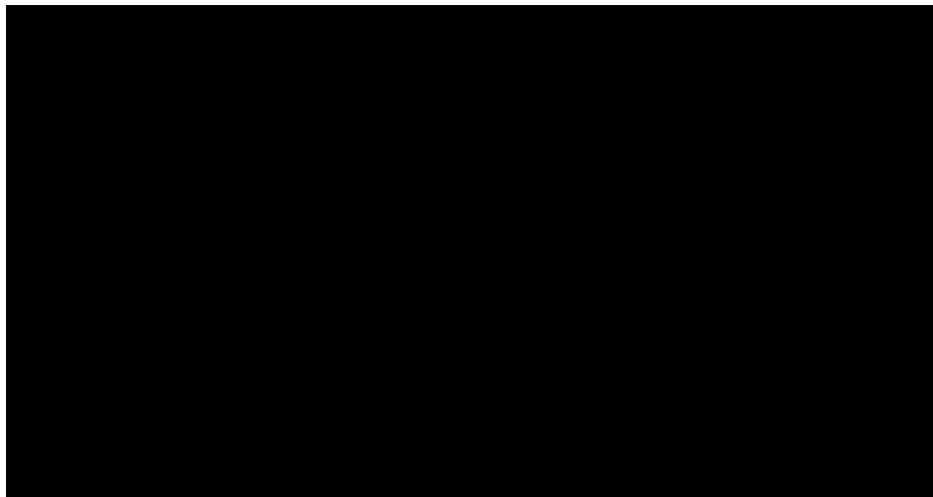
- **Tự do triển khai:** Nếu bạn đang làm việc trong lĩnh vực sản xuất và cần một cơ sở dữ liệu để lưu trữ các thông tin quan trọng, bạn có thể dùng DB2 Express-C.
- **Tự do phân phối:** Nếu bạn đang phát triển một ứng dụng hay một công cụ cần đóng gói với một máy chủ dữ liệu, bạn cũng có thể dùng DB2 Express-C. DB2 Express-C miễn phí ngay cả khi được đóng gói vào trong ứng dụng của bạn, được phân phối mỗi khi bạn bán ứng dụng của mình. Bạn chỉ cần đăng ký với IBM để tái phân phối DB2 Express-C. Tất nhiên việc đăng ký cũng hoàn toàn miễn phí.
- **Không giới hạn:** Trong khi các đối thủ cạnh tranh đưa ra các giới hạn về kích thước cơ sở dữ liệu, nhưng DB2 Express-C thì không đặt giới hạn nào cả. Cơ sở dữ liệu của bạn có thể được tiếp tục mở rộng mà không hề vi phạm điều khoản sử dụng. Cũng không có giới hạn nào về số lượng kết nối hay số người sử dụng trên mỗi máy chủ.

## 1.2 Hỗ trợ kỹ thuật

Nếu bạn có bất kỳ thắc mắc kỹ thuật nào về DB2 Express-C, bạn có thể đăng câu hỏi của bạn lên diễn đàn của DB2 Express-C. Diễn đàn này được điều hành bởi một đội DB2 Express-C, nhưng chính cộng đồng sẽ là những người giải đáp trên cơ sở tự nguyện. IBM cũng cho phép người dùng mua bản quyền theo năm với giá thấp (còn được gọi là Bản quyền 12 tháng hay Bản quyền thời hạn cố định FTL (Fixed Term License)). Việc mua này sẽ cho phép người dùng được hỗ trợ kỹ thuật 24x7 cũng như cập nhật phần mềm. Với khoản phí thấp hằng năm (\$2,995 trên một máy chủ một năm ở Mỹ - có thể thay đổi ở các nước khác) bạn sẽ được hưởng các quyền lợi về hỗ trợ và bảo trì phần mềm cho máy chủ DB2 Express-C, bạn cũng có thêm hai tính năng nữa là: tính sẵn sàng phục hồi sau khi gặp sự cố (HARD - High Availability Disaster Recovery) và di trú dữ liệu truy vấn (SQL replication - để tái tạo lại dữ liệu với các máy chủ DB2 khác).

## 1.3 Các máy chủ DB2

Tất cả các ấn bản máy chủ DB2 đều có chung các thành phần lõi; Các ấn bản này được đóng gói thuận tiện để người sử dụng có thể lựa chọn các tính năng cần thiết với giá cả hợp lý. Hình 1.1 thể hiện sự khác nhau giữa các ấn bản của DB2.



Hình 1.1 - Các máy chủ DB2

Như trên hình 1.1, ấn bản DB2 Express-C cũng giống như ấn bản DB2 Express chỉ thiếu đi một số thành phần. DB2 Express-C là hoàn toàn miễn phí. Cũng như đã nói ở trên, người dùng luôn được sẵn sàng hỗ trợ kỹ thuật từ diễn đàn, hoặc có thể nhận sự hỗ trợ chính thức 24x7 của IBM nếu trả phí 12 tháng.

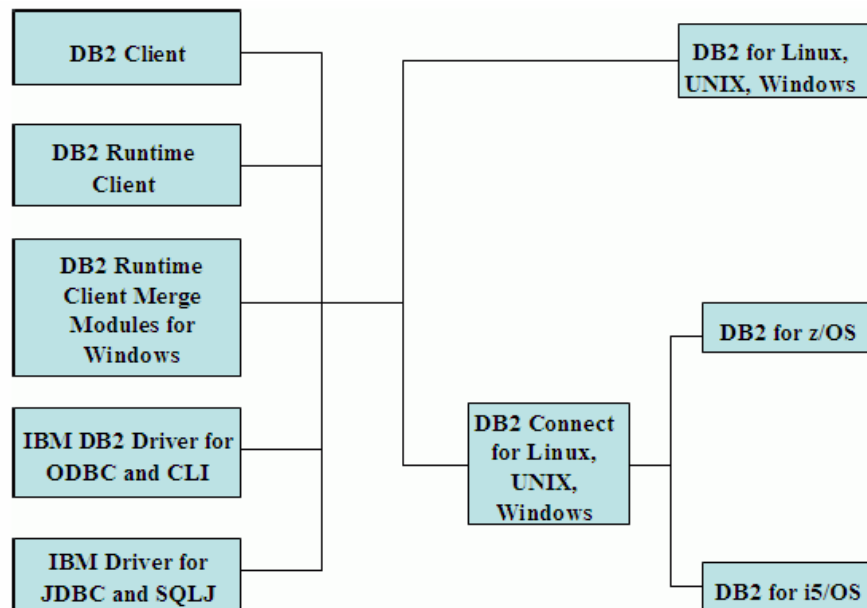
Hình 1.1 cũng cho thấy khả năng nâng cấp từ DB2 Express-C lên bất kì ấn bản máy chủ DB2 nào khác, vì tất cả các máy chủ DB2 đều có chung thành phần lõi. Điều này có nghĩa là tất cả các ứng dụng được phát triển trên một ấn bản vẫn sẽ hoạt động trên các ấn bản khác mà không cần phải chỉnh sửa. Và bất kì kỹ năng nào bạn học được từ một ấn bản đều áp dụng được cho các ấn bản khác.

### 1.4 Máy khách DB2

Một máy khách DB2 bao gồm các chức năng cần thiết để kết nối với máy chủ DB2. Tuy nhiên, không phải lúc nào cũng cần cài đặt một máy khách DB2. Ví dụ, một ứng dụng JDBC (Java DataBase Connectivity) kiểu 4 có thể nối thẳng tới máy chủ DB2, chỉ cần có trình điều khiển thích hợp. Máy khách DB2 gồm những loại sau:

- Máy khách DB2: đầy đủ nhất, bao gồm các công cụ đồ họa, các trình điều khiển.
- Máy khách DB2 runtime client: Có các chức năng cơ bản để kết nối, cũng bao gồm các trình điều khiển.
- Máy khách DB2 runtime kết hợp Mô-đun cho Windows: được sử dụng chủ yếu trên máy khách DB2 runtime như là một phần cài đặt cho các ứng dụng của Windows.

Hình 1.2 cho ta thấy các máy khách khác nhau và các trình điều khiển sẵn có.



**Hình 1.2 - Các máy khách DB2 và các trình điều khiển**

Trên phần bên trái của hình 1.2, tất cả các máy khách DB2 và trình điều khiển được thể hiện. Mặc dù tất cả các máy khách DB2 đều bao gồm trình điều khiển cần thiết, trong DB2 9 chúng tôi vẫn cung cấp những trình điều khiển riêng biệt. Các máy khách DB2 và các trình điều khiển đều miễn phí và có thể được tải về từ trang web DB2 Express-C. Các máy khách và các trình điều khiển có thể được sử dụng để kết nối đến máy chủ DB2 trên nền Linux, Unix, hoặc Windows. Để kết nối đến DB2 trên máy chủ z/OS® hoặc DB2 trên máy chủ i5/OS®, bạn cần phải thông qua máy chủ kết nối DB2 (DB2 Connect nằm ở giữa hình 2). Chúng ta sẽ thảo luận về phần mềm DB2 Connect trong Chương 2.

### 1.5 Tự do phát triển phần mềm ứng dụng

DB2 tạo ra một môi trường phát triển ứng dụng dựa trên các chuẩn và trong suốt đối với họ sản phẩm DB2. Việc sử dụng SQL chuẩn trong các dòng sản phẩm DB2 sẽ cung cấp một tập các giao diện lập trình ứng dụng (API) thông dụng cho việc truy nhập cơ sở dữ liệu.

Thêm vào đó, mỗi sản phẩm DB2 cung cấp các bộ tiền-dịch (pre-compiler) cho phép nhà phát triển nhúng các câu lệnh SQL tĩnh và động trong các chương trình ứng dụng khả chuyển. DB2 còn có một sẵn công cụ quản lý .NET có thể tích hợp với các công cụ Microsoft® Visual Studio.

Các ngôn ngữ và các chuẩn có thể được sử dụng với DB2 là:

- Ruby trên nền Rails
- C/C++(ODBC và SQL nhúng)
- JDBC và SQLJ
- COBOL
- Borland
- Python
- PHP
- Perl
- Ngôn ngữ .NET
- OLE-DB
- ADO
- Dịch vụ Web (Web Service)
- SQL
- Microsoft Office: Excel, Access, Word

### 1.6 Phiên bản và ấn bản DB2 (version and edition)

Nếu bạn mới làm quen với DB2, bạn có thể bị bối rối một chút về sự khác biệt giữa các phiên bản DB2 (version) và các ấn bản DB2 (edition).

Vài năm một lần, IBM công bố một phiên bản mới của DB2. Một phiên bản bao gồm các tính năng mới và đưa những cải tiến đáng kể vào sản phẩm. Hiện tại, phiên bản DB2 8 và DB2 9 được hỗ trợ chính thức bởi IBM. Một phiên bản cũng có thể có một vài lần phát hành, có thể bao gồm một vài chức năng mới nhưng thông thường không đủ rõ ràng để nói rằng đây là một phiên bản mới. Ví dụ 8.1 và 8.2 là các lần phát hành với phiên bản DB2 8. Trong thời gian vừa qua, IBM thường đưa ra một lần phát hành mới của DB2 mỗi năm một lần, tuy nhiên những phiên bản mới thường được đưa ra cách nhau từ 2-3 năm. Lần phát hành mới nhất đến thời điểm hiện tại, V9.1 (trước đây được gọi là DB2 “Viper”) trở thành phiên bản chính thức vào mùa hè năm 2006. Khi cuốn sách này được viết (tháng 9/2007), lần phát hành tiếp theo (có tên là DB2 “Viper 2”) đã có phiên bản beta. Mỗi lần phát hành cũng có thể chứa nhiều mức chỉnh sửa, thường chứa những phần vá lỗi tức là tương ứng với các chương trình sửa lỗi (fixpack), và rất ít khi chứa những tính năng mới. Tại thời điểm này, phiên bản, phát hành, chỉnh sửa (Version-Release-Modification/ V-R-M) của DB2 là 9.1.2 tương đương với phiên bản 9.1 cùng với bộ chương trình sửa lỗi 2.

Trên một nội dung khác, các ấn bản là những lựa chọn hoặc nhóm các gói tính năng trong mỗi phiên bản. Như đã đề cập, một ấn bản là một gói những chức năng khác nhau với giá thành và bản quyền xác định. DB2 phiên bản 9 (cũng được biết tới với tên DB2 9) có nhiều ấn bản; ví dụ: DB2 express-C 9, DB2 Express 9, DB2 Workgroup 9, và DB2 Enterprise 9 (xem hình 1.1)

## 1.7 Chuyển sang ấn bản DB2 khác

Khi cơ sở dữ liệu của bạn cần mở rộng, có thể bạn sẽ cần nâng cấp ấn bản DB2 hỗ trợ cấu hình phần cứng lớn hơn. Nếu trường hợp này xảy ra, rất dễ dàng để nâng cấp lên một ấn bản khác của DB2:

- Nếu bạn nâng cấp lên một ấn bản DB2 khác trên cùng một máy, cài đặt ấn bản DB2 mới đè lên DB2 Express-C, cùng với bản quyền tương ứng. Cơ sở dữ liệu của bạn sẽ không bị xóa (nhưng chúng ta vẫn nên sao lưu cơ sở dữ liệu trước khi nâng cấp).
- Nếu bạn nâng cấp DB2 trên một máy mới, lớn hơn và sử dụng chung một hệ điều hành với máy cũ, hãy cài đặt ấn bản DB2 mới trên máy mới, sao lưu dữ liệu trên máy cũ và khôi phục lại nó trên máy mới. Bạn cũng có thể lưu các cấu hình trên máy cũ (dbm cfg) và cài đặt cấu hình này trên máy mới. Các lệnh sao lưu và khôi phục sẽ được nói rõ hơn trong Chương 11 Sao lưu và Khôi phục, dbm cfg sẽ được nói rõ hơn trong Chương 5, Môi trường DB2.
- Trong bất kì trường hợp nào, ứng dụng của bạn đều không phải sửa chữa để sử dụng DB2.

## 1.8 Bảo trì DB2 Express-C

Như đã nói trước đây, có hai lựa chọn hỗ trợ cho DB2 Express –C:

1. Mua một bản quyền 12 tháng. Điều này sẽ cung cấp cho bạn hỗ trợ 24x7 từ bộ phận hỗ trợ kĩ thuật của IBM, đồng thời còn cho bạn khả năng cài đặt các cập nhật của DB2 (còn được gọi là các chương trình sửa lỗi fixpack).

2. Sử dụng diễn đàn cộng đồng DB2 Express-C trực tuyến. Cách này hoàn toàn miễn phí, tuy nhiên bạn sẽ không có được sự hỗ trợ chính thức nào từ IBM. Đồng thời, với cách này, IBM không cam kết cung cấp những chức năng mới và những bản vá lỗi như thời gian đã định. Định nghĩa về một chương trình sửa lỗi, được nói đến trong Chương 2, cũng sẽ không có ý nghĩa ở đây, thay vì vậy một bản cài đặt mới của DB2 Express-C sẽ được cung cấp trên trang web cho các người dùng tải về. Khi IBM đưa ra bất kì sự chỉnh sửa mới nào trong DB2, bạn có thể tìm thấy bản cài đặt mới này và thay thế cho bản DB2 cũ bạn đang dùng.

## 1.9 Một số phần mềm miễn phí liên quan

Tất cả các phần mềm có thể được tải từ trang tải phần mềm của DB2 Express-C ([www.ibm.com/db2/express/download.html](http://www.ibm.com/db2/express/download.html)) đều là phần mềm miễn phí. Bên cạnh những bản DB2 Express-C (cho Linux và Windows, cả kiến trúc 32 và 64 bit), còn có những phần mềm hữu dụng có thể được tải và sử dụng miễn phí như:

- DB2 Developer Workbench
- DB2 9 Embedded Application Server
- DB2 9 Net Search Extender

Ngoài ra bạn có thể tải về bộ công cụ khởi đầu dựa trên DB2 Express-C từ trang web IBM Alphaworks ([www.alphaworks.ibm.com/datamgmt](http://www.alphaworks.ibm.com/datamgmt)). Bộ công cụ này bao gồm:

- Starter toolkit for DB2 on Rails
- Web 2.0 Starter Toolkits for DB2

### 1.9.1 DB2 Developer Workbench (công cụ phát triển)

DB2 Developer Workbench (DBW) là một môi trường phát triển tích hợp mạnh (IDE) cho phép bạn tạo ra, sửa chữa, gỡ rối, triển khai và kiểm tra các thủ tục được lưu trữ SQL PL và Java, cũng như các hàm do người dùng định nghĩa (User defined function UDFs), các ứng dụng SQLJ, khởi tạo và thực thi các câu lệnh SQL và các truy vấn XML. Công

cụ này dựa trên Eclipse IDE và thay thế IBM Development Center của các phiên bản DB2 trước đây. Chúng ta sẽ thảo luận DBW trong chương 14, Các thủ tục lưu trữ SQL PL.

### 1.9.2 DB2 9 Embedded Application Server (máy chủ nhúng)

Embedded Application Server cho phép bạn chạy các ứng dụng Web được hỗ trợ với DB2 phiên bản 9.1 mà không yêu cầu bạn phải mua thêm một máy chủ ứng dụng riêng biệt. Các ứng dụng Web được hỗ trợ với DB2 phiên bản 9.1 bao gồm:

- Các công cụ Web DB2 dành cho người quản trị cơ sở dữ liệu nền web.
- DB2WebServer, một ứng dụng tự động triển khai các dịch vụ web .NET từ Microsoft Visual Studio đến DB2 Embedded Application Server..

### 1.9.3 DB2 9 Net Search Extender (mở rộng tìm kiếm trên mạng)

Với DB2 9 Net Search Extender bạn có thể thực thi nhanh và chi tiết các tìm kiếm nội dung văn bản (full-text), bao gồm bất cứ tài liệu XML nào được lưu trữ binaire trong DB 9.1

### 1.9.4 Bộ công cụ bắt đầu Rails cho DB2

Bộ công cụ bắt đầu Rail cho DB2 là một tập các sản phẩm và công nghệ được đóng gói tiện lợi cho phép tạo ra nhanh chóng một môi trường để xây dựng các ứng dụng Web DB2 sử dụng công nghệ Ruby trên nền Rails. Tất cả các phần mềm được yêu cầu bao gồm: DB2 Express-C, trình điều khiển DB2 cho Ruby, bộ thích nghi (adapter) DB2 với Rails, bên cạnh đó là các hướng dẫn, ví dụ, và những tài liệu học tập khác. Chúng ta sẽ thảo luận Ruby trên nền Rails ở Chương 17, Sự phát triển trong Java, PHP và Ruby.

### 1.9.5 Bộ công cụ bắt đầu Web 2.0 dành cho DB2

Bộ công cụ bắt đầu Web 2.0 dành cho DB2 là một cách dễ dàng để bắt đầu với DB2, PHP và Dojo. Nó giúp cho bạn triển khai các yêu cầu phần mềm, cung cấp cho bạn các bài hướng dẫn và các bản ứng dụng trình diễn (demo). Hai trong số các ứng dụng trình diễn là Atom Feed Control Panel, nó khởi tạo các Atom lấy từ các bảng DB2, và Web Services Control Panel tạo ra các bao phủ dịch vụ Web REST xung quanh các bảng DB2 của bạn. Cả hai ứng dụng đều dựa trên Dojo hỗ trợ đầy đủ khả năng nổi bật của Ajax và khả năng mở rộng.

# 2

## Chương 2 – Các tính năng liên quan và các sản phẩm

Chương này mô tả các tính năng của DB2 đi kèm trong việc mua giấy phép sử dụng 12 tháng của DB2 Express-C. Ngoài ra còn một số tính năng khác không có trong DB2 Express-C, nhưng là một phần của những ấn bản DB2 khác, trong trường hợp có trả thêm phí.

Những tính năng có trong DB2 Express-C có bản quyền 12 tháng

- Bộ chương trình sửa lỗi (Fixpacks)
- Tính sẵn sàng cao và Khôi phục sau sự cố (*High Availability and Disaster Recovery- HADR*)
- Di trú dữ liệu (SQL hợp nhất)

Những tính năng không có mặt trong DB2 Express-C nhưng có mặt trong các ấn bản DB2 khác là:

Các tính năng của ấn bản DB2 Enterprise có phí:

- Tính năng phân tán cơ sở dữ liệu (DPF)
- Tính năng tối ưu hóa lưu trữ dữ liệu (kể cả nén)
- Điều khiển truy nhập mức cao (an toàn mức cao và được chia nhỏ tốt)
- Tối ưu sự thực thi (kiểm soát các truy vấn, chuyên nghiệp trong thực thi)
- Quản lý dữ liệu đo đạc
- Sự liên kết hợp nhất dữ liệu DB2

DB2 Enterprise cũng bao gồm những tính năng được thêm vào miễn phí như

- Phân tán bảng (phạm vi)
- Các bảng truy vấn được cụ thể hóa (Materialized Query Tables MQT)
- Kỹ thuật bó đa chiều (Multi-dimensional Clustering - MDC)
- Tính sẵn sàng cao và khả năng phục hồi sau sự cố (HADR)
- Bộ tập trung kết nối (Connection Concentrator)

Các tính năng của ấn bản DB2 Workgroup và Express có phí

- Tính sẵn sàng cao
- Quản lý công việc (Bộ tập trung kết nối, bộ kiểm soát truy vấn)
- Tối ưu hiệu năng (MQT, MDC, truy vấn song song)
- Sự liên kết hợp nhất dữ liệu DB2

Các sản phẩm có phí có liên quan tới DB2:

- DB2 Connect
- WebSphere® Federation Server - Máy chủ nhóm WebSphere
- WebSphere Replication Server - Máy chủ di trú WebSphere



## 2.1 Các tính năng bao hàm trong đăng ký sử dụng DB2 Express-C

Phần này điếm qua các chương trình sửa lỗi DB2, sự di trú HADR và SQL.

### 2.1.1 Các chương trình sửa lỗi

Một chương trình sửa lỗi DB2 là một tập các sửa lỗi được áp dụng cho một sản phẩm DB2 đã được cài đặt, nhằm mục tiêu sửa các vấn đề khác nhau được báo cáo sau khi sản phẩm được phát hành. Với một đăng ký sử dụng có bản quyền, các chương trình sửa lỗi có thể được tải xuống và cài đặt miễn phí. Chúng thường được đưa ra ba tháng một lần.

Để tải chương trình sửa lỗi mới nhất, xem trang hỗ trợ kỹ thuật của DB2 tại địa chỉ [http://www.ibm.com/software/data/db2/support/db2\\_9/](http://www.ibm.com/software/data/db2/support/db2_9/)

### 2.1.2 Tính sẵn sàng cao và khôi phục sau sự cố (HADR)

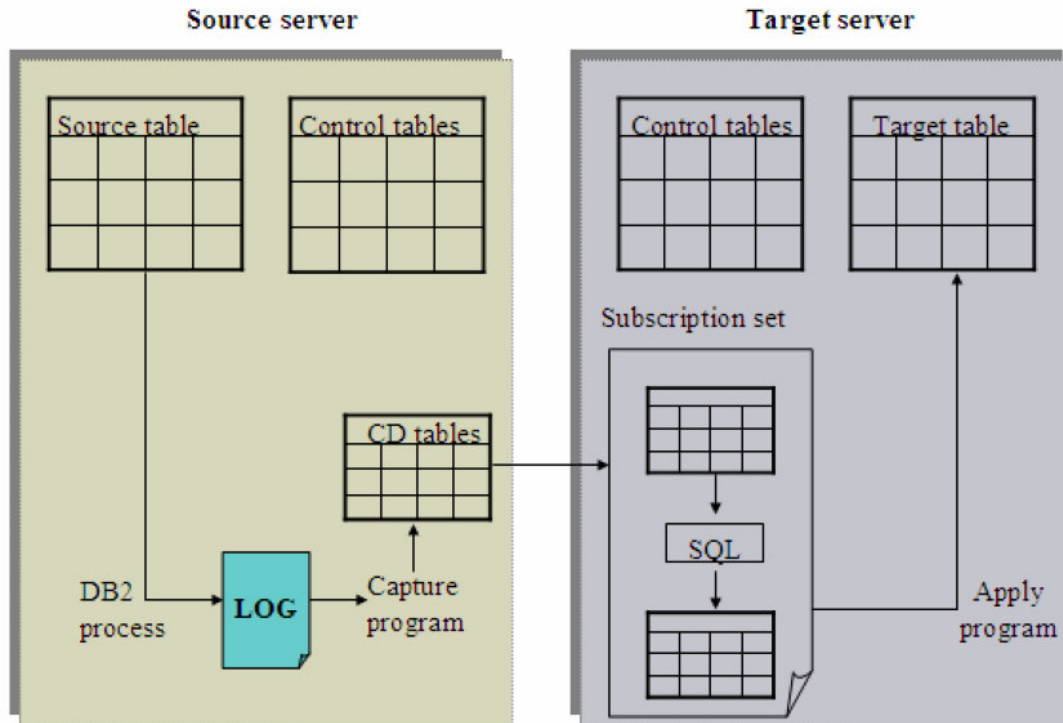
Tính sẵn sàng cao và khôi phục sau sự cố là độ tin cậy của dữ liệu, nó cung cấp một giải pháp sẵn sàng và khôi phục sau sự cố cho sự suy sụp toàn phần hoặc một phần. Một môi trường HADR thông thường bao gồm hai máy chủ dữ liệu, máy chủ chính và phụ (có thể được đặt tại những nơi khác nhau). Máy chủ chính là nơi cơ sở dữ liệu gốc được lưu trữ và truy cập bởi các trình ứng dụng tại máy khách. Trong khi các giao dịch được tiến hành trên cơ sở dữ liệu chính, các bản ghi nhật ký dữ liệu được chuyển tự động đến máy chủ thứ cấp (máy chủ phụ) thông qua hệ thống mạng. Máy chủ thứ cấp có một bản sao của cơ sở dữ liệu chính, thường được tạo ra bằng cách sao lưu cơ sở dữ liệu chính và khôi phục nó tại hệ thống thứ cấp. Khi các bản ghi nhật ký dữ liệu được nhận về, chúng được thực thi lại và áp dụng cho cơ sở dữ liệu thứ hai. Thông qua việc thực hiện lại các bản ghi nhật ký này, cơ sở dữ liệu thứ hai được đồng bộ hóa với cơ sở dữ liệu chính và có thể thay thế cơ sở dữ liệu chính nếu có sự cố xảy ra.

Một giải pháp HADR hỗ trợ DB2 đầy đủ cung cấp cho bạn:

- Khả năng phục hồi cực nhanh, trong suốt tuyệt đối với khách hàng và các ứng dụng tại máy trạm
- Giao dịch đầy đủ từng thành phần nhỏ để tránh mất mát dữ liệu
- Khả năng nâng cấp hệ thống hoặc nâng cấp ứng dụng mà không gây ngắt quãng đối với dịch vụ
- Khả năng thay thế hệ thống từ xa, cung cấp khả năng phục hồi đầy đủ cho địa điểm bị sự cố
- Quản trị dễ dàng với các công cụ đồ họa của DB2
- Tất cả các khả năng trên có ảnh hưởng không đáng kể với hiệu năng tổng thể của hệ thống

### 2.1.3 Khả năng tái tạo dữ liệu (di trú dữ liệu)

Tính năng này cho phép tái tạo dữ liệu giữa máy chủ nguồn nơi các thay đổi dữ liệu được ghi lại, và một máy chủ đích nơi các thay đổi dữ liệu được áp dụng. Hình 2.1 cung cấp tổng quan về cách tái tạo dữ liệu.



Hình 2.1 – Khôi phục SQL

Trong hình 2.1 có hai máy chủ, một máy chủ nguồn và một máy chủ đích. Trên máy chủ nguồn, một chương trình “chụp” ghi nhận lại các thay đổi đối với cơ sở dữ liệu. Trên máy chủ đích, một chương trình “áp dụng” thực hiện các thay đổi cho cơ sở dữ liệu như thật. Khả năng tái tạo dữ liệu hữu dụng cho rất nhiều mục đích cần đến dữ liệu được tái lập, bao gồm giảm tải dung lượng truyền, nạp dữ liệu vào các kho chứa dữ liệu hay các “chợ” dữ liệu (data marts), và kiểm tra lịch sử thay đổi. Sử dụng tính năng tái tạo dữ liệu, bạn có thể tái tạo dữ liệu giữa DB2 Express-C và các máy chủ DB2 khác, bao gồm cả những máy sử dụng các hệ thống khác như Linux, UNIX, z/OS và i5/OS.

## 2.2 Các tính năng không có trong bản DB2 Express-C

Phần này miêu tả một vài tính năng có trong các bản DB2 khác mà không trong DB2 Express-C

### 2.2.1 Phân vùng cơ sở dữ liệu

Tính năng phân vùng cơ sở dữ liệu (DPF) chỉ có trong ấn bản DB2 Enterprise với một mức phí đăng ký bổ sung. Nó cho phép các cơ sở dữ liệu có thể được trải ra trên nhiều phân vùng khác nhau, đặt trên nhiều máy tính. DPF dựa trên kiến trúc không chia sẻ (share-nothing architecture). Mỗi máy tính, do được thêm vào nhóm phân vùng, mang theo sức mạnh xử lý dữ liệu với CPU và bộ nhớ của chúng. DPF đặc biệt hữu dụng trong môi trường máy chủ cơ sở dữ liệu lớn như các kho chứa dữ liệu nơi mà các truy vấn của hệ trợ giúp quyết định (DSS) được thực thi.

### 2.2.2 Bộ tập trung kết nối

Bộ tập trung kết nối là tính năng cho phép hỗ trợ một lượng lớn người dùng kết nối tại một thời điểm. Trước đây, mỗi kết nối cơ sở dữ liệu cần đến một tác tử (agent) cơ sở dữ liệu. Bộ tập trung kết nối giới thiệu khái niệm tác tử lô-gic, cho phép một tác tử xử lý nhiều kết nối. Các tác tử được bàn đến kỹ hơn trong chương 6, kiến trúc DB2.

### 2.2.3 Bộ mở rộng không gian địa lý

Bộ mở rộng không gian địa lý của DB2 tồn tại như một lựa chọn có phí cho ấn bản DB2 Enterprise. Phần mở rộng này làm việc phát triển cho các ứng dụng chính phủ điện tử hoặc các ứng dụng có qui trình nghiệp vụ yêu cầu phân tích các vị trí địa lý được dễ dàng hơn. Bộ mở rộng không gian địa lý của DB2 có thể cấu thành một thế giới địa lý ảo với bất kỳ tỷ lệ nào. Hầu hết các thông tin vị trí được thu thập sử dụng các hệ thống toàn cầu, như hệ thống định vị vệ tinh toàn cầu (GPS), và có thể được thể hiện trong hệ kinh độ/vĩ độ. Các dữ liệu kinh doanh, như địa chỉ, có thể được chuyển đổi thành tọa độ kinh tuyến/vĩ tuyến nhờ bộ mở rộng không gian của DB2 (DB2 Spatial Extender) và các chương trình ứng dụng doanh nghiệp làm việc tốt hơn khi chúng lưu trữ dữ liệu theo định dạng này, để phép chiếu tương ứng lên bản đồ được thống nhất ở các ứng dụng khác nhau: lớp trình bày, để hiển thị và in các bản đồ.

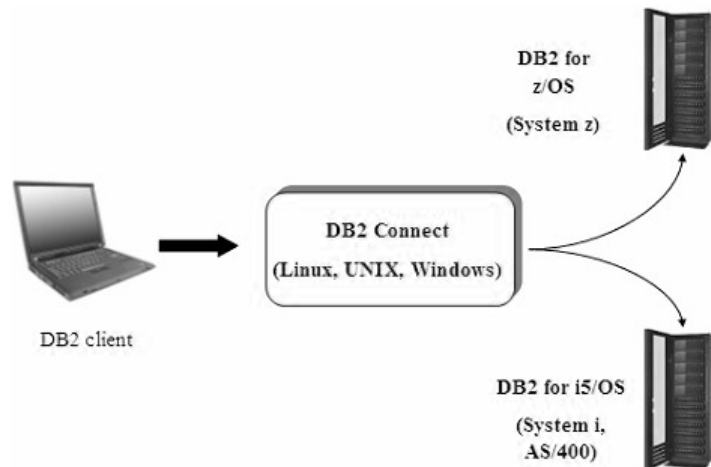
### 2.2.4 Bộ kiểm soát truy vấn

Bộ kiểm soát truy vấn của DB2 là một hệ thống quản lý truy vấn mạnh, có thể điều khiển luồng các truy vấn tới cơ sở dữ liệu DB2 của bạn. Nó cho phép bạn điều chỉnh lại khối lượng công việc của truy vấn dữ liệu, do đó các truy vấn nhỏ và có mức ưu tiên cao có thể chạy nhanh chóng, đảm bảo các tài nguyên hệ thống được sử dụng hiệu quả.

## 2.3 Các sản phẩm có trả phí liên quan đến DB2.

### 2.3.1 DB2 Connect

DB2 Connect là một phần mềm dựa trên việc trả phí cho phép một máy khách DB2 Linux, UNIX hay Windows kết nối đến một máy chủ DB2 z/OS hay DB2 i5/OS như trong hình 2.2. DB2 Connect không được yêu cầu trong quá trình liên kết theo chiều ngược lại, khi bạn kết nối từ DB2 z/OS hay DB2 i5/OS đến một máy chủ DB2 Linux, UNIX hay Windows. DB2 Connect có trong hai ấn bản chính tùy theo sự kết nối mà bạn cần: DB2 Connect Personal Edition và DB2 Connect Enterprise Edition.



Hình 2.2. DB2 Connect

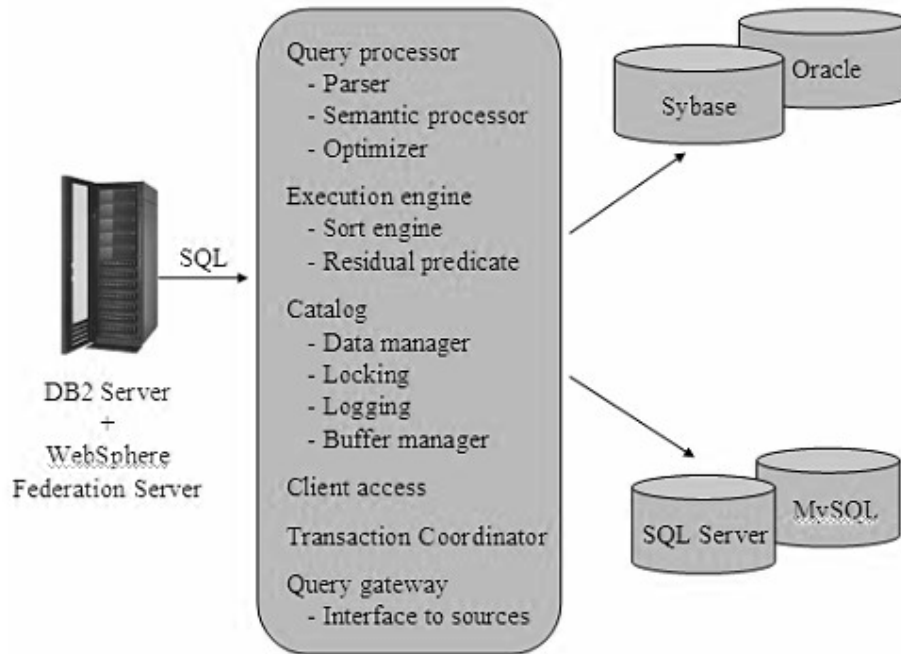
### 2.3.2 WebSphere Federation Server

Được biết đến trước đây là WebSphere Information Integrator (hỗ trợ tích hợp thông tin), WebSphere Federation Server cho phép sự liên kết hợp nhất các cơ sở dữ liệu, điều này nghĩa là bạn có thể chạy các truy vấn cơ sở dữ liệu trên nhiều đối tượng từ các hệ thống cơ sở dữ liệu quan hệ khác nhau. Ví dụ nếu bạn mua WebSphere Federation Server bạn có thể chạy truy vấn dưới đây:

```

SELECT *
FROM Oracle.Table1 A
      DB2.Table2 B
      SQLServer.Table3 C
WHERE
      A.col1 < 100
And   B.col5 = 1000
and   C.col2 = 'Test'
    
```

Hình 2.3 cung cấp sự giới thiệu về khả năng sử dụng của WebSphere Federation Server



Hình 2.3 - WebSphere Federation Server

Đối với các hệ thống quản lý cơ sở dữ liệu quan hệ thuộc họ IBM, sự hỗ trợ liên kết hợp nhất đã được xây dựng sẵn trong DB2 Express-C. Điều này nghĩa là WebSphere Federation Server không cần được đòi hỏi., ví dụ khi bạn muốn chạy một truy vấn giữa hai cơ sở dữ liệu DB2 khác nhau, hoặc giữa một cơ sở dữ liệu DB2 và một cơ sở dữ liệu Informix® (Infomix là một phần trong họ IBM).

### 2.3.3 WebSphere Replication Server

Được biết trước đây là WebSphere Information Integrator (hỗ trợ tích hợp thông tin). WebSphere Replication Server cho phép sự tạo lại SQL trong các bản ghi cơ sở dữ liệu khi các máy chủ không phải của IBM tham gia vào. Nó cũng bao gồm một tính năng được biết đến là Q-Replication cho sự tái tạo dữ liệu sử dụng các hàng đợi thông điệp.

# 3

## Chương 3 - Cài đặt DB2

Để cài đặt ấn bản DB2 Express-C cho Linux hay Windows, đảm bảo rằng các hệ thống của bạn thỏa mãn các điều kiện trước khi cài đặt.

### 3.1 Các điều kiện trước khi cài đặt

Ở khía cạnh phiên bản hệ điều hành, DB2 Express-C có thể được sử dụng trên Linux hay Windows 2000, XP, Vista. Kiến trúc bộ xử lý được yêu cầu là 32 bit, 64 bit và PowerPC (Linux). Nếu bạn cần chạy DB2 trên một nền khác (giống như UNIX), bạn nên mua một trong những ấn bản máy chủ dữ liệu khác đã được nói đến trước đây trong quyển sách này. Các điều kiện của hệ điều hành trước khi cài đặt cho tất cả các ấn bản DB2 cũng được miêu tả trong tài liệu này.

<http://www.ibm.com/software/data/db2/udb/sysregs.html>

Trong khía cạnh về các tài nguyên phần cứng, DB2 Express-C được tối ưu hóa cho các hệ thống lên tới 2 CPU vật lý và 4GB RAM. Các hệ thống này có thể là các hệ thống vật lý hay các hệ thống ảo được tạo ra bởi sự phân tán hoặc việc chạy các phần mềm máy ảo. Bạn có thể dĩ nhiên chạy nó trong các hệ thống nhỏ hơn nếu bạn muốn, ví dụ trong các hệ thống đơn CPU với 1GB RAM.

Để có được những thông tin mới nhất về các điều kiện của phần cứng trước khi cài đặt DB2 Express-C, hãy xem trên trang web DB2 Express-C.

<http://www.ibm.com/software/data/db2/udb/db2express/getstarted.html>

### 3.2 Quyền cài đặt hệ điều hành

Để cài đặt DB2 Express-C trên Linux hay Windows, bạn phải là một người sử dụng hệ điều hành có đủ quyền.

Với **Linux**: bạn cần phải là root (superuser) để cài đặt DB2 Express-C.

Với **Windows**, tài khoản người sử dụng phải thuộc về nhóm Quản trị trên máy mà bạn sẽ thực hiện cài đặt. Một cách khác, một tài khoản người dùng không thuộc nhóm Quản trị có thể được sử dụng, với điều kiện là một thành viên thuộc nhóm Quản trị Windows đầu tiên cấu hình các thiết lập nâng quyền Windows để cho phép một tài khoản người dùng không thuộc nhóm Quản trị có thể thực thi cài đặt.

Với các tài khoản miền (domain) Windows, để xác thực định danh (ID) của người dùng trên máy chủ DB2, ID của người tiến hành cài đặt phải thuộc nhóm Quản trị tại miền (domain) mà các tài khoản này sẽ được tạo. Có thể sử dụng tài khoản sẵn có (built-in) trên hệ thống để tiến hành cài đặt.

Tài khoản người dùng cũng phải có quyền “Truy cập máy tính này qua mạng”

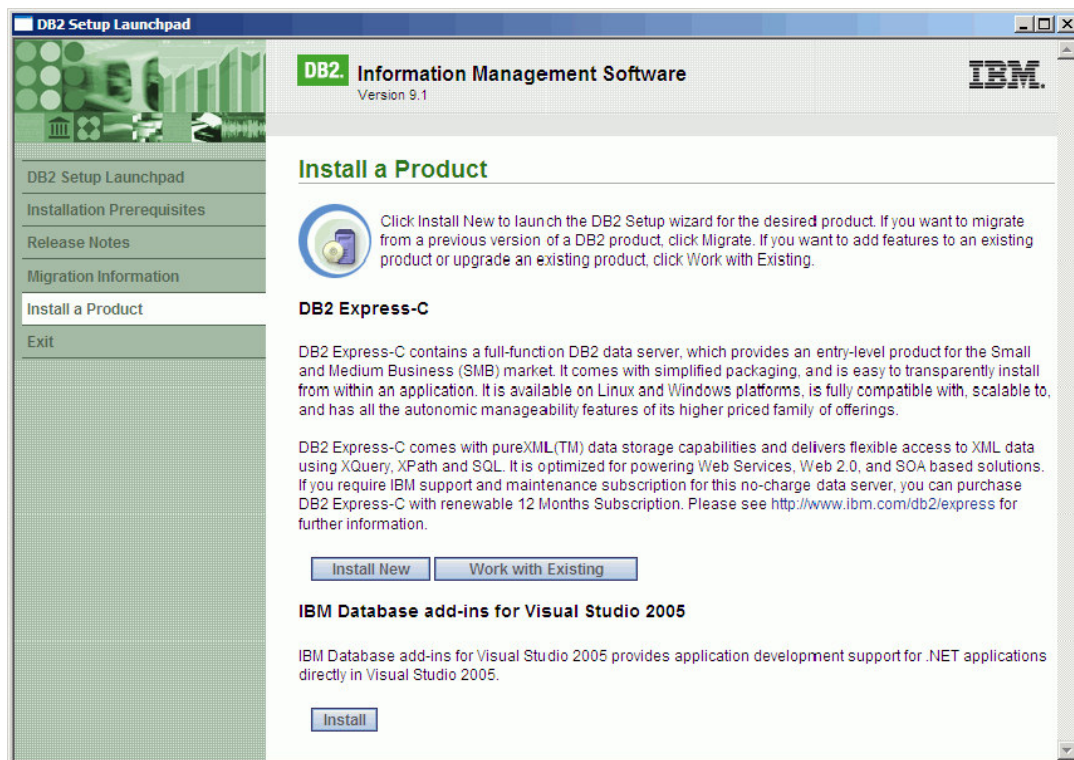
### 3.3 Cài đặt theo hướng dẫn từng bước

Có rất nhiều cách để tiến hành cài đặt DB2 Express-C, nhưng đơn giản nhất là sử dụng cách cài đặt theo hướng dẫn từng bước dựa trên giao diện đồ họa của DB2. Sau khi tải và giải nén DB2 Express-C, bạn có thể tiến hành cài đặt như sau :

- Windows: chạy tệp setup.exe trong thư mục EXP/image.
- Linux: chạy lệnh db2setup trong thư mục exp/disk1.

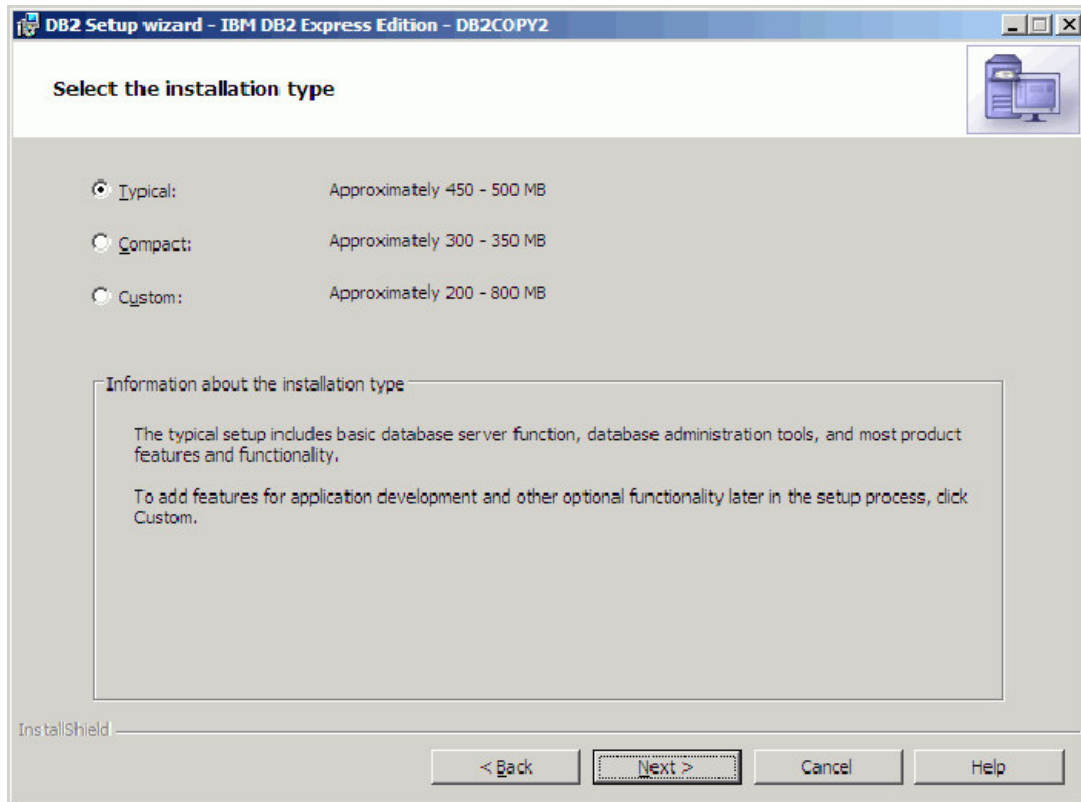
Việc cài đặt DB2 Express-C theo hướng dẫn cài đặt từng bước rất đơn giản. Thông thường, bạn chỉ việc sử dụng các thiết lập mặc định, vì vậy, tất cả các việc cần làm là chấp nhận điều khoản sử dụng, nhấn liên tục nút “Next” rồi nhấn “Finish”. Quá trình cài đặt sẽ kết thúc sau vài phút và bạn có thể bắt đầu sử dụng DB2.

Trên hình 3.1 là cửa sổ bắt đầu cài đặt DB2 (DB2 Setup Launchpad), bạn nên chọn “Install New” để cài đặt một bản sao mới của DB2 lên hệ thống.



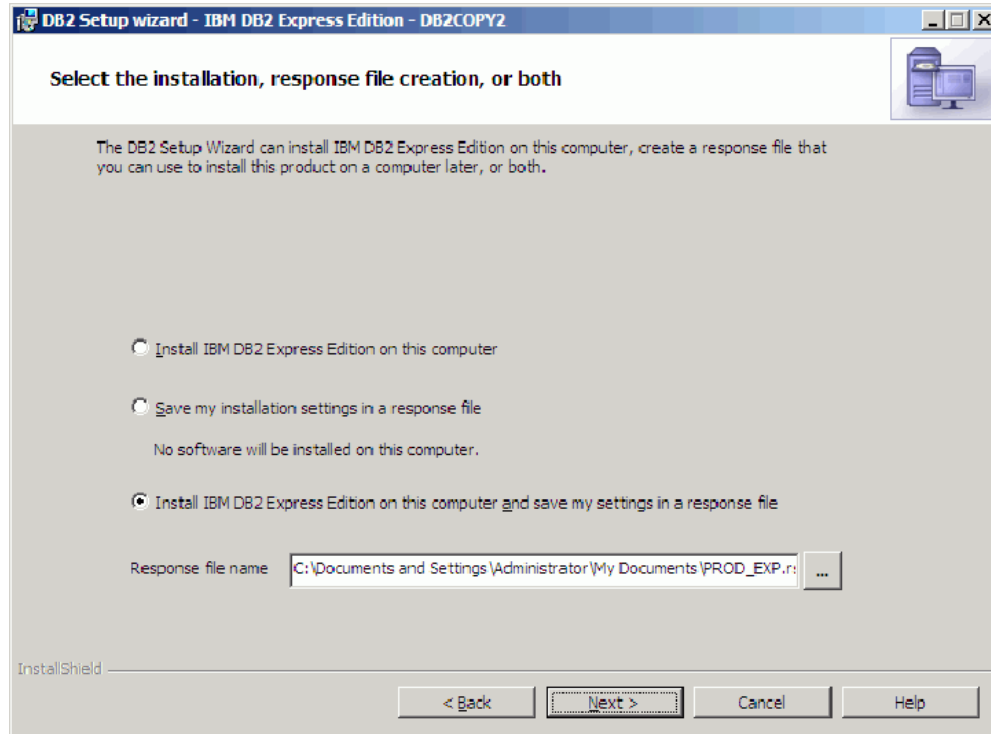
Hình 3.1 – Bắt đầu cài đặt DB2 (DB2 Setup Launchpad)

Sau khi chấp nhận điều khoản sử dụng, thông thường bạn nên chọn kiểu cài đặt “Typical” (mặc định) như trên hình 3.2.



**Hình 3.2 – Các kiểu cài đặt**

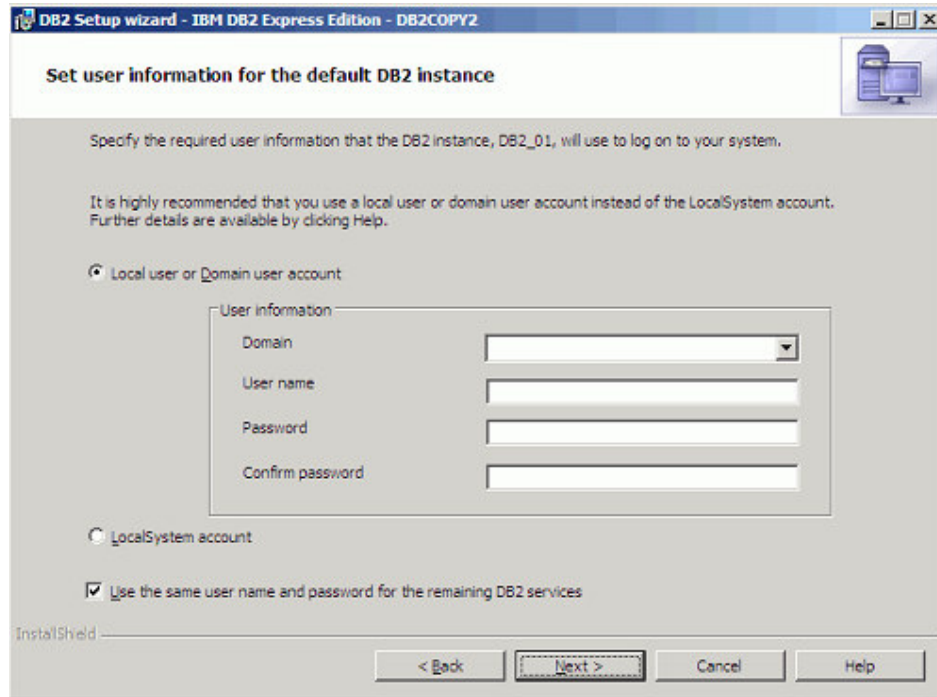
Trong hình 3.3, bạn có các lựa chọn để cài đặt sản phẩm, tạo một tệp biên bản (response file), hoặc cả hai. Các tệp biên bản sẽ được thảo luận trong mục 3.4, Cài đặt hàng loạt (Silent Install). Sử dụng tùy chọn mặc định (Cài đặt IBM DB2 Express Edition trên máy tính này và lưu các thiết lập ra tệp biên bản) là đủ.



**Hình 3.3 – Lựa chọn cài đặt**

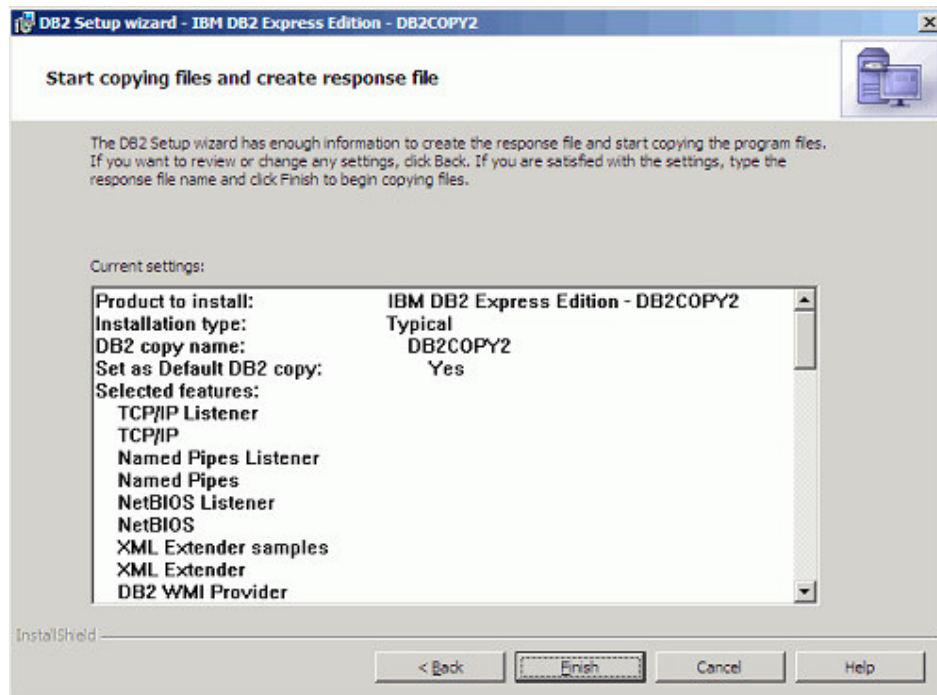
Trong vài màn hình tiếp theo, hãy chọn giá trị mặc định cho đến khi bạn gặp màn hình như trong hình 3.4. Trong màn hình này bạn có thể nhập vào một người dùng đã có để làm việc với bản cài đặt DB2 này (instance) cũng như các dịch vụ khác. Người dùng này phải nằm trong danh sách Quản trị (Local Administrator) của Windows. Nếu như ID của người dùng bạn nhập vào không tồn tại, một người dùng mới sẽ được thêm vào nhóm Quản trị. Bạn có thể để trống “domain” nếu như người dùng bạn nhập vào không thuộc vào một miền nào đó. ID người dùng mặc định được tạo ra trong windows là db2admin, trong Linux là db2inst1.





Hình 3.4 - Nhập thông tin người dùng cho bản DB2 mặc định.

Cuối cùng trong hình 3.5, cài đặt theo hướng dẫn từng bước sẽ hiển thị một danh sách tổng kết những phần nào sẽ được cài đặt cũng như một vài thông tin, thiết lập do bạn nhập vào. Khi bạn nhấn “Finish”, quá trình cài đặt DB2 vào hệ thống của bạn bắt đầu.



Hình 3.5 - Danh sách tổng kết những phần sẽ được cài

### 3.4 Cài đặt hàng loạt (Silent Install)

Trong nhiều trường hợp bạn cần phải cài đặt máy khách DB2 trên nhiều máy tính khác nhau. Hoặc máy chủ dữ liệu DB2 là một phần trong ứng dụng bạn xây dựng và bạn

muốn quá trình cài đặt DB2 là một phần của quá trình cài đặt chung toàn bộ ứng dụng đó. Khi đó, cài đặt hàng loạt là phương pháp phù hợp cho bạn.

DB2 hỗ trợ cài đặt hàng loạt nhờ tệp biên bản (response file) chứa các thông tin cần thiết cho quá trình cài đặt. Dưới đây là một trích đoạn của một tệp biên bản mẫu.

Tệp biên bản mẫu:

```
PROD = UDB_EXPRESS_EDITION
LIC_AGREEMENT=ACCEPT
FILE = C:\Program Files\IBM\SQLLIB\
INSTALL_TYPE = CUSTOM
COMP=TCPIP_DB2_LISTENER_SUPPORT COMP=TCPIP_DB2_CLIENT_SUPPORT
COMP=DB2_SAMPLE_DATABASE
COMP=DB2_SAMPLE_APPLICATIONS
COMP=OLE_DB_SUPPORT
COMP=ODBC_SUPPORT
COMP = JDBC_SUPPORT
COMP = IBM_JRE
...
```

Các phương pháp có thể dùng để tạo ra tệp biên bản:

- Cài đặt DB2 Express-C trước một lần, sử dụng phương pháp cài đặt theo hướng dẫn từng bước. Một trong các lựa chọn đầu tiên khi bạn cài đặt theo phương pháp này cho phép bạn đánh dấu yêu cầu DB2 lưu lại các thông số bạn nhập vào trong quá trình cài đặt vào một tệp biên bản. Khi kết thúc quá trình cài đặt tệp biên bản sẽ được lưu vào thư mục và tên tệp do bạn nhập vào. Vì tệp này là một tệp văn bản cho nên sau đó bạn có thể sửa lại nó tùy ý. Bạn có thể xem lại Hình 3.3 về phương pháp này.
- Chỉnh sửa tệp biên bản mẫu được đóng gói kèm theo DB2. Tệp mẫu này (có phần mở rộng là .rsp) nằm trong thư mục: *db2/platform/samples/ directory*
- Trong Windows, bạn cũng có thể dùng lệnh tạo tệp biên bản:  
db2rspgn -d <thư mục lưu tệp>
- Sau khi bạn đã có tệp biên bản, trong Windows bạn dùng lệnh:  
setup -u <Tên tệp biên bản>
- Đối với Linux, bạn dùng lệnh:  
db2setup -r <Tên tệp biên bản>

## Bài tập nhanh #1: Cài đặt DB2 Express-C và tạo ra cơ sở dữ liệu mẫu

Chú ý: Nếu bạn đang sử dụng DB2 Express-C V9.1.2 hoặc phiên bản cao hơn, nếu bạn chọn cài đặt thông qua chế độ đồ họa cơ sở dữ liệu mẫu có thể được tạo ra cho bạn một cách tự động.

### Mục đích

Trước khi bạn có thể bắt đầu nghiên cứu các tính năng và công cụ có trong DB2 Express-C, bạn phải cài nó trên hệ thống của mình. Bài thực hành này sẽ hướng bạn từng bước cách cài đặt DB2 Express-C trên Window.

### Các bước

1. Lấy về ảnh của DB2 Express-C. Để có thể cài đặt DB2, bạn có thể tải bộ cài dưới dạng tệp .zip tại địa chỉ <http://www.ibm.com/db2/express>. Giải nén các tệp vào một thư mục do bạn đặt tên.

2. Định vị thư mục chính về thư mục chứa các tệp đã giải nén

3. Kích hoạt tiến trình cài đặt. Để kích hoạt tiến trình cài đặt, bạn nhấn đúp vào tệp setup.exe, sau đó từ cửa sổ windows hiện ra bạn nhấn vào tùy chọn Install Product ở phía bên trái của cửa sổ.

4. Chạy cài đặt DB2 theo từng bước (setup wizard). Sau khi bạn chọn Install Product ở bước 3, một thành phần trợ giúp cài đặt gọi là “cài đặt từng bước” sẽ chạy kiểm tra cấu hình phần cứng của hệ thống để tiến hành chuẩn bị cài đặt DB2 Express-C, cũng như kiểm tra xem bạn đã cài DB2 trước đó hay chưa. Sau khi chương trình đã kiểm tra xong bạn nhấn vào nút Next để tiếp tục cài đặt

5. Chấp nhận các điều khoản ràng buộc, bạn hãy đọc kỹ các điều khoản trong giấy phép sử dụng và lựa chọn “I Accept...” và nhấn nút Next để tiếp tục tiến trình cài đặt.

6. Lựa chọn kiểu cài đặt. Trong bài thực hành này bạn hãy chọn lựa kiểu cài đặt là Typical (bạn sẽ thấy rằng đây là kiểu cài đặt mặc định). Hai kiểu cài đặt khác là Compact và Custom. Kiểu Compact là kiểu cài đặt cơ bản còn kiểu Custom cho phép bạn thực hiện một số tùy chọn trong quá trình cài đặt. Nhấn nút Next để tiếp tục.

7. Lựa chọn thư mục cài đặt. Bạn chọn thư mục để cài đặt DB2, chú ý đến dung lượng của ổ cứng. Trong ví dụ này bạn chọn thư mục mặc định để cài đặt DB2 lên hệ thống của bạn như sau:

**Ổ cài đặt:** C:

**Thư mục cài đặt:** C:\Program Files\IBM\SQLLIB

Nhấn nút Next để tiếp tục.

8. Nhập vào thông tin của người sử dụng. Một khi DB2 Express-C đã được cài đặt, những tiến trình DB2 sẽ được chạy dưới tư cách là các dịch vụ của hệ thống. Những dịch vụ này đòi hỏi phải được đăng nhập bằng một tài khoản của hệ thống để có thể thực hiện. Trong môi trường Window bạn sử dụng tài khoản mặc định được đề nghị là db2admin, nếu tài khoản này chưa tồn tại, DB2 sẽ tạo ra cho bạn. Chú ý bạn phải đặt mật khẩu (password) cho tài khoản. Bạn cũng có thể sử dụng một tài khoản sẵn có trong

hệ thống thay vì nhờ DB2 tạo ra một tài khoản mới cho bạn nhưng phải ghi nhớ rằng để có thể chạy các dịch vụ DB2 tài khoản này phải có quyền quản trị (admin). Nhấn nút Next để tiếp tục.

9. Cấu hình DB2 instance. Một instance DB2 có thể được coi như là một “kho chứa cho cơ sở dữ liệu”. Một instance này phải tồn tại trước khi bất kỳ một cơ sở dữ liệu nào được tạo ra và được “đặt vào bên trong kho”. Trong môi trường Window, một instance được gọi là *DB2* tự động được tạo ra, trong khi trong môi trường Linux tên của instance mặc định này là *db2inst1*. Chúng ta sẽ giải thích chi tiết các đối tượng này trong các chương tiếp theo của cuốn sách.

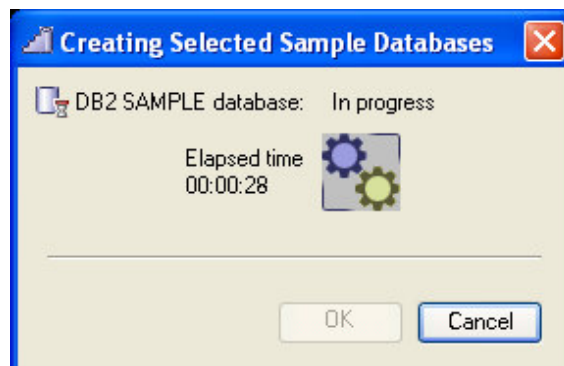
Mặc định, *instance DB2* được cấu hình để lắng nghe kết nối TCP/IP trên cổng kết nối 5000. Cả hai giao thức kết nối (TCP, UDP) và cổng kết nối đều có thể được cài đặt lại khi bạn nhấn vào các nút *Protocols* và *Startup*. Tuy nhiên, trong ví dụ này chúng ta sẽ sử dụng sự cấu hình mặc định đã được cài đặt sẵn và không thay đổi bất cứ cấu hình thuộc về giao thức kết nối hay cổng kết nối nào cả. Nhấn nút Next để tiếp tục.

10. Bắt đầu cài đặt. Để bắt đầu cài đặt bạn nhấn vào nút Install, tất cả các cấu hình và các tệp cần thiết sẽ được cài đặt vào máy của bạn một cách tự động.

11. First Steps. Sau khi quá trình cài đặt kết thúc, một công cụ hỗ trợ cài đặt khác được gọi là First Steps sẽ được chạy và hiển thị trên màn hình. First Step cũng có thể được khởi chạy bằng tay với câu lệnh `db2fs` trong cửa sổ hệ điều hành DOS.

12. Tạo ra cơ sở dữ liệu Mẫu. Trong First Steps, lựa chọn thẻ “Database Creation”, và sau đó theo sau các hướng dẫn của chương trình để tạo ra cơ sở dữ liệu Mẫu. Chú ý bạn phải đảm bảo rằng tùy chọn *XML and SQL objects and data* phải được lựa chọn, và nhấn nút OK.

13. Sau khi bạn nhấn OK một cửa sổ window sau sẽ được hiện ra để thông báo cơ sở dữ liệu mẫu đang được tạo ra cho bạn. Tiến trình này có thể mất một vài phút.



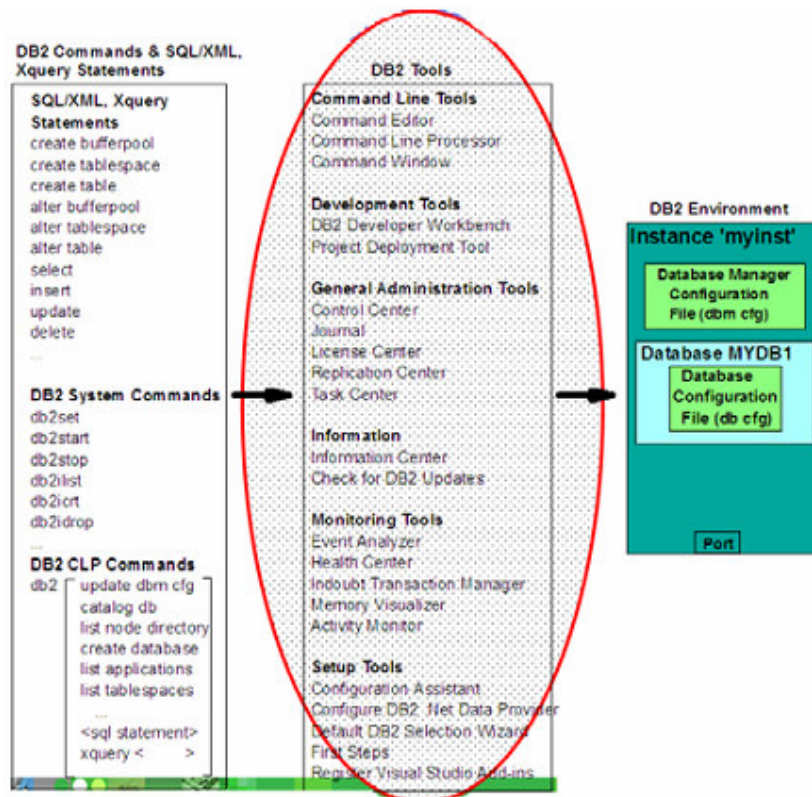
14. Xác nhận cơ sở dữ liệu mẫu đã được tạo ra. Khi việc tạo cơ sở dữ liệu đã hoàn tất, nhấn nút OK và đóng công cụ First Steps. Quay lại Trung tâm điều khiển (Control Center) và kiểm tra rằng một cơ sở dữ liệu tên là SAMPLE đã được tạo ra trong khung cửa sổ Object Tree (cây đối tượng). Bạn có thể phải làm mới lại Trung tâm điều khiển để nhận biết được sự thay đổi này.

15. Khởi động lại máy tính. Mặc dù bước này là không bắt buộc trong các tài liệu hướng dẫn cài đặt DB2, nhưng chúng tôi đề nghị bạn nên khởi động lại hệ thống để đảm bảo rằng tất cả các tiến trình được bắt đầu một cách hoàn hảo và giải phóng các tài nguyên của hệ thống đã bị chiếm dụng trong quá trình cài đặt DB2. Đây là một Tùy chọn.

# 4

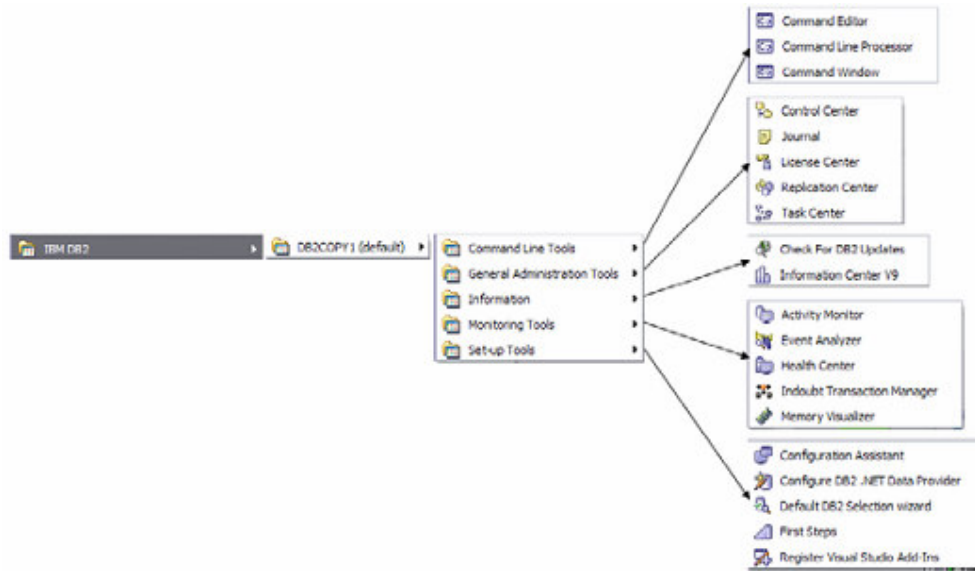
## Chương 4 - Các công cụ DB2

Trong chương này chúng tôi sẽ giới thiệu một số công cụ bạn có thể dùng với DB2. Trong hình 4.1, phía bên trái là các lệnh DB2 khác nhau, SQL, SQL/XML và các câu lệnh XQuery có thể được dùng để tương tác với máy chủ dữ liệu DB2. Phần ở giữa (được đánh dấu bằng hình elip màu đỏ) liệt kê tên của các công cụ khác nhau có thể tương tác với máy chủ dữ liệu DB2. Phía bên phải trong hình vẽ là môi trường DB2 cơ bản bao gồm một thể hiện (instance) DB2, một cơ sở dữ liệu và các tệp cấu hình tương ứng.



Hình 4.1 - Bức tranh tổng thể DB2: Các công cụ DB2

Hình 4.2 liệt kê tất cả các công cụ DB2 sẵn có trong IBM DB2 Start Menu. Hầu hết các công cụ này giống nhau trên cả Linux và Windows.



**Hình 4.2 - Các công cụ DB2 trên IBM DB2 Start Menu**

Bảng 4.1 cung cấp một danh sách các lệnh tắt để gọi một số các công cụ DB2 phổ biến nhất trong cả môi trường Linux và Windows.

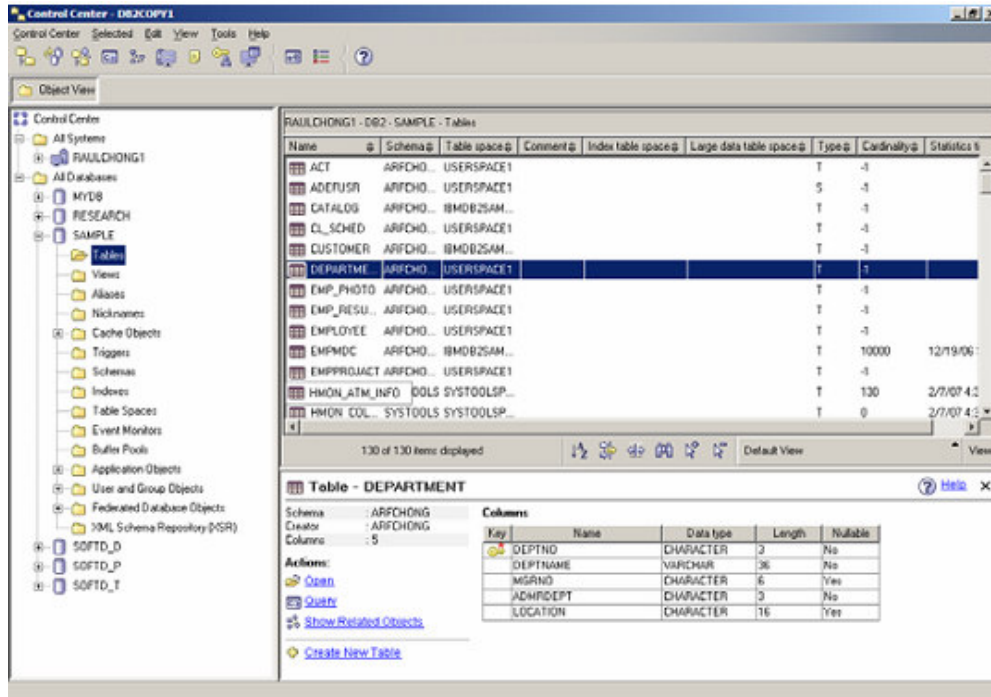
Tên công cụ	Mô tả	Lệnh
Command Editor	Cửa sổ nhập lệnh	db2ce
Command Line processor	Bộ xử lý dòng lệnh	db2
Command Window	Cửa sổ lệnh (Chỉ dành cho môi trường Windows)	db2cmd
Control Center	Trung tâm điều khiển	db2cc
Task Center	Trung tâm tác vụ	db2tc
Health Center	Trung tâm sức khỏe/tình trạng	db2hc
Configuration Assistant	Trợ giúp cấu hình	db2ca
First Steps	Bước khởi đầu	db2fs

**Bảng 4.1 - Các lệnh tắt cho một số công cụ DB2**

#### 4.1 Trung tâm điều khiển (Control Center)

Công cụ chính của DB2 dùng trong việc quản trị cơ sở dữ liệu là Trung tâm điều khiển có giao diện như trong hình 4.3





Hình 4.3 - Trung tâm điều khiển của DB2

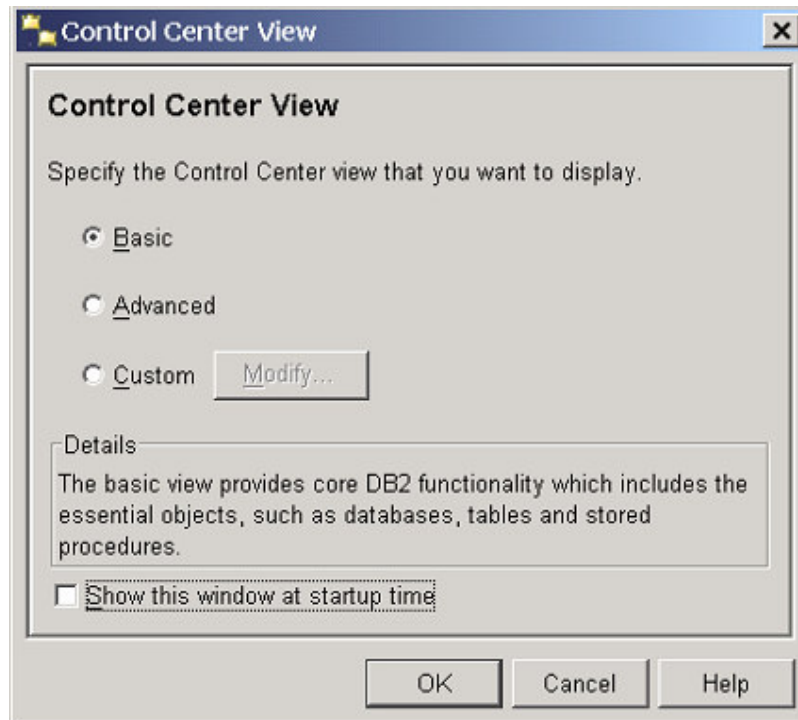
Trung tâm điều khiển là một công cụ quản lý tập trung cho phép bạn:

- Xem xét hệ thống của bạn, các bản cài đặt DB2, các cơ sở dữ liệu và các đối tượng trong cơ sở dữ liệu;
- Tạo, sửa và quản lý cơ sở dữ liệu và các đối tượng cơ sở dữ liệu;
- Khởi động các công cụ đồ họa khác của DB2

Khung bên trái cung cấp một cây phân cấp trực quan phản ánh cấu trúc của các đối tượng cơ sở dữ liệu trong hệ thống của bạn, với các “thư mục” đại diện cho bảng (Tables), thể hiện (Views)... Khi bạn nhấn đúp vào một thư mục (ví dụ thư mục Tables, như trong hình 4.3), khung trên cùng bên phải sẽ liệt kê tất cả các đối tượng liên quan, trong trường hợp này là tất cả các bảng liên quan đến cơ sở dữ liệu SAMPLE. Nếu bạn chọn một bảng nào đó ở khung trên bên phải, thì khung phía dưới bên phải sẽ cung cấp các thông tin chi tiết hơn về bảng đó.

Nhấn chuột phải vào các thư mục/đối tượng khác nhau trong Cây đối tượng (Object tree) sẽ hiện lên một bảng chọn thích hợp với thư mục/đối tượng đó. Ví dụ, nhấn phải chuột lên một cài đặt DB2 và chọn “Configure parameters” cho phép bạn xem và cập nhật tệp cấu hình cho chương trình quản trị cơ sở dữ liệu. Tương tự, nếu bạn nhấn chuột phải vào một cơ sở dữ liệu và chọn “Configure parameters”, bạn có thể xem và cập nhật tệp cấu hình cho cơ sở dữ liệu đó. Môi trường DB2 và các thông số cấu hình sẽ được nhắc tới kỹ hơn ở chương 5, Môi trường DB2.

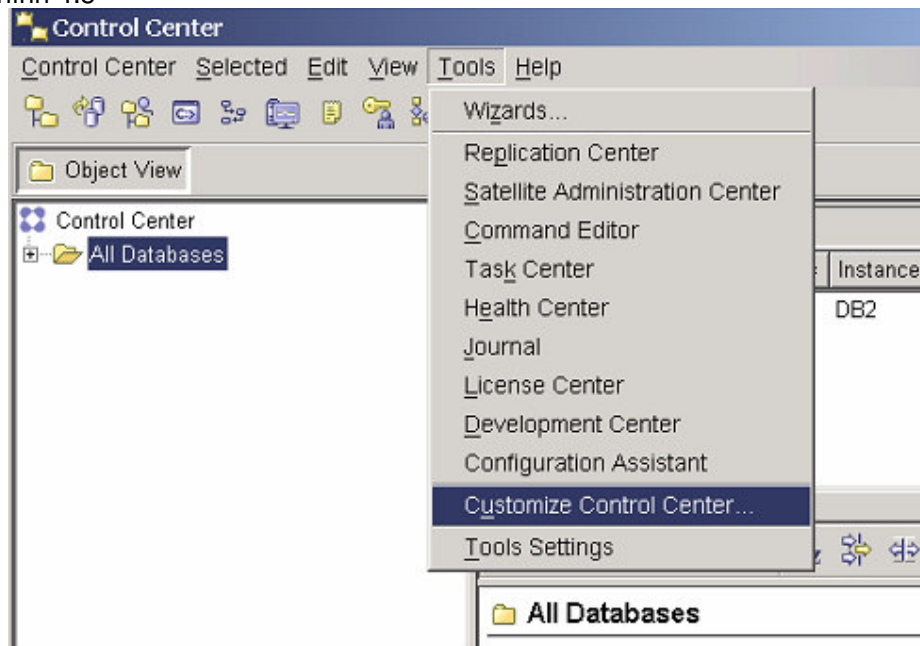
Lần đầu tiên chạy Trung tâm điều khiển, bạn được hỏi và lựa chọn cách trung tâm điều khiển sẽ hiển thị theo cách nào. Ứng với mỗi cách hiển thị, các đối tượng cơ sở dữ liệu sẽ được thể hiện theo kiểu tương ứng. Hình 4.4 là hộp thoại dùng để lựa chọn cách hiển thị của Trung tâm điều khiển.



**Hình 4.4 - Hộp thoại chọn cách hiển thị Trung tâm điều khiển**

Cách hiển thị cơ bản (basic) cung cấp các chức năng nòng cốt của DB2.  
 Cách hiển thị nâng cao (advanced) thể hiện nhiều lựa chọn và chức năng hơn.  
 Cách hiển thị tùy chọn (custom) cho phép bạn tùy biến các hình thức thể hiện các chức năng, các lựa chọn và các đối tượng theo bạn muốn.

Để bật lại hộp thoại này, bạn chọn “Customize Control Center” từ bảng chọn Tools như trong hình 4.5




**Hình 4.5 - Tùy biến Trung tâm điều khiển**



**Khởi động Trung tâm điều khiển**

Có nhiều cách để khởi động Trung tâm điều khiển:

- Sử dụng từ Menu Start của Windows
- Chạy lệnh *db2cc* từ dấu nhắc lệnh
- Nhấn đúp vào biểu tượng Trung tâm điều khiển  trên bất kỳ thanh công cụ nào của các công cụ giao diện đồ họa DB2.
- Từ biểu tượng DB2 trên khay hệ thống của Windows như trong hình 4.6 (Nhấn phải chuột vào biểu tượng DB2 và chọn DB2 Trung tâm điều khiển)

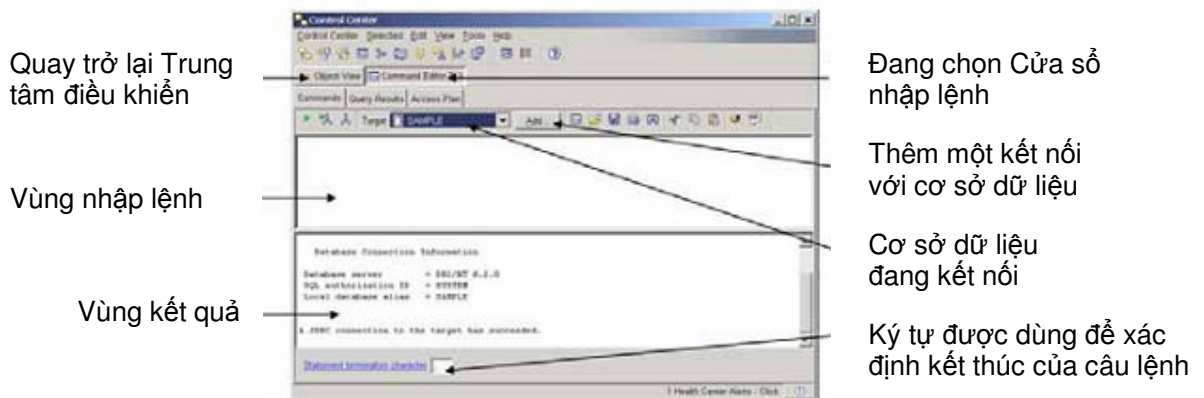


**Hình 4.6 - Khởi động Trung tâm điều khiển DB2 từ khay hệ thống của Windows**

**4.2 Nhập lệnh trực tiếp**

Sử dụng cửa sổ nhập lệnh của DB2 (Command Center), bạn có thể thực thi các lệnh DB2, SQL và các câu lệnh XQuery, phân tích kế hoạch thực hiện của một câu lệnh, xem hay sửa kết quả của câu truy vấn.

Hình 4.7 thể hiện Cửa sổ nhập lệnh cùng với mô tả ngắn gọn các thành phần của nó.



**Hình 4.7 - Cửa sổ nhập lệnh của DB2**

Trong vùng nhập lệnh (input area), bạn có thể nhập nhiều câu lệnh cùng lúc, nhưng cần đảm bảo các câu lệnh kết thúc với ký tự kết thúc do bạn chọn. Khi bạn bấm nút thực hiện (xem hình 4.8), các câu lệnh sẽ được thực hiện lần lượt. Nếu như bạn đánh dấu một câu lệnh nào đó thì chỉ mình câu lệnh này sẽ được thực hiện. Một kết nối tới cơ sở dữ liệu cần phải được thiết lập trước khi thực hiện bất kỳ câu lệnh SQL nào, tuy nhiên bạn cũng có thể chèn câu lệnh thực hiện kết nối vào đoạn lệnh của mình.



Thực hiện câu lệnh  
(Có thể dùng tổ hợp phím  
tắt Ctrl+Enter)


Thực hiện câu lệnh và sinh  
ra một kế hoạch truy cập  
(Sẽ nói kỹ hơn ở phần sau)

Sinh ra một kế hoạch truy cập mà  
không thực hiện câu lệnh  
(Sẽ nói kỹ hơn ở phần sau)

**Hình 4.8 - Cửa sổ nhập lệnh – Thẻ Lệnh**

**Khởi động Cửa sổ nhập lệnh**

Bạn có thể khởi động cửa sổ nhập lệnh bằng một số cách:

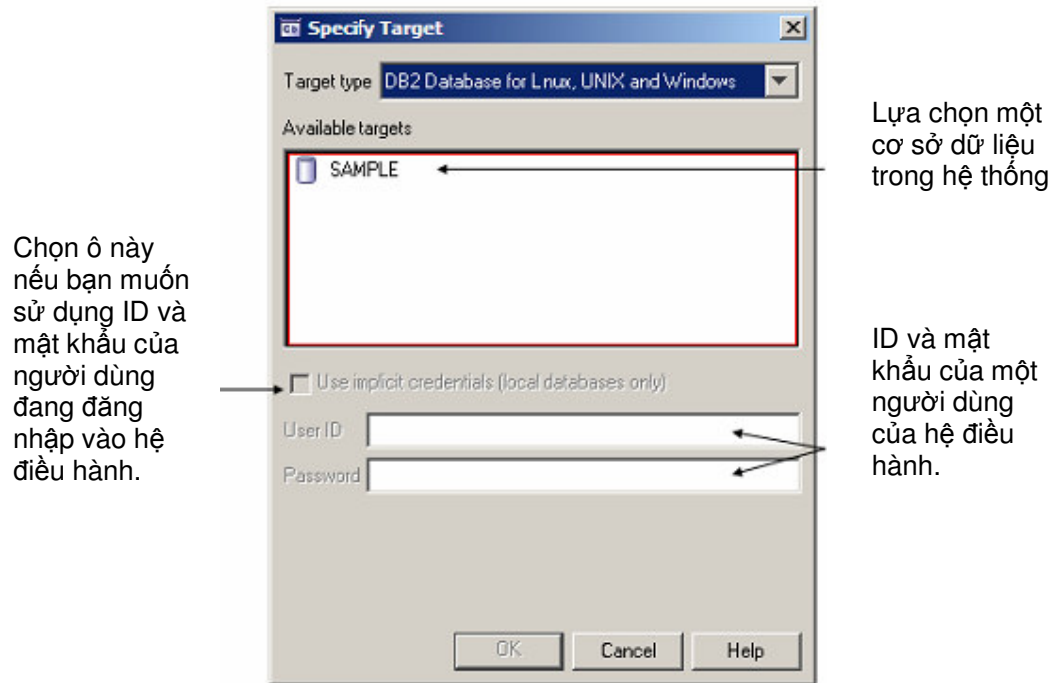
- Từ Menu Start của Windows:  
*Start -> Programs -> IBM DB2 -> Command Line Tools -> Command Editor*
- Từ một cửa sổ nhập lệnh (của windows hoặc linux), gõ *db2ce*
- Từ menu Tools của Trung tâm điều khiển
- Nhúng trong Trung tâm điều khiển:
  - Nhấn phải chuột vào biểu tượng cơ sở dữ liệu SAMPLE trong khung hiển thị các đối tượng của Trung tâm điều khiển sau đó chọn menu Query.
  - Bất kỳ khi nào một đối tượng mà bạn có thể truy vấn, được chọn (cơ sở dữ liệu, bảng..), bạn có thể khởi động Cửa sổ nhập lệnh bằng cách nhấn vào đường dẫn Query trong khung thông tin chi tiết về đối tượng của Trung tâm điều khiển (khung dưới bên tay phải).
- Từ Trung tâm điều khiển, nhấn chọn biểu tượng Cửa sổ nhập lệnh  trên thanh công cụ của Trung tâm điều khiển như trong hình 4.9



**Hình 4.9 - Biểu tượng Cửa sổ nhập lệnh trong Trung tâm điều khiển**

**Thêm một kết nối tới cơ sở dữ liệu**

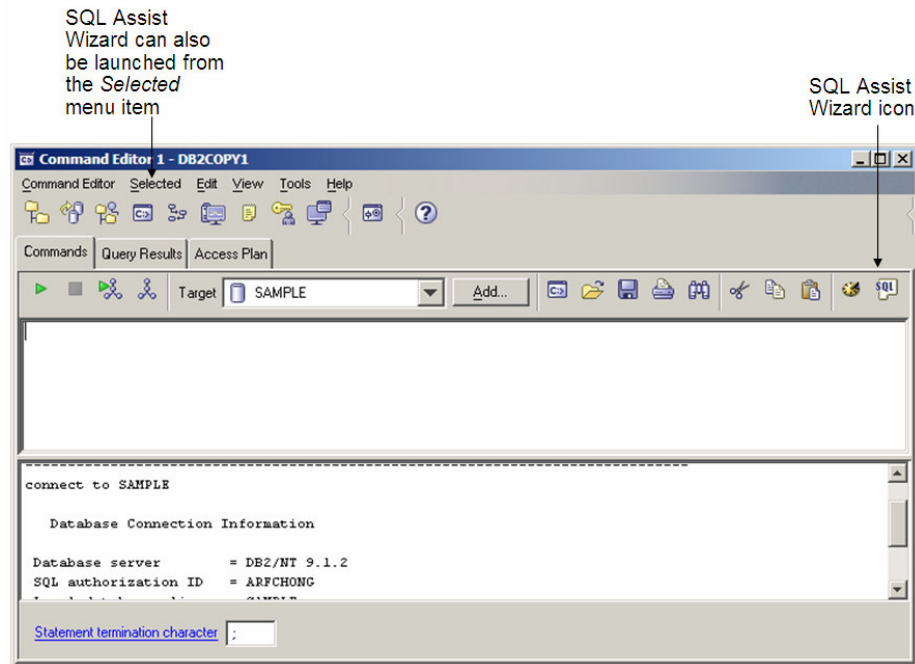
Để thêm một kết nối tới một cơ sở dữ liệu nhấn vào nút Add. Một hộp thoại như trong hình 4.10 sẽ xuất hiện.



Hình 4.10 - Thêm một kết nối cơ sở dữ liệu

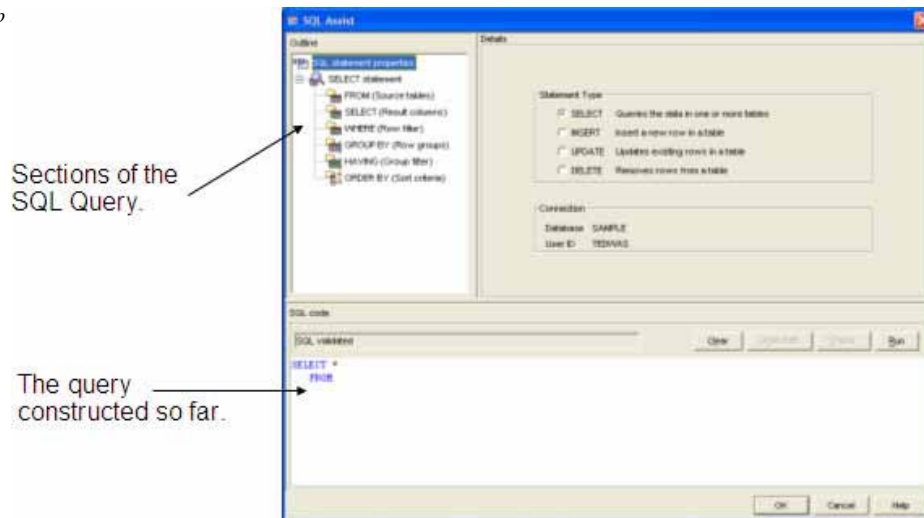
### 4.3. Trình trợ giúp SQL

Nếu bạn không quen thuộc với ngôn ngữ SQL và thích sử dụng một trình hỗ trợ hoặc hướng dẫn tự động sinh mã SQL, lúc này trình trợ giúp “SQL Assist Wizard” là một công cụ sẵn có từ trình soạn thảo dòng câu lệnh (Command Editor) sẽ giúp bạn giải quyết. Như hình 4.11 bên dưới, bạn có thể triệu gọi nó từ Command Editor bằng việc nhấp chuột vào biểu tượng sau với ký hiệu SQL (nổi lên trong hình với vòng tròn đỏ).



**Hình 4.11 – Gọi trình trợ giúp SQL Assist Wizard.**

Hình 4.12 trình bày về trình trợ giúp SQL Assist Wizard. Nó cho phép sử dụng song hành đồng thời cả hai chế độ. Đầu tiên chỉ ra kiểu của câu lệnh SQL bạn hỗ trợ với (SELECT, INSERT, UPDATE, DELETE). Nó còn phụ thuộc vào câu lệnh bạn chọn, các tùy chọn khác nhau sẽ được hiển thị ra. Ở phần cuối cùng của cửa sổ bạn sẽ nhìn thấy làm thế nào để câu lệnh SQL được xây dựng nên giống như bạn chọn những lựa chọn khác trong trình hướng dẫn.



Hình 4.12 – Trình trợ giúp SQL Assist Wizard

#### 4.4 Trình bày nút Show SQL

Hầu hết các công cụ trợ giúp đồ họa (GUI) và các trình hướng dẫn trong DB2 cho phép bạn xem lại câu lệnh hiện thời hoặc câu lệnh SQL đã được thực thi ra kết quả bằng việc sử dụng công cụ hay trình hướng dẫn để thực hiện một công việc. Để xem phần này, nhấp chuột vào nút Show SQL trong công cụ bạn đang làm việc, như hình 4.13 và hình 4.14.



Hình 4.13 – Trình bày nút Show SQL



Hình 4.14 – Đầu ra của một phần trình bày về nút Show SQL.

Khả năng hiển thị lại những câu lệnh SQL và những dòng lệnh là rất có ích trong việc học cú pháp SQL, rút gọn và tái sử dụng những lệnh và những dòng lệnh trong một tệp để sử dụng về sau này. Bạn cũng có thể xây dựng các kịch bản bằng việc dùng lại những lệnh và những dòng lệnh đã được tạo ra này.

#### Bài tập nhanh #2 - Tạo một CSDL mới với Trung tâm điều khiển



##### Mục đích

Trong bài tập nhanh này, bạn sẽ tạo ra một CSDL mới sử dụng trình hướng dẫn tạo CSDL trong Trung tâm điều khiển.

##### Thủ tục thực hiện:

1. Từ ô cây đối tượng của Trung tâm điều khiển, nhấp phải chuột vào thư mục All Databases, chọn danh sách Create Database, và chọn danh sách

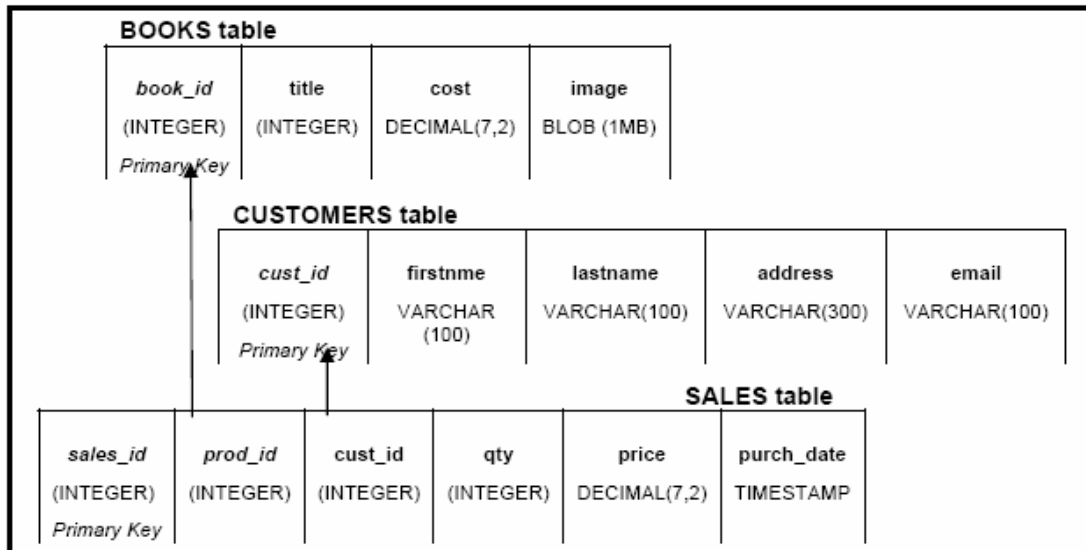
With Automatic Maintenance. Bắt đầu chạy Creat Database Wizard.

2. Đặt tên cho CSDL và vị trí lưu tệp trong trang Name của trình hướng dẫn. Sử dụng các giá trị dưới đây:
  - Tên CSDL: EXPRESS
  - Ổ đĩa mặc định (của Windows): C:
  - Đường dẫn mặc định (của Linux): /home/db2inst1
  - Bí danh: Trường hợp này mặc định là EXPRESS nếu bên trái để trống.
  - Ghi chú: Trường hợp này có thể tùy chọn, bên trái có thể để trống.
 Nhấp chuột vào CSDL có sẵn Enable cho hộp chọn XML, và sau đó nhấp chuột vào nút Next để tiếp tục trang kế của trình hướng dẫn.
3. Trong Specify cần chỉ rõ nơi lưu trang dữ liệu, không thêm bất kỳ sự thay đổi nào, và nhấp Next.
4. Trong trang chọn chiến lược bảo trì, lấy giá trị mặc định (Yes, I can specify an offline...), và nhấp Next.
5. Chỉ định khe thời gian bảo trì trong trang *Timing*. Chỉ rõ hai hoặc hơn 4 giờ một tuần khi DB2 có thể thực hiện tự động duy trì các công việc để giữ việc hoạt động bình thường của CSDL. Bây giờ, cấu hình khe thời gian bắt đầu lúc 1 giờ sáng từ thứ hai tới thứ năm, kéo dài 6 tiếng. Nhấp vào nút Next.
6. Cấu hình thông điệp trên trang Mail Server của trình hướng dẫn. DB2 có thể tự động gửi một email hay một trang nếu có một sự cố hay một điều kiện bất thường được phát hiện thấy. Nếu bạn muốn cài đặt lại điều này, hãy chỉ ra một máy chủ SMTP có sẵn nào đó để DB2 dùng trong việc gửi email. Với bài tập này, chúng ta không có một máy chủ SMTP nào, vì thế cho phép để trống phần này và nhấp nút Next.
7. Xem lại các tùy chọn đã chọn trong trang Summary trong trình hướng dẫn. Nhấp vào nút Finish để bắt đầu quá trình tạo ra CSDL. CSDL được tạo ra thường phải mất một vài phút, trong thời gian này chỉ thị báo tiến trình đang thực hiện sẽ hiện ra.
8. Bây giờ một CSDL đã được tạo, nội trú ngay trong nó là một vài bảng và ít dữ liệu được tạo ra. Để tạo cho bạn sự thuận lợi, hai kịch bản có tên là *quicklab2.db2* và *quicklab2.dat* đã sẵn có để giúp bạn. Kịch bản *quicklab2.db2* chứa những lệnh được dùng để tạo các bảng và do đó phải được chạy đầu tiên. Kịch bản *quicklab2.dat* chứa những câu lệnh chèn dữ liệu vào các bảng. Cả hai kịch bản này có thể tìm thấy trong tệp nén **ex-pressc\_book\_quicklabs.zip** đi cùng cuốn sách này. Để chạy những kịch bản này, sử dụng cửa sổ nhập lệnh.
  - a. Hãy chắc rằng CSDL mới bạn đã tạo được chọn (bôi đen) trong danh sách thả xuống trong thanh công cụ Command Editor. Nếu CSDL mới không hiện lên trong danh sách, sử dụng nút Add để thêm một kết nối tới nó (xem gần hình 4.10)
  - b. Nhấp vào lệnh *Select -> Open* hoặc nhấn nút mở  trong Command Editor và chuyển tới thư mục nơi chứa các kịch bản đã được lưu. Đánh dấu tệp *quicklab2.db2* và nhấp nút OK. Các nội dung của tệp bây giờ được hiển thị trong vùng đầu vào dữ liệu của Command Editor. Chọn nút Run  để chạy kịch bản. Xác

định không có lỗi nào được tìm thấy khi chạy kịch bản này.

9. Lặp lại bước 8 cho tệp *quicklab2.dat*.

CSDL mới này là một ví dụ đơn giản cho một cửa hàng sách trên Internet. Bảng *BOOKS* chứa tất cả thông tin về những cuốn sách được lưu trữ. Bảng *CUSTOMERS* chứa thông tin về mỗi khách hàng của cửa hàng. Cuối cùng bảng *SALES* chứa dữ liệu về việc bán hàng. Bất kỳ lúc nào khách hàng đặt mua một cuốn sách, một bản ghi được tạo ra trong bảng *SALES*. Sơ đồ dưới đây trình bày thiết kế và quan hệ giữa các bảng.



**4.5 Tạo kịch bản**

Nó rất hữu ích để cho phép tạo ra những tệp kịch bản, những tệp thực thi những lệnh DB2 hoặc những câu lệnh SQL lặp lại nhiều lần. Ví dụ, một người quản trị (DBA) có thể muốn chạy một kịch bản đưa ra hàng ngày để kiểm tra số hàng của những bảng quan trọng.

Có hai dạng kịch bản thông thường:

1. Các kịch bản SQL.
2. Các kịch bản (vô) hệ điều hành.

**4.5.1 Các kịch bản SQL**

Các kịch bản SQL bao gồm các câu lệnh truy vấn và các lệnh CSDL. Các kịch bản này là mối quan hệ đơn giản để hiểu và là nền độc lập. Tuy nhiên, các biến hay các tham số không được hỗ trợ.

Cho ví dụ, các lệnh dưới đây được lưu lại trong tệp có tên script1.db2.

```
CONNECT TO EXPRESS;
CREATE TABLE user1.mytable
  ( col1 INTEGER NOT NULL,
    col2 VARCHAR(40),
    col3 DECIMAL(9,2) );
SELECT * FROM user1.mytable FETCH FIRST 10 ROWS ONLY;
```

```
COMMIT;
```

### Tệp script1.db2

Trong phần kịch bản trên, tất cả các câu lệnh là các câu lệnh SQL, và mỗi câu lệnh được cách nhau bởi một dấu phân cách và trong trường hợp này là dấu chấm phẩy. Tên tệp không nhất thiết sử dụng đuôi mở rộng là “db2”. Bất kỳ phần mở rộng nào cũng dùng được.

#### Thực thi các kịch bản SQL:

Một kịch bản SQL có thể được thực thi bằng nhiều cách khác nhau có thể từ cửa sổ lệnh hoặc DB2 Command Window trên Windows, hoặc thông qua Linux shell. Để chạy kịch bản từ DB2 Command Window hoặc Linux shell, bạn có thể dùng các lệnh dưới đây:

```
db2 -t -v -f script1.db2 -z script1.log hoặc,
db2 -tvf script1.db2 -z script1.log
```

Trong câu lệnh này:

- t Chỉ ra các câu lệnh dùng dấu kết thúc mặc định (dấu chấm phẩy).
- v Chế độ “lắm lời”, để db2 hiển thị các lệnh khi thực hiện.
- f Chỉ ra rằng tên tệp sau cờ này là tệp kịch bản.
- z Chỉ ra rằng theo sau thông báo tên tệp sẽ được sử dụng để hiện giá trị đầu ra cho việc phân tích sau này (đây là chọn, nhưng nên làm).

Khi cờ hiệu **-t** được dùng và không có dấu phân cách dòng nào được chỉ rõ, dấu chấm phẩy sẽ được chọn. Có một số hoàn cảnh mà ta cần đến một dấu phân cách khác. Ví dụ một kịch bản chứa mã SQL PL cần dùng một dấu phân cách câu lệnh khác với ký tự mặc định (dấu chấm phẩy), bởi vì những dấu chấm phẩy này đã được dùng bên trong các định nghĩa đối tượng SQL PL để phân cách các câu lệnh thuộc về thủ tục.

Ví dụ, trong tệp kịch bản ở dưới đây có tên “functions.db2”, nó chứa ngôn ngữ định nghĩa dữ liệu DDL để tạo một hàm, một dấu chấm phẩy là cần thiết ở cuối của cú pháp SELECT ở bên trong hàm. Với câu lệnh CREATE FUNCTION dấu phân cách chúng ta phải sử dụng một dấu chấm than (!). Nếu chúng ta lại dùng một dấu chấm phẩy vào việc ngăn cách câu lệnh, khi chạy sẽ xảy ra xung đột từ kịch bản, kết quả là có một lỗi trong thông báo từ DB2.

```
CREATE FUNCTION f1() SELECT
...      ;
... END!
```

### Tệp functions.db2

Để cho DB2 biết rằng một ký tự dấu phân cách khác được dùng, sử dụng cờ hiệu **-d**, tiếp sau là ký tự phân cách mong muốn (dấu !) như trình bày dưới đây:

```
db2 -td! -v -f functions.db2 -z functions.log
```

Để có các mô tả về các cờ hiệu khác bằng cách chạy câu lệnh dưới đây trong Command Window hoặc Linux shell:

```
db2 list command option
```



#### 4.5.2 Các kịch bản với hệ điều hành

Các kịch bản với hệ điều hành cung cấp độ linh hoạt và tính năng mạnh mẽ hơn, như cho ta khả năng thêm vào những đoạn mã lập trình có tính logic. Những đoạn kịch bản này phụ thuộc vào hệ điều hành nền, tuy nhiên chúng có hỗ trợ các tham số và các biến. Dưới đây là ví dụ của một kịch bản trong hệ điều hành Windows.

```
set DBPATH=c;
set DBNAME=PRODEXPR
set MEMORY=25
db2 CREATE DATABASE %DBNAME% ON %DBPATH% AUTOCONFIGURE USING
MEM_PERCENT %MEMORY% APPLY DB AND DBM
db2 CONNECT TO %DBNAME% USER %1 USING %2
del schema.log triggers.log app_objects.log
db2 set schema user1
db2 -t -v -f schema.db2 -z schema.log
db2 -td@ -v -f triggers.db2 -z triggers.log
db2 -td@ -v -f functions.db2 -z functions.log
```

#### Tệp create\_database.bat

Để thực hiện kịch bản với hệ điều hành này từ dòng lệnh, bạn gõ lệnh sau trong Windows:

```
create_database.bat db2admin ibmdb2
```

Trong Windows sử dụng phần mở rộng “bat” để nói cho hệ điều hành biết rằng đây là một tập tin thực thi xử lý theo lô.

Trong Linux, bạn cần thay đổi kiểu trên tập tin để chỉ định rằng tập tin này là một tập tin thực thi bằng cách sử dụng lệnh như chmod +x. Sau đó, bạn có thể chạy tập tin này với cách thức giống như ở phần trên.

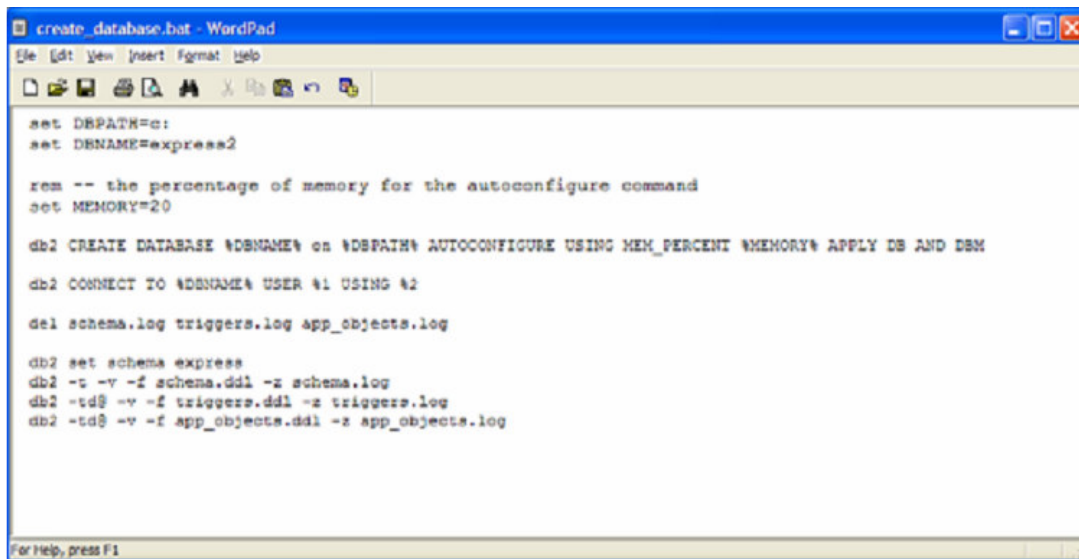
### Bài tập nhanh #3 – Tạo một kịch bản cài đặt cho cơ sở dữ liệu EXPRESS

#### Mục tiêu

Các kịch bản là một cơ chế mạnh mẽ trong việc thực hiện những công việc có tính chất lặp đi lặp lại như tập hợp thông kê cho cơ sở dữ liệu, sao lưu, và triển khai cơ sở dữ liệu. Các kịch bản hệ điều hành có lợi điểm của việc hỗ trợ các tham số kịch bản, điều này giúp cho chúng linh hoạt hơn. Trong bài tập nhanh này, bạn sẽ tạo ra một kịch bản hệ điều hành để triển khai cơ sở dữ liệu EXPRESS. Các kịch bản này sẽ gọi kịch bản với SQL đã được tạo ra trước đó cho các đối tượng cơ sở dữ liệu. Nhằm mục đích tiết kiệm không gian, bài tập nhanh này chỉ hiển thị những kịch bản và câu lệnh dành riêng cho nền Windows. Nếu bạn muốn làm việc trên Linux, hãy đảm bảo rằng bạn thực hiện những thay đổi phù hợp với những sự hướng dẫn sau đây.

#### Thủ tục

1. Mở một chương trình soạn thảo văn bản, như Notepad hay Wordpad và nhập vào thông tin được hiển thị dưới đây.



```

set DBPATH=c:
set DBNAME=express2

rem -- the percentage of memory for the autoconfigure command
set MEMORY=20

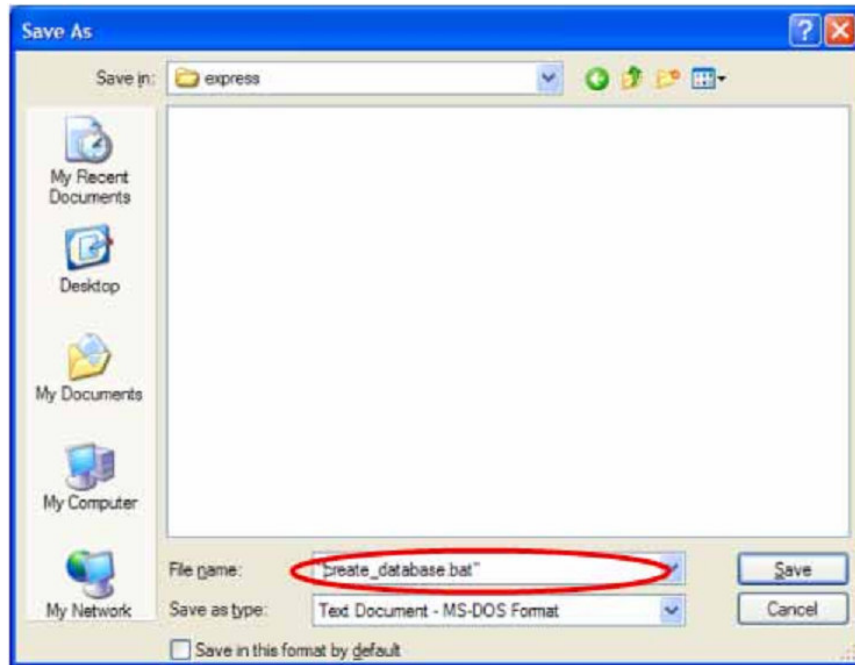
db2 CREATE DATABASE %DBNAME% on %DBPATH% AUTOCONFIGURE USING MEM_PERCENT %MEMORY% APPLY DB AND DSM

db2 CONNECT TO %DBNAME% USER %1 USING %2

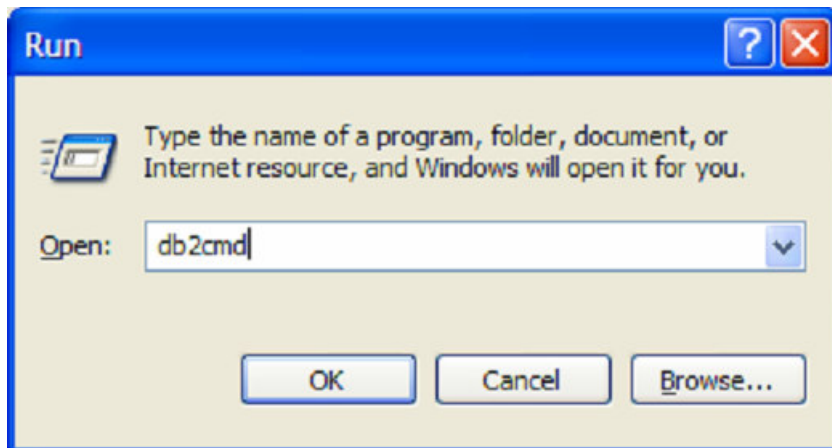
del schema.log triggers.log app_objects.log

db2 set schema express
db2 -t -v -f schema.ddl -z schema.log
db2 -td@ -v -f triggers.ddl -z triggers.log
db2 -td@ -v -f app_objects.ddl -z app_objects.log
    
```

2. Lưu lại tập tin kịch bản trong một thư mục và đặt tên cho tập tin này là create\_database.bat. Trong hộp thoại cửa sổ *Save As*, đảm bảo rằng bạn đã chọn mục tùy chọn *MS-DOS Format* (trong Wordpad). Nếu bạn lưu tập tin này dưới một định dạng khác, Wordpad có thể sẽ tạo thêm một số ký tự ẩn và việc này có thể gây ra những vấn đề trong khi quá trình thực thi của kịch bản. Thêm vào đó, hãy đặt dấu ngoặc kép bao lấy tên của tập tin để đảm bảo rằng Windows không tự động thêm phần mở rộng .TXT cho tập tin như hình vẽ dưới đây.



3. Để chạy kịch bản tương tác với DB2, bạn phải có một môi trường dòng lệnh DB2. Để mở cửa sổ lệnh DB2, hãy vào *Start -> Program Files -> IBM DB2 -> DB2COPY1 (default) -> Command Line Tools -> Command Window*.  
 Cách khác, bạn có thể sử dụng *Start -> Run*, gõ vào *db2cmd* và ấn *enter* như hình vẽ dưới đây



4. Sau đó để chạy kịch bản, nhập vào các lệnh sau:

```
cd C:\express
create_database.bat db2admin ibmdb2
```

```

C:\>cd c:\express
C:\express>create_database.bat db2admin ibndb2_
    
```

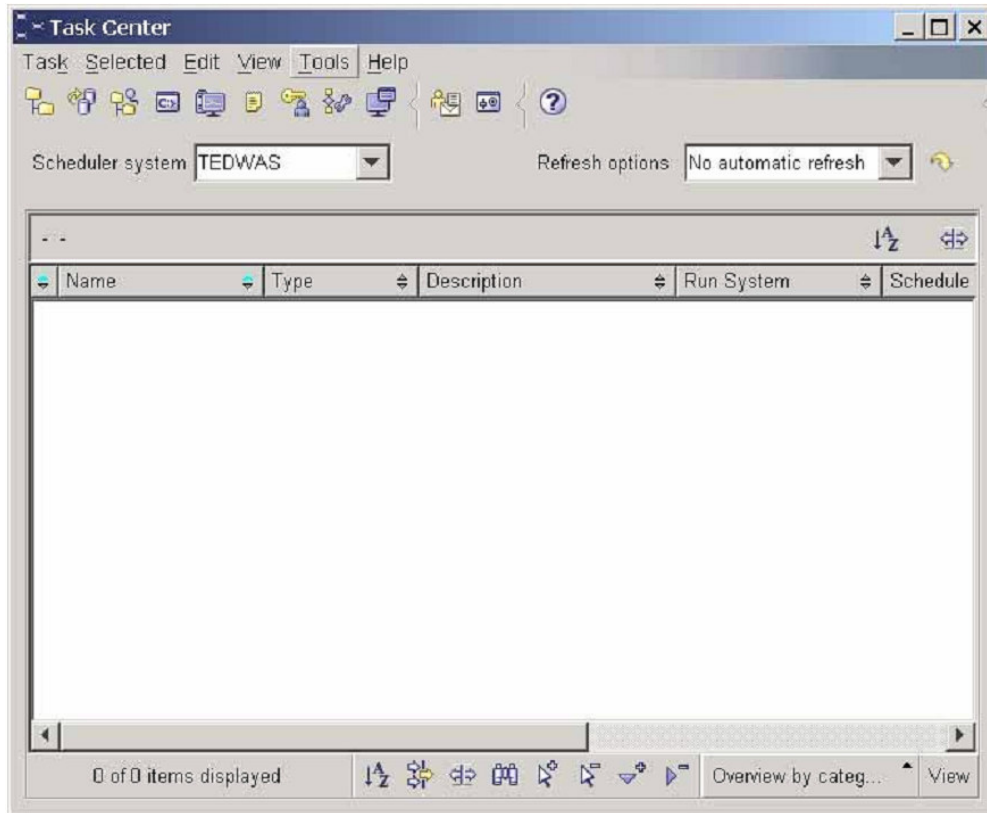
5. Bạn hãy dành chút thời gian để làm quen với kịch bản mà bạn vừa tạo xong. Bạn có hiểu những gì đang xảy ra tại từng dòng không?

6. Hãy cố gắng trả lời những câu hỏi sau đây:

- a. Kết nối cơ sở dữ liệu được thiết lập ở đâu?
- b. Các ký hiệu %1 và %2 nghĩa là gì?
- c. Dòng lệnh sau đây dùng để làm gì? Nó được sử dụng khi nào? Cho mục đích gì?  
SET DBPATH=C:
- d. Dòng lệnh sau đây được dùng để làm gì?  
del schema.log, triggers.log, app\_objects.log
- e. Điều gì sẽ xảy ra khi kịch bản được gọi mà lại không có bất cứ tham số nào?
- f. Tại sao những kịch bản SQL được gọi lại không chứa các câu lệnh CONNECT TO? Chúng kết nối đến cơ sở dữ liệu như thế nào?

### 4.6 Trung tâm tác vụ (Task Center)

Công cụ Trung tâm tác vụ với giao diện đồ họa người dùng cho phép bạn tạo ra những tác vụ: là một tập hợp những hoạt động như chạy các lệnh DB2, các lệnh của hệ điều hành, hay những kịch bản. Các hành động ở phía sau có thể được thực hiện ngay cả khi tác vụ trước thất bại hay thành công. Ví dụ, nếu một tác vụ bao gồm công việc sao lưu một cơ sở dữ liệu quan trọng vào lúc 3h sáng được thực hiện thành công, một email sẽ được gửi đến cho DBA để cung cấp thông tin này. Mặt khác, nếu tác vụ sao lưu thất bại thì Trung tâm tác vụ sẽ đánh dấu trang (page) cho người quản trị biết. Hình 4.15 minh họa Trung tâm Tác vụ



Hình 4.15 – Trung tâm Tác vụ

#### 4.6.1 Cơ sở dữ liệu Tools Catalog

Tất cả các thông tin chi tiết về các tác vụ của bạn và lịch thực hiện tác vụ đó được lưu trữ trong một cơ sở dữ liệu DB2 riêng biệt gọi là cơ sở dữ liệu Tools Catalog. Cơ sở dữ liệu này phải được tồn tại sẵn để có thể lên lịch cho các tác vụ. Để tạo một cơ sở dữ liệu Tools Catalog bạn có thể sử dụng lệnh này:

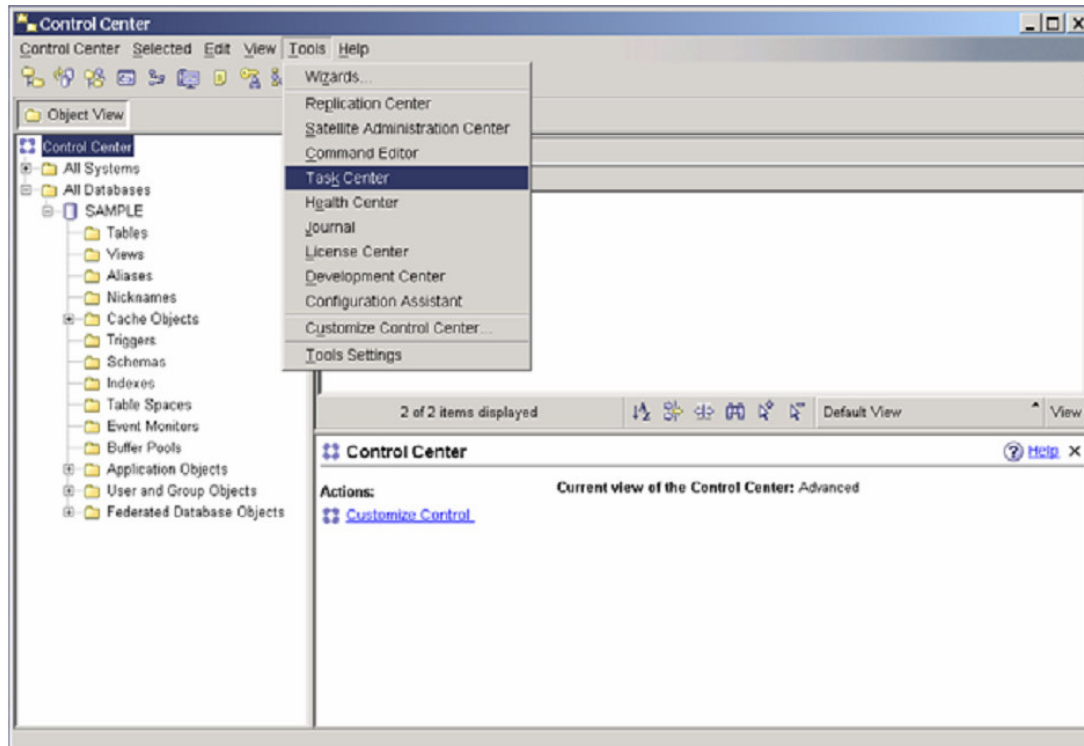
```
CREATE TOOLS CATALOG systools CREATE NEW DATABASE toolsdb
```

Trong ví dụ ở trên, systools là tên sơ đồ (schema name) của tất cả các bảng trong cơ sở dữ liệu, và tên của cơ sở dữ liệu là toolsdb. Chúng ta sẽ nói nhiều hơn về các sơ đồ (schemas) trong chương 8 - Làm việc với các đối tượng cơ sở dữ liệu.

#### Khởi động Trung tâm Tác vụ

Bạn có thể khởi động Trung tâm Tác vụ từ Trung tâm Điều khiển (Control Center) bằng cách nhấn vào *Tools* -> *Task Center* như minh họa trong hình 4.16. Một cách khác, bạn

có thể bắt đầu công cụ này từ menu Start của Windows: *Start -> Programs -> IBM DB -> DB2COPY1 -> General Administration Tools -> Task Center*



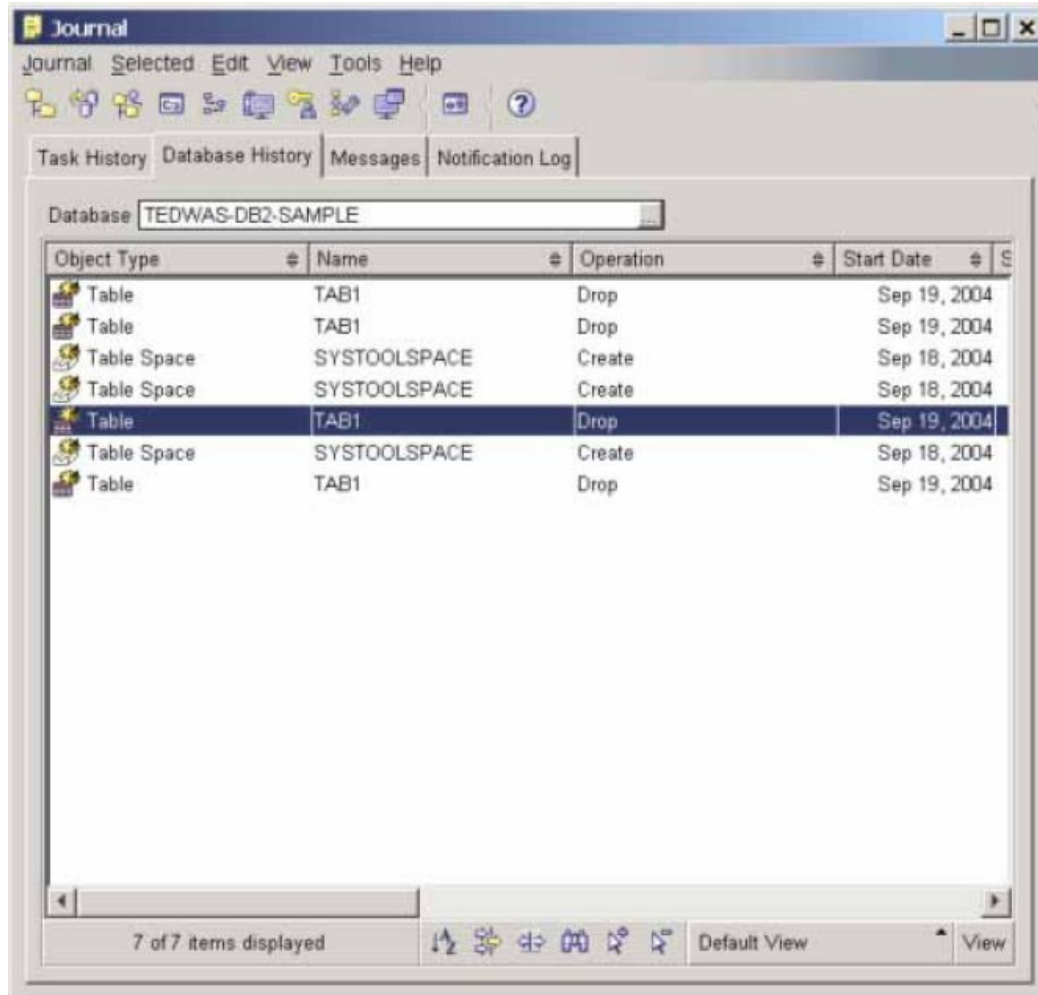
Hình 4.16 – Khởi động Trung tâm Tác vụ

#### Lên lịch với Trung tâm Tác vụ

Bất cứ loại kịch bản nào cũng có thể được lên lịch bằng cách sử dụng Trung tâm Tác vụ (kịch bản đó có thể được hay không được tạo thông qua một công cụ của DB2 có giao diện đồ họa người dùng). Các tác vụ được chạy tại thời điểm đã lên lịch từ hệ thống nơi mà bạn đã tạo cơ sở dữ liệu Tools Catalog. Chúng tôi khuyến khích bạn tự mình khám phá Trung tâm Tác vụ. Việc tạo một tác vụ khá đơn giản.

#### 4.7 Nhật ký

Công cụ Nhật ký có giao diện đồ họa của DB2 cung cấp cho người quản trị cơ sở dữ liệu một nhật ký về các hoạt động trên biểu mẫu trực tuyến (online form). Hình 4.17 mô tả về Nhật ký trong DB2 và Bảng 4.2 mô tả những thông tin mà bạn có thể thu được từ Nhật ký.



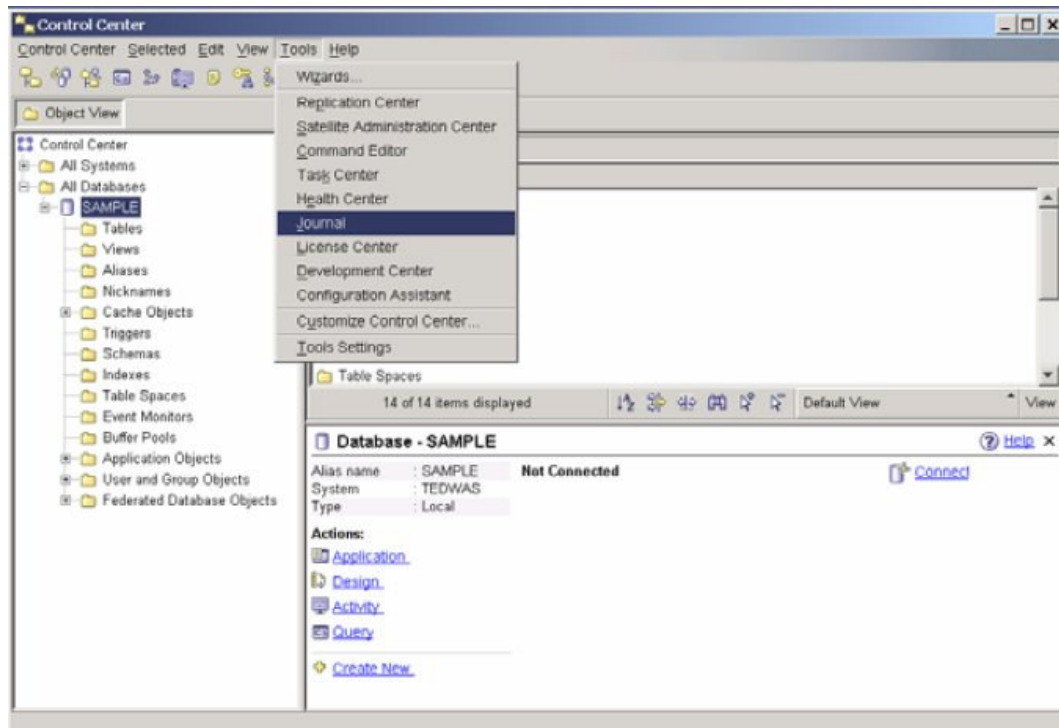
Hình 4.17 – Nhật ký

Kiểu thông tin	Mô tả
Lịch sử tác vụ (Task History)	Tất cả các công việc được lên lịch đã được thực hiện và tình trạng hoạt động của chúng
Lịch sử của cơ sở dữ liệu (Database History)	Bản ghi nhớ về các tác động lên cơ sở dữ liệu (sao lưu, phục hồi, tổ chức lại (REORG), v.v...)
Thông điệp	Nhật ký của các thông điệp được phản hồi từ các công cụ của DB2. Điều này rất hữu ích nếu bạn muốn triệu hồi và đối chiếu các thông điệp lỗi cũ, hoặc nếu bạn đóng hộp thoại quá nhanh hay do trục trặc.
Bản ghi các thông báo	Lưu trữ thông điệp cấp hệ thống. Các lỗi nghiêm trọng được ghi lại ở đây

Bảng 4.2 – Các thông tin được mô tả trong Nhật ký.

**Khởi chạy Nhật ký**

Bạn có thể khởi động Nhật ký từ Trung tâm điều khiển (Control Center) bằng cách bấm chuột vào *Tools -> Journal*, như được mô tả trong Hình 4.18. Bằng cách khác, bạn có thể chạy công cụ này từ trình đơn Start của Windows: *Start -> Programs -> IBM DB2-> DB2COPY1 -> General Administration Tools -> Journal*.



Hình 4.18 – Khởi động Nhật kí.

#### 4.8 Giám sát tình trạng (health)

Giám sát tình trạng là một tác nhân mặc định chạy bên trong bộ máy DB2, giám sát tất cả các mặt của tình trạng cơ sở dữ liệu (bộ nhớ, quản lý không gian, các hoạt động được tự động hóa đã được định rõ trước...). Khi mà một số tính năng của DB2 được thực thi bên ngoài các tham số đã thiết lập, sẽ xuất hiện ngoại lệ, người quản trị cơ sở dữ liệu sẽ được lưu ý tới. Có 3 kiểu cảnh báo:

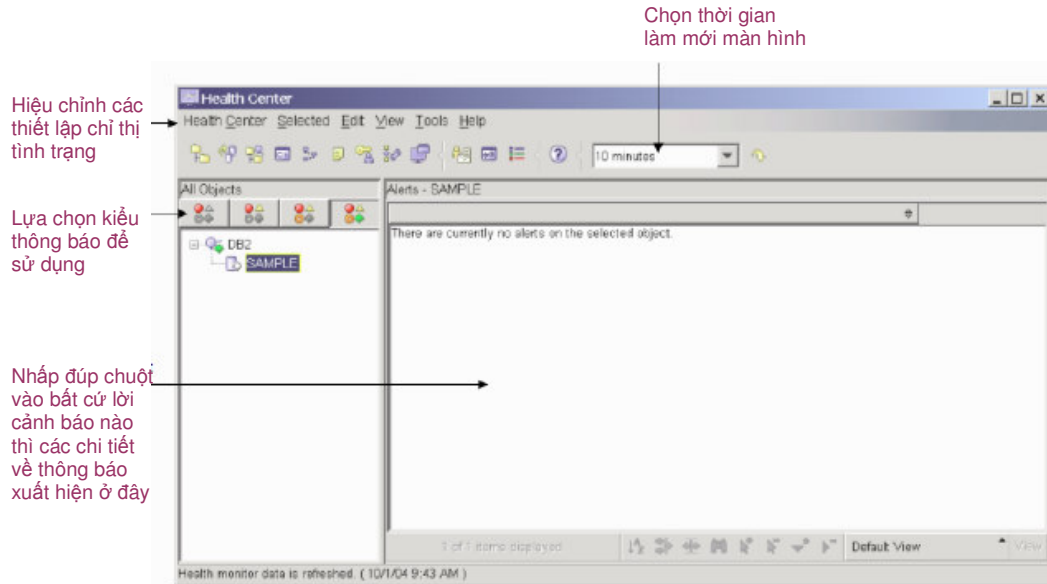
- Chú ý: Tình trạng không bình thường.
- Cảnh báo: Tình trạng chưa nghiêm trọng, không yêu cầu sự quan tâm ngay lập tức nhưng có thể báo hiệu hệ thống không còn trong tình trạng tối ưu.
- Báo động: Tình trạng nghiêm trọng yêu cầu có các hành động can thiệp ngay lập tức.

Giám sát hệ thống có thể bật hoặc tắt sử dụng tham số cấu hình quản lý cơ sở dữ liệu HEALTH\_MON.

##### 4.8.1 Trung tâm tình trạng

Trung tâm tình trạng là một công cụ đồ họa để tương tác với tác nhân Giám sát tình trạng. Công cụ Trung tâm tình trạng phân tích cảnh báo tình trạng trong hệ thống đối với từng thể hiện (instance), cơ sở dữ liệu, và không gian bảng. Hình 4.19 mô tả về công cụ Trung tâm tình trạng.

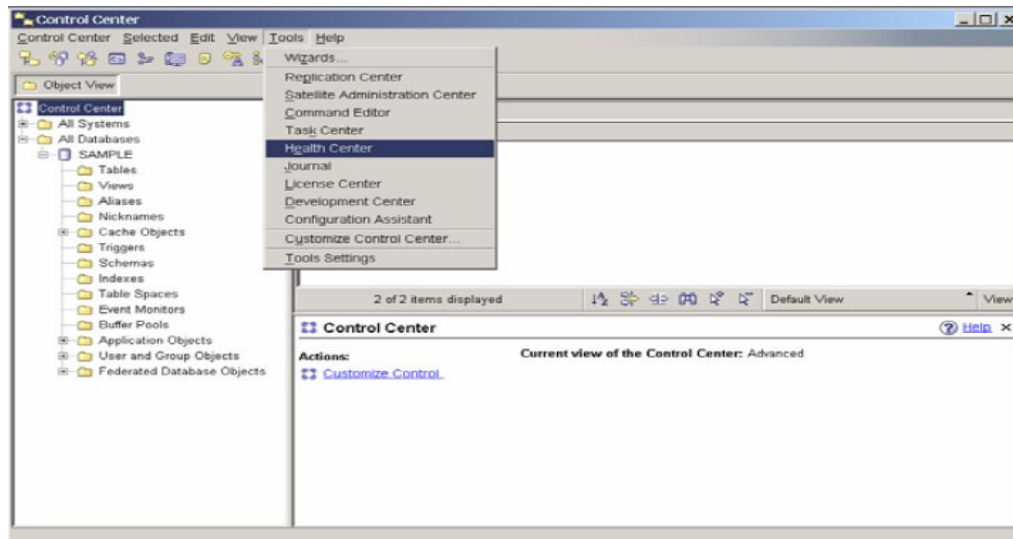




Hình 4.19 – Công cụ Trung tâm tình trạng

**Khởi chạy Trung tâm tình trạng.**

Bạn có thể khởi động Trung tâm tình trạng từ Control Center bằng cách nhấn vào Tools Menu và chọn Health Center. Điều này được biểu diễn qua Hình 4.20. Bạn cũng có thể chạy công cụ này từ *Start -> Programs -> IBM DB2 -> DB2COPY1 -> Monitoring Tools -> Health Center*.



Hình 4.20 – Khởi chạy Trung tâm sức khỏe

**Thiết lập cảnh báo tình trạng**

Một khi Trung tâm tình trạng của bạn đã được khởi động, bạn có thể thiết lập các khai báo cho lời cảnh báo (Alert notification) bằng cách nhấn vào *Health Center menu -> Configure -> Alert Notification* như trong Hình 4.21. Cảnh báo tình trạng cho phép bạn nhập vào danh bạ với địa chỉ email hay số máy nhắn tin của những người cần phải liên lạc nếu xuất hiện cảnh báo.

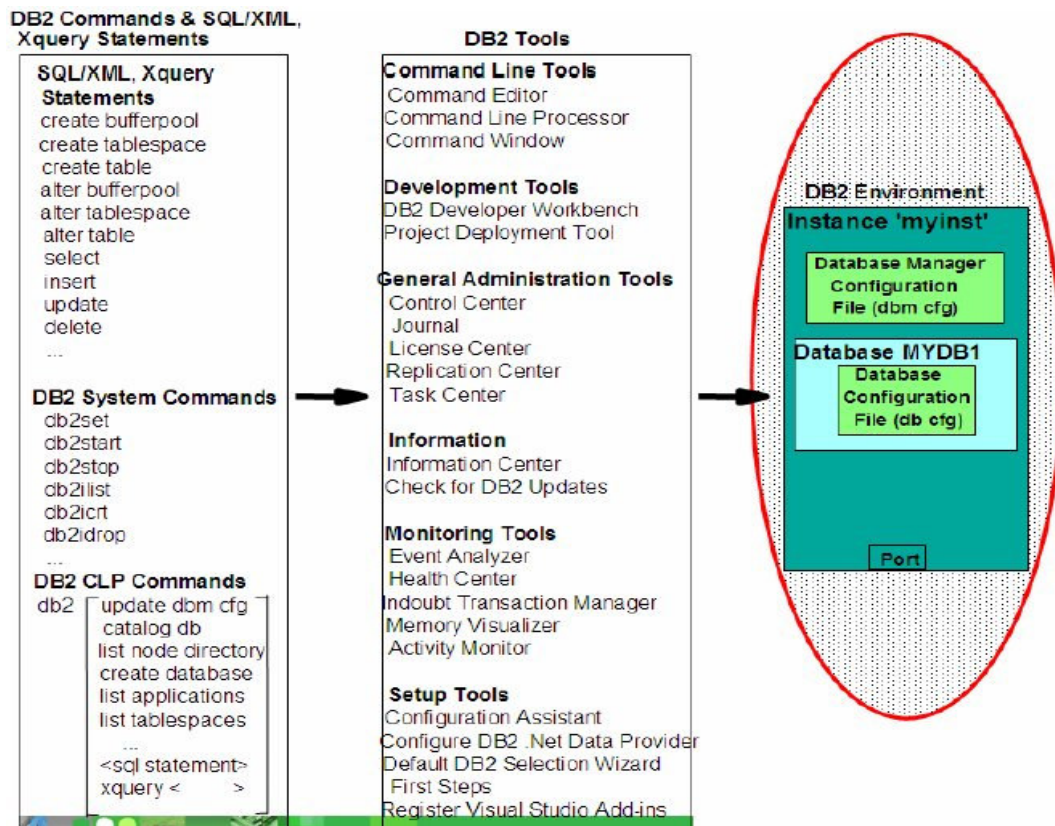


Hình 4.21 – Khai báo cảnh báo

# 5

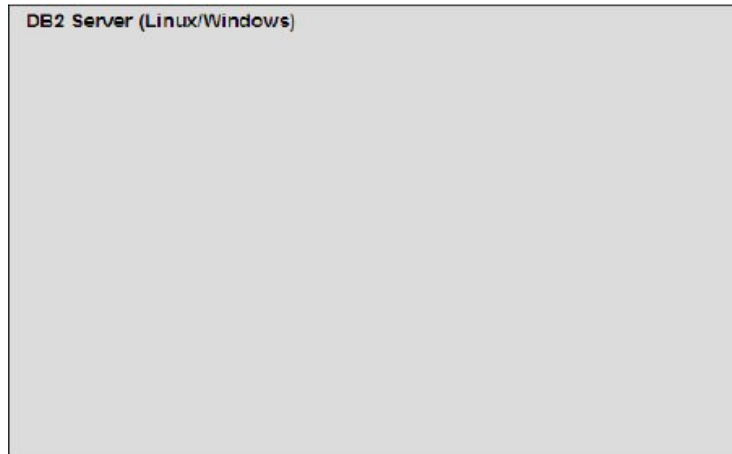
## Chương 5 - Môi trường của DB2

Trong chương này, chúng ta sẽ thảo luận về Môi trường DB2: các thể hiện, cơ sở dữ liệu và các file cấu hình. Hình 5.1 mô tả các vấn đề bao bởi e-líp màu đỏ sẽ được chúng ta thảo luận trong chương này.



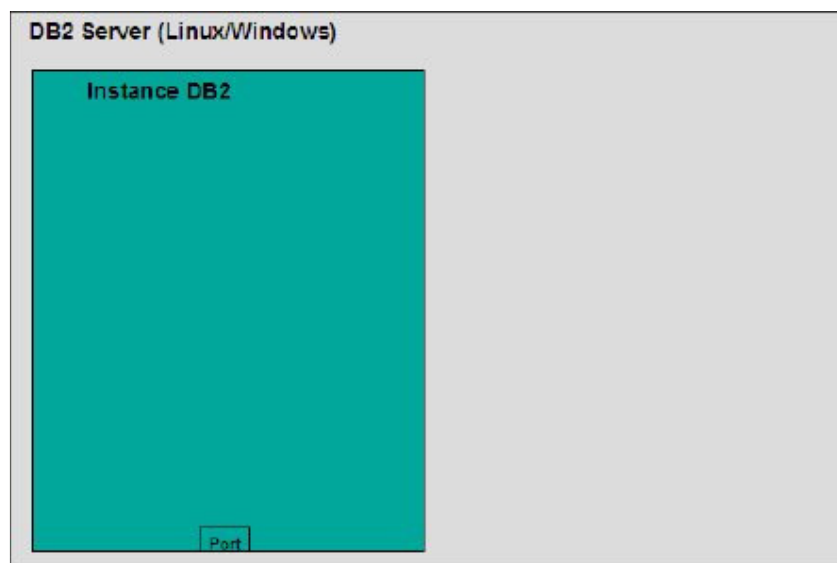
Hình 5.1 - Bức tranh tổng thể về DB2: Môi trường DB2

Để mô tả môi trường DB2, hãy mô tả dần dần từng phần tử thành phần. Hình 5.2 cho thấy hình tượng về máy chủ dữ liệu DB2 sau khi cài đặt DB2 Express-C 9



Hình 5.2 – Hình mẫu về máy chủ dữ liệu DB2 sau khi cài đặt DB2 Express-C 9

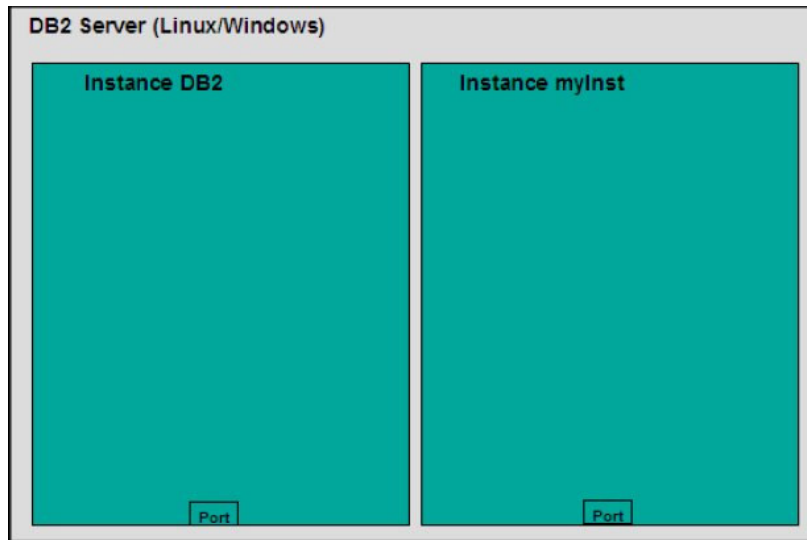
Đây là một phần của quá trình cài đặt trong môi trường Windows, tạo ra một thể hiện mặc định gọi là *DB2* (hay “**db2inst1**” trên môi trường Linux). Điều này được mô tả bằng một hộp màu xanh trong Hình 5.3. Thể hiện này đơn giản là một môi trường độc lập trong đó có thể chạy các ứng dụng và tạo ra các cơ sở dữ liệu. Bạn có thể tạo nhiều thể hiện trên cùng một máy chủ dữ liệu và sử dụng chúng cho các mục đích khác nhau. Ví dụ, một thể hiện có thể được dùng để lưu trữ cơ sở dữ liệu cho sản phẩm, thể hiện khác được dùng để kiểm tra môi trường cơ sở dữ liệu, và một cái khác dùng cho môi trường phát triển. Tất cả các thể hiện này là độc lập với nhau; nghĩa là, các thao tác thực hiện trên một thể hiện sẽ không làm ảnh hưởng tới các thể hiện còn lại.



Hình 5.3 – Thể hiện mặc định DB2 được tạo ra.

Để tạo một thể hiện DB2 mới, sử dụng câu lệnh `bd2icrt <instance name>`, chỗ `<instance name>` được thay thế bởi bất cứ tên nào gồm 8 kí tự. Ví dụ, để tạo ra thể hiện *myinst*, ta sử dụng câu lệnh: `bd2icrt myinst`.

Hình 5.4 cho thấy một thể hiện mới tên *myinst* như một hộp màu xanh riêng biệt



**Hình 5.4 – Một máy chủ DB2 với 2 thể hiện.**

Chú ý rằng mỗi thể hiện có một địa chỉ cổng (port) riêng. Điều này giúp phân biệt giữa các thể hiện khi mà bạn muốn kết nối tới một cơ sở dữ liệu trong một thể hiện nhất định từ một máy khách ở xa. Nếu bạn sử dụng cửa sổ dòng lệnh DB2 (DB2 Command Window), bạn có thể làm bất cứ thể hiện nào của DB2 trở thành một thể hiện hoạt động bằng cách sử dụng một lệnh hệ điều hành trên môi trường Windows:

```
set db2instance = myinst
```

Trong ví dụ này, nếu bạn tạo một cơ sở dữ liệu từ Cửa sổ dòng lệnh, nó sẽ được tạo trong thể hiện **myinst**.

Để liệt kê các thể hiện, gõ lệnh:

```
db2ilist
```

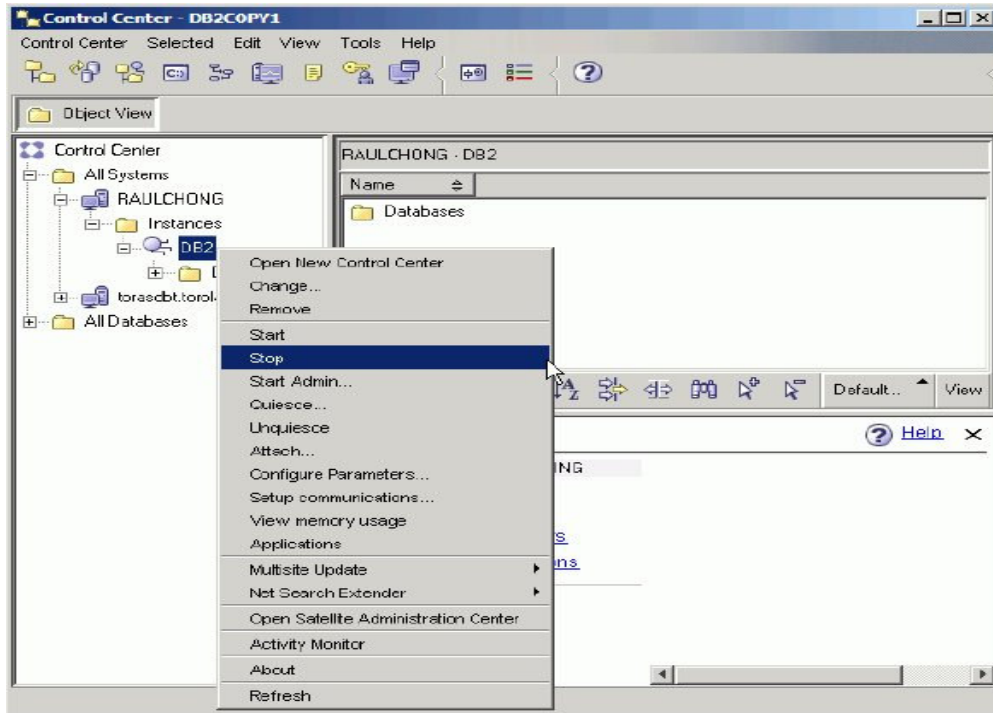
Trên Linux, một thể hiện phải gắn liền với một người dùng trên Linux; do đó, để chuyển đổi giữa các thể hiện, bạn chỉ cần chuyển đổi người dùng (với câu lệnh `su`).

Bảng 5.1 mô tả vài lệnh hữu ích trên thể hiện

Câu lệnh	Mô tả
db2start	Khởi động thể hiện hiện tại
db2stop	Ngừng thể hiện hiện tại
db2icrt	Tạo một thể hiện mới
db2idrop	Loại bỏ một thể hiện
db2ilist	Liệt kê các thể hiện bạn có trong hệ thống
db2 get instance	Liệt kê các thể hiện hoạt động hiện tại

**Bảng 5.1 – Các lệnh DB2 hữu dụng trên thể hiện.**

Vài câu lệnh ở trên có thể thực hiện thay thế thông qua trung tâm điều khiển (Control Center). Ví dụ, trong cửa sổ trung tâm điều khiển, nếu bạn mở rộng thư mục *Instances* và nhấn chuột phải vào thể hiện mong muốn, bạn có thể chọn *Start* cũng giống như sử dụng lệnh `db2start` từ cửa sổ dòng lệnh của DB2, hay *Stop*, cũng như sử dụng lệnh `db2stop` như trong hình 5.5



**Hình 5.5 – Các lệnh trên thẻ hiện trong cửa sổ Control Center**

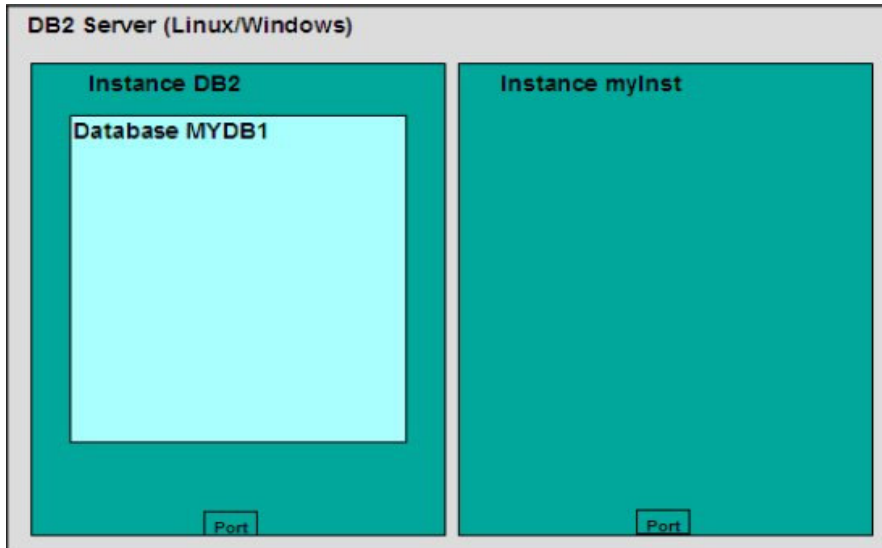
Để tạo một cơ sở dữ liệu trong một thẻ hiện hoạt động, thực hiện lệnh này từ cửa sổ dòng lệnh DB2:

```
db2 create database mydb1
```

Để liệt kê tất cả các cơ sở dữ liệu được tạo, chạy lệnh:

```
db2 list db directory
```

Bên trong bất kì thẻ hiện nào, bạn có thể tạo nhiều cơ sở dữ liệu. Một cơ sở dữ liệu là một tập hợp của các đối tượng như là các bảng, các khung nhìn, chỉ mục và nhiều thứ khác. Các cơ sở dữ liệu là các chủ thể độc lập, và do vậy, không chia sẻ các đối tượng với các cơ sở dữ liệu khác. Hình 5.6 mô tả một cách hình tượng cơ sở dữ liệu “MYDB1” được tạo ra bên trong thẻ hiện “DB2”.



Hình 5.6 – Cơ sở dữ liệu “MYDB1” được tạo trong thể hiện “DB2”.

Bảng 5.2 Một số lệnh bạn có thể sử dụng ở lớp cơ sở dữ liệu.

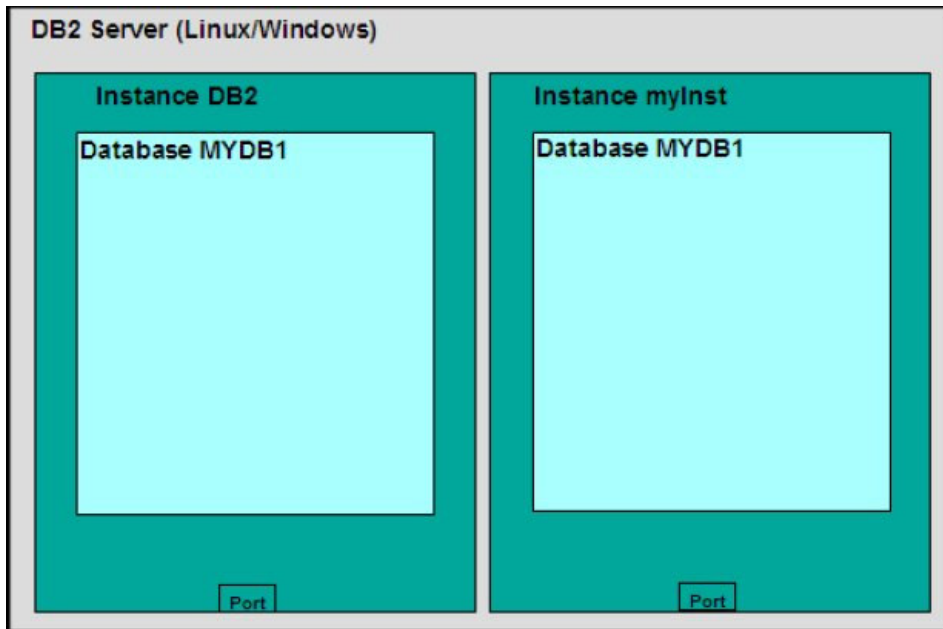
Lệnh/ Câu lệnh SQL	Mô tả
db2 create database	Tạo một cơ sở dữ liệu mới
db2 drop database	Loại bỏ một cơ sở dữ liệu
db2 connect to <database_name>	Kết nối tới cơ sở dữ liệu
db2 create table / create view / create index	Câu lệnh SQL để tạo ra bảng biểu, khung nhìn, và chỉ mục tương ứng

Bảng 5.2 – Các lệnh/câu lệnh SQL trên cơ sở dữ liệu.

Nếu muốn tạo một cơ sở dữ liệu khác cũng có trùng tên (MYDB1) nhưng trong thể hiện “myinst”, thực hiện các lệnh sau đây trên cửa sổ dòng lệnh của DB2

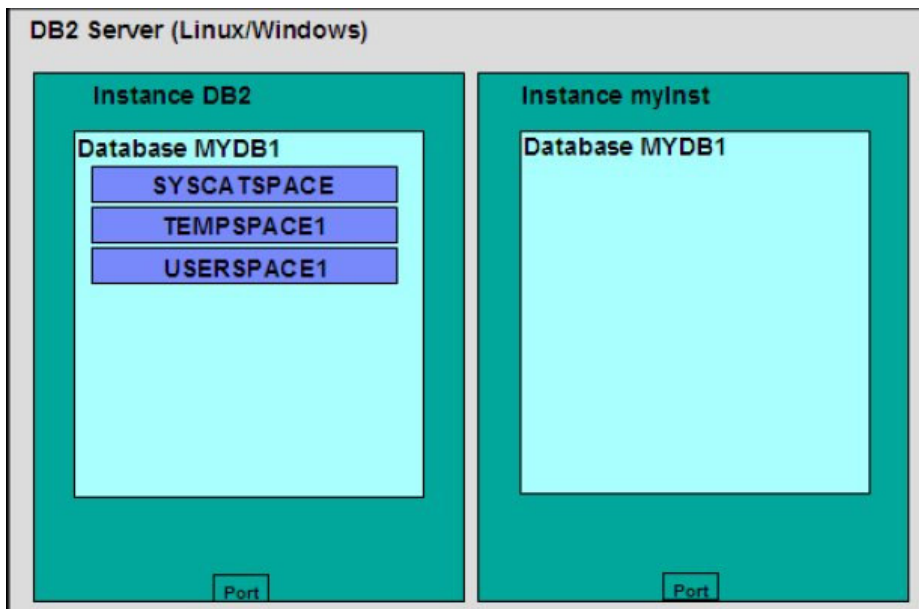
```
db2 list db directory
set db2instance = myinst
db2 create database mydb1
set db2instance = db2
```

Hình 5.7 mô tả cơ sở dữ liệu mới “MYDB1” được tạo trong thể hiện “myinst”.



Hình 5.7 – Cơ sở dữ liệu “MYDB1” trong khung nhìn “myInst”.

Khi một cơ sở dữ liệu được tạo, một vài đối tượng cũng được tạo ra theo mặc định: không gian bảng, bảng, vùng đệm dữ liệu và các tập tin nhật ký (log file). Tạo các đối tượng này sẽ mất một ít thời gian, đó là lý do tại sao lệnh create database yêu cầu vài phút để xử lý. Hình 5.8 mô tả 3 không gian bảng mặc định được tạo. Không gian bảng sẽ được thảo luận chi tiết hơn trong Chương 6, Kiến trúc của DB2; nhưng hiện tại, hãy cứ coi không gian bảng như là các lớp theo logic giữa các bảng logic (logical table), và như tài nguyên hệ thống như là các ổ đĩa và bộ nhớ-- vùng đệm dữ liệu.



Hình 5.8 – Không gian bảng được tạo mặc định khi một Cơ sở dữ liệu được tạo ra.

Không gian bảng SYSCATSPACE chứa các bảng danh mục (Catalog). Bảng danh mục này cũng được biết đến trong các hệ thống quản lý cơ sở dữ liệu quan hệ khác như là từ điển dữ liệu (data dictionary). Về cơ bản, bảng này chứa thông tin hệ thống mà bạn



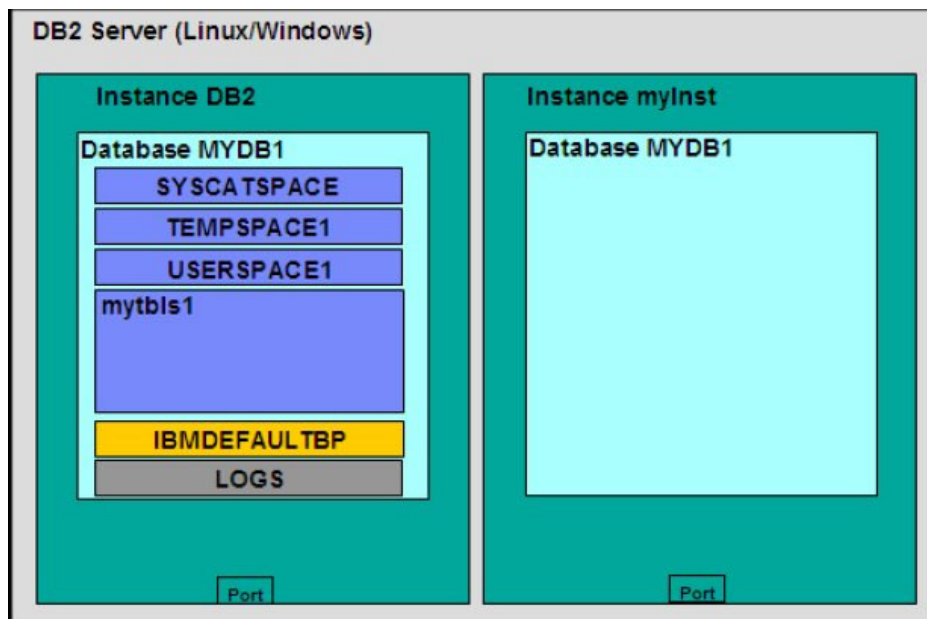
không được sửa đổi cũng như xóa đi; nếu không cơ sở dữ liệu của bạn sẽ hoạt động không chính xác. Không gian bảng TEMPSPACE1 được DB2 sử dụng khi mà cần thêm không gian cho xử lý một số thao tác như là sắp xếp. Không gian bảng USERSPACE1 thông thường được dùng để lưu trữ các bảng cơ sở dữ liệu người dùng khi một bảng được tạo ra mà không có không gian bảng riêng biệt để lưu trữ nó.

Bạn cũng có thể tạo cho riêng mình các không gian bảng khi sử dụng câu lệnh CREATE TABLESPACE. Hình 5.9 mô tả cách tạo không gian bảng MYTBLS1 trong cơ sở dữ liệu MYDB1 của thể hiện DB2. Khi bạn tạo ra một không gian bảng, bạn cũng chỉ định các ổ đĩa và vùng đệm dữ liệu để sử dụng. Do đó, nếu bạn có một bảng “nóng”, đó là một bảng được sử dụng thường xuyên, bạn có thể dùng những ổ đĩa nhanh nhất và phần lớn bộ nhớ thông qua việc ấn định các đặc tính này cho không gian bảng.

Trong hình 5.9, ta thấy có 2 đối tượng khác được mặc định tạo ra: một vùng đệm dữ liệu IBMDEFAULTBP, và các tệp nhật ký.

Vùng đệm dữ liệu về cơ bản là một vùng nhớ được sử dụng bởi cơ sở dữ liệu. Bạn có thể tạo một hay nhiều vùng đệm, nhưng luôn luôn phải có một vùng đệm với kích cỡ một trang nhớ ngang bằng kích thước trang nhớ của không gian bảng có sẵn. Trang nhớ và kích thước trang nhớ sẽ được thảo luận chi tiết hơn vào Chương 6, Kiến trúc của DB2.

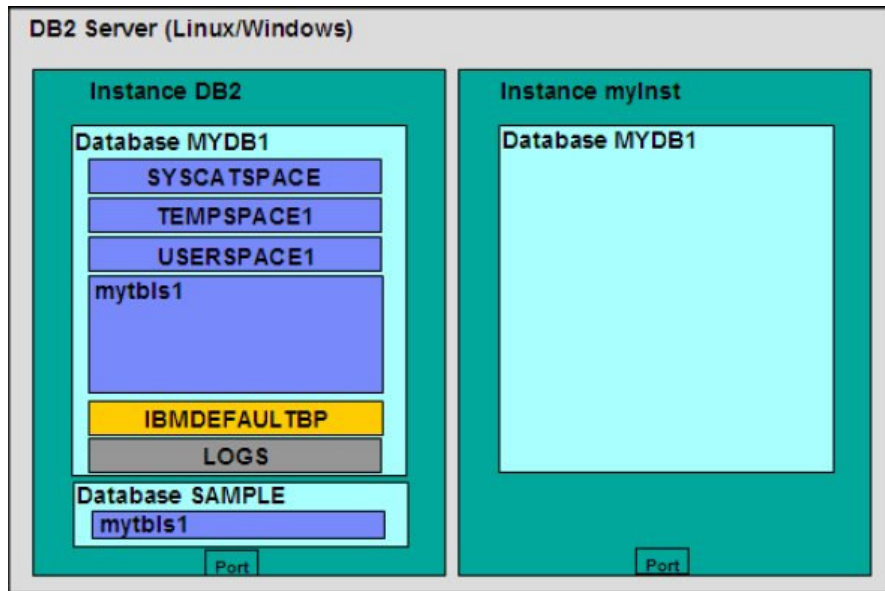
Các tệp nhật ký (log) dùng để phục hồi. Khi bạn làm việc trên một cơ sở dữ liệu, các thông tin không chỉ được lưu trữ trên các ổ đĩa dành cho cơ sở dữ liệu đó, mà trong khi bạn đang làm còn có cả các tệp nhật ký lưu trữ tất cả thao tác xử lý trên dữ liệu đấy. Cứ coi các tệp nhật ký như các tệp lưu tạm mà trên đó lệnh “autosave” được thi hành. Các tệp nhật ký sẽ được nói kĩ hơn ở Chương 11: Sao lưu và phục hồi.



Hình 5.9 – Vùng đệm dữ liệu và các tệp nhật ký.

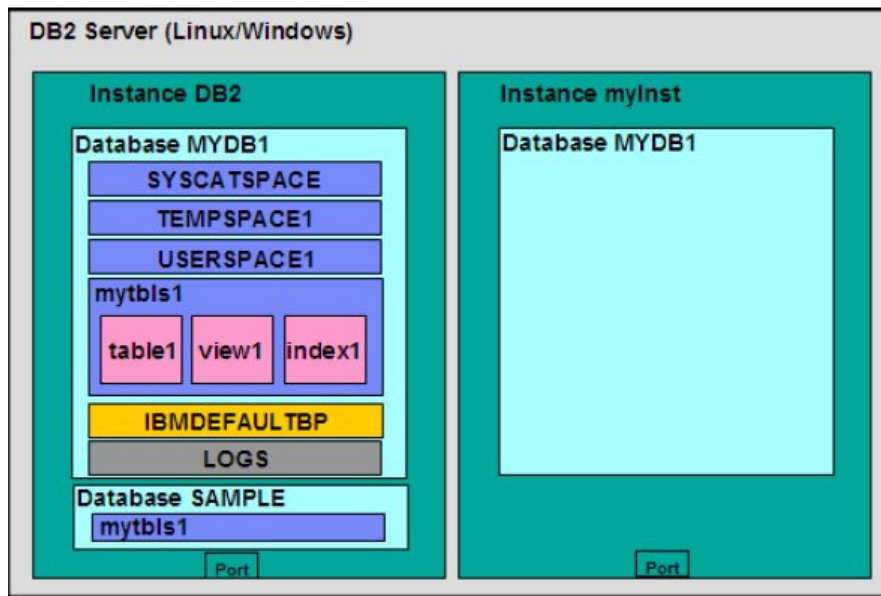
Phần trước, chúng ta đã trao đổi rằng các thể hiện là các môi trường độc lập, và do đó, một cơ sở dữ liệu trùng tên có thể được tạo trong các thể hiện khác nhau. Cũng giống như các thể hiện, các cơ sở dữ liệu cũng là các chủ thể độc lập; do đó, các đối tượng trong một cơ sở dữ liệu này không có liên hệ nào với các đối tượng trong một cơ sở dữ liệu khác. Việc tạo một không gian bảng cùng tên “mytbls1” trong một cơ sở dữ liệu MYDB1 và một cơ sở dữ liệu SAMPLE trong thể hiện DB2 được mô tả như trong

hình 5.10. Điều này là hợp lệ bởi vì các cơ sở dữ liệu là riêng biệt. Lưu ý rằng hình 5.10 không mô tả các đối tượng mặc định khác của một cơ sở dữ liệu SAMPLE do giới hạn không gian trong hình vẽ.



**Hình 5.10 – Vùng bảng có cùng tên trong các cơ sở dữ liệu khác nhau.**

Một khi bạn tạo ra một không gian bảng, bạn cũng có thể tạo các đối tượng trong không gian bảng như bảng biểu, khung nhìn và chỉ mục. Điều này được mô phỏng như trong Hình 5.11.



Hình 5.11 – Bảng, khung nhìn và chỉ mục được tạo trong vùng bảng.

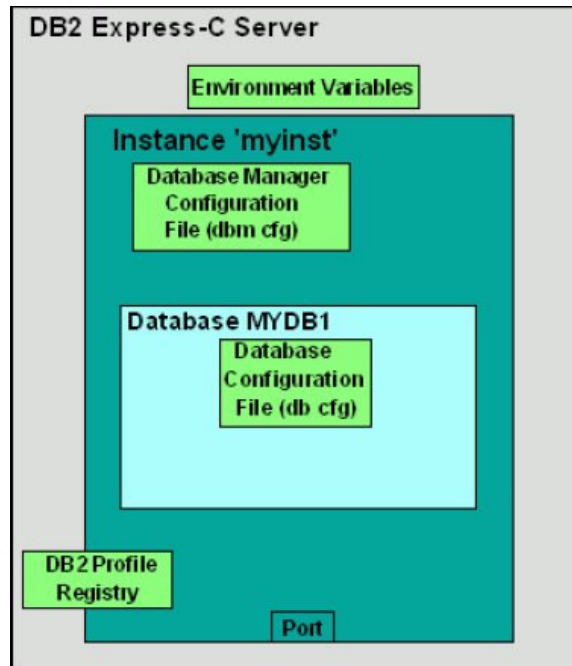
### 5.1 Cấu hình DB2

Ta có thể cấu hình các tham số của DB2 bằng cách sử dụng công cụ Configuration Advisor Tool – công cụ trợ giúp cấu hình. Để kết nối công cụ trợ giúp cấu hình thông qua trung tâm điều khiển, nhấn chuột phải lên một cơ sở dữ liệu và chọn “Configuration Advisor”. Dựa trên cách trả lời của bạn về một số câu hỏi liên quan đến tài nguyên hệ thống, khối lượng công việc, công cụ hỗ trợ cấu hình sẽ cung cấp một danh sách các tham số của DB2 nên thay đổi ứng với các giá trị phù hợp. Nếu bạn muốn biết chi tiết hơn về cấu hình trên DB2, hãy đọc tiếp; nếu không, sử dụng công cụ trợ giúp cấu hình cũng giúp bạn hoàn thành tốt công việc trên DB2.

Một máy chủ DB2 có thể cấu hình trên 4 cấp độ:

- Các biến môi trường.
- Tệp cấu hình quản trị cơ sở dữ liệu (dbm cfg).
- Tệp cấu hình cơ sở dữ liệu (db cfg).
- Đăng ký lý lịch DB2.

Điều này được mô tả trong Hình 5.12. Trong hình, hãy chú ý vị trí của mỗi chiếc hộp. Ví dụ, các biến môi trường được thiết lập ở cấp độ hệ điều hành của máy chủ, trong khi các tham số tệp cấu hình quản lý cơ sở dữ liệu được thiết lập ở lớp thể hiện. Tham số cấu hình cơ sở dữ liệu được quản lý ở lớp cơ sở dữ liệu, và đăng ký lý lịch (profile DB2) được thiết lập hoặc ở hệ điều hành hoặc lớp thể hiện.



Hình 5.12 – Cấu hình DB2.

### 5.1.1 Các biến môi trường

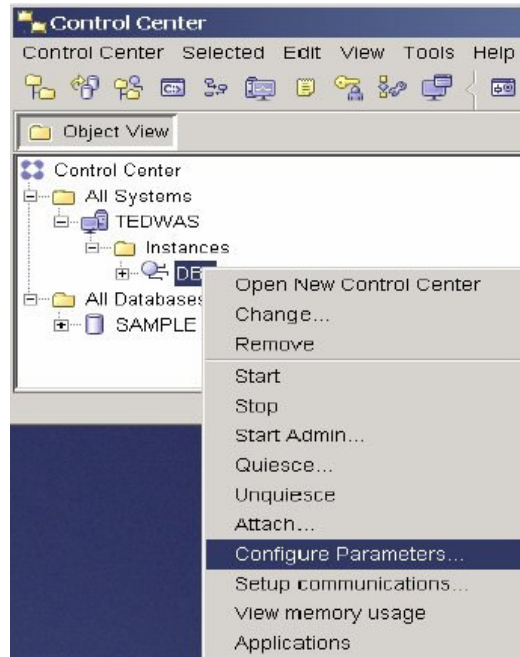
Các biến môi trường là các biến được thiết lập ở hệ điều hành. Một biến môi trường chính là DB2INSTANCE. Biến này cho biết thể hiện đang hoạt động mà bạn đang làm việc, và các lệnh DB2 bạn sẽ áp dụng đến. Ví dụ, để thiết lập thể hiện hoạt động là “myinst” trên Windows, bạn có thể chạy lệnh hệ điều hành sau:

```
set db2instance = myinst
```

### 5.1.2 Tập cấu hình quản trị cơ sở dữ liệu (dbm cfg)

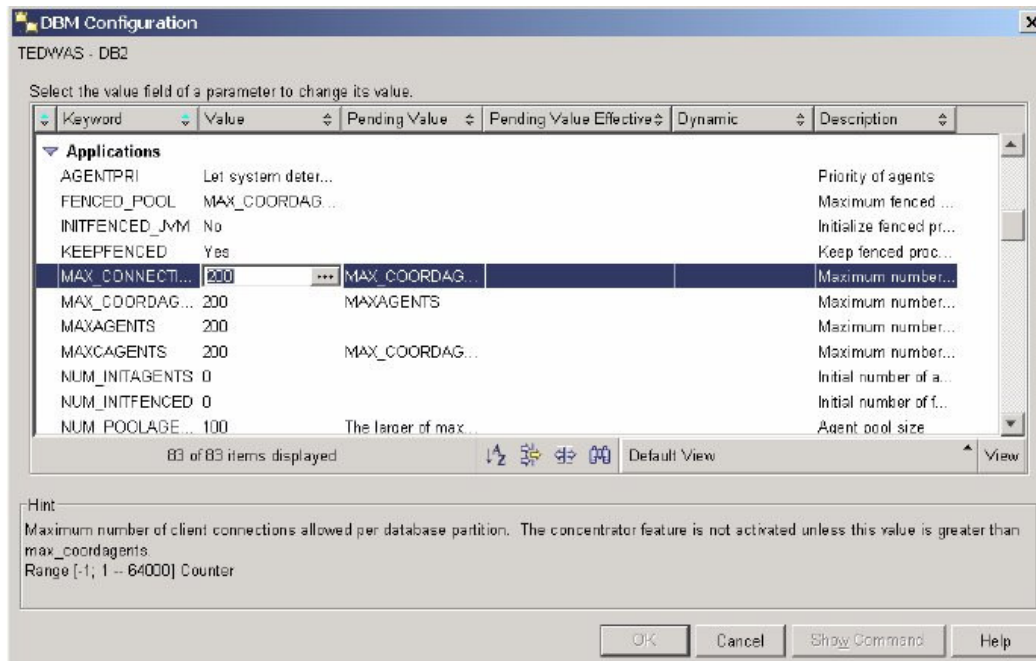
Tập cấu hình quản trị cơ sở dữ liệu (dbm cfg) bao gồm các tham số ảnh hưởng tới thể hiện và tất cả cơ sở dữ liệu chứa trong đó. Tập cấu hình quản trị cơ sở dữ liệu có thể được xem xét hay sửa đổi bằng cách sử dụng dòng lệnh, hoặc thông qua Trung tâm điều khiển của DB2.

Để làm việc với DBM CFG từ Trung tâm điều khiển, lựa chọn đối tượng thể hiện từ thư mục Instance của trung tâm điều khiển, nhấn chuột phải để cho trình đơn bật lên và lựa chọn Configuration Parameters. Công việc này được mô tả trong Hình 5.13.



Hình 5.13 – Cấu hình dbm từ Trung tâm điều khiển.

Sau khi chọn Configuration Parameters, một màn hình sẽ hiển thị lên như trong Hình 5.14 với danh sách các tham số dbm cfg.



Hình 5.14 – Danh sách dbm cfg.

Phần lớn tham số là động, nghĩa là các thay đổi sẽ có hiệu lực ngay lập tức; tuy nhiên, thay đổi một số tham số yêu cầu phải khởi động lại thể hiện để có hiệu lực. Từ công cụ dòng lệnh, điều này có thể thực hiện thông qua các lệnh `db2stop` và `db2start`.

Trước khi một thể hiện ngừng, tất cả các ứng dụng phải được ngắt kết nối, hay tạm dừng. Nếu bạn muốn ngừng một thể hiện ngay lập tức, bạn có thể dùng lệnh `db2stop force`.

Một thể hiện cũng có thể ngừng bằng cách nhấn chuột lên đối tượng thể hiện và lựa chọn Stop hay Start thông qua Control Center.

Bảng 5.3 Mô tả một số lệnh hữu hiệu để quản trị công cụ dbm cfg từ dòng lệnh.

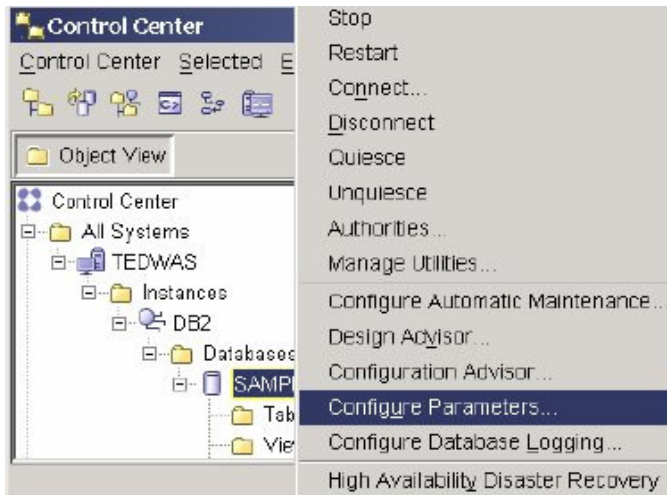
Lệnh	Mô tả
<code>db2 get dbm cfg</code>	Thu thập thông tin về dbm cfg
<code>db2 update dbm cfg using &lt;parameter_name&gt; &lt;value&gt;</code>	Cập nhập giá trị của tham số dbm cfg.

**Bảng 5.3 – Các lệnh điều khiển dbm cfg.**

### 5.1.3 Tập cấu hình cơ sở dữ liệu (db cfg)

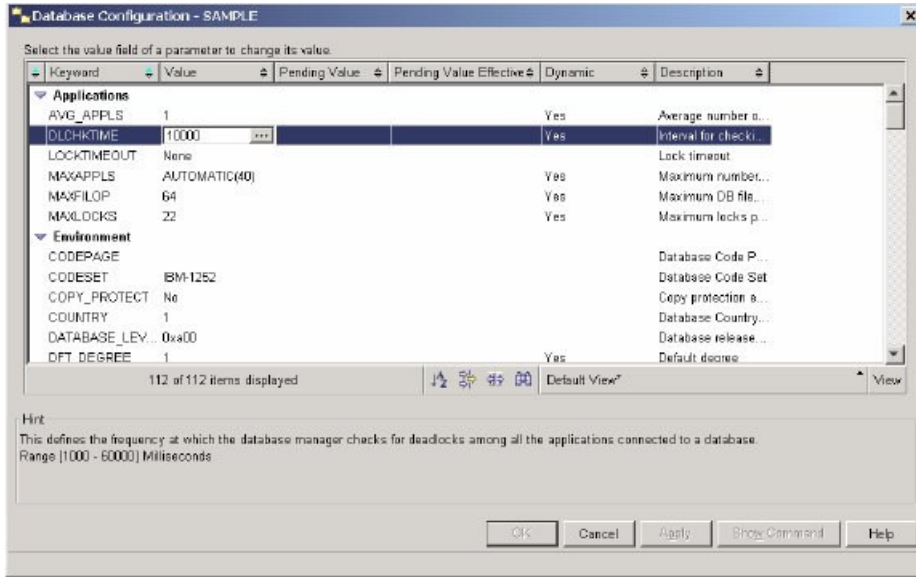
Tập cấu hình cơ sở dữ liệu (db cfg) bao gồm các tham số tác động đến từng cơ sở dữ liệu. Tập cấu hình cơ sở dữ liệu có thể xem xét hay sửa đổi bằng dòng lệnh, hay thông qua Trung tâm điều khiển của DB2.

Để làm việc với DB CFG từ Trung tâm điều khiển, chọn đối tượng cơ sở dữ liệu từ thư mục cơ sở dữ liệu của màn hình Trung tâm điều khiển, nhấn chuột phải để cho trình đơn bật lên và lựa chọn Configuration Parameters. Việc này được mô tả như trong Hình 5.15.



**Hình 5.15 – Cấu hình db cfg từ Trung tâm điều khiển.**

Sau khi chọn Configuration Parameters, một màn hình sẽ hiển thị lên như trong Hình 5.16 với danh sách các tham số db cfg.



**Hình 5.16 – Cấu hình cơ sở dữ liệu.**

Bảng 5.4 Mô tả một số lệnh hữu hiệu để quản lý db cfg từ dòng lệnh.

Lệnh	Mô tả
get db cfg for <database_name>	Thu thập thông tin về db cfg của cơ sở dữ liệu mong muốn.
update db cfg for <database_name> using <parameter_name> <value>	Cập nhật giá trị cho một tham số của db cfg.

**Bảng 5.4 – Lệnh điều khiển db cfg.**

**5.1.4 Đăng ký lý lịch DB2 (profile).**

Các biến đăng ký lý lịch DB2 bao gồm các tham số có thể trên một nền cụ thể và có thể thiết lập thành toàn cục (ảnh hưởng tới tất cả thể hiện), hay ở từng lớp thể hiện (ảnh hưởng riêng từng thể hiện).

Bảng 5.5 Mô tả một số lệnh hữu hiệu để điều khiển các biến đăng ký lý lịch DB2.

Lệnh	Mô tả
db2set -all	Liệt kê tất cả các biến đăng ký lý lịch DB2 hiện tại được thiết lập.
db2set -lr	Liệt kê tất cả biến đăng ký lý lịch DB2.
db2set <parameter>=<value>	Thiết lập một giá trị cho một tham số.

**Bảng 5.5 – Các lệnh điều khiển đăng ký lý lịch DB2.**

Bảng 5.6 Một số biến đăng ký DB2 hay được dùng nhất.



<b>Biến đăng ký</b>	<b>Mô tả</b>
DB2COMM	Chỉ định những quản trị giao tiếp sẽ khởi động khi bắt đầu công cụ quản trị cơ sở dữ liệu.
DB2_EXTSECURITY	Trên Windows, ngăn chặn những người chưa được định danh kết nối tới DB2 bằng cách khóa tệp tin hệ thống DB2.
DB2_COPY_NAME	Lưu trữ tên của bản sao DB2 hiện tại đang dùng. Để chuyển sang một bản sao DB2 đã cài khác, chạy dòng lệnh <i>path\bin\db2envvars.bat</i> . Biến này không thể dùng cho mục đích này.

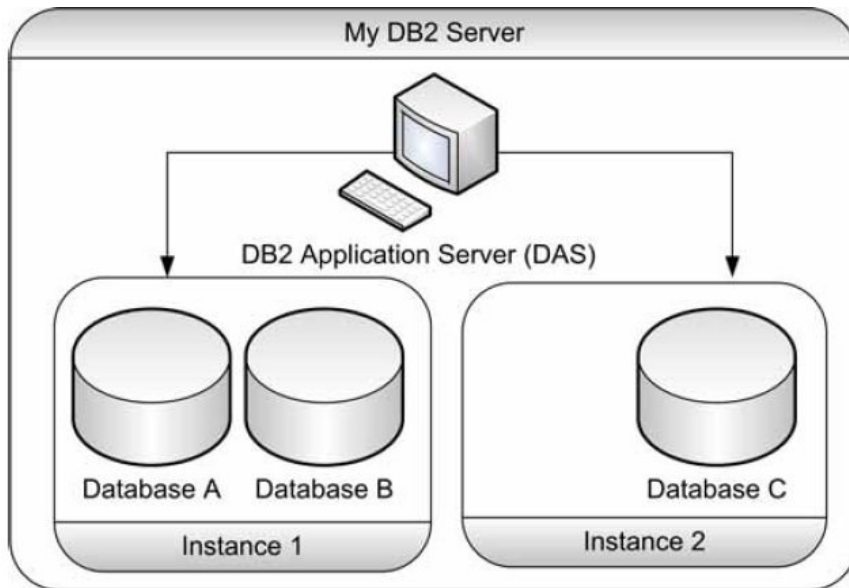
**Bảng 5.6 – Các biến đăng ký DB2 hay được dùng.**

Ví dụ, để cho phép giao tiếp sử dụng giao thức TCP/IP, thiết lập biến đăng ký DB2COMM thành TCPIP như ở dưới:

```
db2set db2comm = tcpip
```

**5.2 Máy chủ quản trị DB2**

Máy chủ quản trị DB2 (DAS) là một quá trình nền chạy trên máy chủ DB2 cho phép các máy khách từ xa có thể quản trị máy chủ DB2 thông qua công cụ đồ họa. Ở đây chỉ có một DAS trên mỗi máy tính như trên Hình 5.16.



**Hình 5.16 – Máy chủ quản trị DB2 (DAS).**



# 6

## Phần II – Học DB2: Quản trị Cơ sở dữ liệu

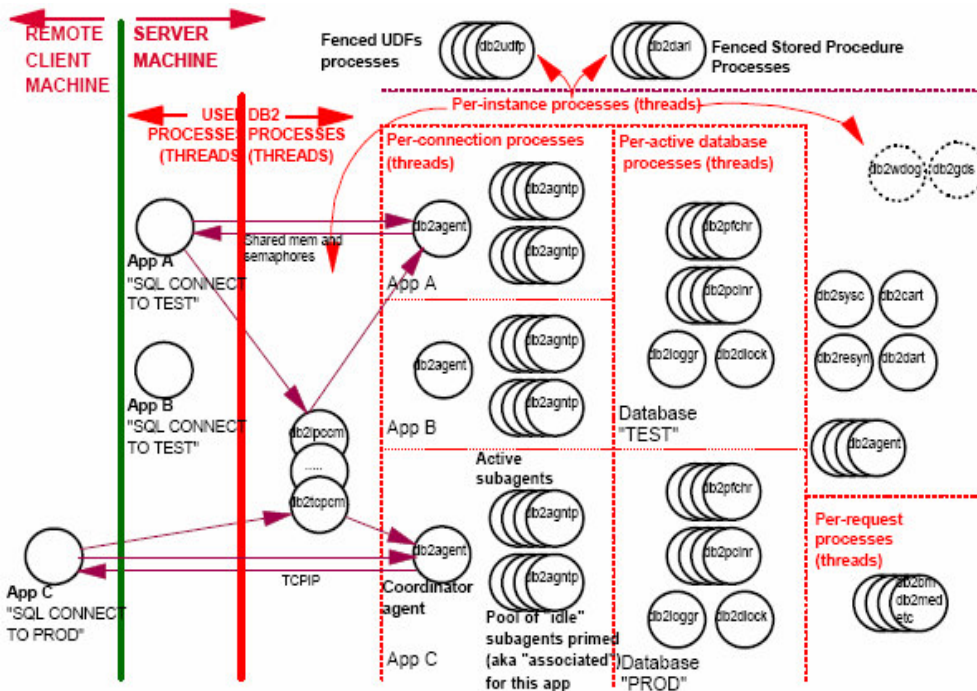
### Chương 6 - Kiến trúc của DB2

Trong chương này chúng ta sẽ thảo luận những nét chính về

- Mô hình xử lý của DB2
- Mô hình bộ nhớ của DB2
- Mô hình lưu trữ của DB2

#### 6.1 Mô hình xử lý của DB2

Hình 6.1 mô tả mô hình xử lý của DB2. Trong hình, bên trái đường thẳng đứng màu xanh tượng trưng cho một máy trạm từ xa (remote client machine), và bên phải tượng trưng cho máy chủ (server machine). Khi một ứng dụng trên máy trạm kết nối đến máy chủ bằng lệnh SQL CONNECT, các bộ đội kết nối (listener) cho giao thức truyền thông sẽ nhận được yêu cầu này và liên lạc với một tác nhân kết hợp DB2 (DB2 coordinator agent). Tác nhân DB2 có chức năng thực hiện các thao tác thay cho DB2. Nếu ứng dụng đó là cục bộ (chạy trên cùng máy chủ với DB2), các bước rất giống nhau, chỉ có tác nhân *db2ipccm* xử lý yêu cầu thay cho *db2tccpm*. Trong vài trường hợp, như khi xử lý song song, một tác nhân db2 có thể sinh ra các tác nhân khác (*db2agntp*). Các tác nhân khác như *db2pfchr*, *db2loggr*, *db2dlock* cũng có thể được sử dụng cho những mục đích khác nhau. Những xử lý thông dụng nhất được mô tả trong bảng 6.1.



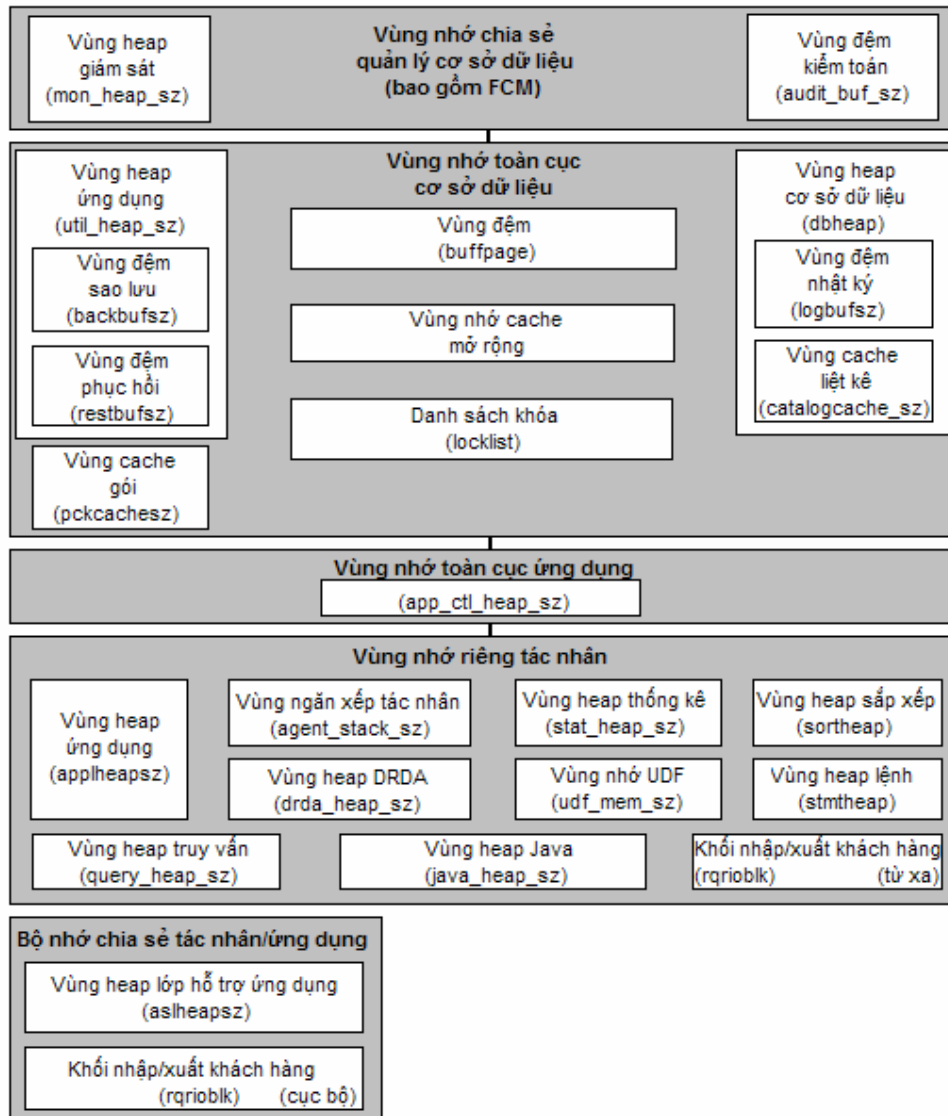
Hình 6.1 – Mô hình xử lý của DB2

Tên xử lý	Mô tả
<b>db2sync</b>	Xử lý của thể hiện cơ sở dữ liệu (một xử lý cho một thể hiện).
<b>db2tccpcm</b>	Truyền thông/listener TCP/IP.
<b>db2agent</b>	Tác nhân kết hợp thực thi những thao tác cơ sở dữ liệu thay cho các ứng dụng (ít nhất một tiến trình trên một kết nối, phụ thuộc vào bộ tập trung kết nối có được kích hoạt không).
<b>db2agntp</b>	Tác nhân con chủ động được sinh ra nếu trường INTRA_PARALLEL là YES. Sẽ thực thi các thao tác cơ sở dữ liệu thay cho ứng dụng. <i>db2agent</i> sẽ kết hợp công việc của các tác nhân con <i>db2agntp</i> .
<b>db2pfchr</b>	Bộ đọc dữ liệu nhập/xuất bất đồng bộ (NUM_IOSERVERS).
<b>db2pclnr</b>	Bộ ghi dữ liệu nhập/xuất bất đồng bộ (NUM_IOCLEANERS).

**Bảng 6.1- Các xử lý thông dụng của DB2**

**6.2 Mô hình bộ nhớ của DB2**

Mô hình bộ nhớ DB2 gồm nhiều vùng nhớ khác nhau tại mức thể hiện, mức cơ sở dữ liệu và mức ứng dụng và tác nhân (hình 6.2).



## Hình 6.2 – Mô hình bộ nhớ của DB2

Khi bắt đầu một thể hiện, vùng nhớ chia sẻ quản lý cơ sở dữ liệu được cấp phát. Thường vùng nhớ này không nhiều. Lần đầu tiên kết nối đến cơ sở dữ liệu, vùng nhớ toàn cục cơ sở dữ liệu sẽ được cấp phát. Trong khối này, vùng đệm (buffer pool) là một trong những thành phần quan trọng nhất, đặc biệt khi cần cải thiện việc thực thi truy vấn. Kích thước của vùng đệm quyết định độ lớn của toàn bộ vùng nhớ toàn cục cơ sở dữ liệu.

Vùng nhớ riêng tác nhân là vùng nhớ được sử dụng cho mỗi tác nhân DB2. Nếu không dùng bộ tập trung kết nối thì mỗi kết nối cần một tác nhân. Một tác nhân sử dụng khoảng 3–5MB. Nếu sử dụng bộ tập trung kết nối, nhiều kết nối có thể dùng chung một tác nhân, giảm nhu cầu sử dụng bộ nhớ vật lý.

### 6.3 Mô hình lưu trữ của DB2

Trong mục này chúng ta sẽ mô tả những khái niệm sau:

- Trang và phạm vi
- Vùng nhớ đệm
- Không gian bảng

#### 6.3.1 Trang và phạm vi

Trang là đơn vị lưu trữ nhỏ nhất của DB2, với các kích thước 4K, 8K, 16K và 32K. Một phạm vi là một nhóm các trang. Tại một thời điểm nếu chỉ làm việc với một trang thì lãng phí khả năng thực thi, do vậy DB2 làm việc với các phạm vi tại một thời điểm. Kích thước của trang và phạm vi được định nghĩa khi làm việc với vùng nhớ đệm và các không gian bảng như trong phần tiếp theo.

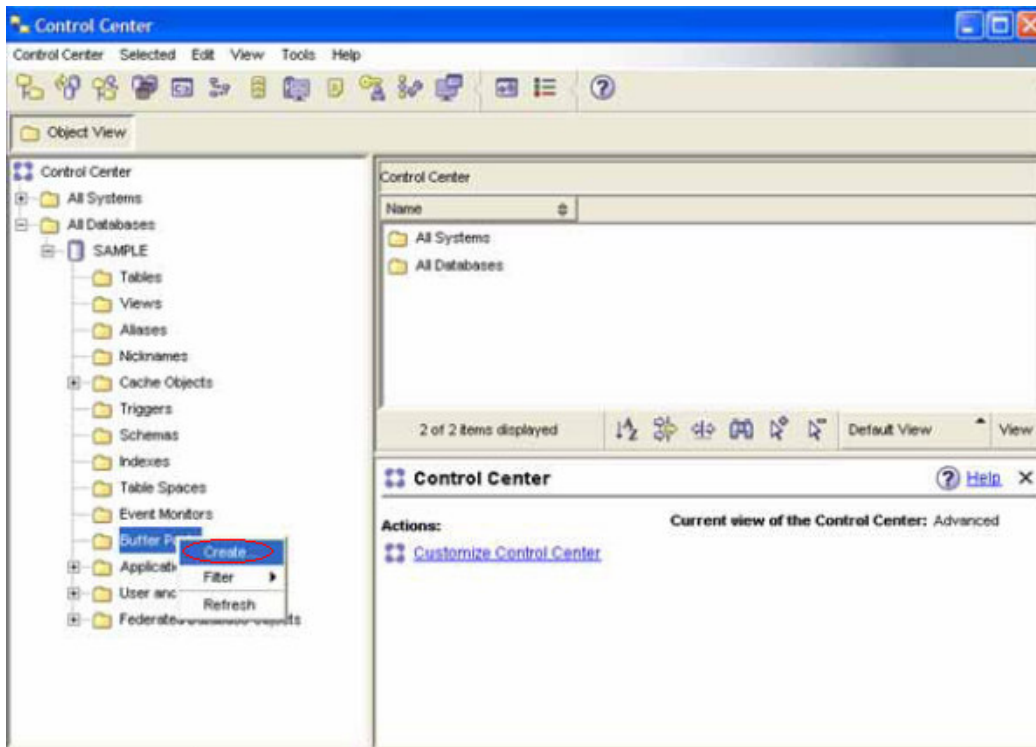
#### 6.3.2 Vùng nhớ đệm

Một vùng nhớ đệm là vùng nhớ thực (real cache) cho dữ liệu bảng và chỉ mục. Nó cải thiện thực thi bằng cách giảm nhập/xuất tuần tự trực tiếp và nâng cấp đọc ghi không đồng bộ. Có nghĩa là DB2 đoán trước trang nào sẽ cần đến, đọc chúng từ đĩa vào vùng đệm, nên chúng sẵn sàng được sử dụng.

Vùng đệm được cấp phát với các đơn vị nhớ là các trang 4K, 8K, 16K, và 32K. Nên có ít nhất một vùng đệm trên một cơ sở dữ liệu, và ít nhất một vùng đệm phù hợp cho một không gian bảng có kích thước trang đã biết.

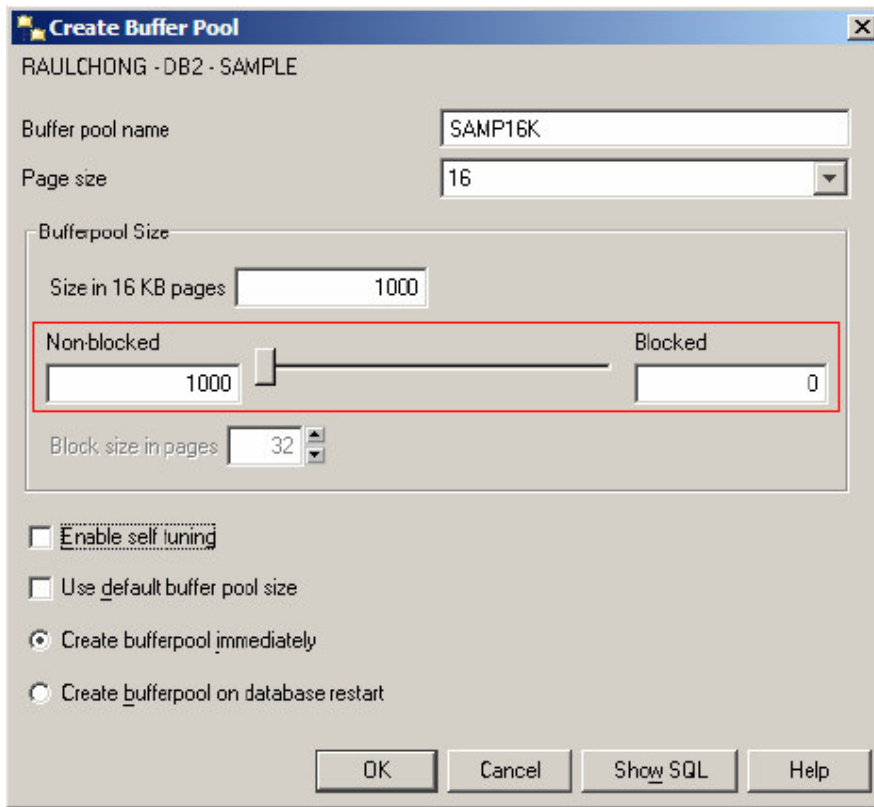
#### Tạo một vùng đệm

Có thể dùng lệnh CREATE BUFFERPOOL để tạo vùng đệm. Cũng có thể vào trung tâm điều khiển, nhấn chuột phải lên thư mục Buffer pool trong cơ sở dữ liệu đã biết và chọn Create như trong hình 6.3.



Hình 6.3 – Tạo một vùng đệm

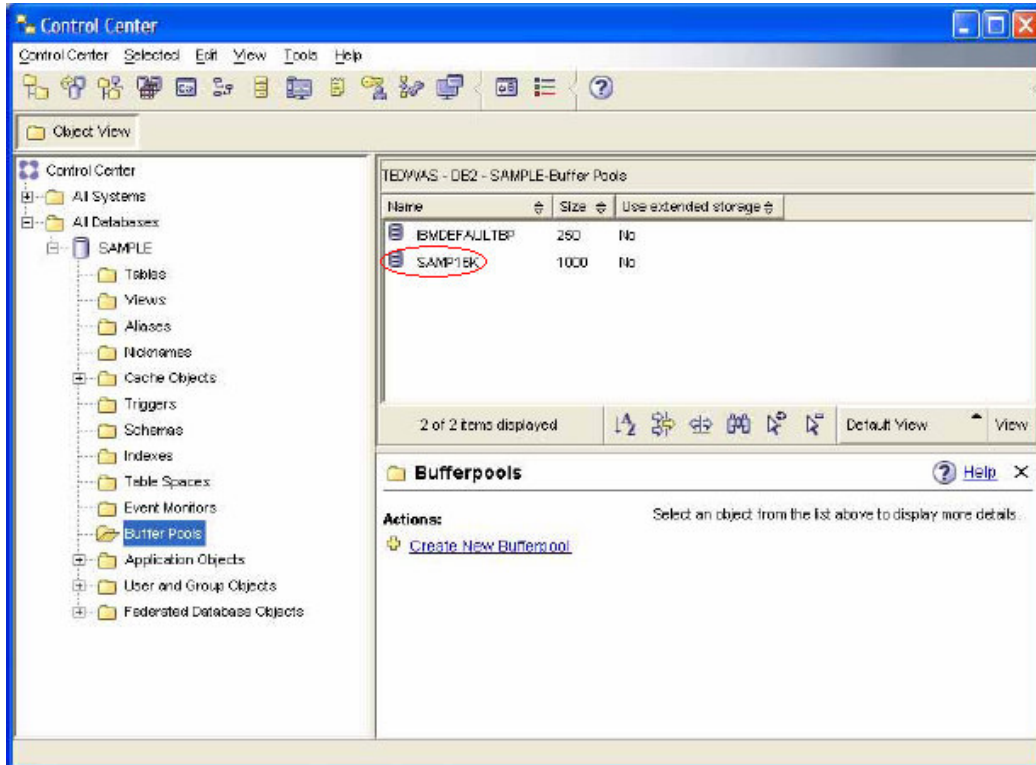
Sau khi chọn Create, hộp thoại Create Buffer Pool xuất hiện như trong hình 6.4.



**Hình 6.4 - Hộp thoại tạo vùng đệm**

Phần lớn các mục trong hình 6.4 được tự giải thích. Các trường 'Non-blocked' và 'Blocked' chỉ định số lượng trang tồn tại là không đóng khối (non-blocked) hay đóng khối (blocked). Vùng đệm dựa trên khối bảo đảm rằng các trang liên tục trên đĩa di chuyển vào vùng đệm cũng liên tục trong một vùng đóng khối. Cách này có thể cải thiện việc thực thi. Số lượng trang không thể lớn hơn 98% số lượng trang của vùng đệm. Đặc biệt, giá trị 0 sẽ vô hiệu hóa nhập/xuất khối.

Mỗi lần vùng đệm được tạo ra, nó sẽ được trình bày trong trung tâm điều khiển như trong hình 6.5.



Hình 6.5 – Trung tâm điều khiển sau khi tạo vùng đệm ‘SAMP16K’

### 6.3.3 Các không gian bảng

Không gian bảng là một là một giao diện logic giữa các bảng logic và các vùng nhớ của hệ thống (bao gồm vùng nhớ đệm và các đĩa). Bạn có thể sử dụng câu lệnh CREATE TABLESPACE để tạo một không gian bảng mà bạn đặc tả:

- Kích thước trang của không gian bảng (4KB, 8KB, 16KB, or 32KB). Kích thước trang này phải tương ứng với kích thước trang của vùng đệm.
- Tên của vùng đệm mà liên kết với không gian bảng
- Kích thước của phạm vi
- Kích thước nạp trước

#### Các loại không gian bảng

Có ba loại không gian bảng:

- Thông thường  
Loại này dành cho các bảng của người dùng. Ví dụ như USERSPACE1 là một không gian bảng thông thường vì nó được tạo ra một cách mặc định.
- Lớn  
Loại này được sử dụng tùy chọn để phân chia các dữ liệu LOB về không gian bảng của nó. Nó cũng được sử dụng để lưu trữ các dữ liệu XML để tạo ra các CSDL có hỗ trợ PureXML – sử dụng UNICODE và XML làm kiểu của các cột. Các không gian bảng loại lớn là mặc định.
- Tạm thời  
Có hai loại không gian bảng tạm thời:

- ▶ Không gian bảng tạm thời của hệ thống  
Đây là loại không gian bảng được sử dụng bởi DB2 cho các thao tác nội bộ (hệ thống) như sắp xếp. Ví dụ TEMPSPACE1 là không gian bảng được tạo ra tự động khi bạn tạo một CSDL, nó là một không gian bảng tạm thời của hệ thống.
- ▶ Không gian bảng tạm thời của người dùng  
Đây là loại không gian bảng được sử dụng để tạo ra các bảng tạm thời toàn cục do người dùng định nghĩa. Chúng thường hay bị nhầm lẫn với các không gian bảng tạm thời của hệ thống.

### **Quản lý các không gian bảng**

Các không gian bảng có thể được phân loại theo cách quản lý. Việc này có thể được chỉ định trong câu lệnh CREATE TABLESPACE

#### **Không gian bảng được quản lý bởi hệ thống**

Loại không gian bảng này được gọi là System Managed Storage (SMS). Điều này có nghĩa là hệ điều hành sẽ quản lý không gian nhớ này. Chúng dễ dàng được quản lý và được chứa trong các thư mục hệ thống tệp. Không gian bảng loại này không được cấp phát trước, nhưng dung lượng các tệp được tăng lên tự động. Một khi bạn đã xác định các chỗ chứa (thư mục), thì chúng sẽ được cố định lúc tạo ra và không thể thay đổi về sau, trừ khi việc phục hồi tái định hướng được dùng. Khi sử dụng các không gian bảng SMS, dữ liệu, chỉ mục, dữ liệu LOB không thể được trải trên các không gian bảng khác nhau.

#### **Không gian bảng được quản lý bởi CSDL**

Loại không gian bảng này được gọi là Database Managed Storage (DMS). Điều này có nghĩa là DB2 sẽ quản lý loại không gian lưu trữ này. Việc quản lý đòi hỏi sự can thiệp của người quản trị cơ sở dữ liệu. Các chỗ chứa (container) có thể là các tệp được cấp phát trước hoặc các thiết bị thô (thiết bị thô được hiểu như thiết bị lưu trữ được ghi theo định hướng ký tự, không cần hệ thống tệp của hệ điều hành). Với các thiết bị thô, dữ liệu được ghi trực tiếp mà không qua bộ đệm của hệ điều hành.

Các chỗ chứa (hệ thống tệp) có thể được thêm, bỏ, thay đổi kích thước. DMS có hiệu suất tốt nhất, vì dữ liệu, chỉ mục hay dữ liệu LOB có thể được nằm trên các không gian bảng khác nhau để nâng cao hiệu suất.

#### **Không gian bảng được quản lý tự động**

Loại không gian bảng này được quản lý bởi bộ nhớ tự động, ích lợi của chúng là sử dụng tương tự như các không gian bảng SMS, nhưng lại có hiệu suất tốt và tính linh hoạt của không gian bảng DMS. Do đó trong DB2 9, đây là loại không gian bảng mặc định. Với loại không gian bảng này, người dùng chỉ cần phải chỉ ra nhóm các thiết bị lưu trữ logic. Không phải định nghĩa tường minh chỗ chứa. Các chỗ chứa được tạo tự động theo các đường dẫn. Việc tăng kích thước hay bổ sung thêm chỗ chứa mới hoàn toàn do DB2 quản lý.

Để cho phép lưu trữ tự động, đầu tiên bạn cần tạo một CSDL với chức năng lưu trữ tự động được kích hoạt (mặc định) và liên kết các đường dẫn lưu trữ tới nó (CSDL). Sau khi tạo xong, nếu cần thiết bạn có thể tái định nghĩa các đường dẫn lưu trữ bằng cách sử dụng thao tác RESTORE cơ sở dữ liệu. Sau đó bạn có thể tạo các không gian bảng sử dụng lưu trữ tự động (đây là chế độ mặc định)

#### **Ví dụ lưu trữ tự động**



Đầu tiên, hãy tạo một CSDL với lưu trữ tự động được cho phép theo các ví dụ sau:

Lưu trữ tự động được cho phép theo mặc định:

```
CREATE DATABASE DB1
```

Lưu trữ tự động được chỉ định tường minh:

```
CREATE DATABASE DB1 AUTOMATIC STORAGE YES
```

Lưu trữ tự động được cho phép theo mặc định nhưng đường dẫn được chỉ ra:

```
CREATE DATABASE DB1 ON /data/path1, /data/path2
```

Lưu trữ tự động không được phép:

```
CREATE DATABASE DB1 AUTOMATIC STORAGE NO
```

Tiếp theo, chúng ta tạo không gian bảng với lưu trữ tự động được cho phép theo các ví dụ sau:

Lưu trữ tự động cho không gian bảng được cho phép theo mặc định:

```
CREATE TEMPORARY TABLESPACE TEMPTS
```

Lưu trữ tự động được chỉ định một cách tường minh cho không gian bảng:

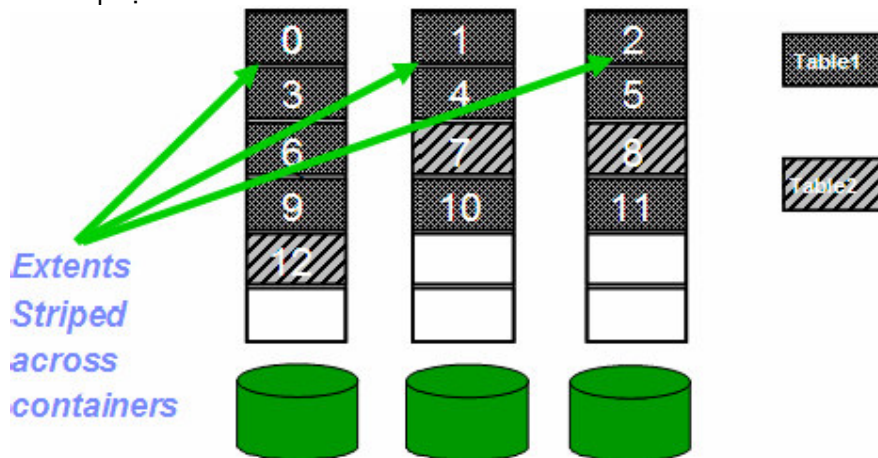
```
CREATE TABLESPACE TS2 MANAGED BY AUTOMATIC STORAGE
```

Lưu trữ tự động được chỉ định không tường minh, kích thước ban đầu được cấp phát, kích thước tăng theo từng lần, và kích thước tối đa có thể tăng đến.

```
CREATE TABLESPACE TS1
    INITIALSIZE 500 K
    INCREASESIZE 100 K
    MAXSIZE 100 M
```

**Dữ liệu được lưu trữ như thế nào trong không gian bảng**

Theo mặc định, DB2 sẽ ghi vào các phạm vi đĩa (disk extent) tại thời điểm chia băng (stripe) qua các chỗ chứa. Ví dụ nếu bạn có một không gian bảng 4K với kích thước mở rộng là 8, sử dụng 3 chỗ chứa thô trong không gian bảng DMS, điều này có nghĩa rằng 32K dữ liệu (4K x 8 trang trên một phạm vi = 32K) sẽ được ghi vào một đĩa trước khi ghi vào đĩa tiếp theo. Điều này được chỉ ra trong Hình 6.6. Chú ý rằng các bảng không chia sẽ các phạm vi.

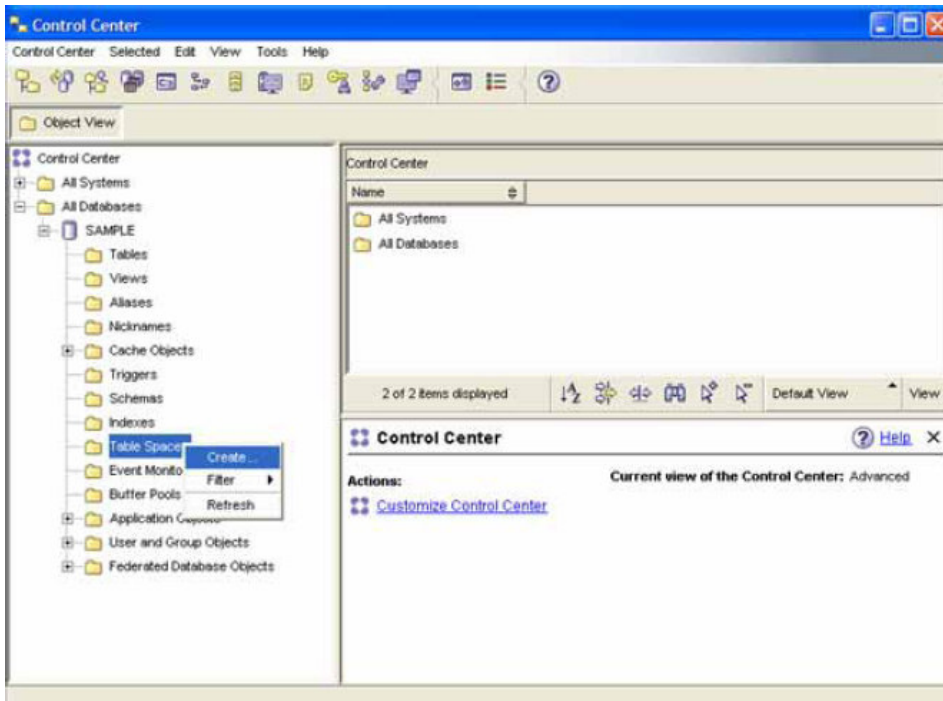


Hình 6.6 – Ghi dữ liệu vào không gian bảng

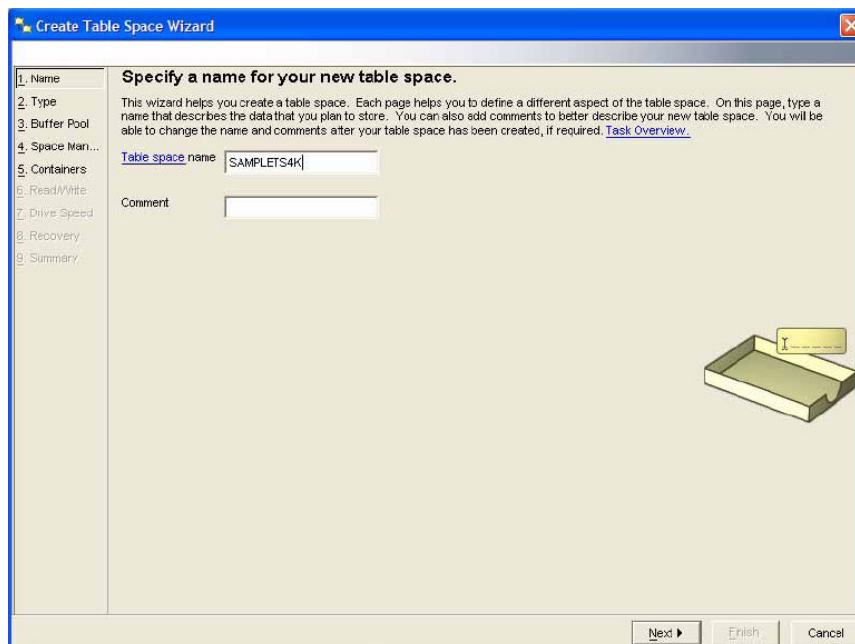


### Tạo không gian bảng sử dụng trung tâm điều khiển

Để tạo một không gian bảng từ trung tâm điều khiển, kích chuột phải vào thư mục Table Spaces trong CSDL được chọn và chọn Create theo Hình 6.7. Trợ giúp từng bước tạo không gian bảng sẽ xuất hiện như Hình 6.8.



Hình 6.7 – Tạo không gian bảng từ trung tâm điều khiển



Hình 6.8 – Tạo một không gian bảng sử dụng trợ giúp

# 7

## Chương 7 – Kết nối máy khách DB2

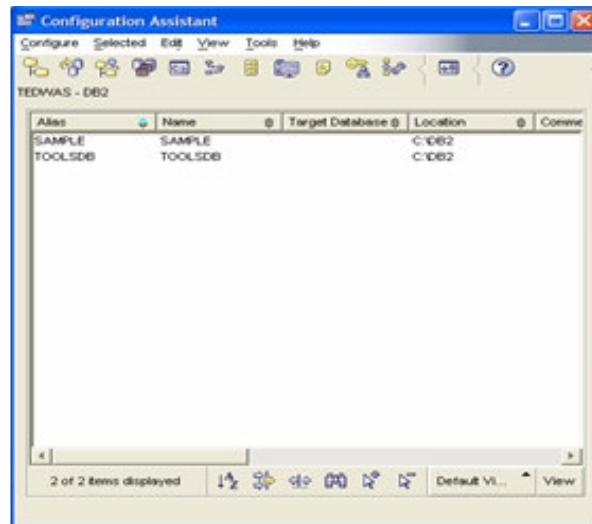
Chương này trình bày các bước thiết lập cần thiết để kết nối từ máy khách DB2 đến máy chủ DB2 sử dụng giao thức TCPIP. Chú ý rằng một máy chủ DB2 cũng được xem như một thành phần máy khách, vì thế máy chủ DB2 cũng có hành vi của một máy khách khi kết nối đến một máy chủ DB2 khác. Hiện tại có một số cách khác nhau để thiết lập kết nối máy khách DB2; tuy nhiên, trong phạm vi của chương này chúng ta chỉ thảo luận một phương thức đơn giản nhất đó là sử dụng công cụ Trợ giúp cấu hình (Configuration Assistant).

### 7.1 Trợ giúp cấu hình

Sử dụng giao diện trực quan của công cụ Trợ giúp cấu hình, bạn có thể dễ dàng cấu hình kết nối giữa một máy khách DB2 và một máy chủ DB2.

Để thực thi Trợ giúp cấu hình trên Windows, bạn có thể chọn: *Start -> Programs -> IBM DB2 -> DB2COPY1 -> Set-up Tools -> Configuration Assistant.*

Nếu sử dụng dòng lệnh, bạn có thể bắt đầu công cụ này bằng cách sử dụng lệnh `db2ca`. Hình 7.1 mô tả công cụ trợ giúp cấu hình.



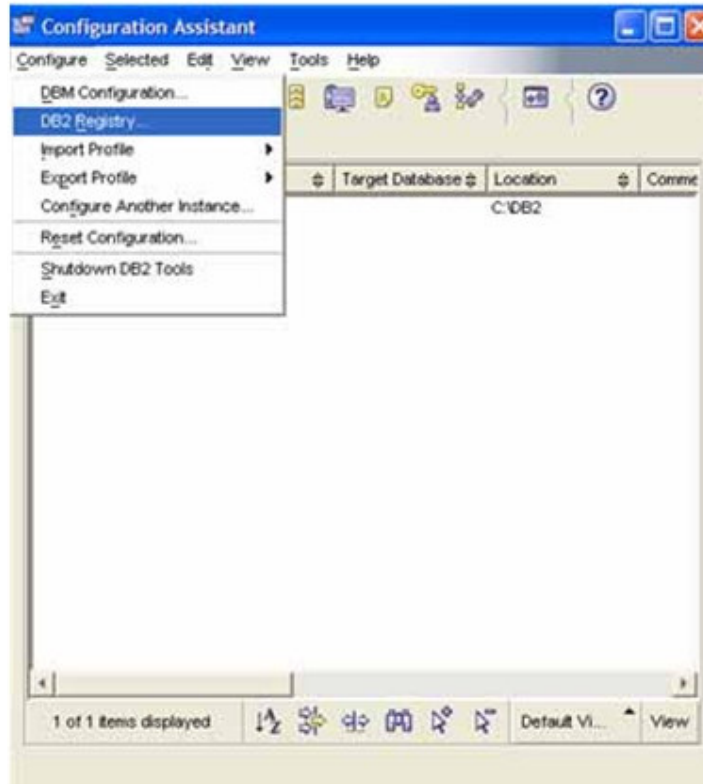
Hình 7.1 - Công cụ trợ giúp cấu hình

#### 7.1.1 Những cài đặt bắt buộc trên máy chủ

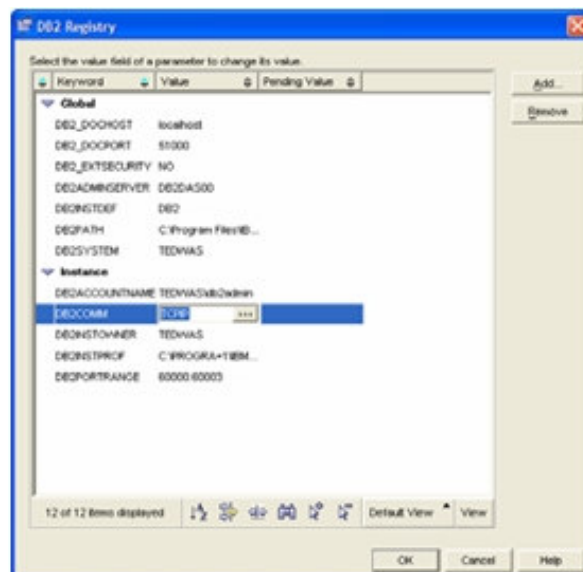
Có hai việc cần được thiết lập tại máy chủ:

- 1) Biến DB2COMM

Biến đăng kí này xác định giao thức truyền thông được sử dụng để lắng nghe các yêu cầu đến từ máy khách. Bộ giao thức TCP/IP là bộ các giao thức truyền thông được sử dụng phổ biến nhất. Khi thay đổi giá trị của tham số này, hệ thống cần được khởi động lại. Để xem và thay đổi giá trị của biến DB2COMM trong trợ giúp cấu hình, chọn *Configure -> DB2 Registry* như được mô tả trong Hình 7.2 và Hình 7.3



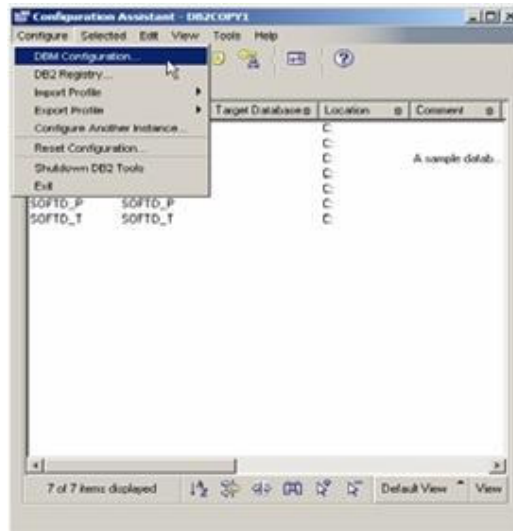
Hình 7.2 – Truy cập DB2 Registry



Hình 7.3 – Kiểm chứng giá trị của biến DB2COMM trong DB2 Registry

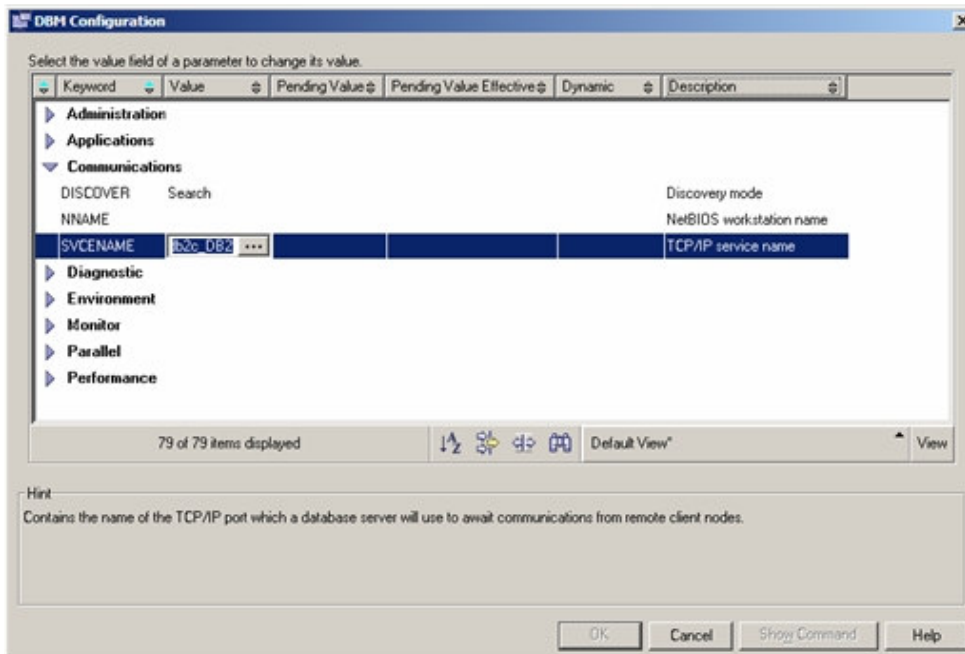
2) Biến SVCENAME

Tham số dbm cfg sẽ được thiết lập đến tên dịch vụ (như được định nghĩa trong tệp dịch vụ TCPIP) hoặc đến số hiệu cổng giao tiếp để sử dụng khi bạn muốn truy cập cơ sở dữ liệu của thể hiện này. Từ công cụ trợ giúp cấu hình, chọn *Configure -> DBM configuration* như được mô tả trong Hình 7.4



Hình 7.4 – Xem xét dbm cfg từ Trợ giúp cấu hình

Một khi bạn đã vào trong cửa sổ cấu hình của DBM, tìm đến mục Communications, và tìm kiếm tên SVCENAME. Bạn có thể thay đổi giá trị của SVCENAME thành một chuỗi hay thành số hiệu cổng giao tiếp khi cần thiết. Điều này được mô tả trong Hình 7.5



Hình 7.5 – Xem xét tham số dbm cfg của SVCENAME

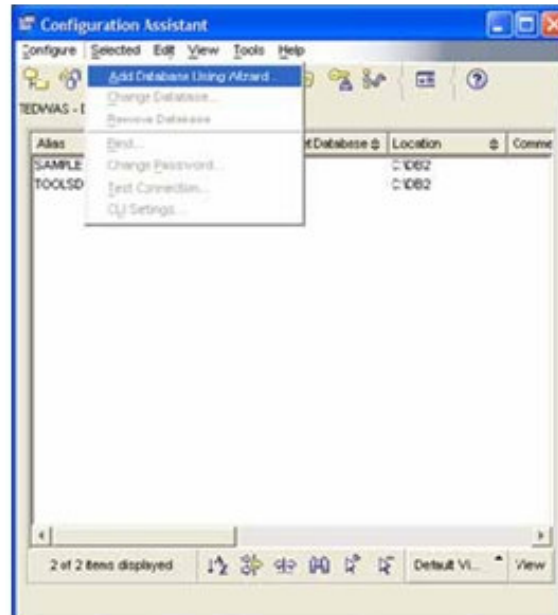
### 7.1.2 Những cài đặt cần thiết trên máy khách

Tại máy khách, bạn cần biết những thông tin sau:

1. Tên cơ sở dữ liệu mà bạn muốn kết nối đến
2. Số hiệu cổng giao tiếp của thể hiện DB2 tại máy chủ nơi cơ sở dữ liệu được cài đặt. Bạn cũng có thể sử dụng tên của dịch vụ, miễn là tồn tại một thực thể tương

- ứng trong tệp dịch vụ TCPIP
3. ID của người dùng hệ thống và mật khẩu để đăng nhập vào cơ sở dữ liệu. ID của người sử dụng này cần được định nghĩa trước tại máy chủ

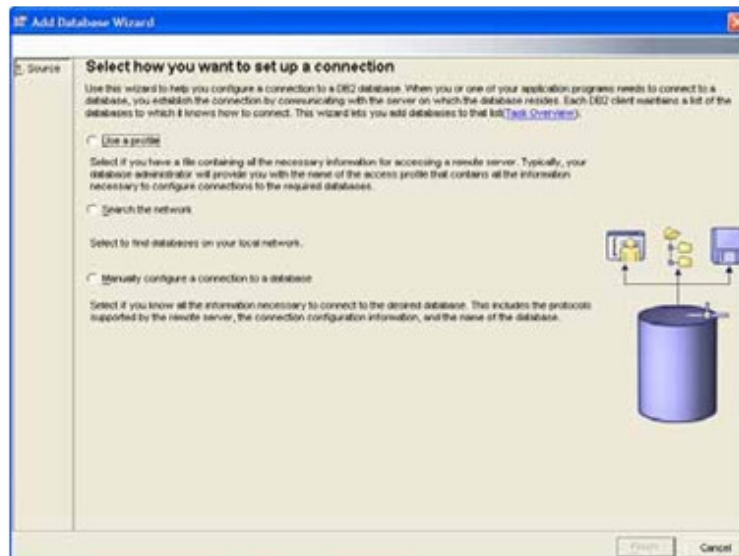
Những thông tin trên có thể được nhập vào từ máy khách DB2 sử dụng Configuration Assistant. Đầu tiên, thực hiện chức năng Add Database Wizard bằng cách kích chọn *Selected -> Add Database Using Wizard*, như được mô tả trong Hình 7.6



Hình 7.6 – Sử dụng chức năng Add Database Wizard

Bạn cũng có thể thực hiện chức năng này bằng cách nhấp chuột phải vào vùng trống trong cửa sổ trợ giúp cấu hình và chọn “Add Database Using Wizard”.

Hình 7.7 mô tả chức năng Add Database Wizard:



Hình 7.7 – Thêm CSDL (Add Database Wizard)

Trong Add Database Wizard, có ba tùy chọn:

**Sử dụng lý lịch (Profile)**

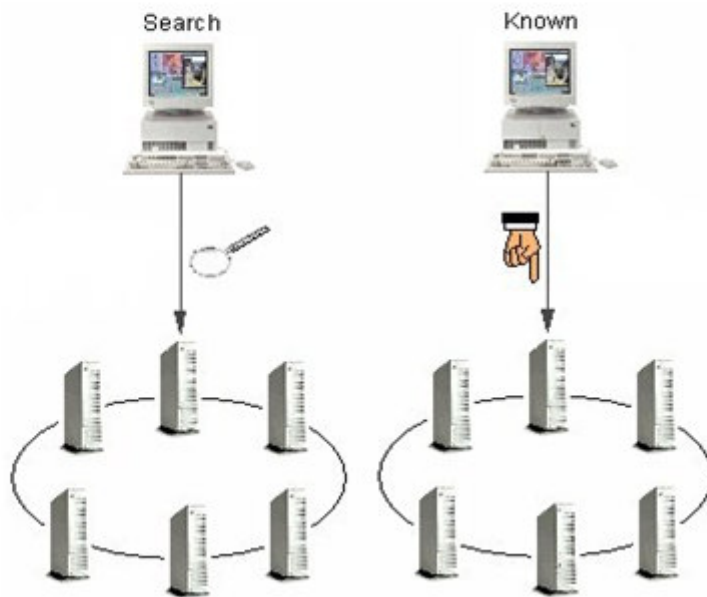
Có thể xảy ra nhiều tình huống khi bạn cần định cấu hình nhiều máy khách kết nối đến cùng một máy chủ DB2. Trong những tình huống như vậy, tùy chọn này thuận tiện để thực hiện tất cả các cấu hình từ một máy khách, và lưu cấu hình này vào trong một tệp “lý lịch”. Với tệp này, bạn có thể nạp tất cả thông tin trực tiếp đến những máy khách khác. Trong Hình 7.7, nếu bạn chọn “Use a Profile”, bạn sẽ nạp thông tin từ một “lý lịch” đang tồn tại. Chi tiết của tùy chọn được cung cấp trong mục sau của chương này sẽ mô tả cách tạo lý lịch cho máy khách và máy chủ.

**Tìm kiếm trong hệ thống mạng**

Phương pháp này cũng được biết đến như “Sự khám phá” (Discovery), yêu cầu DB2 tìm kiếm hệ thống mạng tương ứng với một máy chủ, thể hiện và cơ sở dữ liệu cho trước. Để phương pháp này hoạt động, DAS phải được chạy trên mọi máy chủ DB2 nơi mà cơ sở dữ liệu được tìm thấy. Với phương pháp này, có hai cách để thực hiện việc tìm kiếm:

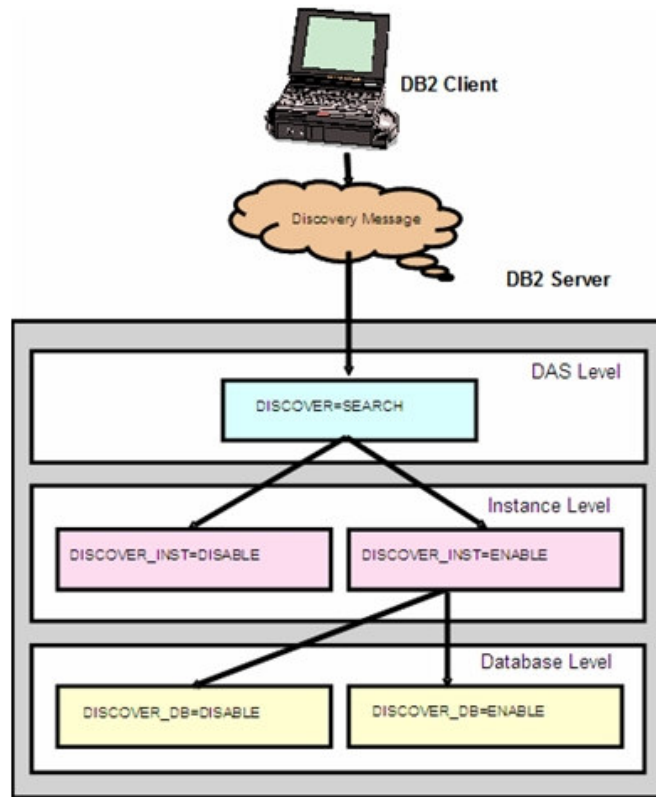
- Tìm kiếm (Search):  
 Tìm kiếm trên toàn bộ hệ thống mạng. Phương pháp này không được khuyến khích sử dụng nếu hệ thống mạng của bạn lớn và có nhiều bộ tập trung mạng (Hub/Switch), vì nó sẽ mất nhiều thời gian truy vấn dữ liệu từ mỗi hệ thống
- Đã biết (Known):  
 Tìm kiếm trên hệ thống mạng cho một máy chủ đã biết tại địa chỉ mà bạn cung cấp.

Hai phương pháp được minh họa trong Hình 7.8



**Hình 7.8 – Phương pháp tìm kiếm Search và Known**

Trong một số trường hợp, người quản trị hệ thống không muốn các máy khách được tìm kiếm trên mạng với cơ sở dữ liệu lưu trữ những thông tin cần bảo mật. Điều này có thể được ngăn chặn tại cấp độ DAS, cấp độ thể hiện hay cơ sở dữ liệu. Hình 7.9 cung cấp chi tiết về vấn đề này.



**Hình 7.9 – Cấu hình các tham số cho phép dùng Sự khám phá (discovery)**

Hình 7.9 mô tả những cấp độ khác nhau trong đó bạn có thể cho phép hay ngăn cản sử dụng Sự khám phá. Tại cấp độ DAS, bạn có thể gán tham số cho trước DISCOVER một giá trị là SEARCH hay KNOWN. Tại cấp độ thể hiện, tham số dbm cfg là DISCOVER\_INST có thể được thiết lập thành DISABLE hay ENABLE. Cuối cùng, tại cấp độ cơ sở dữ liệu, tham số DISCOVER\_DB cũng có thể được thiết lập là ENABLE hay DISABLE. Việc thiết lập những tham số này một cách hợp lý sẽ cung cấp cho bạn cách thức tìm ra cơ sở dữ liệu.

**Cấu hình một cách thủ công kết nối đến cơ sở dữ liệu**

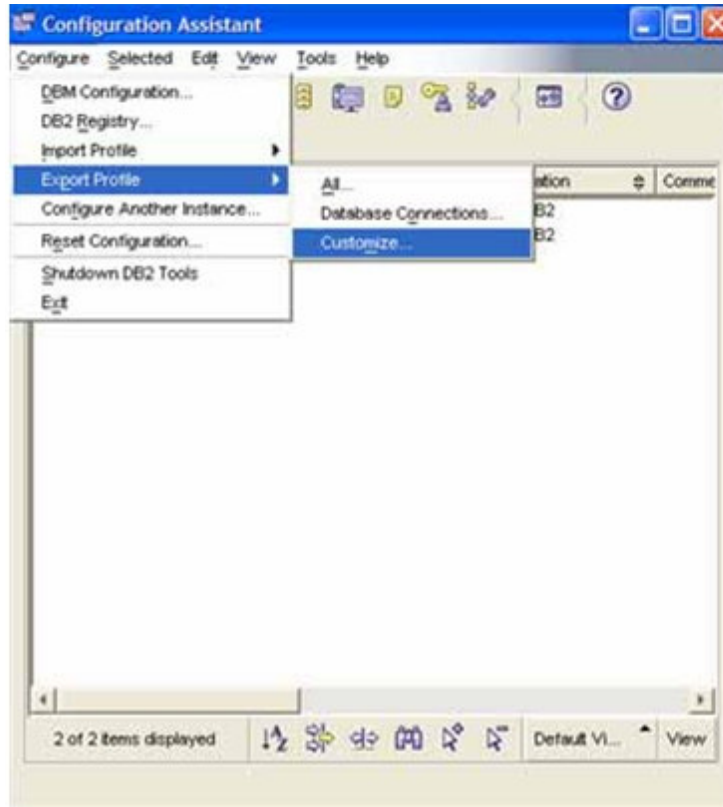
Sử dụng phương pháp này, bạn thực hiện thủ công việc thêm tên máy chủ, số hiệu cổng giao tiếp và thông tin về cơ sở dữ liệu cho công cụ Trợ giúp cấu hình, nơi sẽ phát sinh ra “danh mục” câu lệnh để thực thi việc cấu hình kết nối. Công cụ Trợ giúp cấu hình sẽ không kiểm tra thông tin có chính xác không. Chính bạn sẽ biết nó không đúng khi bạn không kết nối được đến máy chủ. Tương tự, bạn phải chắc rằng ID và mật khẩu bạn cung cấp để kết nối từ xa đến cơ sở dữ liệu là đúng. Quá trình xác minh thông tin được mặc định thực hiện trên máy chủ DB2 mà bạn đang cố gắng kết nối đến, vì vậy, bạn phải cung cấp một ID người dùng cùng với mật khẩu đã được định nghĩa trên máy chủ đó.

**7.1.3 Tạo lý lịch cho máy khách và máy chủ**

Nếu bạn đang cấu hình cho một số lượng lớn máy chủ hay máy khách, hơn nữa việc thiết lập cho mỗi máy độc lập nhau, bạn có thể chỉ cần thiết lập cho một máy duy nhất, sau đó xuất ra một lý lịch (xuất tệp cấu hình chẳng hạn), khi đó áp dụng lý lịch này đến các máy khách/ máy chủ khác. Điều này giúp tiết kiệm rất nhiều thời gian cho người quản trị hệ thống khi thiết lập môi trường.

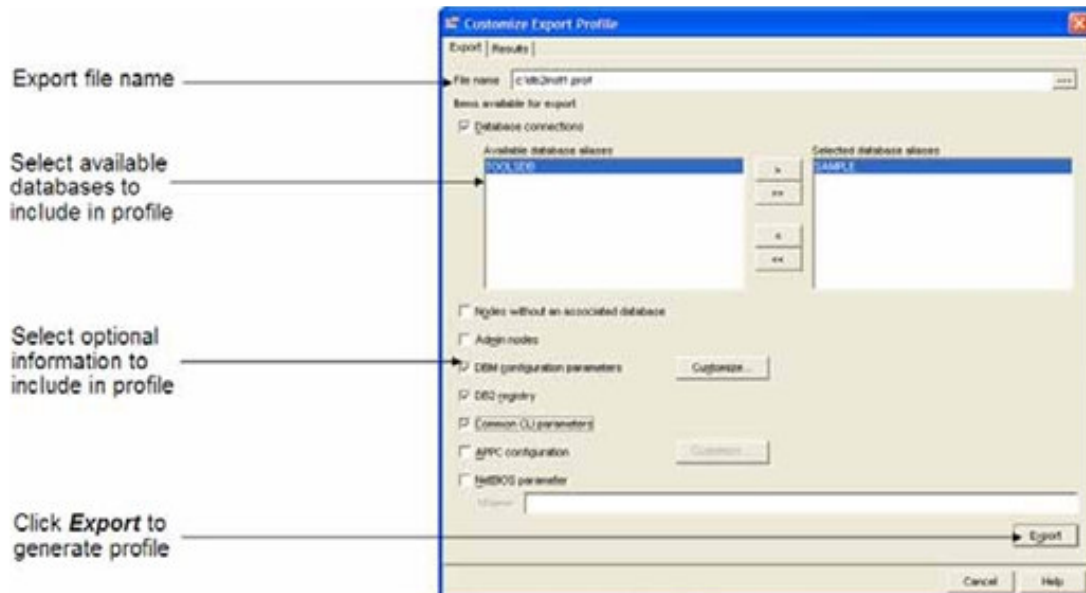
Để tạo một lý lịch tùy chỉnh theo yêu cầu, từ Trợ giúp cấu hình, kích vào trình đơn *Configure*, sau đó chọn *Export Profile -> Customize*, như được mô tả trong Hình 7.10





Hình 7.10 – Xuất một lý lịch

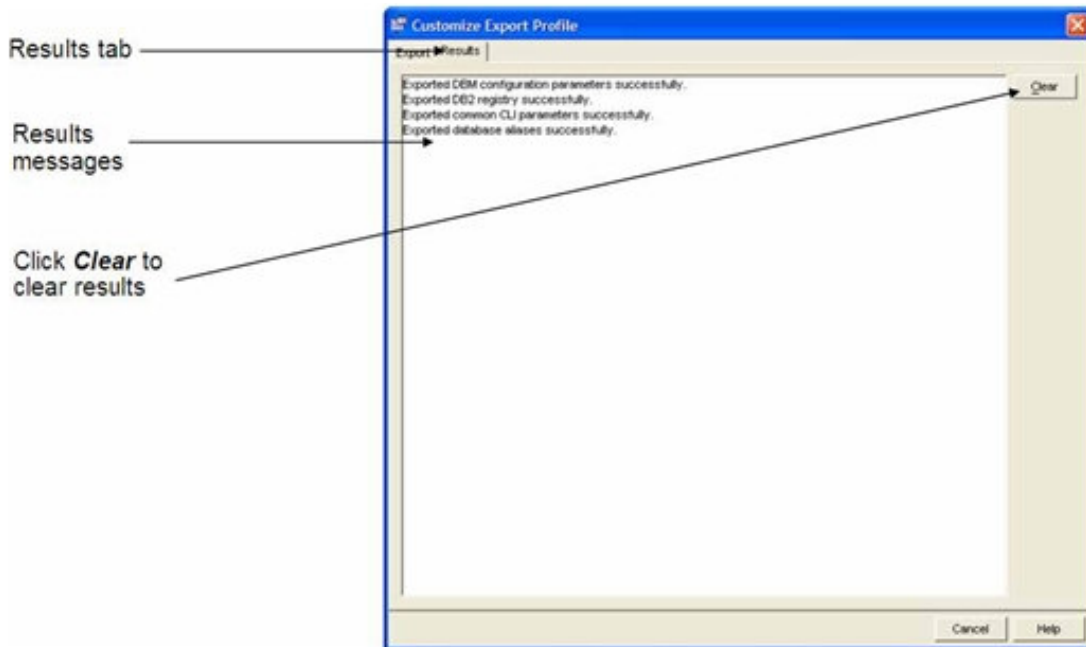
Hình 7.11 mô tả những trường cần được hoàn thành để xuất một lý lịch



Hình 7.11 – Hộp thoại tùy chỉnh xuất lý lịch

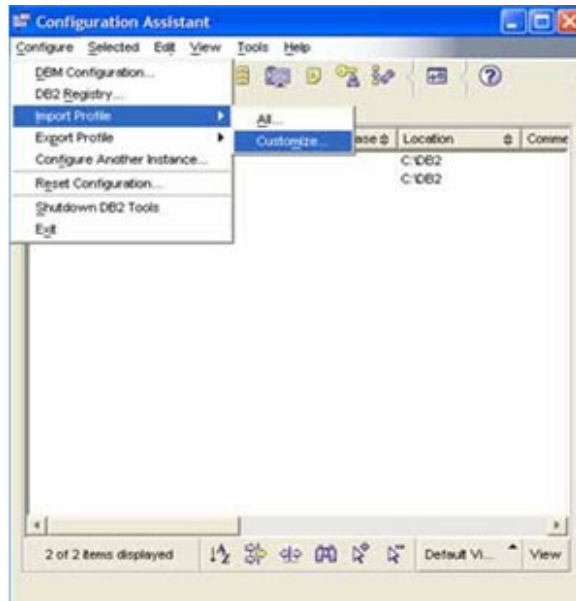
Hình 7.12 mô tả kết quả sau khi kích “Export” trong hộp thoại Customize Export Profile.





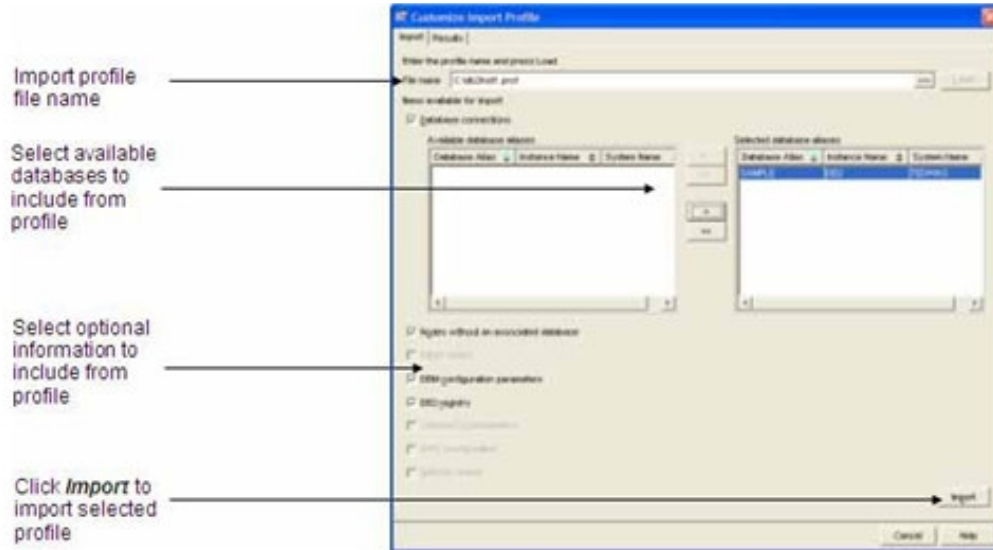
**Hình 7.12 – Xuất kết quả lý lịch**

Để nhập một lý lịch đã được tùy chỉnh từ Trợ giúp cấu hình, kích vào trình đơn *Configure*, sau đó chọn *Import Profile -> Customize*, như được mô tả trong Hình 7.13



**Hình 7.13 – Nhập một lý lịch**

Hình 7.14 mô tả những trường cần được hoàn thành để nhập một lý lịch



Hình 7.14 – Nhập lý lịch được tùy chỉnh

## Bài tập nhanh #4 – Sử dụng Trợ giúp thiết lập cấu hình

### Mục tiêu

Chương trình Trợ giúp thiết lập cấu hình có thể được sử dụng để cấu hình các kết nối cơ sở dữ liệu từ xa một cách nhanh chóng và dễ dàng. Trong bài tập nhanh này, bạn sẽ tạo chỉ mục cho một cơ sở dữ liệu nằm trên một máy chủ DB2 ở xa (remote DB2 server - được thể hiện bằng máy trạm của người ngồi bên cạnh bạn, sử dụng cả hai cách **Tim kiếm** và **Khám phá**). Một khi cơ sở dữ liệu đó đã được liệt kê, bạn sẽ có thể truy cập cơ sở dữ liệu này giống như là nó đang ở ngay trên hệ thống máy của mình. DB2 thực hiện tất cả các quá trình giao tiếp một cách thầm lặng.

### Ghi chú

Bài tập nhanh này giả định rằng bạn đang làm việc trong 1 mạng máy tính. Nếu không phải như vậy, bạn luôn có thể sử dụng chính máy tính của mình vừa làm máy chủ vừa làm máy khách và làm theo các chỉ dẫn cho công việc cấu hình dưới đây để kết nối đến chính hệ thống của bạn.

### Thủ tục

1. Hỏi người bên cạnh (hay người hướng dẫn bạn) các thông tin sau đây :

Thông tin cơ sở dữ liệu từ xa:

(PR)	Giao thức	<u>TCPIP</u>
(IP)	Địa chỉ IP hay tên máy chủ	_____
(PN)	Số cổng của thể hiện (instance)	_____
(DB)	Tên của cơ sở dữ liệu	<u>SAMPLE</u>

Các mẹo :

- Trong môi trường Windows, để lấy tên của máy ta gõ lệnh `hostname` từ cửa sổ dòng lệnh
  - Trong môi trường Windows, để lấy địa chỉ IP ta gõ lệnh `ipconfig` từ cửa sổ dòng lệnh
2. Mở chương trình Trợ giúp thiết lập cấu hình (Mẹo: chương trình này có thể truy cập thông qua lệnh Start)
  3. Mở bảng chọn *Selected* và chọn mục *Add Database Using Wizard*
  4. Trong trang *Source* của trình trợ giúp, chọn mục *Manually Configure a Connection to a Database*. Nhấn nút *Next* để di chuyển đến trang tiếp theo của trình trợ giúp.
  5. Trong trang *Protocol* của trình trợ giúp, chọn mục TCP/IP. Nhấn nút *Next* để di chuyển đến trang tiếp theo của trình trợ giúp.
  6. Trong trang *TCP/IP* của trình trợ giúp, nhập vào tên đầy đủ của máy chủ hoặc là địa chỉ IP mà bạn đã ghi lại trong bước 1. Nhập vào số cổng bạn đã ghi trong bước 1. Nhấn nút *Next* để di chuyển đến trang tiếp theo của trình trợ giúp.

Ghi chú: Tùy chọn *Service Name* có thể được sử dụng nếu bạn có một mục (entry) trong tập tin Services nội bộ với một cổng có số được định nghĩa dựa trên cổng mà máy chủ từ xa đang lắng nghe. Khi bạn sử dụng tùy chọn này, DB2 sẽ tìm kiếm trong tập tin dịch vụ trên máy của bạn chứ không phải trên máy chủ. Bạn cần phải thêm một mục vào tập tin này nếu bạn muốn sử dụng tùy chọn trên.

7. Trong trang *Cơ sở dữ liệu* của trình trợ giúp, nhập vào tên của cơ sở dữ liệu được định nghĩa trên máy chủ từ xa mà bạn đã ghi lại trong bước 1 ở trường *Database Name*. Để ý việc trường *Database Alias* được điền một cách tự động với cùng giá trị đó như thế nào. Bí danh cơ sở dữ liệu (database alias) là một cái tên mà những ứng dụng nội bộ trên máy của bạn sẽ sử dụng để kết nối đến cơ sở dữ liệu này. Bởi vì bạn đã có một cơ sở dữ liệu trên máy của mình được định nghĩa sẵn với tên là *SAMPLE*, DB2 sẽ không cho phép bạn tạo tên một cơ sở dữ liệu khác có cùng tên. Vì thế bạn phải sử dụng một tên bí danh khác. Trong ví dụ này, hãy đổi bí danh cơ sở dữ liệu thành *SAMPLE1*. Bạn có thể nhập vào một lời chú thích tùy ý về cơ sở dữ liệu này nếu bạn muốn. Nhấn nút *Next* để di chuyển đến trang tiếp theo của trình trợ giúp.
8. Trong trang *Data source* của trình trợ giúp, bạn có thể (không bắt buộc) đăng ký cơ sở dữ liệu (nguồn dữ liệu) mới này như một nguồn dữ liệu ODBC. Việc này sẽ tự động đăng ký cơ sở dữ liệu đó trong chương trình quản lý ODBC của Windows (Windows ODBC Manager) cho bạn. Trong ví dụ này, bỏ dấu đánh ở mục "*Register this database for ODBC*" tại vì bạn sẽ không sử dụng ODBC. Nhấn nút *Next* để di chuyển đến trang tiếp theo của trình trợ giúp.
9. Trong trang *Node Options* của trình trợ giúp, xác định hệ điều hành của máy chủ nơi mà cơ sở dữ liệu từ xa đang nằm tại đó. Bởi vì tất cả các máy trạm trong phòng máy này đều sử dụng Microsoft Windows, hãy đảm bảo rằng mục *Windows* trong danh sách sổ xuống được chọn. Trường *Instance name* nên được đặt là *DB2*. Nếu trường này chưa được đặt thì hãy đặt giá trị cho nó là *DB2*. Nhấn nút *Next* để di chuyển đến trang tiếp theo của trình trợ giúp.
10. Trang *System Options* của trình trợ giúp cho bạn cơ hội để đảm bảo rằng hệ thống và tên máy chủ đã được điền chính xác và xác nhận những cài đặt của hệ thống. Nhấn nút *Next* để di chuyển đến trang tiếp theo của trình trợ giúp.
11. Trang *Security Options* của trình trợ giúp cho phép bạn xác định nơi bạn muốn việc xác thực người dùng xảy ra và phương thức bạn muốn sử dụng. Chọn mục *Use authentication value in server's DBM Configuration*. Cách làm này sẽ sử dụng phương thức được xác lập bởi tham số *AUTHENTICATION* trong tập tin cấu hình của thể hiện (instance) ở xa. Nhấn nút *Finish* để tạo danh mục cơ sở dữ liệu từ xa và đóng trình trợ giúp lại. Một hộp thoại xác nhận sẽ hiện ra. Nhấn nút *Test Connection* để đảm bảo bạn có thể kết nối thành công đến cơ sở dữ liệu. Thêm vào đó, để đảm bảo tên người dùng và mật khẩu mà bạn cung cấp là hợp lệ trên máy chủ ở xa (bởi vì có thể tham số *AUTHENTICATION* của máy chủ được đặt giá trị là *SERVER*). Nếu việc thử kết nối thành công, bạn đã tạo danh mục cơ sở dữ liệu ở xa một cách thành công. Nếu việc kết nối không thành công, bạn hãy quay trở lại trình trợ giúp và đảm bảo rằng tất cả các giá trị đã được xác lập đúng (Nhấn nút *Change* để quay lại những cài đặt trong trình trợ giúp)
12. Mở trung tâm điều khiển và thử xem những bảng khác nhau trong cơ sở dữ liệu từ xa vừa mới được tạo danh mục
13. Quay trở lại chương trình Trợ giúp thiết lập cấu hình và thử tạo danh mục với một cơ sở dữ liệu khác, lần này sử dụng tùy chọn *Search the Network*. Làm từng bước trong trình trợ giúp giống như bạn đã cấu hình kết nối bằng tay. Lưu ý rằng, trên những mạng lớn, việc tìm kiếm có thể sẽ tốn thời gian để trả về các kết quả

# 8

## Chương 8 - Làm việc với các đối tượng dữ liệu

Chương này thảo luận các đối tượng dữ liệu như là dạng sơ đồ (schemas), dạng bảng (table), khung nhìn (view), chỉ mục (indexes), chuỗi (sequences)... Một số các đối tượng dữ liệu cao cấp như là các bẫy sự kiện (triggers), các hàm do người dùng định nghĩa (UDFs) và những thủ tục được lưu trữ sẽ được thảo luận trong chương 14, SQL PL lưu trữ các thủ tục, và chương 15, trong dòng SQL PL, UDFs, Triggers.

### 8.1 Sơ đồ

Sơ đồ là không gian tên (*name spaces*) cho tập hợp các đối tượng dữ liệu. Chức năng chính của chúng như sau:

- Cung cấp cho bạn các chỉ dẫn của quyền sở hữu các đối tượng hay các mối quan hệ đối với một ứng dụng.
- Gom nhóm các đối tượng có mối quan hệ với nhau.

Tất cả các đối tượng dữ liệu trong DB2 có hai phần tên theo tiêu chuẩn; sơ đồ là nửa phần đầu tiên của tên:

```
<tên_sơ đồ>.<tên_đối tượng>
```

Tên đối tượng theo quy chuẩn phải là duy nhất. Khi bạn kết nối cơ sở dữ liệu và tạo hay tham chiếu đến một đối tượng mà không định rõ tên sơ đồ, DB2 sẽ lấy mã tài khoản mà bạn kết nối với cơ sở dữ liệu làm tên sơ đồ. Ví dụ, nếu bạn kết nối dữ liệu mẫu với tên sử dụng là "arfchong", và tạo một bảng bằng cách sử dụng phát biểu CREATE TABLE

```
CREATE TABLE artist ...
```

Tên theo quy chuẩn đầy đủ của bảng được tạo ra sẽ là arfchong.artists.

### 8.2 Bảng

Một bảng là một tập hợp các dữ liệu có mối quan hệ với nhau một cách hợp lý theo các cột và hàng. Phát biểu bên dưới cung cấp cho bạn ví dụ về việc làm cách nào để tạo ra một bảng bằng cách sử dụng phát biểu CREATE TABLE

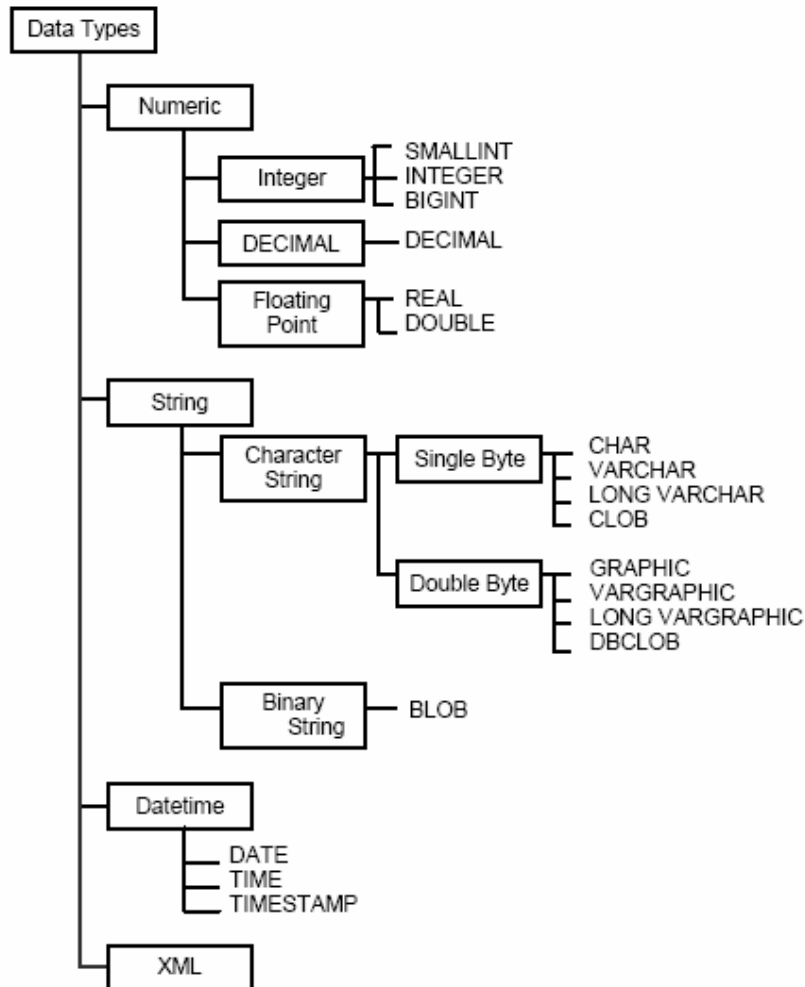
```
CREATE TABLE artists
{
    artno          SMALLINT    not null,
    name           VARCHAR (50) with default 'abc',
    classification CHAR (1) not null,
    bio            CLOB (100K) logged,
    picture        BLOG (2M) not logged compact
}
```

In mytblsl

Trong các phần tiếp theo đây, chúng ta có thể mô tả các phần chính của phát biểu CREATE TABLE này.

### 8.2.1 Kiểu dữ liệu

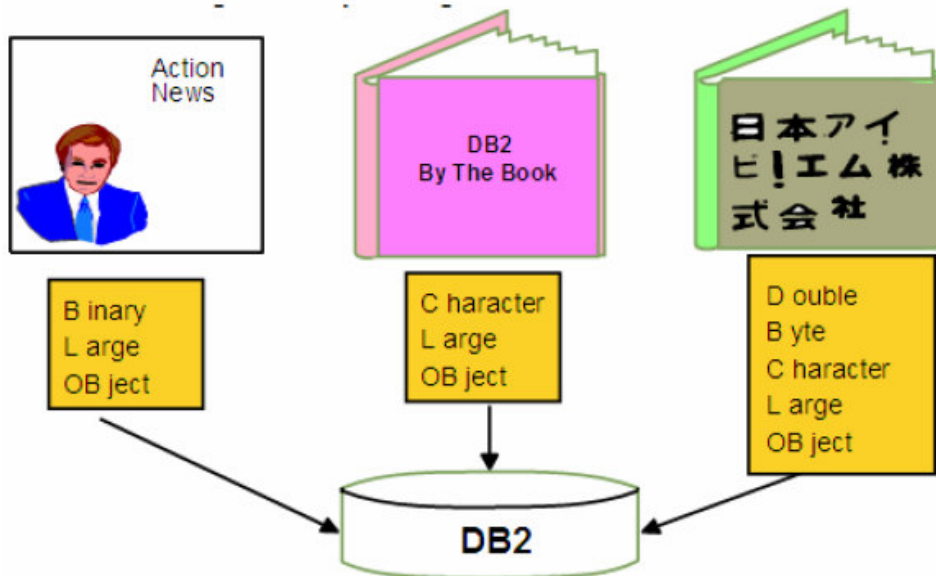
Hình 8.1 Danh sách các kiểu dữ liệu được hỗ trợ trong DB2



Hình 8.1 – Các kiểu dữ liệu của DB2

#### Các kiểu dữ liệu đối tượng lớn (LOB)

Các kiểu dữ liệu đối tượng lớn được dùng để lưu trữ các chuỗi lý tự lớn, các chuỗi nhị phân lớn, hay các tập tin được chỉ ra trong hình 8.2



Hình 8.2 – Các kiểu dữ liệu LOB

Các đối tượng kiểu nhị phân lớn này thường được viết tắt để cho dễ hiểu: Một đối tượng nhị phân lớn phải là một BLOB, một đối tượng ký tự lớn là CLOB, và đối tượng ký tự lớn 2 byte cũng được biết như là DBCLOB.

**Các kiểu dữ liệu được định nghĩa bởi người dùng**

DB2 cho phép bạn có thể tự định nghĩa kiểu dữ liệu của chính mình, dựa trên kiểu dữ liệu của hệ thống. Điều này được xem như là việc có bộ kiểu dữ liệu do người dùng tự định nghĩa (UDTs). UDTs được dùng khi:

- Có nhu cầu thiết lập ngữ cảnh cho các giá trị
- Có nhu cầu cần ép kiểu dữ liệu DB2.

Các phát biểu theo sau đây mô tả một ví dụ về việc sử dụng UDTs như thế nào và khi nào

```
CREATE DISTINCT TYPE POUND AS INTEGER WITH COMPARISONS
CREATE DISTINCT TYPE KILOGRAM AS INTEGER WITH COMPARISONS
CREATE TABLE person
{
    F_name      VARCHAR (30),
    Weight_p    POUND NOT NULL,
    Weight_k    KILOGRAM NOT NULL
}
```

Trong ví dụ này, có 2 UDTs được tạo ra là: POUND and KILOGRAM. Cả hai đều được xây dựng dựa trên kiểu dữ liệu sẵn có của hệ thống *INTEGER*. Mệnh đề *WITH COMPARISONS* được định nghĩa như là phần cú pháp chỉ ra rằng việc thay các hàm với cùng tên cũng giống như việc các kiểu dữ liệu sẽ được tạo ra.

Bảng *person* dùng 2 UDTs mới trong cột *weight\_p* và *weight\_k* theo tuần tự. Nếu bây giờ chúng ta đưa ra phát biểu sau:

```
SELECT F_NAME FROM PERSON
WHERE weight_p > weight_k
```

Bạn sẽ nhận được một thông báo lỗi vì việc so sánh 2 cột với kiểu dữ liệu khác nhau. Thậm chí `weight_p` và `weight_k` dùng dữ liệu là POUND và KILOGRAM, theo thứ tự, cả hai trường này đều được dựa trên kiểu dữ liệu là INTEGER. Tuy nhiên bằng việc tạo UDTs thì bạn không thể thực hiện loại so sánh này. Đây chính là những gì bạn muốn, bởi vì trong cuộc sống thực, làm thế nào để so sánh giữa pound và kilogram? Nó sẽ vô lý.

Trong ví dụ kế tiếp, bạn muốn so sánh cột `weight_p` với kiểu dữ liệu integer. Tuy nhiên hai kiểu dữ liệu này khác nhau, vì thế bạn sẽ nhận được thông báo lỗi trừ khi bạn sử dụng một hàm đã ép kiểu.

Như bạn thấy trong phát biểu bên dưới, chúng tôi sẽ dùng hàm ép kiểu `POUND()` để điều so sánh này có thể thực hiện được. Như đã nêu ở trên hàm ép kiểu `POUND()` được tạo ra với UDT khi triệu gọi WITH COMPARISONS của phát biểu `CREATE DISTINCT TYPE`.

```
SELECT F_NAME FROM PERSON
WHERE weight_p > POUND (30)
```

### Các giá trị NULL

Một giá trị null đại diện cho một trạng thái không xác định. Tuy nhiên, phát biểu `CREATE TABLE` có thể định nghĩa một cột bằng cách sử dụng mệnh đề `NOT NULL`. Điều này đảm bảo rằng cột này sẽ chứa giá trị dữ liệu được biết đến. Bạn cũng có thể chỉ định giá trị mặc định cho cột nếu `NOT NULL` được khai báo. Phát biểu kế tiếp cung cấp ví dụ cho điều này:

```
CREATE TABLE Staff
{
    ID SMALLINT NOT NULL,
    NAME VARCHAR (9),
    DEPT SMALLINT NOT NULL with default 10,
    JOB CHAR (5),
    YEARS SMALLINT,
    SALARY DECIMAL (7, 2),
    COMM DECIMAL (7, 2) with default 15
}
```

### 8.2.2 Cột khoá chính (Identity columns)

Cột khoá chính này là một con số được tạo ra tự động, số này là duy nhất cho mỗi dòng. Chỉ có một cột khoá chính cho một bảng.

Có hai cách tạo ra giá trị cho cột khoá chính, điều này phụ thuộc vào việc nó được định nghĩa như thế nào:

- Luôn luôn được sinh ra: những giá trị luôn luôn được DB2 sinh ra. Các ứng dụng thì không được phép cung cấp những giá trị cụ thể.
- Sinh ra mặc định: giá trị có thể được cung cấp cụ thể bởi một ứng dụng hoặc, nếu không có giá trị cho trước, thì DB2 sẽ sinh ra. DB2 không thể đảm bảo được tính duy nhất. Tùy chọn này được dự định phục vụ cho việc truyền dữ liệu, và cho việc unload và reload của một bảng.

Hãy xem qua ví dụ sau:



```
CREATE TABLE subscriber (subscriberID INTEGER GENERATED ALWAYS AS
    IDENTITY (START WITH 100 INCREMENT BY 100),
    firstname VARCHAR (50),
    lastname VARCHAR (50))
```

Trong ví dụ này, cột subscriberID là một INTEGER được định nghĩa như là một cột khoá chính thuộc loại luôn luôn được sinh ra tự động. Giá trị được sinh ra sẽ bắt đầu từ 100, và nó sẽ được tăng lên với bước là 100.

### 8.2.3 Các đối tượng tuần tự (Sequence object)

Các đối tượng tuần tự sinh ra một con số duy nhất dựa vào cơ sở dữ liệu. Không giống như các cột khoá chính, các đối tượng tuần tự này là những bảng độc lập, xem ví dụ minh họa ở phát biểu sau:

```
CREATE TABLE t1 (salary int)
CREATE SEQUENCE myseq
    START WITH 10
    INCREMENT BY 1
    NO CYCLE
INSERT INTO t1 VALUES (nextval for myseq)
INSERT INTO t1 VALUES (nextval for myseq)
INSERT INTO t1 VALUES (nextval for myseq)
SELECT * FROM t1
```

SALARY

```
-----
      10
      11
      12
3 record(s) selected
```

```
SELECT prevval for myseq FROM sysibm.sysdummy1
```

```
1
-----
      12
1 record(s) selected
```

PREVVAL cung cấp cho bạn giá trị hiện tại của dãy tuần tự, trong khi NEXTVAL cung cấp giá trị kế tiếp.

Ví dụ bên trên cũng sử dụng SYSIBM.SYSDUMMY1. đây là một bảng danh mục hệ thống mà nó chỉ có một cột và một hàng. Nó có thể được dùng trong tình huống như một câu truy vấn yêu cầu trả về chỉ một giá trị. Các bảng danh mục hệ thống được mô tả trong phần kế tiếp.

### 8.2.4. Các bảng danh sách hệ thống

Mỗi cơ sở dữ liệu có các bảng danh sách hệ thống của riêng nó. Các bảng này lưu trữ siêu dữ liệu về các đối tượng cơ sở dữ liệu. Bạn có thể truy vấn những bảng này như bất kỳ bảng cơ sở dữ liệu bình thường khác.

Ba sơ đồ được sử dụng để nhận dạng các bảng danh sách hệ thống:

- **SYSIBM:** các bảng cơ sở, tối ưu hóa sử dụng DB2
- **SYSCAT:** khung nhìn (view) dựa trên bảng SYSIBM, tối ưu sự dễ sử dụng.

- SYSSTAT: các số liệu thống kê cơ sở dữ liệu

Sau đây là một vài ví dụ về các khung nhìn của danh mục:

- SYSCAT.TABLES
- SYSCAT.INDEXES
- SYSCAT.COLUMNS
- SYSCAT.FUNCTIONS
- SYSCAT.PROCEDURES

### 8.2.5. Khai báo các bảng tạm thời

Khai báo bảng tạm thời là các bảng được tạo ra trong bộ nhớ, được sử dụng bởi một ứng dụng và được tự động xóa khi ứng dụng kết thúc. Những bảng này chỉ có thể được truy nhập bởi ứng dụng tạo ra chúng. Không có bản ghi nào được sinh trong bảng danh sách của DB2. Truy nhập những bảng này có hiệu quả rất cao bởi vì không có sự tranh chấp danh sách, các hàng không bị khóa, không ghi nhật ký (tùy chọn), và không kiểm tra quyền. Cũng có hỗ trợ chỉ mục cho các bảng tạm thời này, đó là, bất cứ chỉ mục chuẩn nào cũng có thể được tạo trên một bảng tạm thời. Bạn cũng có thể chạy lệnh RUNSTATS với các bảng này.

Khai báo các bảng tạm thời ở bên trong một không gian sử dụng của bảng tạm thời, đây là cái phải được định nghĩa trước khi tạo ra bất cứ bảng tạm thời nào. Các phát biểu dưới đây cung cấp một ví dụ làm thế nào để tạo ba khai báo bảng tạm thời:

```
CREATE USER TEMPORARY TABLESPACE apptemps
    MANAGED BY SYSTEM USING ('apptemps');
```

```
DECLARE GLOBAL TEMPORARY TABLE employees
    LIKE employee NOT LOGGED;
```

```
DECLARE GLOBAL TEMPORARY TABLE tempdept
    (deptid CHAR(6), deptname CHAR(20))
    ON COMMIT DELETE ROWS NOT LOGGED;
```

```
DECLARE GLOBAL TEMPORARY TABLE tempprojects
    AS (fullselect) DEFINITION ONLY
    ON COMMIT PRESERVE ROWS NOT LOGGED
    WITH REPLACE IN TABLESPACE apptemps;
```

Khi một khai báo bảng tạm thời được tạo ra, sơ đồ của nó là SESSION, và phải được chỉ tường minh. ID người dùng được sử dụng để tạo một bảng tạm thời sẽ có tất cả quyền trên bảng. Mỗi ứng dụng tạo một bảng tạm thời sẽ quyền độc lập của riêng nó được chỉ ra trong Figure 8.5.

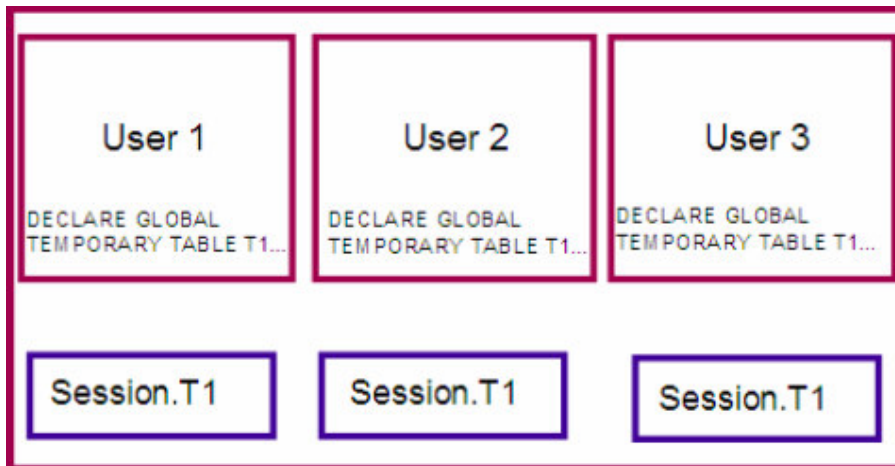


Figure 8.5 – Phạm vi của các bảng tạm thời toàn cục công khai

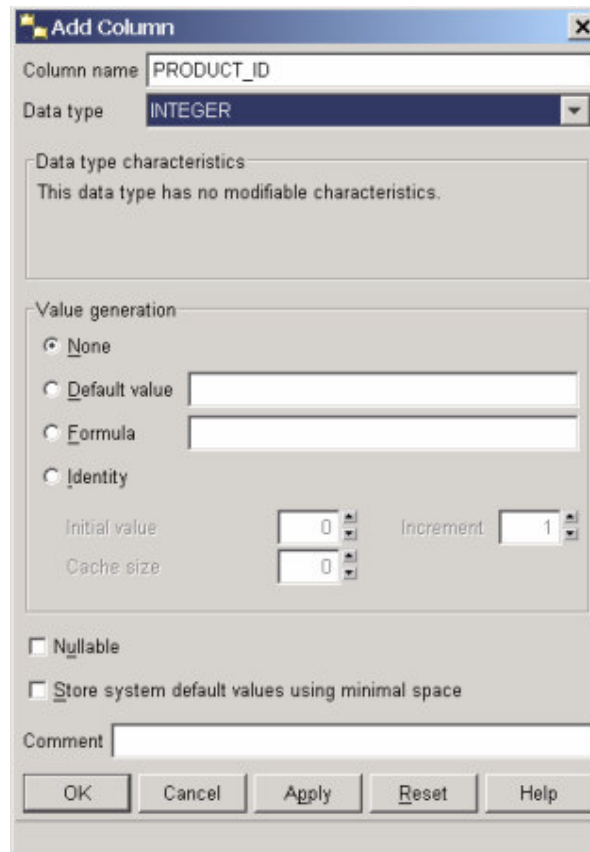
### Bài tập nhanh #5 - Khởi tạo một bảng mới

#### Mục tiêu

Đến đây, bạn có thể sử dụng những bảng có sẵn trong cơ sở dữ liệu *SAMPLE* để minh họa các khái niệm. Cuối cùng, bạn sẽ cần tạo một cơ sở dữ liệu gồm các bảng của riêng mình. Trong phần bài tập này, bạn sẽ sử dụng tạo bảng theo hướng dẫn từng bước để tạo hai bảng mới trong cơ sở dữ liệu *SAMPLE*.

#### Thủ tục

1. Mở Create Table Wizard như đã chỉ ra ở trên trong phần giới thiệu.  
(Control Center -> All Databases -> *SAMPLE* -> (nhấn phải chuột) Tables object -> Create ...)
2. Định nghĩa tên bảng, định nghĩa cột và các ràng buộc. Bảng sẽ được sử dụng để lưu trữ thông tin về cung cấp văn phòng phẩm, được sử dụng bởi một dự án trong cơ sở dữ liệu *SAMPLE*. Mỗi lần văn phòng phẩm được mua, một hàng sẽ được thêm vào trong bảng này. Bảng sẽ có sáu cột:
  - product\_id: nhận dạng duy nhất của sản phẩm đang được mua
  - description: mô tả về sản phẩm
  - quantity: số lượng mua
  - cost: giá của sản phẩm
  - image: một bức ảnh của sản phẩm (nếu dùng được)
  - project\_num: Mã số của dự án mua sản phẩm này
3. Ở trang đầu tiên của hướng dẫn từng bước, đối với tên sơ đồ, gõ vào ID bạn sử dụng để đăng nhập, và sử dụng tên bảng là *SUPPLIES*. Bạn cũng có thể gõ vào một ghi chú tùy ý. Nhấn vào nút Next để tiếp tục tới trang sau của hướng dẫn.
4. Từ trang này, bạn có thể thêm các cột vào bảng. Nhấn nút *ADD* để thêm các cột.



Gõ “*product\_id*” vào Column name và chọn kiểu là *INTEGER*. Bỏ dấu kiểm ở *Nullable*, và nhấn vào nút *Apply* để định nghĩa cột.

Lặp lại bước này cho những cột còn lại của bảng sử dụng các tùy chọn được chỉ trong bảng ở trên. Khi tất cả các cột đã được thêm vào, nhấn nút *OK* và danh sách các cột bạn vừa tạo sẽ được tổng kết. Nhấn nút *Next* để tiếp tục tới trang kế tiếp của hướng dẫn.

**Tên cột**

product\_id (completed)  
 description  
 quantity  
 cost  
 image  
 project\_num

**Thuộc tính**

INTEGER, NOT NULL  
 VARCHAR, length 40, NOT NULL  
 INTEGER, NOT NULL  
 DECIMAL, Precision 7, Scale 2, NOT NULL  
 BLOB, 1MB, NULLABLE, NOT LOGGED  
 CHAR, length 6, NOT NULL

Chú ý: Tùy chọn NOT LOGGED có thể được chỉ rõ khi khai báo các cột LOB. Điều này là bắt buộc với các cột có kích thước lớn hơn 1GB. Nó cũng được khuyên dùng cho các cột LOB lớn hơn 10MB bởi vì sự thay đổi các cột lớn có thể làm đầy tệp này rất nhanh. Ngay cả nếu NOT LOGGED được sử dụng, sự thay đổi các tệp LOB trong quá trình giao dịch xảy ra, vẫn có thể quay lui về trạng thái lúc đầu. Cũng chú ý rằng một mình cột hình ảnh được định nghĩa như một cột “NULLABLE”. Bạn có nghĩ tại sao cột được định nghĩa như thế?

- Ở đây, tất cả thông tin bắt buộc cho việc tạo bảng đều được cung cấp. Bằng cách nhảy sang các trang khác, bạn coi như đang chọn các giá trị mặc định cho những tùy chọn đó. Bạn có thể thêm các khóa và ràng buộc sau khi một bảng vừa được tạo.

6. Thêm một ràng buộc tới bảng để hạn chế các giá trị trên cột *quantity*. Trên trang *Constraint* của cửa sổ thao tác, nhấn nút *ADD*. Trong trường *Check Name*, gõ vào: *valid\_quantities*. Trong trường *Check Condition*, gõ vào: *quantity > 0*.

Nhấn nút *OK*. Bạn nên nhìn một tổng kết của ràng buộc bạn vừa thêm vào trong trang *Constraint* của cửa sổ thao tác. Nhấn nút *Next* để tiếp tục tới trang tiếp theo của màn hình thao tác.

7. Bạn có thể tiếp tục hoàn thành thao tác, thay đổi các tham số khác của bảng. Hoặc, bạn có thể nhảy đến trang *Summary*, hay đơn giản là nhấn vào nút *Finish* để tạo bảng.
8. Từ trung tâm điều khiển, nhấn vào thư mục *Tables* dưới cơ sở dữ liệu *SAMPLE* trong khung cây đối tượng. Bảng bạn vừa tạo bây giờ sẽ xuất hiện trong danh sách. Có thể phải làm tươi cửa sổ trung tâm điều khiển để thấy được sự thay đổi này.

### 8.3 Các khung nhìn (Views)

Các khung nhìn là sự trình bày hay hiển thị của dữ liệu trong bảng. Dữ liệu của các khung nhìn không được chứa riêng biệt, nhưng sẽ được tồn tại khi được gọi tới. Với các khung nhìn lồng nhau, một khung nhìn được tạo làm nền cho các khung nhìn khác. Các thông tin về khung nhìn ở trong mục lục khung nhìn DB2: SYSCAT.VIEWS, SYSCAT.VIEWDEP và SYSCAT.TABLES. Sau đây là cách tạo và sử dụng khung nhìn:

```
CONNECT TO MYDB1;
CREATE VIEW MYVIEW1 AS SELECT ARTNO, NAME, CLASSIFICATION
FROM ARTISTS;

SELECT * FROM MYVIEW1;
```

Kết quả đầu ra:

Artno	Name	Classification
10	HUMAN	A
20	MY PLANT	C
30	THE STORE	E

### 8.4 Các chỉ mục (Index)

Chỉ mục là tập các khóa được sắp xếp theo mục ở các dòng trong bảng. Chỉ mục cho phép trùng, và nó được phép thay đổi. Một số đặc điểm bạn có thể định nghĩa chỉ mục:

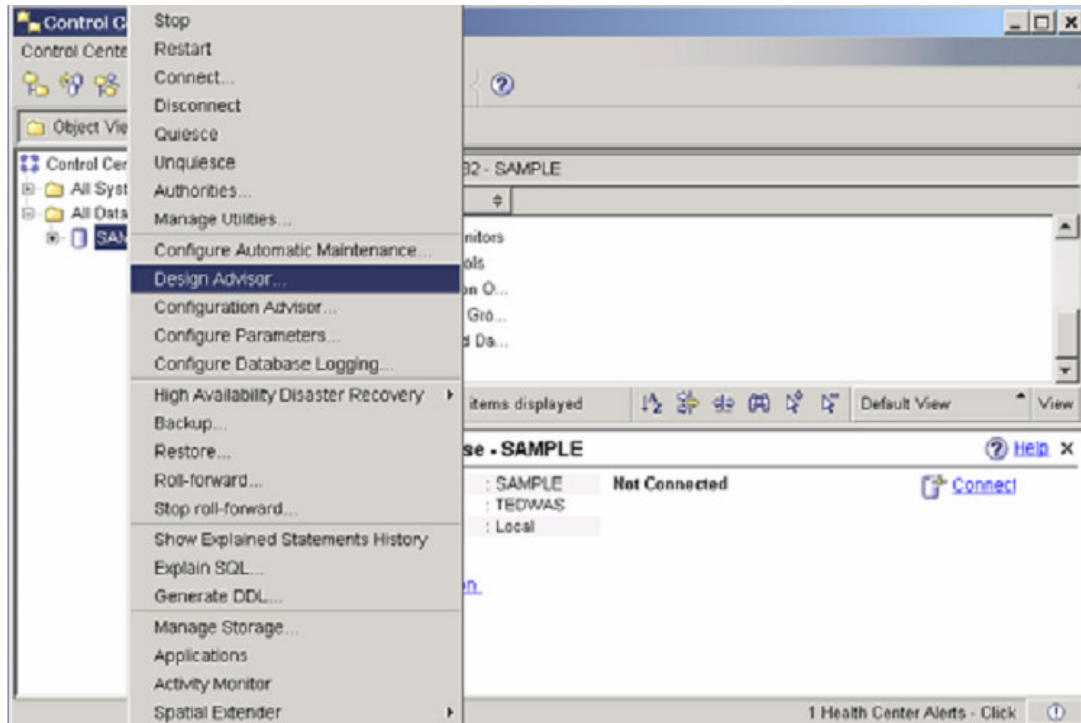
- Kiểu chỉ mục tăng hoặc giảm.
- Khóa của chỉ mục có thể là duy nhất hoặc không duy nhất.
- Nhiều cột có thể sử dụng làm chỉ mục (được gọi là các chỉ mục ghép).
- Nếu chỉ mục và các dữ liệu vật lý hợp lại giống như các chuỗi chỉ mục, được gọi là các bó chỉ mục.

Ví dụ:

```
CREATE UNIQUE INDEX artno_ix ON artists (artno)
```

### 8.4.1 Trợ giúp thiết kế (Design Advisor)

Trợ giúp thiết kế là một công cụ khá tốt để giúp bạn tối ưu thiết kế cơ sở dữ liệu cho một công việc đã biết. Trợ giúp thiết kế có thể giúp bạn bằng cách thiết kế các chỉ mục cho bạn, Materialized Query Tables (MQTs), Multi-dimension clustering (MDC) và các tính năng phân tán cơ sở dữ liệu. Trợ giúp thiết kế được gọi từ Trung tâm điều khiển, bấm chuột phải trên cơ sở dữ liệu và chọn “Design Advisor” như hình 8.6:



Hình 8.6 – Thực hiện trợ giúp thiết kế từ Trung tâm điều khiển

Hình 8.7 Cho thấy trợ giúp thiết kế. Theo sau là các bước hướng dẫn đạt được cách thiết kế gợi ý từ DB2.



Hình 8.7 – Trợ giúp thiết kế

### 8.5 Tính toàn vẹn tham chiếu

Toàn vẹn tham chiếu cho phép cơ sở dữ liệu quản lý được các quan hệ giữa các bảng. Bạn có thể thiết lập kiểu cha-con (parent-child) của các quan hệ giữa các bảng như hình 8.8. Trong hình có hai bảng DEPARTMENT và EMPLOYEE được liên kết bởi mã số của phòng (department). Cột WORKDEPT trong bảng EMPLOYEE chỉ có thể chứa mã số của PHÒNG đã tồn tại trong bảng DEPARTMENT. Đó là lý do trong ví dụ này, bảng DEPARTMENT được gọi là bảng cha, bảng EMPLOYEE được gọi là bảng con hoặc bảng phụ thuộc. Hình sau trình bày cách dùng CREATE TABLE để tạo bảng EMPLOYEE cần cho thiết lập các quan hệ.

DEPARTMENT table (Parent table)

DEPTNO (Primary key) Or unique constraint	DEPTNAME	MGRNO
-------------------------------------------------	----------	-------

EMPLOYEE table (Dependent table)

EMPNO (Primary key)	FIRSTNAME	LASTNAME	WORKDEPT (Foreign key)	PHONENO
------------------------	-----------	----------	---------------------------	---------

```

create table employee (empno .....
primary key empno
foreign key workdept
reference department on delete no action)
in DMS01
    
```

Hình 8.8 - Ví dụ về toàn vẹn tham chiếu giữa các bảng.

Trong toàn vẹn tham chiếu, thường sử dụng các khái niệm sau:

<b>Khái niệm</b>	<b>Mô tả</b>
Bảng cha (Parent table)	Bảng dữ liệu điều khiển chứa khóa chính
Bảng phụ thuộc (Dependant table)	Bảng dữ liệu phụ thuộc vào bảng cha. Chứa khóa ngoại, mỗi dòng tồn tại trong bảng phụ thuộc phải phù hợp với các dòng trong bảng cha.
Khóa chính (Primary key)	Định nghĩa khóa chính của bảng cha, nó không được có giá trị NULL và phải là duy nhất. Khóa chính có thể là một hoặc nhiều cột trong bảng.
Khóa ngoại (Foreign key)	Tham chiếu đến khóa ngoại của bảng cha.

Dữ liệu trong một bảng có thể liên kết với dữ liệu của một hoặc nhiều bảng thông qua toàn vẹn tham chiếu. Các ràng buộc có thể được áp đặt để dữ liệu phù hợp với đặc tính nào đó hoặc các quy tắc trong kinh doanh của dữ liệu. Ví dụ, nếu một cột của bảng là giới tính, nó có thể được ép để chứa một trong hai giá trị là "M" cho nam và "F" cho nữ.

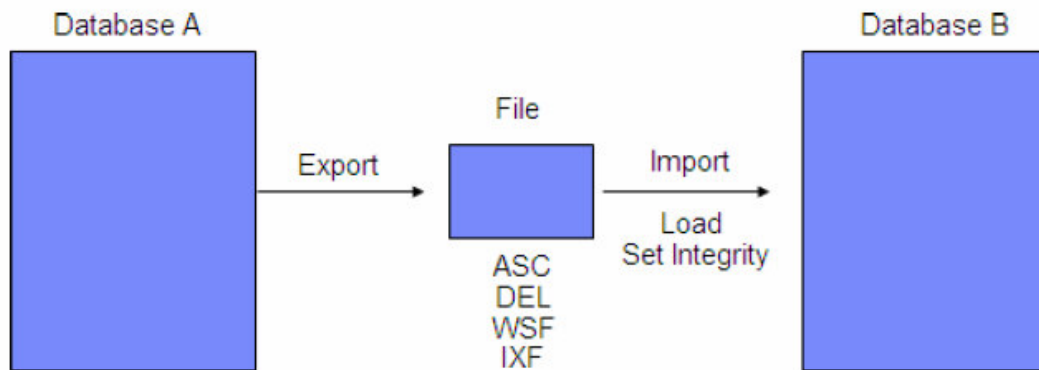


# 9

## Chương 9 – Công cụ di chuyển dữ liệu

Những công cụ hay câu lệnh được mô tả trong phần này được sử dụng để di chuyển dữ liệu trong cùng một cơ sở dữ liệu hay qua những cơ sở dữ liệu khác nhau nằm trên cùng một nền (platform) hoặc có thể khác nền.

Hình 9.1 mô tả tổng quan về công cụ di chuyển dữ liệu



Hình 9.1 - Công cụ di chuyển dữ liệu.

Trong hình 9.1 có 2 cơ sở dữ liệu: A và B. Bằng cách sử dụng công cụ EXPORT, ta có thể kết xuất dữ liệu từ một bảng sang một tệp. Tệp có thể thuộc các định dạng sau:

ASC = ASCII

DEL = Delimited ASCII – ASCII có định giới hạn.

WSF = Worksheet format – Định dạng kiểu bảng tính.

IXF = Integrated Exchange Format – Định dạng trao đổi thống nhất.

Những tệp dạng ASC hoặc DEL là những tệp văn bản và có thể mở và xem lại bằng bất kỳ trình soạn thảo văn bản nào. WSF là một loại định dạng mà có thể di chuyển dữ liệu sang bảng tính chẳng hạn Excel hay là Lotus® 1-2-3. IXF là một định dạng không chỉ bao gồm dữ liệu mà còn là một ngôn ngữ định nghĩa dữ liệu (Data Definition Language-DDL). Điều này rất tiện dụng, vì khi một bảng cần được xây dựng lại, ta có thể xây dựng một cách trực tiếp từ tệp có định dạng IXF, còn những tệp ở định dạng khác thì ta không thể làm được việc này.

Khi dữ liệu được kết xuất sang dạng tệp, công cụ IMPORT có thể được dùng để chuyển dữ liệu từ dạng tệp sang dạng bảng. Bảng phải tồn tại sẵn dưới khuôn mẫu ASC, DEL và WSF, ngoại trừ định dạng IXF. Một phương thức khác là công cụ LOAD dùng để chuyển nạp dữ liệu vào trong bảng. Công cụ LOAD thì nhanh hơn vì nó đi trực tiếp đến những trang cơ sở dữ liệu mà không cần tương tác đến hệ thống DB2; tuy nhiên, phương thức này không thể tạo ra những kiểm tra ràng buộc đối với dữ liệu, và sẽ

không thực hiện được các bất sự kiện (nếu có). Để đảm bảo một cách kiên định việc nạp dữ liệu dùng công cụ LOAD, lệnh SET INTEGRITY thường được sử dụng theo sau.

Những phần tiếp theo sau đây sẽ mô tả những công cụ EXPORT, IMPORT và LOAD một cách chi tiết.

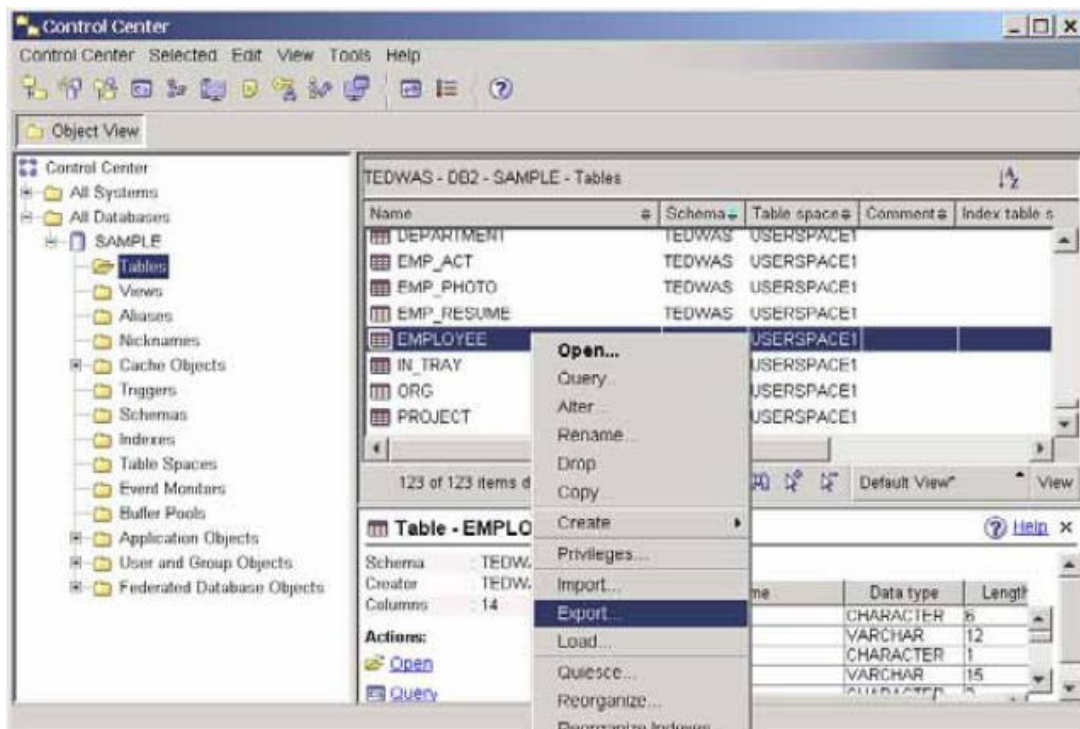
### 9.1 Công cụ EXPORT (xuất)

Công cụ EXPORT được dùng để trích dữ liệu từ bảng thành một tệp như đã đề cập trước đây. Bên dưới đây, một thao tác SQL SELECT được gọi thực thi. Ví dụ dưới đây kết xuất thành tệp *employee.ixf* thuộc định dạng IXF có 10 dòng từ bảng *employee*.

```
EXPORT TO employee.ixf OF IXF
SELECT * FROM employee
FETCH FIRST 10 ROWS ONLY
```

Chúng tôi khuyến khích bạn nên thử chạy ví dụ trên. Bảng *employee* là một bộ phận của cơ sở dữ liệu SAMPLE, vì thế trước hết bạn nên kết nối đến cơ sở dữ liệu này, nó đã được tạo ra trong chương trước.

Nếu bạn thích làm việc với những công cụ có giao diện đồ họa (GUI), công cụ EXPORT cũng có thể được thực hiện từ Trung tâm điều khiển (Control Center) như trong hình 9.2 bên dưới.



**Hình 9.2 - Hiện thị hộp thoại bảng EXPORT**

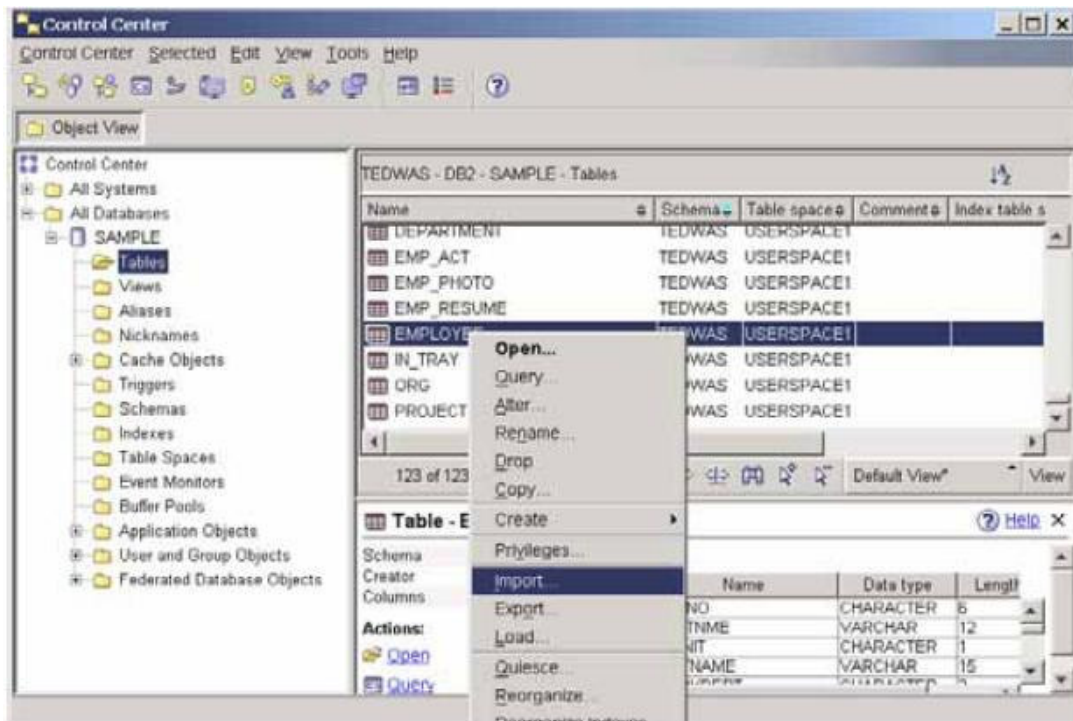
Như đã thấy trên hình, trước hết bạn phải chọn bảng *employee* bằng cách nhấn vào nó một lần, và kế đó thì nhấn chuột phải trên bảng để mở một bảng chọn con (popup menu) từ đó bạn có thể chọn Export. Sau khi chọn mục này, cửa sổ hướng dẫn từng bước sẽ hiện lên. Chỉ cần đơn giản làm theo những bước cung cấp sẵn trong hướng dẫn này để hoàn thành quá trình thực hiện.

### 9.2 Công cụ IMPORT (nhập)

Công cụ IMPORT được dùng để nạp dữ liệu từ một tệp sang một bảng như đã nói trước đây. Đằng sau nó, một thao tác SQL INSERT được thực thi thực sự. Khi một lệnh INSERT được thực thi, các bất sự kiện sẽ được kích hoạt, tất cả các ràng buộc sẽ có hiệu lực ngay lập tức, và vùng nhớ đệm của cơ sở dữ liệu sẽ được sử dụng. Ví dụ sau đây nạp tất cả dữ liệu từ một tệp *employee.ixf* có định dạng IXF vào trong bảng *employee\_copy*. Lựa chọn REPLACE\_CREATE là một trong những lựa chọn có sẵn với công cụ IMPORT. Lựa chọn này sẽ thay thế nội dung của bảng *employee\_copy* nếu như nó đã tồn tại trước khi công cụ IMPORT được thực thi, hoặc là nó sẽ tạo một bảng và nạp dữ liệu nếu bảng chưa tồn tại trước đó. Chúng tôi khuyến khích bạn nên chạy thử ví dụ dưới đây, nhưng trước hết bạn phải chạy công cụ EXPORT trong phần trước.

```
IMPORT FROM employee.ixf OF IXF
REPLACE_CREATE
INTO employ_copy
```

Nếu bạn thích làm việc với Trung tâm điều khiển, bạn có thể thực hiện công cụ IMPORT bằng cách chọn bất kỳ bảng nào, nhấn chuột phải trên nó, và chọn chức năng Import như hình 9.3



Hình 9.3 – Hộp hội thoại thực hiện công cụ IMPORT

### 9.3. LOAD ( nạp)

Công cụ LOAD là một cách nạp dữ liệu từ tệp sang bảng nhanh hơn. Như đã thảo luận trước đây, công cụ LOAD không đi qua hệ thống DB2, vì thế các bất sự kiện không được kích hoạt, vùng đệm sẽ không được dùng và các ràng buộc sẽ không có hiệu lực. Mặt khác, một thao tác LOAD sẽ nhanh hơn IMPORT vì nó là một công cụ nạp dữ liệu cấp thấp trực tiếp truy xuất đến trang dữ liệu trên đĩa. Nó làm việc theo 3 giai đoạn: LOAD, BUILD và DELETE.

Ví dụ sau đây nạp tất cả các dữ liệu từ tệp *employee.ixf* có định dạng IXF vào trong bảng *employee\_copy*. Lựa chọn REPLACE là một trong những lựa chọn có sẵn trong

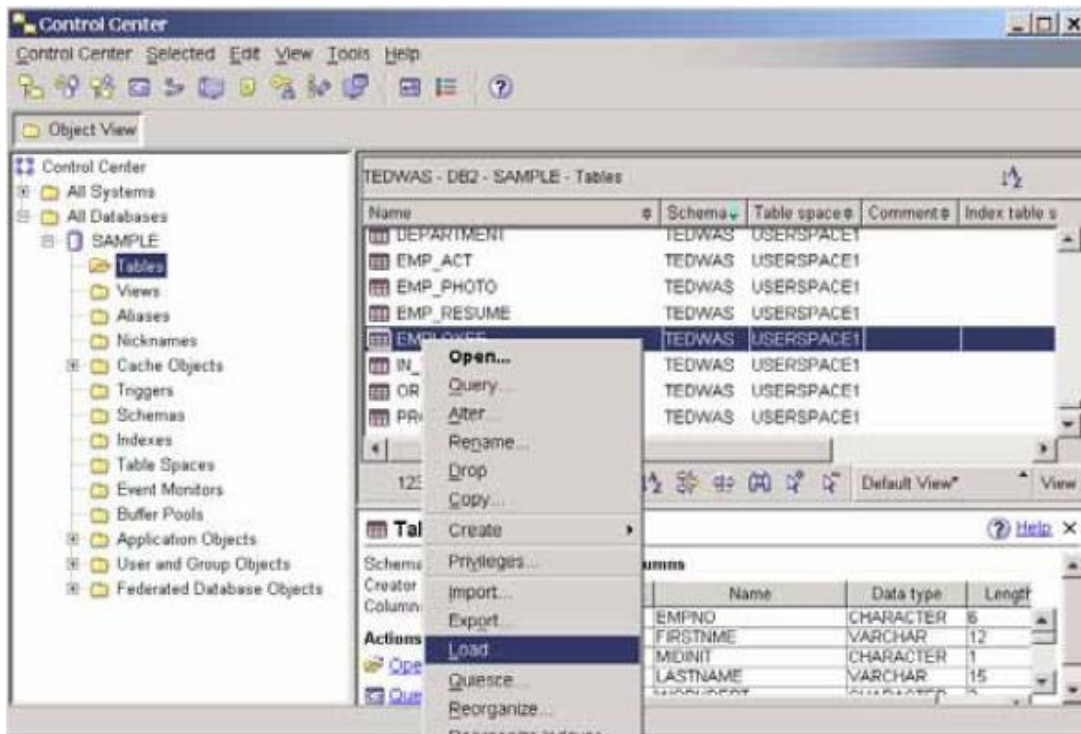
LOAD. Trong trường hợp này, nó được sử dụng để REPLACE tất cả các dữ liệu có trong bảng *employe\_copy*.

```
LOAD FROM employe.ixf OF IXF
REPLACE INTO employ_copy
```

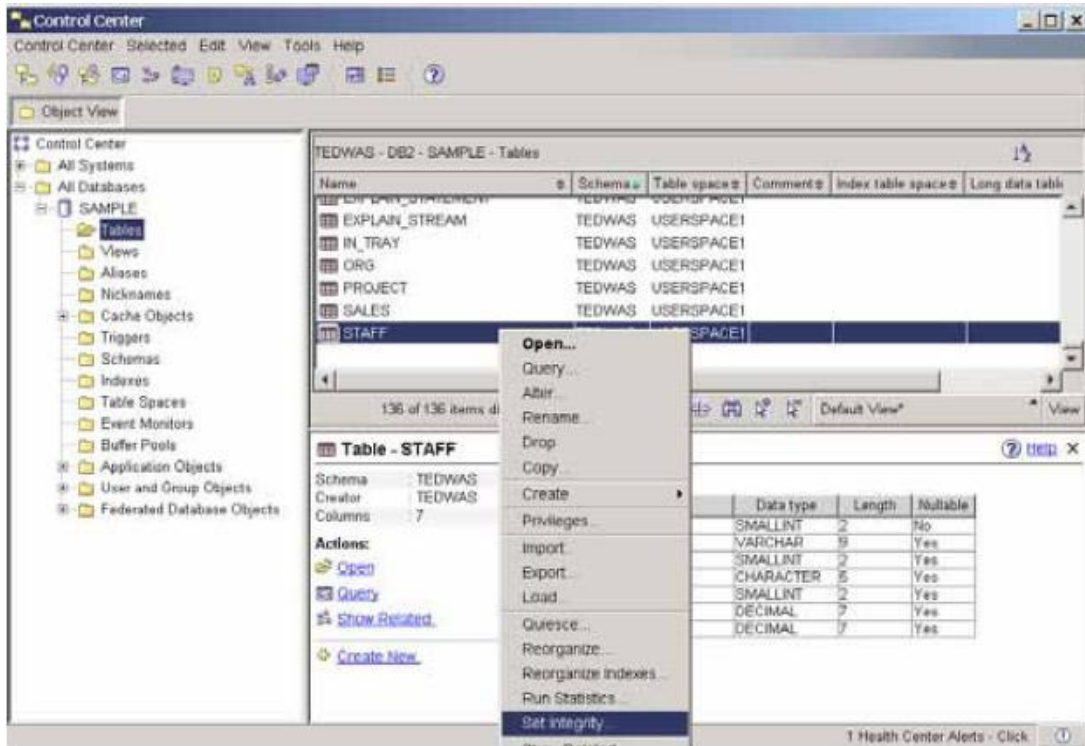
Sau khi thực thi câu lệnh trên, không gian của bảng, nơi mà bảng của bạn trú ngụ, có thể được đưa đến trạng thái CHECK PENDING. Điều này có nghĩa là bạn cần phải chạy câu lệnh SET INTEGRITY để kiểm tra tính chắc chắn dữ liệu của bạn. Ví dụ sau đây sẽ chỉ cho bạn làm thế nào.

```
SET INTEGRITY FOR employe_copy
ALL IMMEDIATE UNCHECKED
```

Nếu bạn thích làm việc với Trung tâm điều khiển hơn, bạn có thể thực hiện công cụ LOAD và SET INTEGRITY như trong hình 9.4 và 9.5 tương ứng.



Hình 9.4 - Thực hiện công cụ LOAD



Hình 9.5 - Thực hiện công cụ SET INTEGRITY

#### 9.4 Công cụ db2move

Tại một thời điểm, công cụ EXPORT, IMPORT, và LOAD chỉ làm việc trên một bảng. Mặc dù bạn có thể viết một kịch bản để phát sinh các lệnh trên cho từng bảng trong cơ sở dữ liệu, nhưng công cụ db2move sẽ giúp bạn làm điều đó. Công cụ db2move chỉ làm việc với tệp IXF, và tên tệp sẽ tự động được phát sinh bởi db2move. Ví dụ bên dưới sẽ biểu diễn chạy db2move với export, import như thế nào để sử dụng cơ sở dữ liệu SAMPLE:

```
db2move sample export
db2move sample import
```

Trung tâm điều khiển không có tùy chọn cho db2move.

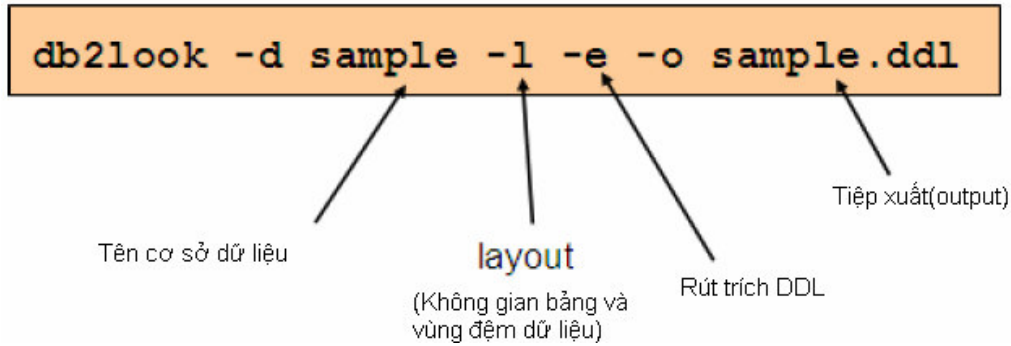
#### 9.5 Công cụ db2look

Trong khi các công cụ EXPORT, IMPORT, LOAD và db2move cho phép bạn di chuyển dữ liệu từ một bảng này tới một bảng khác, trong cùng một cơ sở dữ liệu hay qua vài cơ sở dữ liệu, công cụ db2look có thể được sử dụng để rút trích các câu lệnh DDL, các cơ sở dữ liệu tĩnh, các đặc tính không gian bảng cho một cơ sở dữ liệu, và lưu trữ chúng trong một tệp kịch bản mà có thể sau này được chạy trên một hệ thống khác. Ví dụ, nếu bạn muốn sao chép một cơ sở dữ liệu từ máy chủ DB2 đang chạy trên Linux sang một máy chủ DB2 đang chạy trên Windows; đầu tiên bạn chạy công cụ db2look trên máy chủ DB2 Linux để lấy được cấu trúc của cơ sở dữ liệu và lưu cấu trúc này trên một tệp kịch bản. Sau đó bạn chép tệp kịch bản này sang máy chủ DB Windows, và thực thi kịch bản đó để bắt đầu xây dựng bản sao cơ sở dữ liệu. Tới đây, cấu trúc của cơ sở dữ liệu đó đã được sao chép. Bước kế tiếp, sẽ chạy công cụ db2move với tùy chọn export trong máy chủ DB2 Linux, và sau đó chép tất cả các tệp được phát sinh sang máy chủ DB2 Windows, sau đó thực thi db2move với tùy chọn là import hoặc load. Hoàn thành bước này, cơ sở dữ liệu của bạn sẽ được sao chép toàn bộ từ một máy chủ này sang một máy chủ khác hoạt động trên một nền tảng khác.



Kịch bản trên cần thiết khi làm việc với cơ sở dữ liệu trên các nền tảng khác nhau như Linux và Windows. Nếu cả hai máy chủ đều hoạt động trên cùng một nền tảng, bạn sẽ dùng những lệnh sao lưu dự phòng và khôi phục sẽ làm cho xử lý dễ dàng hơn và rõ ràng hơn. Những lệnh sao lưu và khôi phục sẽ được mô tả chi tiết hơn trong chương sau của cuốn sách này.

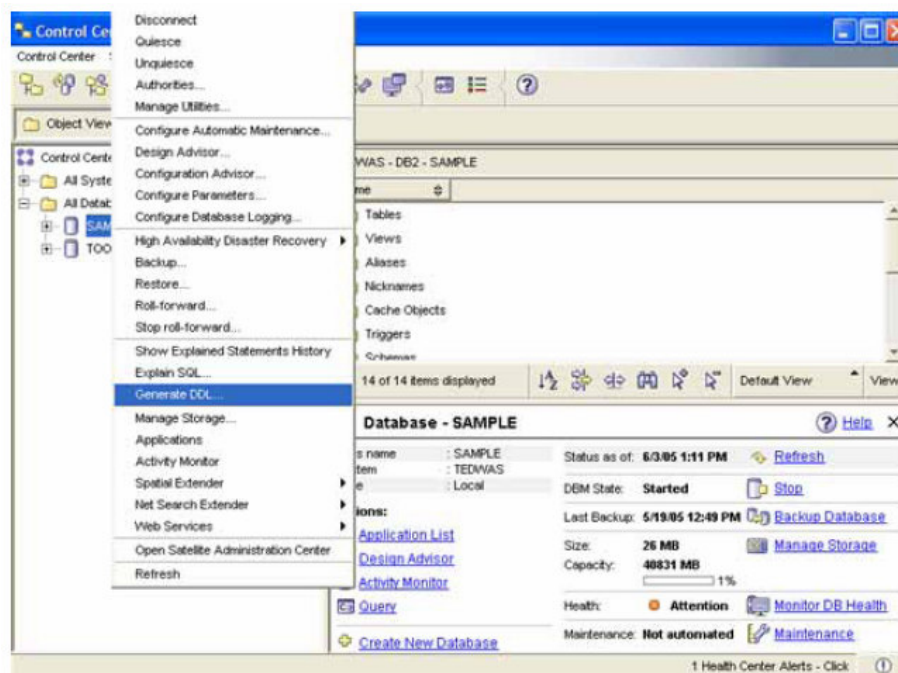
Những ví dụ sau sẽ rút trích không gian bảng và bố trí vùng đệm dữ liệu, cùng với những lệnh DDL từ cơ sở dữ liệu SAMPLE, và lưu trữ chúng trong tệp *sample.ddl*. Chúng tôi khuyến khích bạn chạy lệnh bên dưới và xem lại tệp kết quả xuất (văn bản) “sample.ddl”.



Lệnh db2look có quá nhiều tùy chọn không thể mô tả hết trong cuốn sách này; tuy nhiên bạn có thể sử dụng cờ `-h` để lấy được phân mô tả tóm lược các tùy chọn sẵn có:

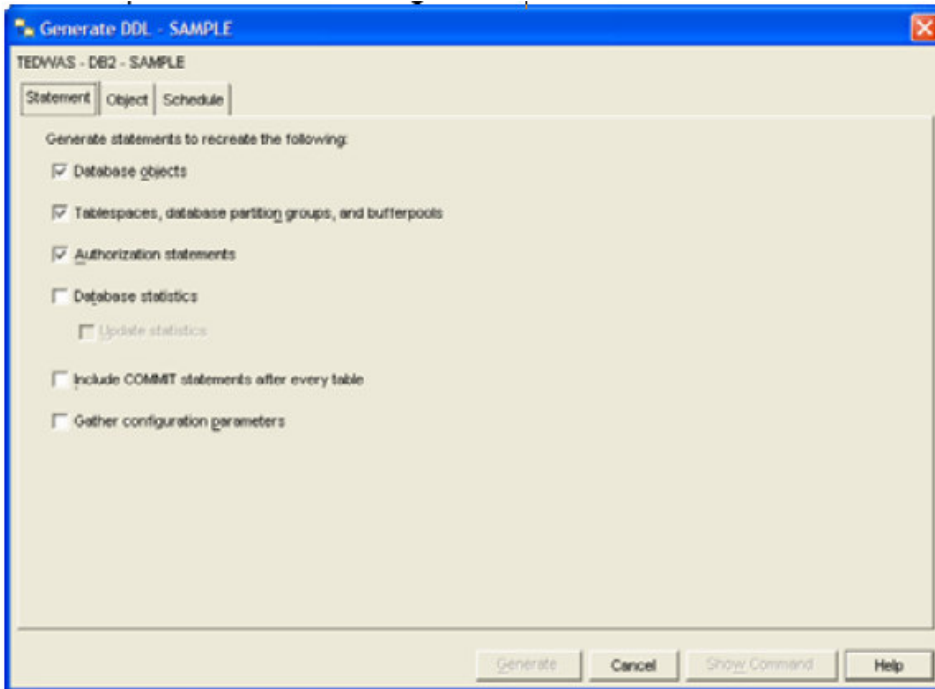
`db2look -h`

Công cụ db2look cũng có thể được gọi từ trung tâm điều khiển như biểu diễn trong Hình 9.6



Hình 9.6 : Trích DDL từ Control Center

Trong Hình 9.6 chọn cơ sở dữ liệu mà bạn muốn lấy DDL, nhấn chuột phải lên nó, và chọn “Generate DDL”. Cửa sổ Generate DDL xuất hiện, hiển thị một số tùy chọn rút trích, thể hiện như trong hình 9.7.



Hình 9.7- Trích DDL từ Control Center

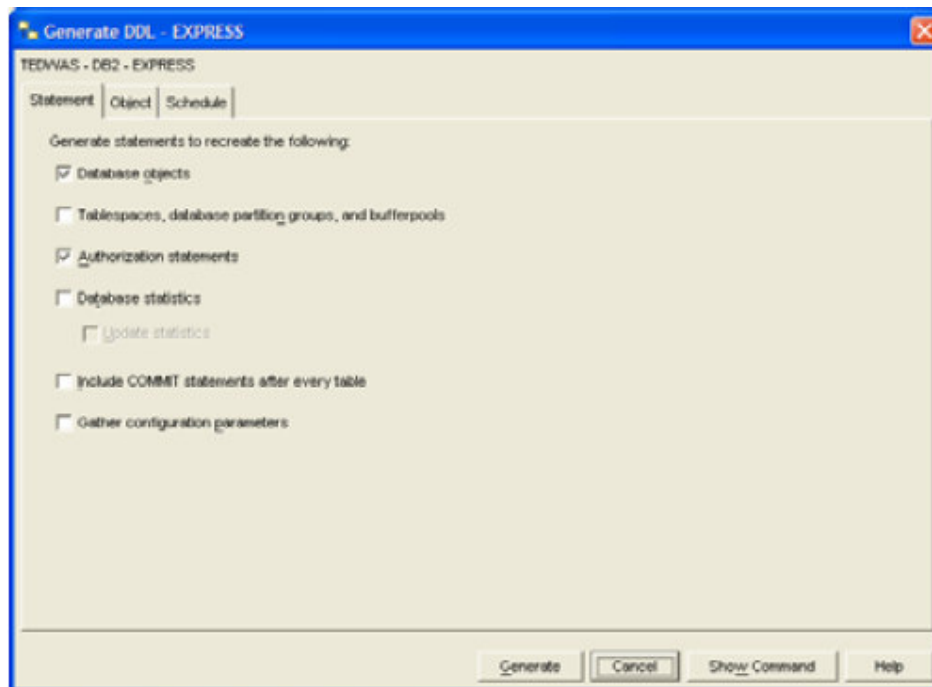
## Bài tập nhanh #6 - Trích xuất DDL từ cơ sở dữ liệu EXPRESS

### Mục tiêu:

Khi bạn sao lưu một cơ sở dữ liệu, mục tiêu của bạn là phải tạo lại cơ sở dữ liệu bằng cách đơn giản và hợp lý. Thông thường sử dụng kịch bản SQL, nó có thể được thực hiện sau khi DB2 được cài đặt. Trong bài tập nhanh này, bạn sẽ rút ra những định nghĩa đối tượng từ cơ sở dữ liệu EXPRESS (đã tạo trong bài tập nhanh #3) sử dụng Trung tâm điều khiển.

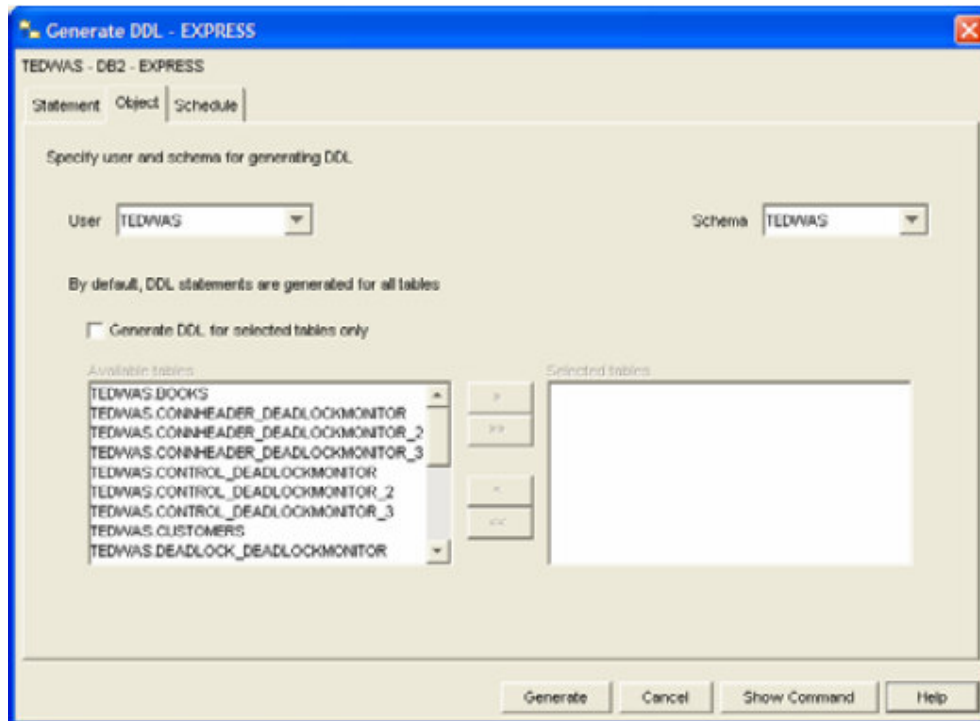
### Quy trình:

1. Mở Trung tâm điều khiển.
2. Nhấn chuột phải vào cơ sở dữ liệu *EXPRESS* trên cây đối tượng và chọn mục *Generate DDL*. Việc này mở cửa sổ hộp thoại *Generate DDL*.
3. Trong cửa sổ *Generate DDL*, chỉ rõ những tùy chọn cho *Generate DDL* như dưới đây. Nếu bạn tạo ra thêm những đối tượng bổ sung trong môi trường của bạn như không gian bảng, bộ đệm..., bạn sẽ chọn chúng ở đây. Khi bạn không tạo ra bất kỳ loại đối tượng nào, đừng đánh dấu vào hộp kiểm. Thống kê cơ sở dữ liệu không được thêm vào vì môi trường sản xuất có khả năng chứa một tập hợp thống kê khác hơn môi trường phát triển. Tương tự, những thông số cấu hình sẽ có khả năng khác đi một chút. Trong môi trường mà bạn sở hữu, nếu mọi thứ cấu hình chính xác theo cách nó được triển khai, bạn có thể chọn thêm vào cái tùy chọn bổ sung.

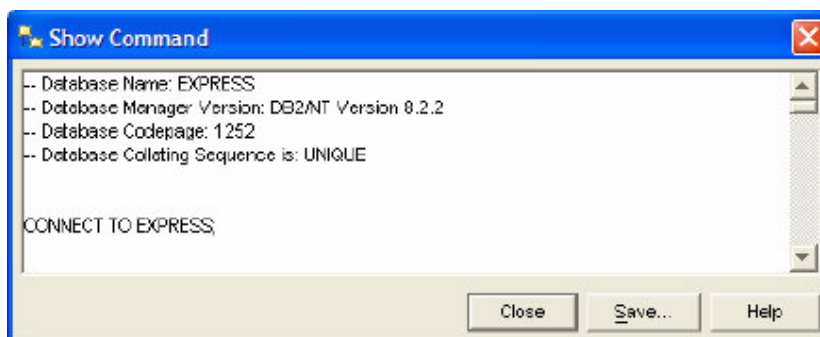


4. Chuyển qua trang *Object*. Bạn có thể chỉ định chọn những đối tượng mà bạn muốn phát sinh ra DDL. Trong trường hợp này, chọn người dùng và giản đồ mà bạn dùng để tạo ra tất cả các đối tượng của bạn và phát sinh ra DLL cho tất cả các đối tượng trong giản đồ này. Nhấn nút *Generate* để bắt đầu tạo DLL.





5. Xem lại kết quả DDL. Kết quả của bước thực hiện trước là một kịch bản với tất cả các câu lệnh SQL cho các đối tượng được chọn. Bạn sẽ tổ chức các kịch bản này vào trong những nhóm hợp lý.
6. Tạo thư mục C:\express trong tập tin hệ thống và lưu tập tin phát sinh DDL vào trong thư mục này với tên là schema.ddl, Nhấn nút "Save".



7. Mở tệp được lưu gần đây nhất trong cửa sổ nhập lệnh. (Gợi ý: từ cửa sổ nhập lệnh, chọn *File -> Open*)
8. Mặc dù chúng tôi chỉ thực sự muốn DDL của các bảng, bạn sẽ thấy các DDL cho cả những đối tượng khác nữa cũng được sinh ra. Chuyển tất cả những khai báo CREATE TRIGGER tới một tệp mới riêng biệt được gọi là `triggers.ddl`. Thậm chí dù chúng tôi chỉ tạo ra có một bất sự kiện cũng để riêng ra, nó là một thói quen tốt để phân biệt các đối tượng theo kiểu của nó.
9. Bây giờ chúng tôi cũng khuyến cáo loại bỏ:

- Các lệnh cơ sở dữ liệu CONNECT TO.
- Các lệnh DISCONNECT.

Bạn phải có hai kịch bản tại điểm này:

C:\express\schema.ddl DDL cho các bảng, các cách hiển thị, các chỉ mục và các ràng buộc.

C:\express\triggers.ddl DDL cho các bẫy sự kiện

#### 10. Chuẩn bị kịch bản cho sự triển khai:

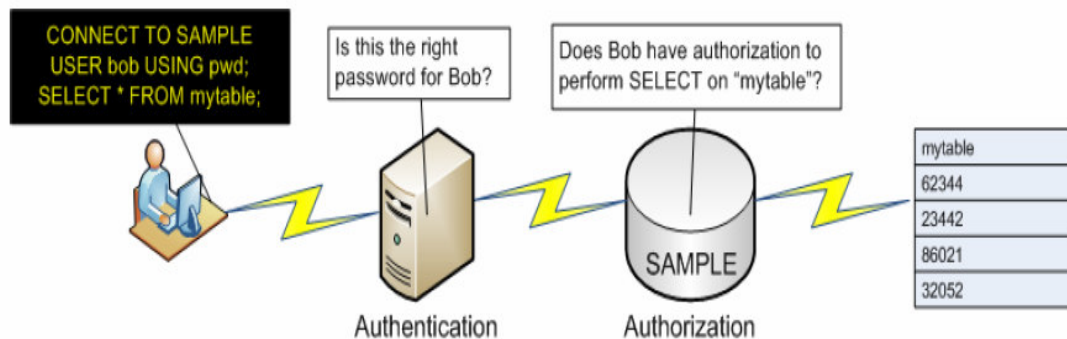
- Xóa những chú thích không cần thiết (ví dụ: -- CONNECT TO...)
- Tách các hàm và các thủ tục vào những tệp tin riêng (rất hữu dụng khi có quá nhiều hàm và thủ tục). Bạn cũng có thể nhóm chúng lại theo chức năng hay ứng dụng (ví dụ: stringfunc.ddl, billing.ddl, math.ddl, , ...)

#### 11. Bạn có thể phải chú ý đến những kí tự đặc biệt được dùng để giới hạn kết thúc các bẫy, các hàm và các thủ tục (@). Điều đó là cần thiết để đặt giới hạn kết thúc của lệnh CREATE <...i t...ng> khác với kết thúc của một khai báo thủ tục nằm bên trong đối tượng.

# 10

## Chương 10 – Bảo mật cơ sở dữ liệu

Chương này bàn về sự bảo mật được xử lý như thế nào trong DB2. Hình 10.1 cung cấp một cái nhìn tổng quan cơ bản



Hình 10.1 – Tổng quan về sự bảo mật trong DB2

Như trong hình 10.1, sự bảo mật DB2 có hai phần:

### Định danh

Nó là một tiến trình qua đó kiểm tra tính hợp lệ của thông tin đăng nhập của người dùng. Tiến trình định danh được thực thi bởi một chương trình bảo mật bên ngoài DB2 (điển hình là bởi một hệ điều hành, một phương thức định danh của một mạng nào đó, hoặc là một phần mềm định danh được thiết kế theo yêu cầu). Tiến trình định danh của hệ điều hành được xác lập mặc định. Khi sử dụng tiến trình định danh của hệ điều hành, tên người dùng (userid) và mật mã được chuyển tới cơ sở dữ liệu ở máy chủ (như là một phần của câu lệnh kết nối). Sau đó, cơ sở dữ liệu ở máy chủ sẽ gọi tiến trình định danh của hệ điều hành để kiểm tra tính hợp lệ của tên người dùng và mật mã.

### Quyền hạn

Ở phần này, DB2 kiểm tra xem người dùng, đã qua định danh ở trên, có thể thi hành thao tác người đó yêu cầu hay không. Thông tin về quyền hạn được lưu trữ trong một danh mục DB2 và một tập tin cấu hình DBM.

Ví dụ, trong hình 10.1, người dùng 'bob' kết nối với cơ sở dữ liệu SAMPLE bằng câu lệnh :

```
CONNECT TO sample USER bob USING pwd
```

Cả "bob" và "pwd" được chuyển tới hệ điều hành hoặc một chương trình định danh bên ngoài để kiểm tra xem tên người dùng "bob" đã được định nghĩa chưa và mật mã đó có

được nối với người dùng này hay không. Nếu phần này thành công, hệ điều hành sẽ trả lại quyền kiểm soát bảo mật cho DB2. Tiếp theo, khi người dùng “bob” thi hành lệnh sau:

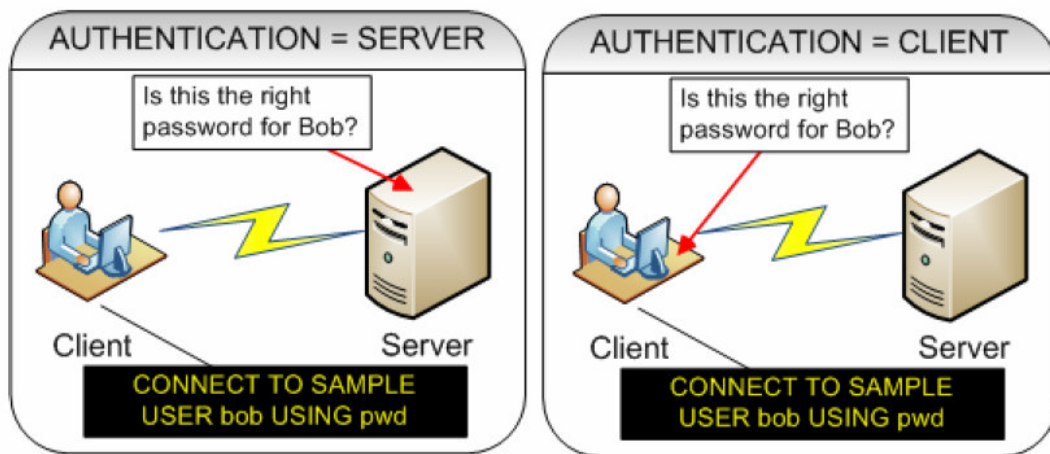
```
SELECT * FROM mytable
```

Bấy giờ DB2 tiếp tục dùng quyền kiểm soát bảo mật để thi hành việc kiểm tra quyền hạn và xác nhận người dùng “bob” có quyền SELECT trên bảng “mytable” hay không. Nếu quyền đó không hợp lệ, DB2 sẽ trả về một thông báo lỗi, ngược lại câu lệnh sẽ được thực thi dựa trên bảng “mytable”.

**10.1 Định danh**

Mặc dù tiến trình định danh thực sự được thực hiện bởi hệ điều hành (hoặc một chương trình bảo mật khác bên ngoài), DB2 vẫn quyết định tầng mà tiến trình định danh xảy ra.

Tham số AUTHENTICATION trong DBM CFG, thiết lập ở máy chủ DB2, có một dãy các giá trị hợp lệ. Ví dụ khi tham số được thiết lập là SERVER (giá trị mặc định), tiến trình định danh sẽ được thực thi bởi hệ điều hành hoặc chương trình bảo mật bên ngoài ở trên máy chủ. Tuy nhiên, nếu AUTHENTICATION được thiết lập là CLIENT, tiến trình định danh sẽ được thực thi bởi hệ điều hành hoặc chương trình bảo mật bên ngoài ở trên máy khách. Vấn đề này được minh họa ở hình 10.2



Hình 10.2 – Nơi tiến trình định danh xảy ra

Tham số AUTHENTICATION có thể được thiết lập bằng một trong những giá trị được liệt kê trong bảng 10.1

Lệnh	Mô tả
SERVER (mặc định)	tiến trình định danh diễn ra ở máy chủ
CLIENT	tiến trình định danh diễn ra ở máy khách
SERVER_ENCRYPT	tương tự như SERVER nhưng tên của người dùng và mật mã được mã hóa
KERBEROS	tiến trình định danh diễn ra dùng cơ chế bảo mật Kerberos
SQL_AUTHENTICATION_DATAENC	tiến trình định danh máy chủ cộng thêm sự kết nối phải dùng sự mã hóa

	dữ liệu
SQL_AUTHENTICATION_DATAENC_CMP	giống như ở trên nhưng sự mã hóa dữ liệu chỉ dùng khi có sẵn
GSSPLUGIN	tiến trình định danh dùng cơ chế bảo mật bên ngoài dựa trên GSS API

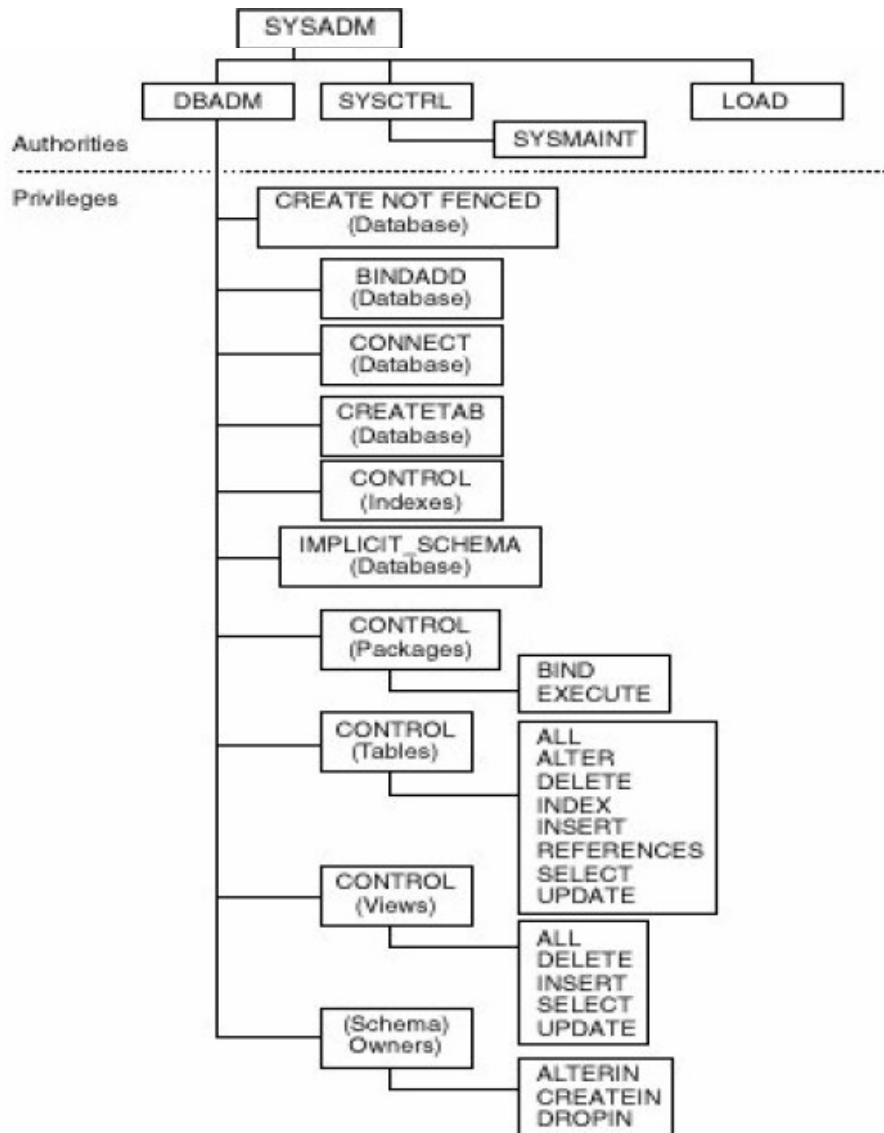
**Bảng 10.1 - Những giá trị tham số AUTHENTICATION hợp lệ**

### 10.2 Quyền hạn

Quyền hạn bao gồm các đặc quyền và quyền được lưu trữ trong những bảng hệ thống DB2 và được DB2 quản lý.

Một đặc quyền cho phép người dùng thi hành một loại thao tác đơn dựa trên cơ sở dữ liệu như là CREATE, UPDATE, DELETE, INSERT, ....

Một quyền là một hành động được xác định trước bao gồm nhiều đặc quyền. Hình 10.3 minh họa các quyền và đặc quyền khác nhau trong DB2.



Hình 10.3 - Những quyền và đặc quyền

Bảng 10.2 nêu ra những hàm khác nhau mà mỗi quyền có thể thực hiện. Như bạn thấy SYSADM có hầu hết các quyền trong khi SYSMON có ít quyền nhất.

Chức năng	SYSADM	SYSCTRL	SYSMAINT	SYSMON	DBADM	LOAD
Cập nhật DBM CFG	Y					
Cấp phép/Hủy bỏ DBADM	Y					
Thiết lập/Thay đổi SYSCTRL	Y					
Thiết lập/Thay đổi SYSMAINT	Y					
Thiết lập/Thay đổi SYSMON	Y					
Buộc người dùng ra khỏi cơ sở dữ liệu	Y	Y				

Tạo/Bỏ cơ sở dữ liệu	Y	Y				
Khôi phục thành cơ sở dữ liệu mới	Y	Y				
Cập nhật DB CFG	Y	Y	Y			
Sao lưu cơ sở dữ liệu/không gian bảng	Y	Y	Y			
Khôi phục thành cơ sở dữ liệu đã tồn tại	Y	Y	Y			
Khôi phục tiến	Y	Y	Y			
Bắt đầu/kết thúc một thể hiện	Y	Y	Y			
Phục hồi không gian bảng	Y	Y	Y			
Thực hiện theo vết	Y	Y	Y	Y		
Lấy hình ảnh màn hình	Y	Y	Y			
Truy vấn trạng thái không gian bảng	Y	Y	Y			
Lược bớt tập tin nhật ký	Y	Y	Y			
Quiesce không gian bảng	Y	Y	Y		Y	Y
Nạp bảng	Y				Y	Y
Thiết lập/gỡ bỏ trạng thái treo	Y				Y	
Tạo/hủy bỏ sự kiện màn hình	Y				Y	

**Bảng 10.2 – Các quyền và đặc quyền DB2**

Để cấp quyền SYSADM, SYSCTRL, hoặc SYSMANT cho một nhóm, những tham số SYSADM\_GROUP, SYSCTRL\_GROUP, hoặc SYSMANT\_GROUP trong DBM CFG phải được gán cho nhóm hệ điều hành

Ví dụ, để cấp quyền SYSADM cho nhóm hệ điều hành 'db2admns', bạn có thể dùng câu lệnh này:

```
update dbm cfg using SYSADM_GROUP db2admns
```

Mỗi thể hiện DB2 có những định nghĩa về nhóm quyền riêng.

Trên Windows, những tham số này được mặc định là rỗng, điều đó có nghĩa là nhóm những người quản trị Windows là SYSADM. Trên Linux, nhóm những người sở hữu thể hiện được mặc định là nhóm SYSADM.

**10.3 Quyền quản trị cơ sở dữ liệu DBADM**

DBADM (DataBase Administrator) là người có quyền cao nhất về quản trị cơ sở dữ liệu. Đây không phải là quyền ở cấp độ cụ thể; do đó không được nêu ra trong phần trước. Để gán quyền quản trị cơ sở dữ liệu DBADM, ta sử dụng lệnh GRANT, thí dụ:

```
connect to sample
grant DBADM on database to user <userid>
```

Trong thí dụ trên, trước hết bạn cần kết nối đến cơ sở dữ liệu, ở đây cơ sở dữ liệu có tên là “sample”, và sau đó gán quyền DBADM cho người dùng. Để có thể gán quyền quản trị cơ sở dữ liệu DBADM, bạn phải là SYSADM.

Ghi chú: DBADM không có quyền tạo các không gian bảng, cho dù những không gian bảng này là các đối tượng bên trong cơ sở dữ liệu. Nguyên nhân là vì một không gian bảng có liên quan đến các vùng chứa dữ liệu (containers) như đĩa và các vùng đệm (bộ nhớ) đó là các tài nguyên vật lý của hệ thống.

#### 10.4 Nhóm PUBLIC

DB2 định nghĩa một nhóm bên trong được gọi là PUBLIC. Bất cứ người dùng nào đã được định danh bởi hệ điều hành hoặc dịch vụ chứng thực mạng đều được ngầm định là thành viên của nhóm PUBLIC. Khi một cơ sở dữ liệu được tạo ra, một số quyền cụ thể sẽ được gán một cách tự động cho nhóm PUBLIC:

- CONNECT,
- CREATETAB,
- IMPLICIT\_SCHEMA,
- BINDADD

Để thêm vào mức độ bảo mật, chúng ta nên thu hồi, hủy bỏ mọi quyền từ nhóm PUBLIC như sau:

```
REVOKE CONNECT ON DATABASE FROM PUBLIC
REVOKE CREATETAB ON DATABASE FROM PUBLIC
REVOKE IMPLICIT_SCHEMA ON DATABASE FROM PUBLIC
REVOKE BINDADD ON DATABASE FROM PUBLIC
```

#### 10.5 Lệnh GRANT và REVOKE

Lệnh GRANT và REVOKE là lệnh SQL chuẩn, được dùng để gán hoặc hủy bỏ quyền của một người dùng hoặc một nhóm. Sau đây là một số ví dụ về các câu lệnh này: Gán quyền SELECT trên bảng T1 cho người dùng USER1:

```
GRANT SELECT ON TABLE T1 TO USER user1
```

Gán tất cả các quyền trên bảng T1 cho nhóm GROUP1:

```
GRANT ALL ON TABLE T1 TO GROUP group1
```

Hủy bỏ tất cả các quyền trên bảng T1 của GROUP1:

```
REVOKE ALL ON TABLE T1 FROM GROUP group1
```

Gán quyền EXECUTE trên thủ tục p1 cho người dùng USER1:

```
GRANT EXECUTE ON PROCEDURE p1 TO USER user1
```

Hủy bỏ quyền EXECUTE trên thủ tục p1 của người dùng USER1:

```
REVOKE EXECUTE ON PROCEDURE p1 FROM USER user1
```

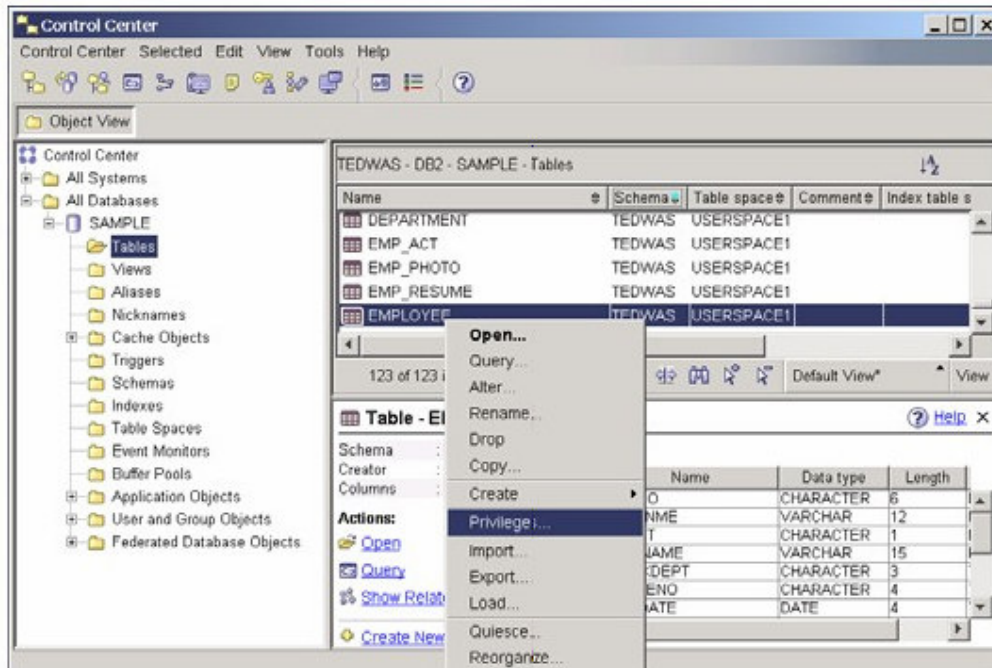
#### 10.6 Kiểm tra việc cấp quyền và quyền

Cách tốt nhất để kiểm tra việc cấp quyền và kiểm tra quyền là thông qua trung tâm điều



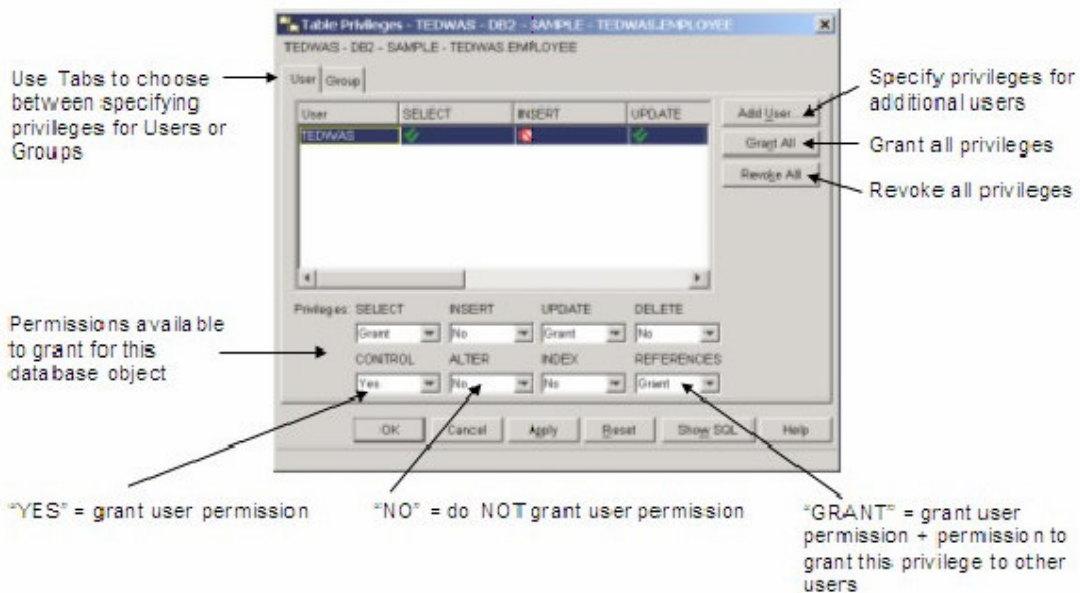
khởi.

Hình 10.4 mô tả cách mở hộp thoại Table Privileges cho bảng EMPLOYEE từ trung tâm điều khiển.



Hình 10.4 – Mở hộp thoại Privileges

Trong hình 10.4, bạn chọn bảng mong muốn, nhấp phải chuột vào bảng này và chọn Privileges. Một khi được chọn, hộp thoại Privileges hiển thị như hình 10.5. Hình này có mô tả các trường và các thành phần của hộp thoại.



Hình 10.5 – Hộp thoại Privileges

Một cách khác, bạn có thể truy vấn các thể hiện danh mục DB2 SYSCAT là nơi chứa các thông tin về việc cấp quyền. Ví dụ, nếu bạn muốn biết người dùng DB2ADMIN có quyền SELECT trên bảng T2 hay không, và muốn biết ai đã gán quyền này, bạn có thể thực hiện câu truy vấn như sau:

```
SELECT grantor, grantee, selectauth
   FROM syscat.tabauth
  WHERE tabname = 'T2'
```

GRANTOR	GRANTEE	SELECTAUTH
ARFCHONG	DB2ADMIN	Y

Trong thí dụ trên, người dùng ARFCHONG đã gán quyền SELECT cho người dùng DB2ADMIN.

### 10.7 Xem xét quyền trên nhóm

Để việc quản trị DB2 dễ dàng hơn, ta nên đặt những người dùng vào các nhóm và gán các quyền cần thiết cho các nhóm này.

Khi một nhóm được gán quyền, các thành viên của nhóm sẽ được gán các quyền ngầm định được kế thừa với tư cách thành viên nhóm.

Khi một người dùng bị xóa khỏi nhóm, người này sẽ mất các quyền ngầm định của nhóm, nhưng vẫn còn giữ các quyền được gán tường minh. Nói một cách khác các quyền được cấp một cách tường minh phải được hủy bỏ một cách tường minh.

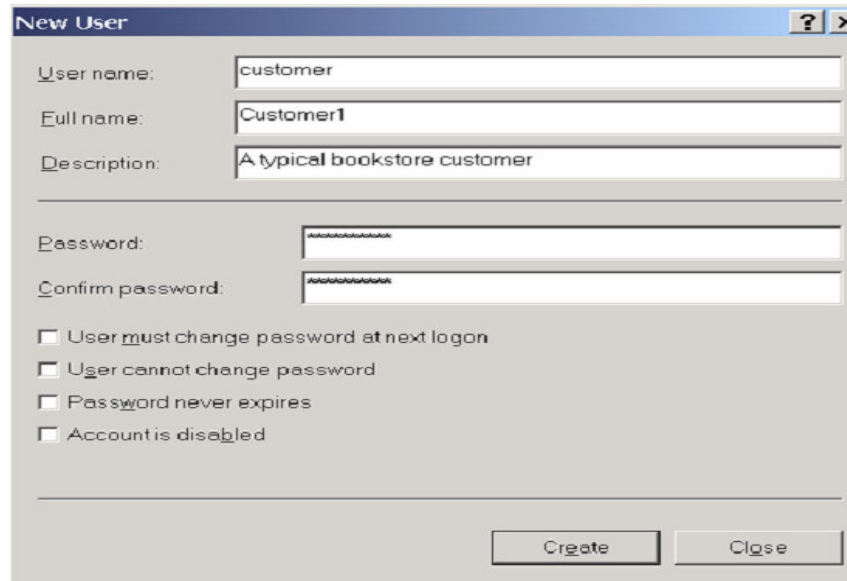
## Bài tập nhanh # 7: Cho phép và hủy bỏ (phân quyền) quyền người sử dụng.

### Mục tiêu:

Cho đến lúc này, bạn đã sử dụng một tài khoản của người quản trị cho việc sử dụng các lệnh với cơ sở dữ liệu. Tài khoản này có thể truy nhập tới mọi tiện ích, dữ liệu và những đối tượng cơ sở dữ liệu đầy đủ. Bởi vậy, điều quan trọng là cần bảo vệ tài khoản này để tránh sự nhiễu loạn hay mất mát dữ liệu có chủ tâm. Trong đa số những trường hợp, bạn sẽ muốn tạo ra những tài khoản người dùng khác hoặc những nhóm với một thiết lập quyền hạn chế hơn. Trong bài tập này, bạn sẽ tạo ra một tài khoản người sử dụng mới, rồi gán nó những đặc quyền xác định.

### Thủ tục:

1. Mở cửa sổ Windows Computer Management bằng cách kích chuột phải vào biểu tượng *My Computer* trên máy tính, và chọn *Manage*.
2. Chọn mục *Local Users and Group* ở nhóm *System Tools* nằm bên trái của cửa sổ Computer Management, nhấn chuột phải lên *User* và chọn mục *New User*.
3. Trong cửa sổ *New User*, nhập các thông tin sau đây: trong mục *User name* nhập "customer" và trong mục *Full name* nhập "Customer1". Trong trường *Description* nhập "A typical bookstore customer". Trong trường *Password* và *Confirm password* nhập "ibmdb2". Bỏ dấu đánh ở mục "người dùng cần đổi mật mã ở lần đăng nhập sau" (User must change password on next logon) và nhấn nút *Create* để tạo người dùng mới



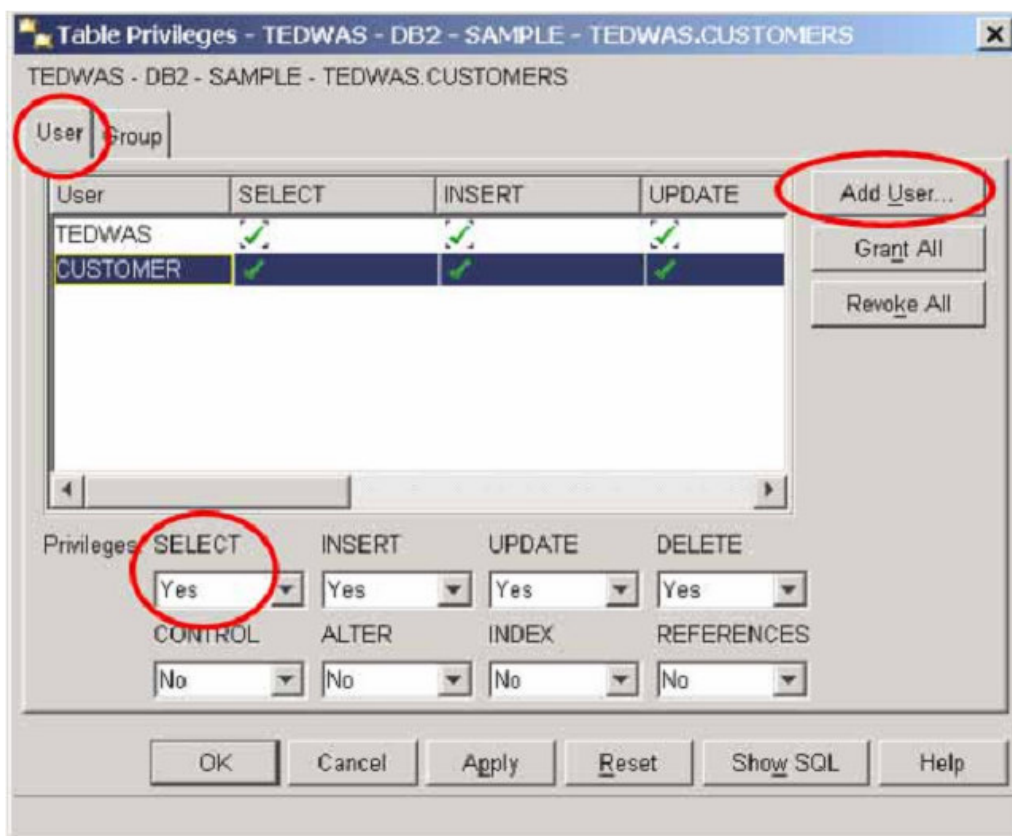
4. Chọn chế độ cao cấp (Advanced) bằng cách chọn mục *Customize Control Center* trên thực đơn *Control Center Tool*, chọn tiếp tùy chọn *Advanced* và nhấn *OK*. Bảo đảm khung nhìn làm việc hiện tại ở chế độ *Advanced*.

5. Mở rộng cây đối tượng của Trung tâm điều khiển ở bên trái cửa sổ, chọn *All Database -> EXPRESS -> Tables*.

6. Phân những quyền cần có cho người dùng vừa mới tạo ra: Từ danh sách các bảng trong cơ sở dữ liệu *EXPRESS*, nhấn chuột phải lên bảng *CUSTOMERS* và chọn mục *Privileges* để hiển thị cửa sổ hội thoại “Table Privileges”

7. Kích vào nút Add User và chọn người dùng “customer” vừa mới tạo ra ở bước 3. Kích vào nút OK để đóng hộp thoại Add User.

8. Bạn chú ý rằng người dùng “customer” đã được thêm vào danh sách người dùng, nhưng chưa được gán thêm các quyền. Để cấp quyền cho người dùng này, chọn Yes trong hộp thoại kéo xuống của tất cả các phần SELECT, INSERT, UPDATE và DELETE. Một người dùng Internet cần phải có khả năng thực hiện add/update/delete dữ liệu tài khoản của họ. Chúng ta không cấp cho người sử dụng những quyền khác bởi vì không cần thiết đối với họ. Nhấn nút Ok để đóng cửa sổ hộp thoại Table Privileges và chấp nhận những thay đổi vừa làm.



9. Lập lại từng bước từ 7-9 cho bảng *BOOKS* và bảng *SALES*. Đối với bảng *BOOKS*, chỉ cung cấp quyền SELECT bởi vì người dùng “customer” không nên có khả năng để điều chỉnh và phục hồi dữ liệu trong kho. Đối với bảng *SALES*, chỉ cho quyền SELECT và INSERT. Người dùng không nên có quyền DELETE hoặc UPDATE bởi vì chỉ những nhân viên mới cần phải có sự truy nhập để sửa đổi những giao dịch bán hàng.

10. Thủ kết nối đến cơ sở dữ liệu bằng tài khoản của người dùng “customer” đã tạo ở trên. Thử để SELECT với bảng khách hàng. Điều gì xảy ra? Thử DELETE hay UPDATE dữ liệu trong bảng *SALES*. Điều gì xảy ra?

*Trong bài tập nhanh này, chúng ta chỉ tạo một người dùng là “customer”, tuy nhiên, ứng dụng của bạn có thể chứa đựng nhiều kiểu người dùng khác. Thử nghiệm tạo ra những người dùng khác và*

*gán cho nó những đặc quyền. Bạn cũng có thể tạo ra những nhóm của các người dùng và phân quyền cho các nhóm đó, hơn là làm với mỗi người dùng riêng lẻ.*

# 11

## Chương 11 – Sao lưu và khôi phục

Trong chương này chúng ta thảo luận về nhật ký cơ sở dữ liệu DB2, làm thế nào để sao lưu toàn bộ hay một phần cơ sở dữ liệu sử dụng công cụ BACKUP và làm thế nào để khôi phục dữ liệu sử dụng công cụ RESTORE.

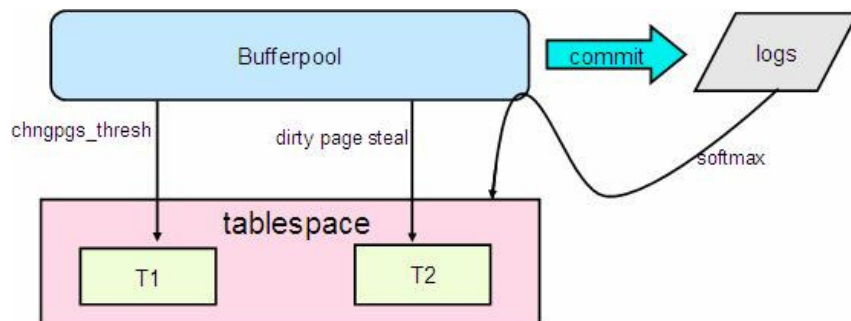
### 11.1 Nhật ký cơ sở dữ liệu

Nếu bạn đang làm việc với một trình soạn thảo văn bản, mỗi lần bạn muốn lưu tài liệu phải nhấn vào nút SAVE. Trong cơ sở dữ liệu, câu lệnh COMMIT sẽ làm điều đó. Mỗi lần câu lệnh COMMIT được thực hiện thì mọi thay đổi tạo thành dữ liệu và được lưu trong nhật ký cơ sở dữ liệu.

Tương tự, khi bạn làm việc với một tài liệu văn bản, đôi khi bạn thấy ở dưới góc phải màn hình có một thông điệp ngắn “auto-saving”. Trong cơ sở dữ liệu, điều này được thực hiện bởi lệnh UPDATE, INSERT và DELETE.

Ở phần trên ta đã đề cập đến nhật ký cơ sở dữ liệu. Nhật ký cơ sở dữ liệu được lưu trữ trên đĩa cứng và được dùng ghi lại những hành động của những giao dịch. Nếu hệ thống hay cơ sở dữ liệu bị hư hại, chúng được dùng nó để lấy lại những giao dịch đã được gửi đi trong suốt quá trình khôi phục dữ liệu.

Hình 11.1 minh họa luồng xử lý khi làm việc với cơ sở dữ liệu trong những mẫu nhật ký



Hình 11.1 - Nhật ký cơ sở dữ liệu

Trong hình 11.1 ta thấy không gian bảng (tablespace) và nhật ký (logs), chúng đều được lưu trữ trên đĩa cứng, mặc dù ta đề xuất rằng chúng không lưu trữ trên cùng đĩa cứng. Ví dụ, khi câu lệnh UPDATE được thực hiện, những trang cho những dòng trong câu lệnh sẽ được lưu trong vùng đệm dữ liệu (bộ nhớ). Những thay đổi, cập nhật sẽ được thể hiện trong vùng đệm dữ liệu, những giá trị cũ và mới sẽ được lưu trữ trong những tập tin nhật ký cơ sở dữ liệu, đôi khi tức thời, đôi khi một vùng đệm nhật ký đầy. Nếu câu lệnh COMMIT được thi hành sau câu lệnh UPDATE thì những giá trị cũ và mới sẽ được

cất trong những tập tin nhật ký ngay lập tức. Xử lý này cũng được lập lại tương tự cho câu lệnh SQL khác truy xuất đến cơ sở dữ liệu. Khi điều kiện được cụ thể thỏa mãn, như là việc đạt đến ngưỡng của sự thay đổi trong tham số CHNGPGS\_THRES thì những trang trong vùng đệm dữ liệu sẽ được “chi tiết hóa” hay ghi lên thành không gian bảng trên đĩa cứng. Tham số CHNGPGS\_THRES chỉ định tỉ lệ phần trăm của vùng đệm dữ liệu với những trang “dirty” đó là những trang chứa những thay đổi.

Về khía cạnh hiệu năng, ta không có cảm giác thực hiện 2 lần ghi cho mỗi lệnh COMMIT: Một là ghi đến nhật ký cơ sở dữ liệu và một ghi đến không gian bảng trên đĩa cứng, đó là lý do tại sao “sự chi tiết hóa” của dữ liệu đến không gian bảng trên đĩa cứng chỉ xảy ra khi tham số “chngpgs\_thres” đạt ngưỡng.

**11.2 Các kiểu nhật ký**

Có hai kiểu nhật ký:

**Nhật ký sơ cấp:**

Được cấp phát sẵn và số nhật ký sơ cấp sẵn sàng được xác định bởi tham biến db cfg LOGPRIMARY

**Nhật ký thứ cấp:**

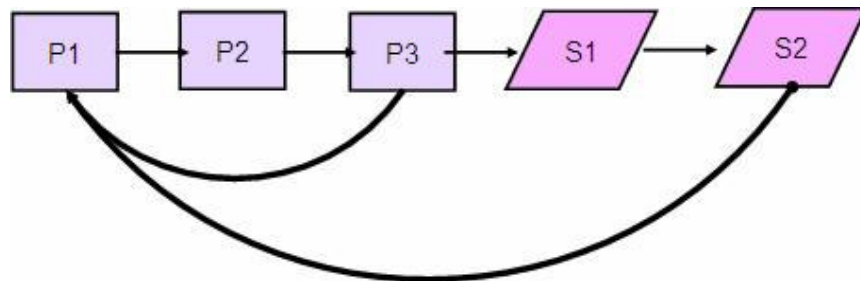
Phần nhật ký thứ cấp này được DB2 cấp phát động khi cần. Số nhật ký thứ cấp được gán giá trị bởi tham biến db cfg LOGSECOND. Việc cấp phát động một nhật ký sẽ gây ra sự lãng phí vì vậy nhật ký thứ cấp sẽ được xóa khi tất cả kết nối đến cơ sở dữ liệu đã bị ngắt

**11.3 Các kiểu ghi nhật ký**

Có hai kiểu ghi nhật ký: Ghi tuần tự quay vòng (mặc định) và ghi lưu trữ

**11.3.1 Ghi tuần tự quay vòng**

Hình 11.2 mô tả luồng xử lý khi ghi tuần tự quay vòng



**Hình 11.2 - Xử lý nhật ký sơ cấp và thứ cấp**

Trong hình 11.2 có 3 nhật ký sơ cấp, vì vậy chúng ta gán LOGPRIMARY bằng 3. Để cho đơn giản trong ví dụ này chỉ có 1 giao dịch. Khi giao dịch được thực hiện không gian sử dụng bắt đầu ghi vào tập tin nhật ký P1 và tiếp theo sau là P2. Nếu có một lệnh COMMIT được thực hiện và thông tin sau đó được xuất đến không gian bảng trên đĩa cứng, sau đó P1 và P2 có thể được ghi đè dữ liệu khác vì thông tin cũ không cần thiết nữa. Nếu, mặt khác, giao dịch đã sử dụng P1, P2, P3 và vẫn cần nhiều không gian nhật ký hơn nữa bởi vì giao dịch chưa được COMMIT, thì một nhật ký thứ cấp được cấp phát (S1 trên hình vẽ) một cách tự động. Nếu giao dịch vẫn còn tiếp tục, nhiều nhật ký thứ cấp được cấp phát cho đến khi số nhật ký thứ cấp được cấp phát đạt ngưỡng tối đa



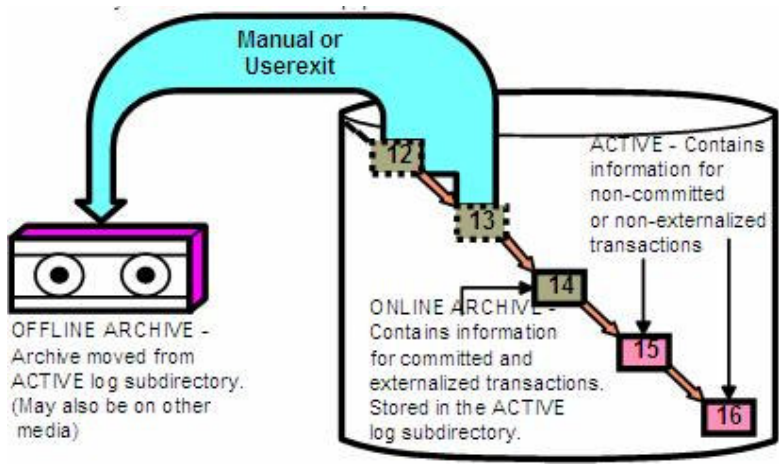
LOGSECOND. Nếu vẫn chưa đủ, một thông điệp lỗi chỉ ra điều kiện hết khả năng cấp phát nhật ký và giao dịch sẽ được quay lui (hồi xử).

**11.3.2 Nhật ký lưu trữ**

Nhật lưu trữ cũng được biết đến như là ghi giữ lại, những tập tin nhật ký không bị ghi đè nhưng chúng được lưu trữ trực tuyến hay ngoại tuyến. Những nhật ký lưu trữ trực tuyến được lưu trữ với những nhật ký động cần cho khôi phục khi hệ thống bị sự cố. Những nhật ký lưu trữ ngoại tuyến được lưu trong băng từ và có thể làm việc với thủ tục USEREXIT. Để có thể ghi nhật ký lưu trữ cần gán tham số db cfg LOGRETAIN là YES.

Cách ghi lưu trữ thường được dùng trong những hệ thống sản phẩm. Những nhật ký được lưu trữ cho phép khôi phục cơ sở dữ liệu sớm nhất có thể bởi những tập tin nhật ký cũ nhất trong hầu hết các tình huống. Với cách ghi lưu trữ người quản trị cơ sở dữ liệu có thể khôi phục những lỗi gây ra bởi con người. Một ví dụ cụ thể là khi một người trong hệ thống bắt đầu thực hiện một giao tác không đúng kéo dài hàng ngày, khi vấn đề được phát hiện thì người quản trị cơ sở dữ liệu sẽ khôi phục lại hệ thống cũ vào thời điểm trước khi vấn đề được xảy ra. Tuy nhiên, vẫn còn có một số thao tác bắt buộc phải làm bằng tay để đảm bảo việc khôi phục được chính xác.

Cách ghi lưu trữ là cần thiết đối với quá trình khôi phục lại và khôi phục trực tuyến. Hình 11.3 mô tả luồng xử lý của ghi lưu trữ

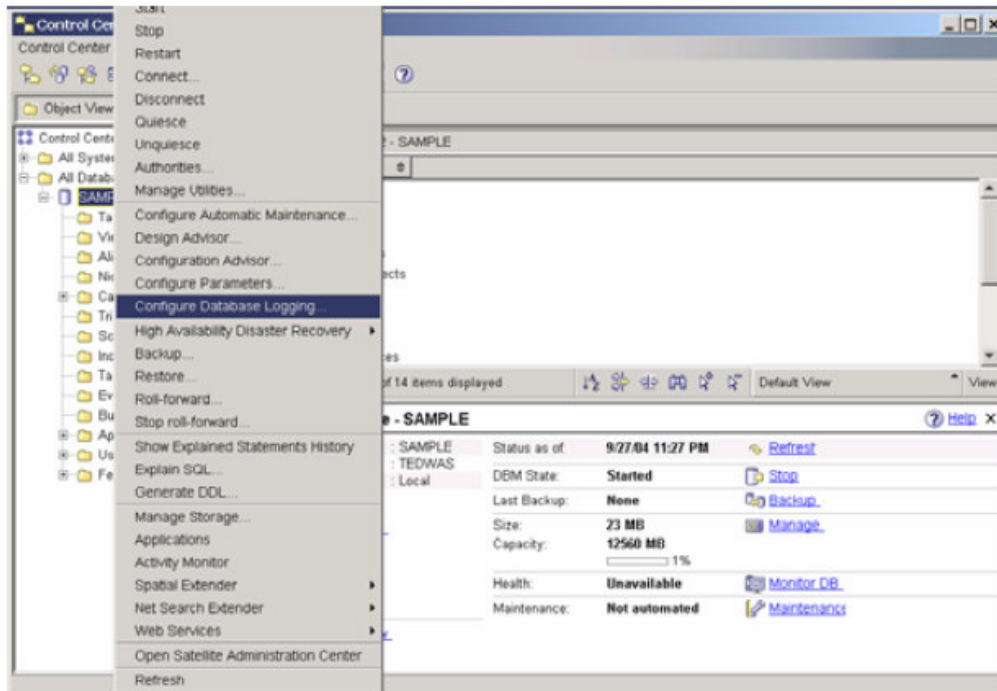


Hình 11.3 - Ghi lưu trữ

**11.4 Nhật ký cơ sở dữ liệu từ Trung tâm điều khiển**

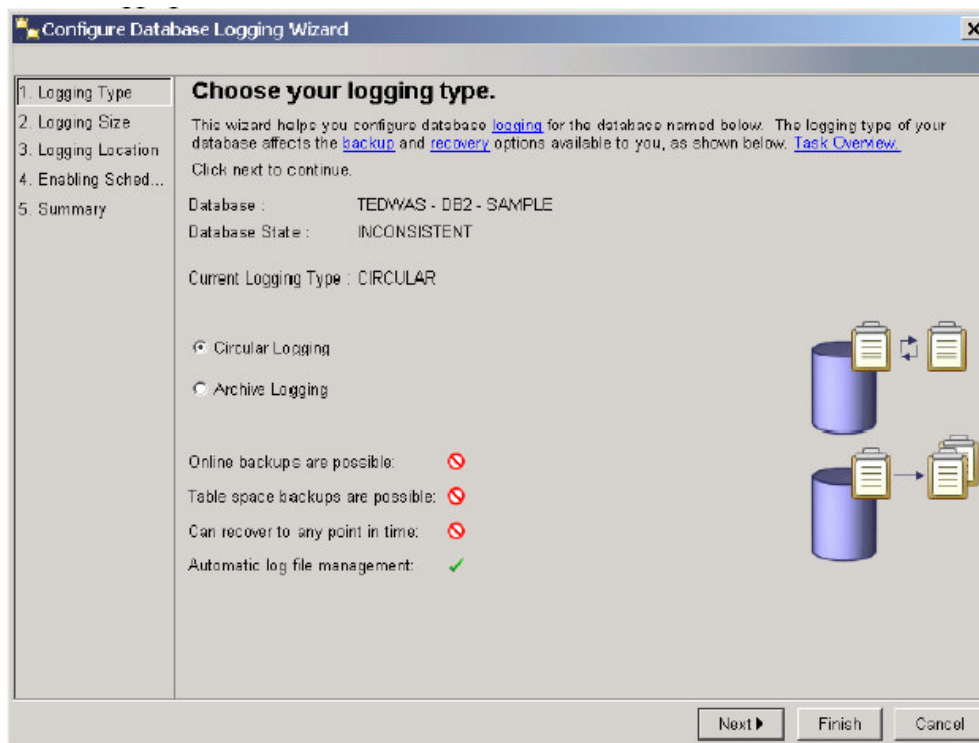
Bạn có thể cấu hình nhật ký cơ sở dữ liệu từ trung tâm điều khiển bằng cách nhấn chuột phải trên cơ sở dữ liệu và chọn "Configure Database Logging"





Hình 11.4 - Nhật ký cơ sở dữ liệu từ Trung tâm điều khiển

Hình 11.5 thể hiện màn hình thiết lập cấu hình nhật ký cơ sở dữ liệu, tại đây ta có thể chọn kiểu ghi nhật ký quay vòng (circular) hoặc ghi lưu trữ



Hình 11.5 - Ghi nhật ký cơ sở dữ liệu theo hướng dẫn từng bước

### 11.5 Các tham số của nhật ký

Có những tham số DB CFG liên quan đến việc tạo bản ghi nhật ký. Bảng 11.1 liệt kê danh sách các tham số chính.

Tham số	Mô tả
logbufsz	Lượng bộ nhớ được sử dụng như một bộ đệm cho các bản ghi nhật ký trước khi ghi chúng vào đĩa.
logfilsz	Kích thước của mỗi nhật ký đã cấu hình, có dung lượng 4KB
logprimary	Số bản nhật ký chính với kích thước logfilsz sẽ được tạo
logsecond	Số tệp nhật ký phụ được tạo và sử dụng cho việc khôi phục nếu cần thiết
logpath/newlogpath	Vị trí các nhật ký hoạt động hoặc lưu trữ tương lai được đặt.
mirrologpath	Để bảo vệ các nhật ký trên đường dẫn chính khỏi sự cố hỏng đĩa hay vô tình xóa, bạn có thể chỉ định một thiết lập đồng nhất cho các nhật ký được lưu giữ trên một đường dẫn phụ (ảnh của đường dẫn nhật ký chính)
loghead	Tên tệp nhật ký hoạt động hiện hành
userexit	Cho phép chương trình userexit sao chép nhật ký ngoại tuyến
softmax	Giới hạn sự mất mát của việc khôi phục hỏng
logretain	Cho phép chế độ ghi nhật ký lưu trữ
overflowlogpath	Tương tự tùy chọn OVERFLOW LOG PATH của lệnh ROLLFORWARD; tuy nhiên, thay vì chỉ định tùy chọn OVERFLOW LOG PATH cho mỗi lệnh ROLLFORWARD được đưa ra, bạn có thể thiết lập tham số cấu hình này một lần
blk_log_dsk_ful	Thiết lập tránh việc tràn đĩa phát sinh khi DB2 không thể tạo ra một tệp nhật ký mới trong đường dẫn nhật ký đang hoạt động. Thay vì đó, DB2 sẽ cố gắng tạo ra tệp nhật ký mỗi năm phút đến khi nó thành công. Khi không còn trở ngại, SQL chỉ-đọc có thể tiếp tục.
maxlog	Phần trăm số nhật ký hoạt động tối đa cách nhau qua giao dịch
num_log_span	Số tệp nhật ký hoạt động cho một UOW kích hoạt

**Bảng 11.1- Các tham số nhật ký**

**11.6 Sao lưu cơ sở dữ liệu**

Lệnh sao lưu dữ liệu DB2 cho phép bạn tạo một bản sao chép cơ sở dữ liệu lúc lệnh được thực thi. Cú pháp đơn giản nhất bạn có thể sử dụng để chạy lệnh này là:

```
BACKUP DATABASE <dbname> [ TO <path> ]
```

Hầu hết các lệnh và công cụ có thể được thực hiện trực tuyến hay ngoại tuyến. Trực tuyến nghĩa là để người sử dụng khác có thể được kết nối và thực hiện các thao tác trên cơ sở dữ liệu trong khi bạn thực thi lệnh của mình. Ngoại tuyến là không người nào được kết nối đến cơ sở dữ liệu trong khi bạn thực hiện thao tác. Để cho phép một thao tác trực tuyến, thêm từ khóa ONLINE vào cú pháp lệnh, nếu không thì ngầm định bạn sẽ làm việc trong chế độ ngoại tuyến.

Ví dụ, nếu muốn sao lưu cơ sở dữ liệu *sample* đến đường dẫn C:\BACKUPS bạn có thể đưa ra lệnh này từ trình đơn lệnh DB2 của Window/Linux:

**db2 BACKUP DB sample TO C:\BACKUPS**

Chú ý rằng thư mục C:\BACKUPS phải tồn tại trước khi thực thi lệnh. Cũng chắc chắn là không có kết nối nào đến cơ sở dữ liệu khi bạn thực thi lệnh trên, nếu không bạn sẽ nhận một thông báo lỗi bởi một sao lưu ngoại tuyến không thể được thực hiện khi có các kết nối.

Trong trường hợp để tìm ra những kết nối đến cơ sở dữ liệu, đưa ra lệnh DB2 này từ trình đơn Window/Linux:

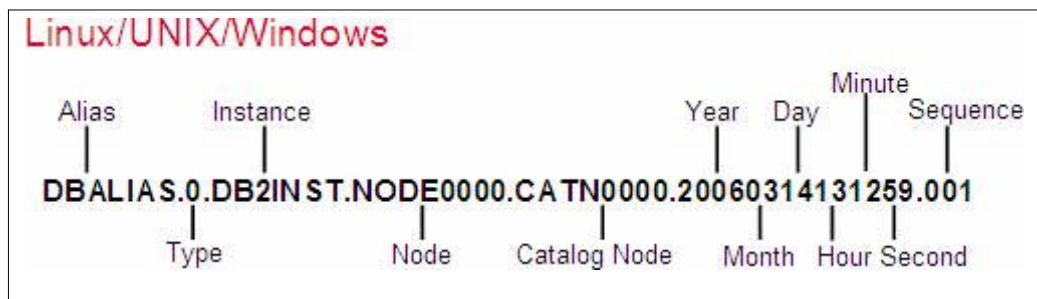
**db2 list applications**

Để buộc ngắt tất cả kết nối đến cơ sở dữ liệu, đưa ra lệnh từ trình đơn lệnh DB2 Window hoặc của Linux:

**db2 force application all**

Bạn có thể không cần chạy lệnh cuối cùng trong môi trường sản xuất với nhiều người sử dụng, nếu không bạn sẽ nhận những cuộc gọi từ những đồng nghiệp nóng tính đấy! Cũng lưu ý rằng lệnh cuối cùng chạy theo cách thức không đồng bộ. Nghĩa là khi bạn cố gắng chạy lệnh sao lưu ngay sau đó, nó có thể không làm việc. Đợi vài giây, và lặp lại lệnh sao lưu nếu bạn đã gặp lỗi lúc đầu.

Sau khi thực thi thành công lệnh sao lưu, một tệp mới chứa ảnh cơ sở dữ liệu sao lưu được tạo ra. Tên của tệp này theo quy ước được chỉ trong hình 11.6



**Hình 11.6 - Quy ước đặt tên ảnh sao lưu**

Loại “0” có nghĩa đó là bản sao lưu đầy đủ. Loại “3” chẳng hạn, nghĩa là chỉ một sao lưu bằng biểu trống. Nút được cố định với NODE0000 cho cơ sở dữ liệu không phân tán, là trường hợp cho tất cả ấn bản DB2 trừ ấn bản thương mại DB2 Enterprise Edition với tính năng DPF. Nút catalog cũng được cố định với CAT0000. Tham khảo sách hướng dẫn DB2 để có chi tiết hơn

Khi có vài bản sao lưu được ghi lại và lưu trữ trên cùng đường dẫn thì mốc thời điểm ghi lại ở cuối tên tệp được sử dụng để phân biệt giữa các ảnh sao lưu. Như chúng ta sẽ thấy ở phần tiếp theo, lệnh RESTORE có thể sử dụng mốc thời điểm này để phục hồi một bản sao đã định.

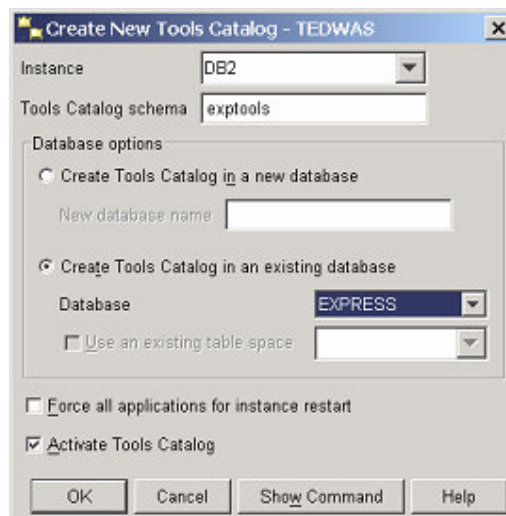
## Bài tập nhanh #8 - Lập lịch sao lưu

### Mục đích:

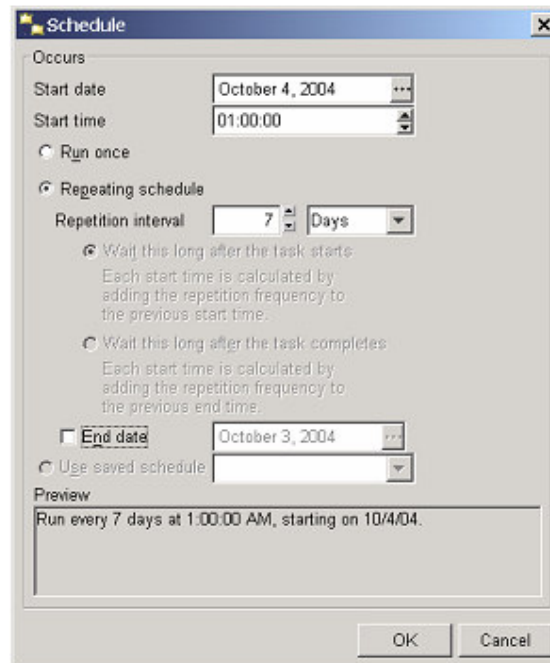
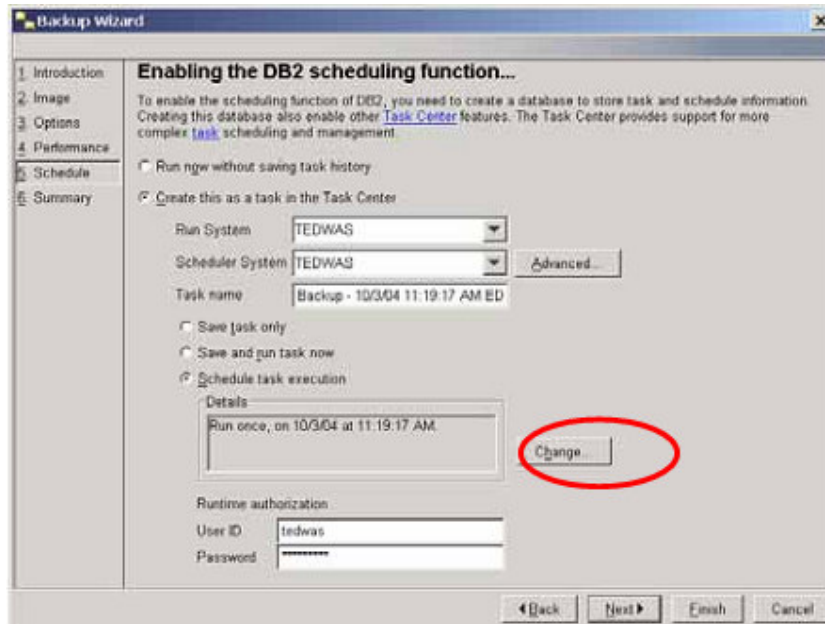
Mặc dù DB2 có một vài khả năng tự động duy trì cơ sở dữ liệu, nhưng đôi khi bạn cũng muốn hiệu chỉnh chúng khi có một hoạt động cụ thể diễn ra. Trong phần bài tập nhanh này, bạn sẽ hiệu chỉnh lịch sao lưu cho cơ sở dữ liệu *EXPRESS*.

### Các thủ tục :

1. Từ cây đối tượng ở Trung tâm điều khiển, bạn chọn *All Database*. Nhấp chuột phải trên cơ sở dữ liệu *EXPRESS* và chọn biểu tượng *Backup*. Công cụ từng bước *Backup* được khởi động.
2. Trong mục *Introduction* sẽ tóm tắt trạng thái hiện tại của cơ sở dữ liệu bao gồm thời điểm sao lưu cuối cùng và phương thức truy cập. Bấm *Next* để chuyển tới mục tiếp theo của công cụ *Backup*.
3. Trong mục *Image*, chọn đích đến của dữ liệu sẽ được sao lưu. Thông thường, bạn nên chọn một ổ đĩa khác với ổ đĩa đang lưu cơ sở dữ liệu hiện hành. Bây giờ, tạo một thư mục mới trong hệ thống có tên là *C:\db2backup*, đây sẽ là thư mục chứa cơ sở dữ liệu sao lưu. Hãy chọn biểu tượng *File System* trong danh sách thả *Media Type*. Bấm nút *Add*, chọn thư mục bạn vừa tạo ra, rồi bấm *OK*. Sau đó bấm *Next* để chuyển sang mục tiếp theo.
4. Bạn có thể mở mục *Options and Performance* tuy nhiên bạn nên để chế độ mặc định vì DB2 mặc định cho phép sao lưu ở chế độ tối ưu nhất. Tiếp theo bạn chọn chọn mục *Schedule*.
5. Trong mục *Schedule*, nếu như trình lập lịch chưa được kích hoạt thì bạn hãy kích hoạt chúng. Chọn hệ thống để tạo danh mục các công cụ và thực hiện chúng. Chỉ ra một giản đồ cho danh mục công cụ và chọn để tạo nó trong cơ sở dữ liệu *EXPRESS* hiện hành. Danh mục công cụ sẽ chứa dữ liệu biến đổi về tất cả các tác vụ lập lịch. Bấm nút *OK* để tiếp tục. Bấm *Next* để chuyển sang mục tiếp theo sau khi danh mục công cụ đã được tạo lập.



- Trong mục *Schedule*, tạo ra một mục lục cho việc thi hành các tác vụ. Lập lịch sao lưu mỗi ngày vào lúc 1 giờ sáng. Bấm *Next* để chuyển sang mục kế tiếp.



- Trong mục *Summary*, bạn sẽ xem lại các tác vụ sẽ được tạo ra. Sau đó bấm *Finish* để tạo ra tác vụ.
- Chạy Trung tâm tác vụ (Task Center) để xem hoặc sửa chữa các tác vụ sao lưu vừa mới tạo ra.

### 11.7 Khôi phục cơ sở dữ liệu

Khôi phục cơ sở dữ liệu bao hàm việc khôi phục cơ sở dữ liệu từ các bản ghi dự phòng và nhật ký. Nếu như bạn chỉ khôi phục cơ sở dữ liệu từ bản ghi dự phòng thì sẽ tạo ra các cơ sở dữ liệu như đã tồn tại ở thời điểm sao lưu.

Nếu việc lưu trữ bản ghi được cho phép trước việc sao lưu, thì bạn không chỉ khôi phục lại bằng cách sử dụng ảnh sao lưu, mà còn từ các tệp nhật ký. Ở phần tiếp theo ta sẽ thấy cách khôi phục dữ liệu cuộn cho phép bạn có thể khôi phục từ các bản ghi dự phòng và sau đó ghép vào phần cuối cùng của các tệp nhật ký, hoặc một điểm cụ thể theo thời gian.

Chú ý rằng từ khôi phục (recovery) được dùng nhiều trong phần này, nhưng lệnh khôi phục lại là RESTORE.

#### 11.7.1 Các kiểu khôi phục dữ liệu

Có ba loại khôi phục dữ liệu:

##### Khôi phục dữ liệu hỏng:

Giả thiết rằng bạn đang làm việc với cơ sở dữ liệu DB2 trên máy tính thì đột ngột mất điện, chuyện gì sẽ xảy ra với cơ sở dữ liệu của bạn?

Khi bạn khởi động lại máy tính và khởi động DB2, chương trình khôi phục dữ liệu hỏng sẽ được tự động thực thi. DB2 sẽ tự động chạy lệnh RESTART DATABASE và sẽ đọc và phục hồi các giao dịch khi trước ở trạng thái kích hoạt. Sau khi hoàn thành, bạn sẽ được đảm bảo rằng, cơ sở dữ liệu của bạn ở trạng thái phù hợp nhất.

##### Khôi phục dữ liệu phiên bản (hoặc khôi phục ảnh):

Có nghĩa là, bạn sẽ phục hồi cơ sở dữ liệu của bạn từ ảnh một bản sao. Do đó cơ sở dữ liệu của bạn được trả về trạng thái ngay trước khi nó được sao lưu. Mọi tác động lên cơ sở dữ liệu đó sau thời điểm sao lưu đều bị mất.

##### Khôi phục dữ liệu theo thời gian:

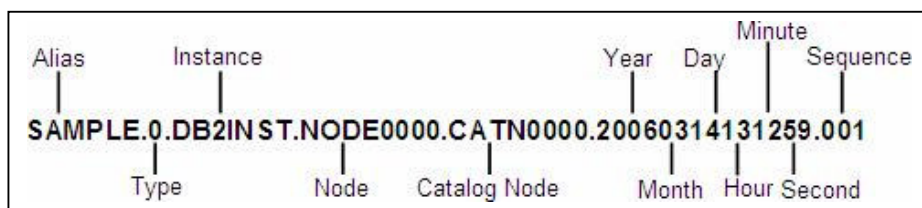
Kiểu khôi phục dữ liệu này vừa cho phép bạn khôi phục cơ sở dữ liệu từ một bản sao, vừa khôi phục dữ liệu tại một thời điểm xác định bằng lệnh ROLLFORWARD. Khôi phục dữ liệu kiểu này làm giảm sự mất mát dữ liệu một cách tối đa.

#### 11.7.2 Khôi phục cơ sở dữ liệu

Dùng lệnh RESTORE để khôi phục dữ liệu từ một bản lưu. Dưới đây là cấu trúc đơn giản nhất cho câu lệnh này:

```
RESTORE DATABASE <tên c. s. d. li.u> [from <...ng d.n>] [taken at <th.i-.i.m>]
```

Ví dụ: Nếu bạn có một tệp sao lưu có tên như sau:



Bạn sẽ gõ lệnh như sau:

```
RESTORE DB sample FROM <...ng d.n> TAKEN AT 2006314131259
```

### 11.8 Các hoạt động khác với BACKUP và RESTORE

Dưới đây là danh sách một vài điều bạn nên biết về lệnh BACKUP và RESTORE. Chúng tôi khuyên bạn nên đọc lại cuốn hướng dẫn về DB2 để có thêm thông tin chi tiết.

- Sao lưu cơ sở dữ liệu ở chế độ 32 bit còn khôi phục dữ liệu ở chế độ 64 bit.
- Khôi phục cả cơ sở dữ liệu đang tồn tại.
- Khôi phục dữ liệu tới một ổ đĩa khác với vùng cần khôi phục.
- Khôi phục hoặc sao lưu dữ liệu theo không gian bảng, thay vì làm với toàn bộ cơ sở dữ liệu.
- Cho phép sao lưu những thay đổi (Delta) và sao lưu tăng dần (incremental). Cách sao lưu thay đổi chỉ sao lưu những thay đổi tiếp theo, trong khi đó cách sao lưu dần tăng thì ghi nhận toàn bộ những thay đổi và tích hợp chúng vào mỗi ảnh sao lưu.
- Sao lưu các bản “chụp” dữ liệu (flash copy) (đòi hỏi về phần cứng tương thích)
- Khôi phục bảng đã mất (nếu tùy chọn được lập cho bảng)
- Sao lưu ở hệ điều hành này và khôi phục ở hệ điều hành khác là không thể. Dùng db2look và db2move cho trường hợp này.

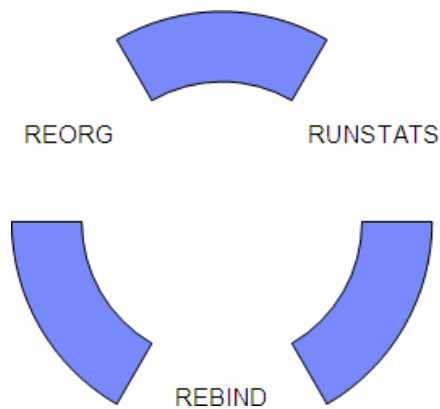
# 12

## Chương 12 - Nhiệm vụ bảo trì

Trong chương này chúng ta sẽ bàn về các nhiệm vụ cần thiết để giữ cho CSDL của bạn duy trì hoạt động được tốt. Tất cả mọi thứ trong DB2 đều tự động thực hiện những công việc này. Bản DB2 Express-C cũng giống như tất cả các bản DB2 hiện thời đều chứa đựng tất cả các khả năng tự động này. Khả năng tự quản lý là một lợi ích lớn làm nhỏ gọn và làm vửa phải số lượng nhân viên trong những công ty, những người không cần thuê làm liên tục làm quản trị để quản lý máy chủ dữ liệu. Khác với việc làm bằng tay, nếu một người quản trị được thuê, anh ấy hay cô ấy sẽ có nhiều thời gian rảnh rỗi để thực hiện việc nâng cao hoạt động làm gia tăng giá trị cho công ty.

### 12.1 REORG, RUNSTATS, REBIND

Có ba công việc duy trì chính trong DB2, như hình 12.1 mô tả: REORG, RUNSTATS và REBIND.



**Hình 12.1 – Các công việc duy trì: REORG, RUNSTATS, REBIND**

Hình 12.1 trình bày những công việc duy trì được thực hiện theo một chu trình hình tròn. Nếu một REORG được thực hiện, nó được khuyên là nên chạy một RUNSTATS, theo sau là một REBIND. Sau một thời gian, các bảng trong một cơ sở dữ liệu đã được thay đổi theo các lệnh như UPDATE, DELETE và INSERT. Tại thời điểm đó, chu kỳ mới lại được bắt đầu với REORG.

#### 12.1.1 Lệnh REORG

Qua thời gian, bạn thực hiện các tính toán INSERT, UPDATE, DELETE trên cơ sở dữ liệu của bạn, dữ liệu của bạn bắt đầu bị phân đoạn, chia nhỏ ra thành các trang cơ sở dữ liệu. Lệnh REORG sẽ tái ghép những mảnh vụn cũng như những khoảng trống lại và sẽ tổ chức lại dữ liệu để làm cho việc truy vấn dữ liệu trở nên hiệu quả hơn. Các bảng hay thay đổi sẽ có lợi nhất khi dùng REORG. Bạn có thể làm chỉ mục REORG như với



các bảng, và một REORG có thể được thực hiện trực tuyến hay không trực tuyến.

Lệnh REORG không trực tuyến thực hiện nhanh hơn và hiệu quả hơn, nhưng không cho phép truy nhập tới bảng, trong khi lệnh REORG trực tuyến cho phép truy nhập tới bảng, nhưng có thể làm tiêu hao một chút tài nguyên hệ thống; đây là những công việc tốt nhất cho những bảng nhỏ.

Cú pháp:

```
REORG TABLE <tablename>
```

Ví dụ:

```
REORG TABLE employee
```

Lệnh REORGCHK có thể được dùng trước một REORG để xác định bảng hay chỉ số cần được sửa.

### 12.1.2 Lệnh RUNSTATS

Cơ chế tối ưu của DB2 (Optimizer) là “khối óc” của DB2. Nó tìm những đường dẫn truy nhập hiệu quả nhất để định vị và truy vấn dữ liệu. Cơ chế tối ưu là hệ thống đánh giá tri thức, và sử dụng phân tích thống kê các đối tượng cơ sở dữ liệu được lưu trong các bảng danh mục cho hiệu suất khai thác cơ sở dữ liệu lớn nhất. Ví dụ, các bảng danh mục có các phân tích thống kê về có bao nhiêu cột hiện diện trong một bảng, có bao nhiêu hàng trong nó, có bao nhiêu và những kiểu gì của chỉ số sẵn có với một bảng, ...

Thông tin về phân tích thống kê không được cập nhật tự động. Chính vì thiết kế kiểu này, nên bạn không muốn DB2 cập nhật những thống kê sau mỗi thao tác trên cơ sở dữ liệu, nó sẽ phản tác dụng với yêu cầu về hiệu năng của hệ thống. Thay vào đó, DB2 cung cấp lệnh RUNSTATS để cập nhật những thống kê này. Nó giữ cho các thống kê cơ sở dữ liệu luôn cập nhật. Cơ chế tối ưu của DB2 có thể tạo ra những thay đổi gốc rễ cơ bản trong đường dẫn truy nhập nếu bảng chỉ có một dòng so với bảng có một triệu dòng. Khi những thống kê cơ sở dữ liệu được cập nhật, DB2 có thể chọn một phương án truy nhập tốt hơn. Tần suất của các thống kê tập hợp này sẽ xác định được dữ liệu trong bảng thay đổi như thế nào.

Cú pháp:

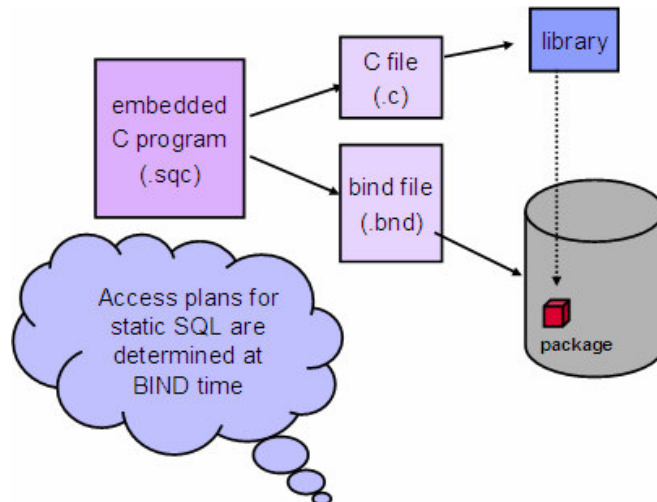
```
RUNSTATS ON TABLE <schema.tablename>
```

Ví dụ:

```
RUNSTATS ON TABLE myschema.employee
```

### 12.1.3 BIND/REBIND

Sau khi chạy thành công lệnh RUNSTATS, không phải tất cả các câu truy vấn sẽ được dùng những thống kê mới nhất. Những sơ đồ truy nhập SQL tĩnh được xác định khi bạn đưa ra lệnh BIND, vì các thống kê được dùng tại thời điểm đó không giống nhau như thời điểm hiện tại. Hình 12.2 sẽ giúp chúng ta hiểu ý tưởng này.



**Hình 12.2 – Tiến trình kết nối SQL tĩnh.**

Trong hình 12.2 một chương trình nhúng viết bằng C (được lưu trữ như một tệp với phần mở rộng là “sqc”) được tiền biên dịch. Sau khi tiền biên dịch, hai tệp được sinh ra, một tệp “.c” chứa mã C với tất cả chú thích SQL, và một tệp “.bnd” chứa tất cả các câu lệnh SQL. Tệp C với phần mở rộng “.c” được biên dịch như thường lệ với trình biên dịch C, tạo một “thư viện” như trình bày phía trên bên phải của hình. Tệp “.bnd” giống như vậy, nó được đóng gói lưu trong cơ sở dữ liệu. Đóng gói thì tương đương với biên dịch các câu lệnh SQL với những sơ đồ truy nhập tốt nhất được xác định trên các thống kê đã có tại thời điểm này, và sau đó lưu chúng thành gói.

Bây giờ, điều gì sẽ xảy ra nếu một triệu dòng được chèn vào bên trong bảng sử dụng trong chương trình nhúng trên? Sau khi chèn, nếu một lệnh RUNSTATS được thực thi, những thống kê sẽ được cập nhật, tuy nhiên gói sẽ không tự động cập nhật để tính toán lại đường dẫn cơ sở trên những thống kê mới nhất. Lệnh db2rbind có thể được dùng để đóng gói lại các gói trên với thống kê mới nhất.

Cú pháp:

```
db2rbind database_alias -l <logfile>
```

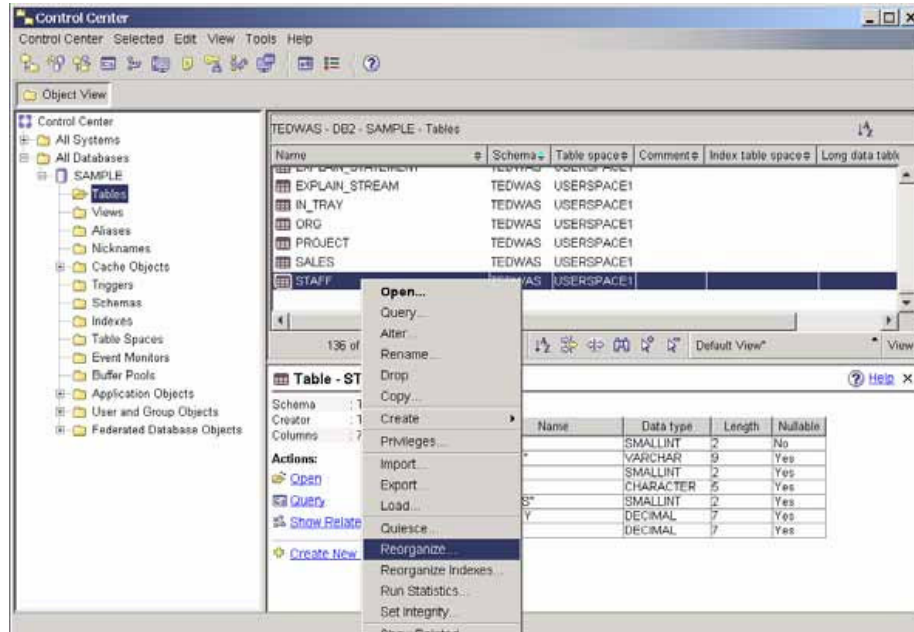
Ví dụ:

Để đóng gói lại tất cả các gói của cơ sở dữ liệu *sample* và lưu nó trong nhật ký tại tệp mylog.txt, thực hiện lệnh sau:

```
db2rbind sample -l mylog.txt
```

### 12.1.4 Các công việc duy trì từ Trung tâm điều khiển

Từ Trung tâm điều khiển bạn có thể REORG và RUNSTATS. Hình 12.3 trình bày cách thực hiện.

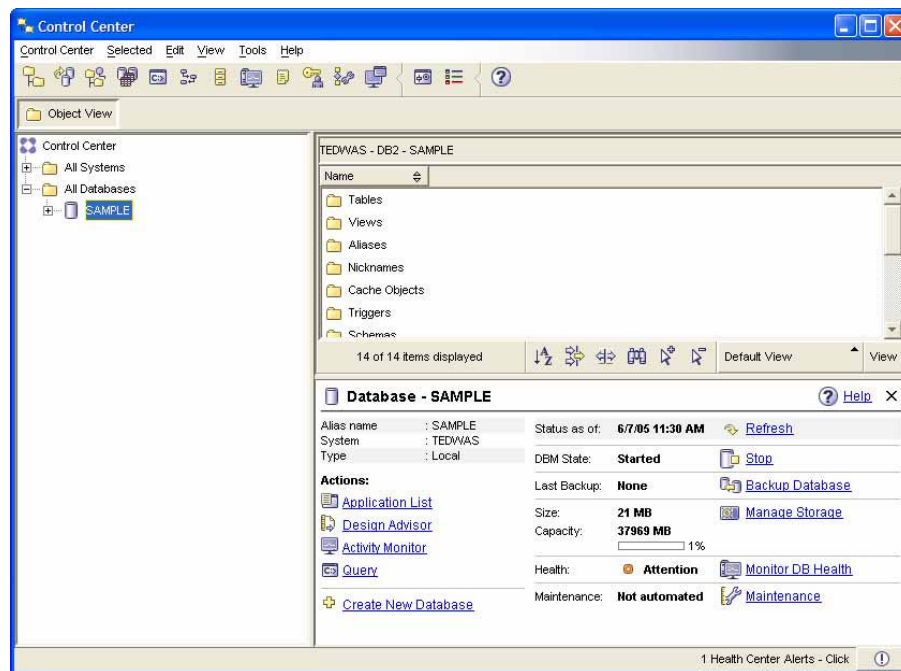


**Hình 12.3 – REORG và RUNSTATS từ Trung tâm điều khiển**

Bạn chọn bảng mà bạn muốn tổ chức lại, nhấp phải chuột và chọn tổ chức lại (cho REORG) hay chạy thống kê (cho RUNSTATS)

**Hiển thị thông số cơ sở dữ liệu:**

Khi bạn chọn một cơ sở dữ liệu, khung nhìn hoạt động cơ sở dữ liệu ở phía dưới bên phải của Trung tâm điều khiển sẽ cung cấp thông tin về cơ sở dữ liệu như là về kích thước, của lần sau cùng, dù nó có được bảo trì tự động hay không. Việc hiển thị này cho phép bạn nhanh chóng xác định những duy trì cần thiết cho cơ sở dữ liệu của bạn. Hình 12.4 hiển thị những thông tin này.



**Hình 12.4 – Hiển thị thông số cơ sở dữ liệu từ Trung tâm điều khiển**

## 12.2 Các lựa chọn bảo trì

Có 3 cách để thực hiện việc bảo trì hệ thống.

### 1. Bảo trì thủ công

Bạn thực hiện hoạt động bảo trì này một cách thủ công khi có nhu cầu phát sinh.

### 2. Tạo ra một đoạn kịch bản (script) để thực hiện việc bảo trì hệ thống.

Bạn có thể tạo các kịch bản với những lệnh bảo trì và lập lịch hợp lý cho nó hoạt động.

### 3. Tự động bảo trì

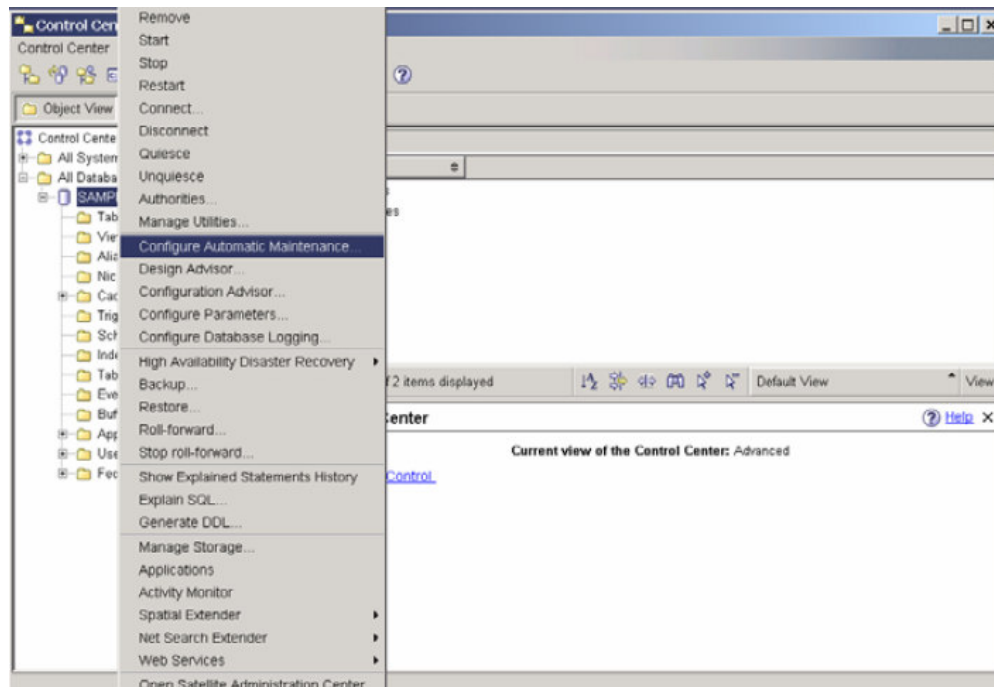
DB2 tự động giám sát công việc bảo trì cho bạn (bằng lịch REORG, RUNSTATS, BACKUP)

Trong phần này, chúng ta tập trung chủ yếu vào cách bảo trì tự động.

Hoạt động bảo trì tự động bao gồm những bước sau đây:

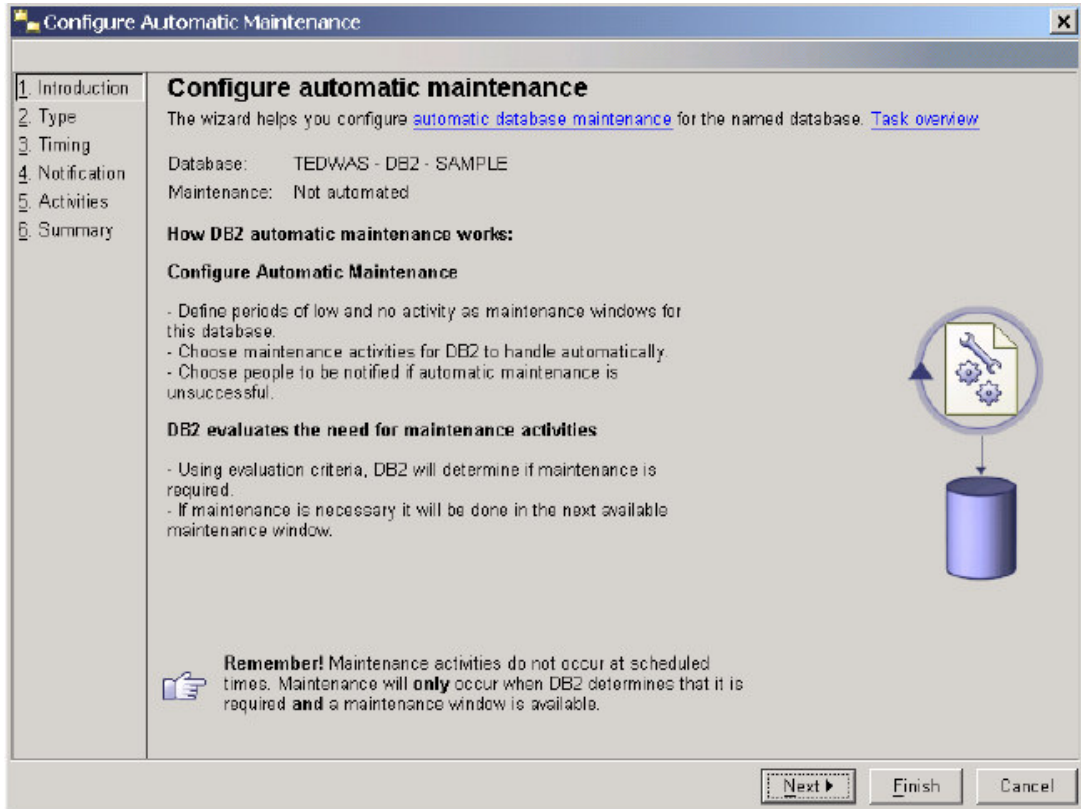
- Người dùng định nghĩa một *cửa sổ bảo trì (khung thời gian)* tại thời điểm đó hệ thống sẽ thực hiện công việc bảo trì mà không gây ra sự xáo trộn. Ví dụ, nếu hệ thống có ít nhất các hoạt động vào ngày chủ nhật, từ 2h sáng tới 4h sáng, thì khung thời gian này sẽ hoạt động như là một cửa sổ bảo trì.
- Có 2 loại cửa sổ bảo trì chính: một là để thực hiện trực tuyến, còn loại kia là để thực hiện ngoại tuyến.
- DB2 chỉ thực hiện bảo trì một cách tự động khi cần thiết và ở đang trong thời gian kích hoạt của cửa sổ bảo trì.

Từ giao diện điều khiển chính, bạn có thể mở “Configure Automated Maintenance Wizard” như hình minh họa 12.5.



Hình 12.5 – Mở Configure Automated Maintenance Wizard

Hình 12.6 sẽ hiện ra cửa sổ “Cấu hình để bảo trì hệ thống một cách tự động theo hướng dẫn”.



Hình 12.6 – Cấu hình để bảo trì hệ thống một cách tự động theo hướng dẫn

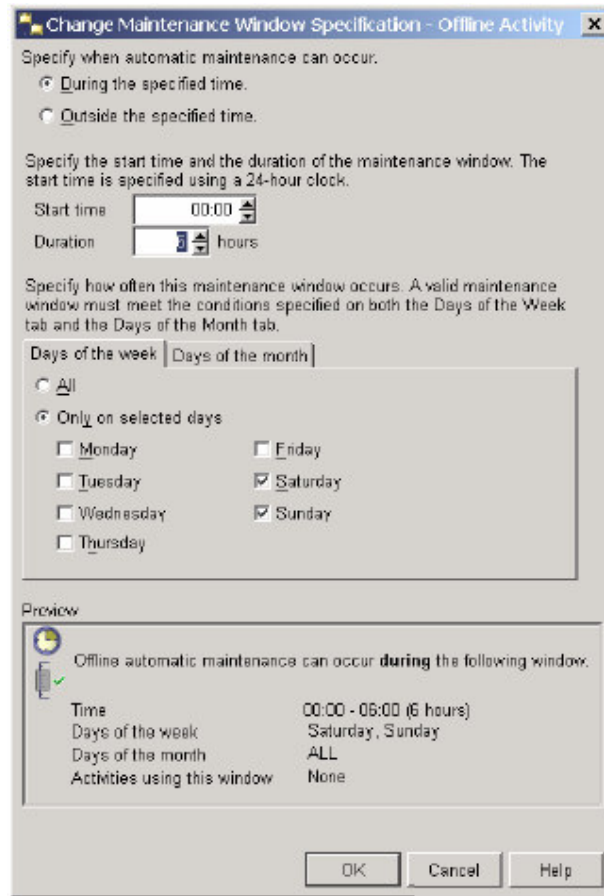
## **Bài tập nhanh #9 – Cấu hình việc bảo trì tự động**

### Mục đích

Trong bài tập nhanh này, với vài thao tác đơn giản, bạn sẽ cấu hình cho việc tự động bảo trì trên cơ sở dữ liệu mẫu DB2.

### Các thủ tục

1. Từ cây đối tượng của Trung tâm điều khiển, nhấn chuột phải vào cơ sở dữ liệu SAMPLE và chọn mục "Configure Automatic Maintenance". Nó sẽ mở ra cửa sổ để thực hiện việc bảo trì tự động theo hướng dẫn.
2. Trang *Introduction* của "hướng dẫn" hiển thị những cài đặt hiện thời của chức năng tự động bảo trì. Nếu bạn tạo ra cơ sở dữ liệu với chức năng tự động bảo trì, khi đó nó sẽ tự động được cấu hình. Bạn có thể dùng "hướng dẫn" này để cấu hình lại chức năng tự động bảo trì. Bấm Next để chuyển tới trang tiếp theo của "hướng dẫn".
3. Trang *Type* của "hướng dẫn" yêu cầu bạn chọn giữa vô hiệu hóa tất cả các tùy chọn tự động lưu giữ và thay đổi cấu hình của việc tự động bảo trì. Chọn một chức năng để thay đổi cài đặt hiện tại của tự động lưu giữ. Bấm Next.
4. Trang *Timing* (thời gian) của "hướng dẫn" sẽ yêu cầu bạn chỉ rõ cửa sổ bảo trì. Cấu hình cho cửa sổ ngoại tuyến sẽ được thực hiện vào mỗi tối thứ 7 và chủ nhật từ giữa đêm tới 6h sáng như hình bên dưới. Bấm nút Change bên cạnh cửa sổ bảo trì ngoại tuyến và chọn thời gian như mong muốn. Sau khi hoàn thành những thông tin cần thiết, bấm nút OK để trở lại "hướng dẫn". Thoát khỏi cửa sổ trực tuyến (bảo trì trực tuyến có thể mở bất kỳ lúc nào). Bấm nút Next.



5. Đến trang *Notification* (cảnh báo) của “hướng dẫn”, bạn có thể cài đặt một sự tương tác trong trường hợp có một hoạt động tự động bảo trì bị lỗi. Bây giờ bỏ qua bước đó. Bấm Next.
6. Đến trang *Actives* của “hướng dẫn”, bạn có thể chọn là có tự động hóa các công việc cá nhân hay không cũng như là được thông báo về các hoạt động cụ thể khác. Trong ví dụ này, phải chắc rằng tất cả các hộp kiểm “Automate” đều được chọn và các hộp kiểm “Notify” đều không được chọn. Bấm Next.
7. Trước khi thực hiện trang kế tiếp của “hướng dẫn”, bạn nên cấu hình vị trí lưu dự phòng của cơ sở dữ liệu. Lý tưởng nhất là bạn nên lưu trữ ở một ổ đĩa vật lý khác trong trường hợp đĩa bị lỗi. Từ trang *Actives*, chọn chức năng “Backup database”, sau đó bấm nút “Configure Settings”.
8. Trên trang *Backup Criteria* của cửa sổ “Configure Settings”, chọn chức năng “Balance Database Recoverability with Performance”. Trên trang *Backup Location*, chọn vị trí lưu dự phòng và bấm nút “Change”. Chỉ ra một vị trí khác để thực hiện lưu dự phòng (phải bảo đảm là đủ bộ nhớ trong ổ đĩa đó). Trên trang *Backup Mode*, bảo đảm là “Offline Backup” đã được chọn. Bấm OK để đóng trang *Backup Criteria*. Bấm nút Next.
9. Trang *Summary* (tóm tắt) của “hướng dẫn” chứa một bảng tóm tắt của những lựa chọn mà bạn đã chọn. Bấm nút Finish để chấp nhận và thi hành những thay đổi đó.

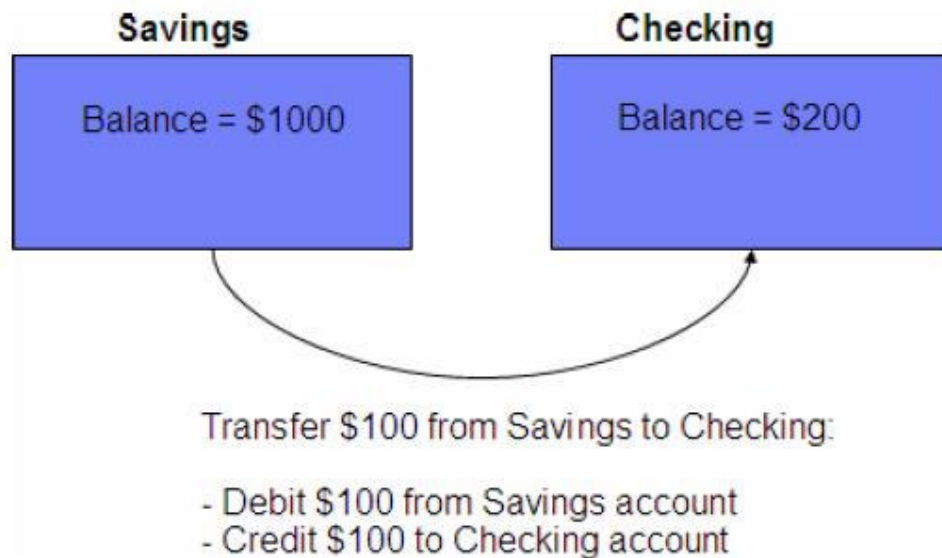
# 13

## Chương 13 – Truy cập đồng thời và khóa

Chương này tập trung thảo luận làm thế nào để cho phép nhiều người dùng truy cập vào cùng một dữ liệu giống nhau tại cùng một thời điểm mà không làm ảnh hưởng lẫn nhau, và giữ cho quá trình hoạt động của họ phù hợp. Chúng ta sẽ thảo luận về khái niệm của các giao dịch, truy cập đồng thời và khóa.

### 13.1 Các giao dịch

Một giao dịch hay một đơn vị công việc gồm có một hay nhiều dòng lệnh SQL mà khi thực hiện cần phải được coi là một đơn vị riêng lẻ. Có nghĩa là nếu một trong những dòng lệnh trong giao dịch thất bại thì toàn bộ giao dịch thất bại, và bất kỳ dòng lệnh nào được thi hành đến điểm thất bại thì sẽ được quay lui (hồi xử), đưa dữ liệu về giá trị, trạng thái trước đó. Một giao dịch kết thúc với dòng lệnh COMMIT, nó cũng có nghĩa là báo hiệu bắt đầu một giao dịch mới. Hình 13.1 cung cấp cho chúng ta ví dụ của một giao dịch



**Hình 13.1 – Một ví dụ về giao dịch**

Trong hình 13.1, ví dụ bạn muốn chuyển 100 đô la từ tài khoản tiết kiệm (savings) của bạn sang tài khoản séc (checking). Việc đó đòi hỏi phải thực hiện công việc này:

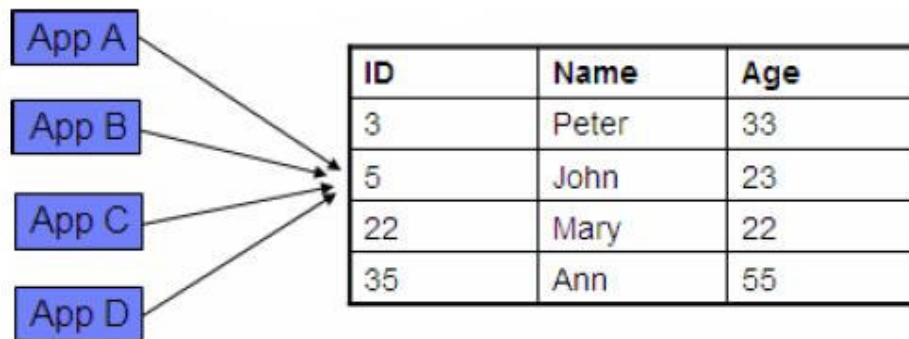
Trừ 100 đô la ở tài khoản tiết kiệm  
 Cộng 100 đô la vào tài khoản séc



Nếu dãy các việc trên không được xem xét như là một đơn vị công việc - một giao dịch, hình dung điều gì sẽ xảy ra nếu có lỗi (xảy ra) sau khi trừ tiền ở tài khoản tiết kiệm, nhưng không cộng được tiền vào tài khoản séc. Bạn sẽ mất 100 đô la!

### 13.2 Truy cập đồng thời

Truy cập đồng thời ý nói rằng nhiều người có thể làm việc ở cùng một thời điểm trên cùng một cơ sở dữ liệu. DB2 được thiết kế như là một cơ sở dữ liệu nhiều người dùng. Việc truy cập dữ liệu phải được kết hợp chính xác và rõ rệt sử dụng một cơ chế để bảo đảm tính toàn vẹn và chắc chắn của dữ liệu. Xem ví dụ 13.2



**Hình 13.2 - Một ví dụ của truy cập đồng thời và sự cần thiết cho việc điều khiển truy cập đồng thời**

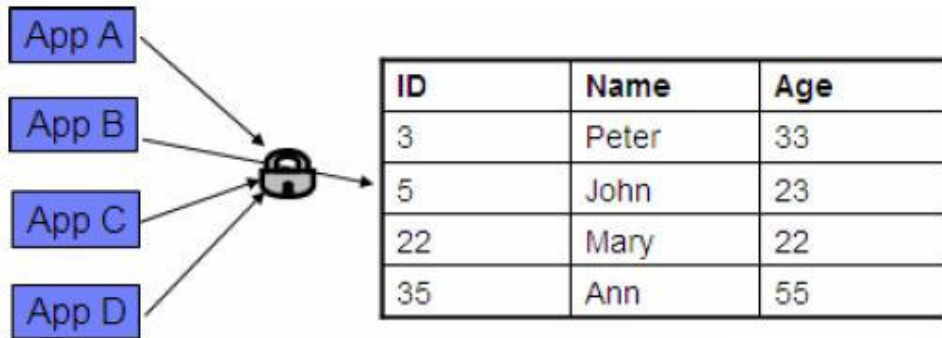
Trong hình 13.2, có 4 ứng dụng, App A, App B, App C và App D đang cố gắng truy cập đến cùng một dòng (dòng 2) trong bảng dữ liệu. Không có điều khiển truy cập đồng thời, tất cả ứng dụng có thể thực hiện các thao tác đối với dòng dữ liệu này. Giả định rằng tất cả ứng dụng đang cập nhật cột Age cho dòng 2 với giá trị khác, ứng dụng thực hiện cập nhật sau cùng sẽ là "người chiến thắng" trong trường hợp này. Hiển nhiên trong ví dụ này một điều khiển truy cập đồng thời nào đó được đòi hỏi để đảm bảo chắc chắn cho kết quả. Điều khiển truy cập đồng thời này dựa vào việc sử dụng khóa bản ghi.

Khái niệm khóa và truy cập đồng thời có liên quan chặt chẽ với nhau. Khóa dùng để tạm thời ngăn những ứng dụng thao tác trên chúng cho đến khi một thao tác kết thúc. Nếu chúng ta có nhiều khóa hơn trong một hệ thống thì việc truy cập đồng thời sẽ ít đi. Nói một cách khác, nếu chúng ta có ít khóa hơn trong hệ thống thì sẽ có nhiều truy cập đồng thời hơn.

Những khóa này sẽ tự động xuất hiện để hỗ trợ một thao tác và sẽ được giải phóng khi thao tác kết thúc (khi sử dụng lệnh COMMIT hoặc ROLLBACK). Các khóa có thể đặt trên các bảng hoặc các dòng. Có hai loại khóa cơ bản:

- Khóa chia sẻ (khóa S) - được sử dụng khi ứng dụng muốn đọc và ngăn cản cập nhật từ những dòng khác
- Khóa dành riêng (khóa X) – được sử dụng khi một ứng dụng cập nhật, thêm hay xóa một dòng

Bây giờ chúng ta để ý đến hình 13.3, nó có vẻ giống như hình 13.2 nhưng bây giờ nó xuất hiện thêm một khóa



Hình 13.3 – Một ví dụ của truy cập đồng thời, và sự cần thiết của khóa

Trong ví dụ hình 13.3, nếu App B truy cập dòng thứ 2 đầu tiên, và thực hiện việc UPDATE, App B giữ một khóa dành riêng (khóa X) trên dòng đó. Khi App A, App C và App D cố gắng truy cập vào dòng 2, chúng sẽ không thể UPDATE dòng này bởi vì khóa X. Việc kiểm tra này giúp bảo đảm tính toàn vẹn và chắc chắn của dữ liệu.

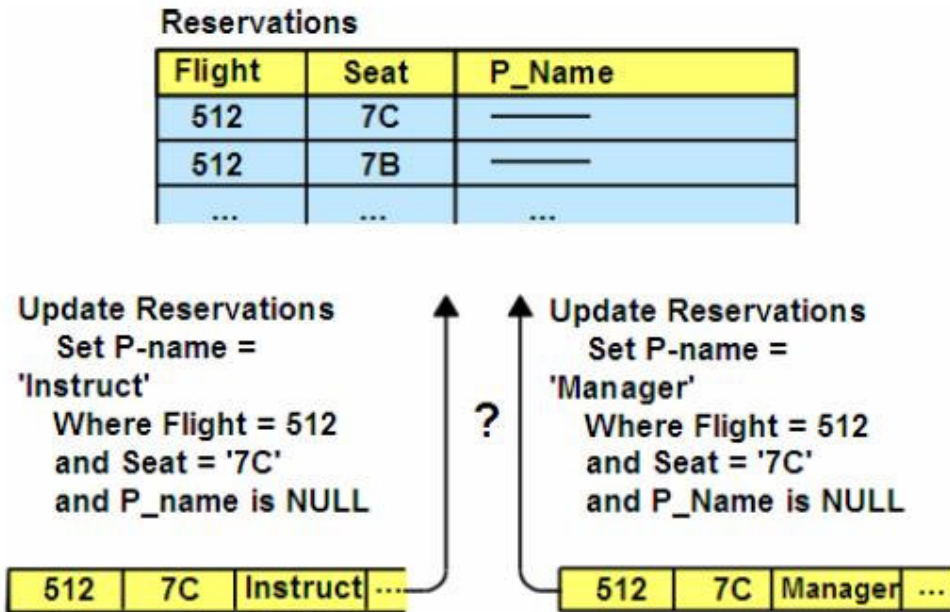
**13.3 Các vấn đề khi không kiểm tra truy cập đồng thời**

Khi không có điều khiển truy cập đồng thời, chúng ta có thể gặp những vấn đề dưới đây

- Mất dữ liệu cập nhật
- Đọc với không cam kết
- Đọc không lặp lại
- Đọc thừa

**13.3.1 Mất dữ liệu cập nhật**

Mất dữ liệu cập nhật là một vấn đề tương tự như giải thích trước trong mục này, khi ứng dụng thực hiện cập nhật sau cùng sẽ là “người chiến thắng”.



Hình 13.4 – Mất dữ liệu cập nhật

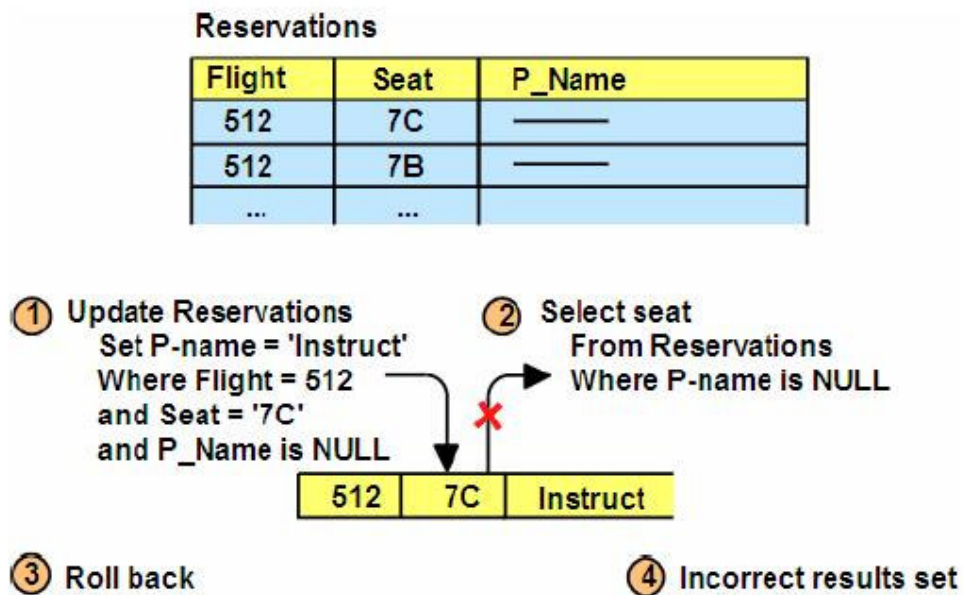
Trong hình 13.4 có hai ứng dụng cố gắng cập nhật trên cùng một dòng. Bên trái là App1, và bên phải là App2. Và chuỗi việc sau đó là:

1. App1 cập nhật một dòng
2. App2 cập nhật dòng tương tự
3. App1 hoàn tất
4. App2 hoàn tất

Dữ liệu cập nhật của App1 sẽ bị mất khi App2 cập nhật lại nó, do đó xảy ra việc “Mất dữ liệu”

**13.3.2 Đọc với không cam kết**

Một hành động đọc không cam kết hoặc “**đọc có ý xấu**” đề cập đến việc một ứng dụng đọc những thông tin mà không **được cam kết**, và tất nhiên việc này thì không được đúng đắn.



Hình 13.5 – Đọc với không cam kết

Hình 13.5 kéo theo những sự việc này:

1. App1 cập nhật một dòng
2. App2 đọc giá trị mới của dòng đó
3. App1 cuộn lại những thay đổi trên dòng đó

App2 thì đang đọc với dữ liệu không được cam kết, và kể từ đó dữ liệu không hợp lệ và được gọi là “đọc với không cam kết”

**13.3.3 Đọc không lặp lại**

Đọc không lặp lại ngụ ý rằng bạn không đạt được kết quả giống nhau sau khi thực hiện việc đọc giống nhau với những thao tác giống nhau.

FLIGHT	SEAT	NAME	DESTINATION	ORIGIN
512	7B	---	DENVER	DALLAS
....				
....				
814	8A	---	SAN JOSE	DENVER
....				
134	1C	---	HONOLULU	SAN JOSE
....				....

Hình 13.6 – Đọc không lặp lại

Trong hình 13.6, dường như nếu bạn cố gắng ghi một chuyến bay từ Dallas đến Honolulu. Một dãy các sự việc là:

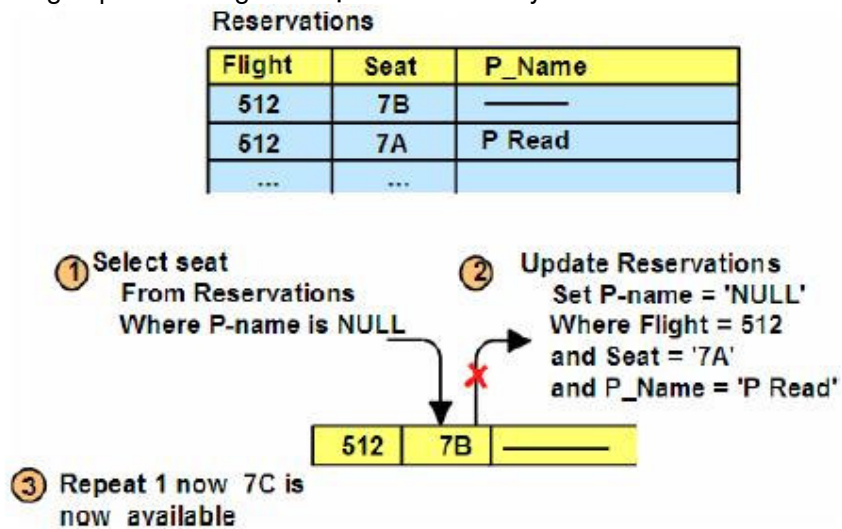
1. App1 mở một bảng (được hiểu như bảng kết quả) hiện hành như bạn thấy trong hình 13.6
2. App2 xóa một dòng trong bảng (ví dụ, dòng với destination “San Jose”)
3. App2 hoàn tất những thay đổi
4. App1 đóng và mở lại bảng

Trong trường hợp này, từ lúc App1 không được cung cấp dữ liệu tương tự đang đọc lặp lại, nó không thể sao chép lại bảng dữ liệu, đó chính là vấn đề được gọi là “đọc không lặp lại”

**13.3.4 Đọc thừa**

Vấn đề đọc không có thực giống như vấn đề đọc không lặp lại, nhưng khác nhau là ở những lộ trình kế tiếp, bạn có thể thu được những dòng vừa thêm vào mà đúng ra thì chúng (dòng) ít hơn.

Hình 13.7 cung cấp cho chúng ta ví dụ của vấn đề này



Hình 13.7 - Đọc thừa

Hình 13.7 trình bày chuỗi sự kiện:

1. App1 mở một bảng
2. App2 thêm một dòng vào cơ sở dữ liệu ở nơi mà nó có đủ tiêu chuẩn của bảng
3. App2 hoàn tất những thay đổi
4. App1 đóng và lại tiếp tục mở bảng

Trong trường hợp này, App1 sẽ không thấy dữ liệu tương tự như đang đọc lặp lại, nó sẽ lấy được nhiều dòng hơn. Vấn đề này được gọi là đọc thừa

### 13.4 Các mức cô lập

Bạn có thể nghĩ các mức cô lập như là nơi giữ các chính sách khóa, dựa vào sự lựa chọn mức cô lập, bạn có thể có những tác động khác nhau cho cơ sở dữ liệu khóa một ứng dụng

DB2 cung cấp những mức độ bảo vệ khác nhau để cô lập dữ liệu:

- Đọc với không cam kết (UR)
- Ổn định con trỏ (CS)
- Đọc ổn định (RS)
- Đọc lặp lại (RR)

#### 13.4.1 Đọc với không cam kết

Đọc với không cam kết cũng được hiểu như đọc có ý xấu. Đó là mức độ cô lập thấp nhất, và cung cấp mức độ truy cập đồng thời cao nhất. Không có sự khóa dòng nào tồn tại trên những thao tác đọc, nếu các ứng dụng khác không xâm nhập để xóa hoặc sửa một bảng, và thao tác cập nhật giữ nhiệm vụ như là nếu chúng ta dùng cấp độ cô lập ổn định con trỏ.

Những vấn đề vẫn tồn tại với cấp độ cô lập này:

- Đọc với không cam kết
- Đọc không lặp lại
- Đọc thừa

Những vấn đề được ngăn ngừa với cấp độ cô lập này là:

- Mất dữ liệu khi cập nhật

#### 13.4.2 Ổn định con trỏ

Ổn định con trỏ là cấp độ cô lập mặc định. Nó cung cấp một mức độ khóa tối thiểu. Về cơ bản, với mức cô lập này thì dòng "hiện hành" của con trỏ (cursor) bị khóa. Nếu đó là dòng chỉ đọc, thì khóa sẽ được giữ cho đến khi một dòng mới được nạp hoặc đơn vị làm việc được kết thúc. Nếu dòng được cập nhật, chiếc khóa sẽ được giữ cho đến khi đơn vị làm việc được kết thúc

Những vấn đề vẫn tồn tại với mức độ cô lập này:

- Đọc không lặp lại
- Đọc thừa

Những vấn đề được ngăn ngừa với mức độ cô lập này

- Mất dữ liệu cập nhật
- Đọc không cam kết

### 13.4.3 Đọc ổn định

Với đọc ổn định, tất cả các dòng mà ứng dụng làm việc trong một phạm vi đều bị khóa. Cho một con trỏ, nó khóa tất cả các dòng đủ tiêu chuẩn trong tập hợp kết quả. Ví dụ nếu bạn có một bảng chứa 10.000 dòng và câu truy vấn trả về 10 dòng, sau đó chỉ có 10 dòng đó bị khóa. Cơ chế đọc ổn định sử dụng một mức độ khóa vừa phải.

Vấn đề vẫn tồn tại với mức độ cô lập này:

- Đọc thừa

Những vấn đề được hạn chế với mức độ cô lập này là:

- Mất dữ liệu cập nhật
- Đọc với không cam kết
- Đọc không lặp lại

### 13.4.4 Đọc lặp lại

Cơ chế đọc lặp lại là mức độ cô lập cao nhất. Nó cung cấp cấp độ khóa cao nhất, và truy cập đồng thời tối thiểu nhất. Khóa được giữ trên tất cả các dòng, xử lý việc xây dựng một tập hợp kết quả. Tức là những dòng không cần thiết trong tập hợp kết quả cuối cùng có thể bị khóa. Không một chương trình nào khác có thể cập nhật, xóa, hoặc thêm mới một dòng mà có thể ảnh hưởng đến tập hợp kết quả cho đến khi một đơn vị công việc được hoàn tất. Cơ chế đọc lặp lại bảo đảm rằng những câu truy vấn tương tự nhau được đưa ra bởi một ứng dụng trong một đơn vị làm việc sẽ nhận được kết quả giống nhau tại mỗi thời điểm.

Vấn đề vẫn còn tồn tại với mức độ cô lập này:

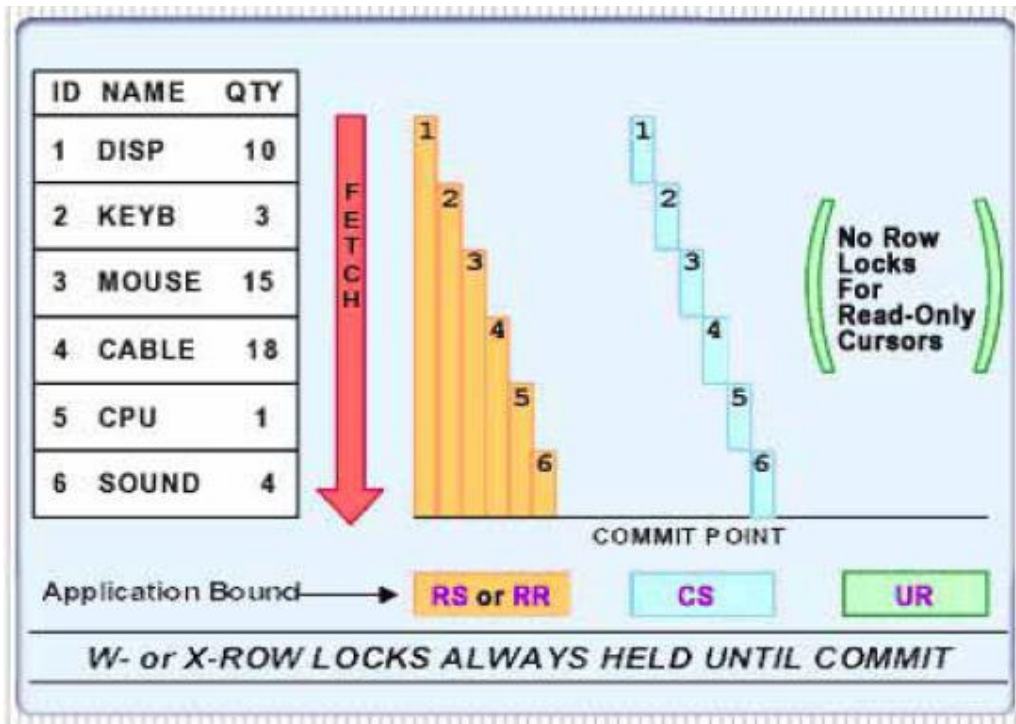
- không có

Những vấn đề được ngăn ngừa với mức độ cô lập này:

- Mất dữ liệu khi cập nhật
- Đọc không cam kết
- Đọc không lặp lại
- Đọc thừa

### 13.4.5 So sánh các mức độ cô lập dữ liệu

Hình 13.8 so sánh sự khác nhau mà các mức độ cô lập mang lại. Trong hình, chúng ta nhìn thấy mức độ cô lập “đọc không cam kết” không khóa gì cả. Mức độ cô lập ổn định con trỏ giữ một cái khóa dòng 1 khi nó đang lấy dòng đó về, nhưng sẽ giải phóng nó sớm rồi nó lấy dòng 2 về,... Đối với mức độ cô lập đọc ổn định hoặc đọc lặp lại, bất cứ dòng nào lấy về sẽ bị khóa, và khóa sẽ không được giải phóng cho đến khi kết thúc một giao tác (khi hoàn tất)



Hình 13.8 – So sánh các mức độ cô lập

**13.4.6 Đặt mức cô lập**

Những mức cô lập có thể được chỉ rõ tại nhiều mức :

- Phiên làm việc trong ứng dụng
- Kết nối
- Câu lệnh

Những mức cô lập bình thường được xác định tại phiên hoặc tại mức ứng dụng. Nếu không chỉ rõ được mức cô lập trong ứng dụng của các bạn, nó mặc định tới con trỏ ổn định. Chẳng hạn, bảng 13.1 cho thấy những mức cô lập khả dĩ cho một chương trình .NET hay JDBC và làm sao khi đặt những thuộc tính này phải phù hợp với một mức cô lập của DB2:

DB2	.NET	JDBC
Uncommitted Read (UR)	ReadUncommitted	TRANSACTION_READ_UNCOMMITTED
Cursor Stability (CS)	ReadCommitted	TRANSACTION_READ_COMMITTED
Read Stability (RS)	RepeatableRead	TRANSACTION_REPEATABLE_READ
Repeatable Read (RR)	Serializable	TRANSACTION_SERIALIZABLE

**Table 13.1 – Sự so sánh của thuật ngữ đặt mức cô lập**

Mức cô lập được phát biểu có thể dùng với mệnh đề WITH {mức cô lập}. Chẳng hạn như:

```
SELECT ... WITH {UR | CS | RS | RR}
```



Một ứng dụng cần xác định “sơ bộ” số hàng trong một bảng. Hiệu suất rất là quan trọng. Con trỏ được yêu cầu mức cô lập ổn định với ngoại lệ của một câu lệnh SQL:

```
SELECT COUNT(*) FROM tab1 WITH UR
```

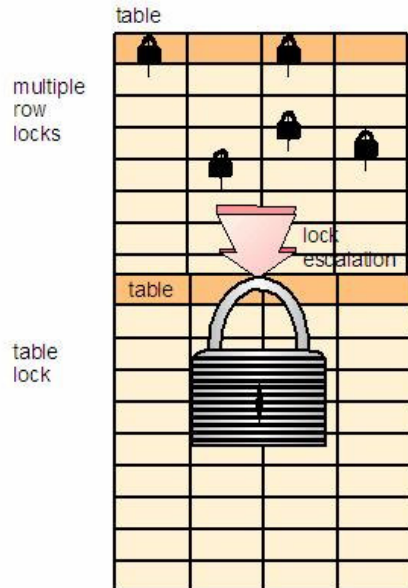
Với SQL nhúng, các mức thiết lập được đặt tại thời điểm đóng gói, còn với SQL động các mức được đặt ở thời gian chạy.

Việc chọn sử dụng mức cô lập nào phụ thuộc vào ứng dụng của các bạn. Nếu ứng dụng của các bạn không yêu cầu có những giá trị như ở ví dụ, chọn cô lập UR. Nếu ứng dụng của các bạn yêu cầu điều khiển rất chặt trên dữ liệu làm việc với nó, chọn cô lập RR.

### 13.5 Khóa leo thang

Mỗi khóa do DB2 tạo ra sẽ sử dụng lượng bộ nhớ nào đó. Khi bộ tối ưu cho rằng nó tốt hơn là để một khóa trên toàn bộ bảng, thay vì khóa nhiều hàng, sự leo thang khóa xuất hiện.

Hình 13.9 minh họa cho điều này:



Hình 13.9 – Khóa leo thang

Có hai tham số cấu hình cơ sở dữ liệu chính liên quan đến khóa leo thang:

- **LOCKIST** - Số lượng của bộ nhớ (trong trang 4k) dự trữ để quản lý những khóa cho mọi ứng dụng được nối. Sự mặc định là năm mươi lần trang 4K (200K) trên Windows.
- **MAXLOCKS** - Phần trăm cực đại của toàn bộ danh sách khóa cho một ứng dụng. Mặc định là 22%.

Bởi vậy, nếu những giá trị ngầm định được sử dụng, khóa leo thang xuất hiện khi một ứng dụng đơn yêu cầu hơn 44K trong tổng bộ nhớ khóa (200K \* 22% = 44K). Nếu sự leo thang khóa xuất hiện thường xuyên với sự thiết lập này, tăng giá trị của LOCKIST và MAXLOCKS. Khóa leo thang không tốt cho hiệu năng nhưng lại giảm sự xung đột. Tập nhật ký triệu chứng của DB2 có thể được dùng để xác định liệu có



phải sự leo thang khóa đang xuất hiện. (db2diag.log, thường tệp này nằm trong thư mục: C:\Program Files\IBM\SQLLIB\DB2)

### 13.6 Kiểm soát khóa

Bạn có thể theo dõi những sự sử dụng khóa bằng ảnh chụp nhanh khóa ứng dụng DB2. Để bật những ảnh chụp nhanh cho khóa, ta dùng lệnh:

```
UPDATE MONITOR SWITCHES USING LOCK ON
```

Sau khi được cho phép, thông tin theo dõi sẽ được tập hợp. Để thu được một báo cáo của khóa vào một thời gian nhất định, dùng lệnh sau:

```
GET SNAPSHOT FOR LOCKS FOR APPLICATION AGENT ID <handle>
```

Hình 13.9: Ảnh chụp nhanh kết quả của một ví dụ sử dụng ứng dụng khóa

```

Application Lock Snapshot

Snapshot timestamp           = 11-
05-2002 00:09:08.672586

Application handle           = 9
Application ID               =
*LOCAL.DB2.00B9C5050843
Sequence number              = 0001
Application name              =
db2bp.exe
Authorization ID             =
ADMINISTRATOR
Application status           = UOW
Waiting
Status change time          = Not
Collected
Application code page        = 1252
Locks held                    = 4
Total wait time (ms)         = 0

List Of Locks
Lock Name                    =
0x050007000480010000000000052
Lock Attributes               = 0x00000000
Release Flags                 = 0x40000000
Lock Count                    = 255
Hold Count                    = 0
Lock Object Name              = 98308
Object Type                   = Row
Tablespace Name               = TEST4K
Table Schema                  = ADMINISTRATOR
Table Name                    = T2
Mode                          = X
    
```

Figure 13.9 – Ảnh chụp nhanh khóa của các ứng dụng

### 13.7 Chờ khóa

Khi hai hoặc nhiều các ứng dụng cần thực hiện một phép toán trên cùng một đối tượng, một trong số đó có thể phải đợi để nhận được khóa. Theo mặc định một ứng dụng sẽ

đợi vô hạn. Thời gian chờ đợi một ứng dụng của một khóa được kiểm soát bởi tham số cấu hình cơ sở dữ liệu. Giá trị ngầm định của tham số này là -1 (đợi vô hạn).

Thanh ghi CURRENT LOCK TIMEOUT cần được dùng để đặt thời gian chờ khóa cho một kết nối đã cho. Theo mặc định, thanh ghi này được đặt giá trị của LOCKTIMEOUT. Sử dụng SET LOCK TIMEOUT để thay đổi giá trị của nó. Một khi giá trị của thanh ghi đã được đặt cho một kết nối, nó sẽ tồn tại qua mọi giao dịch.

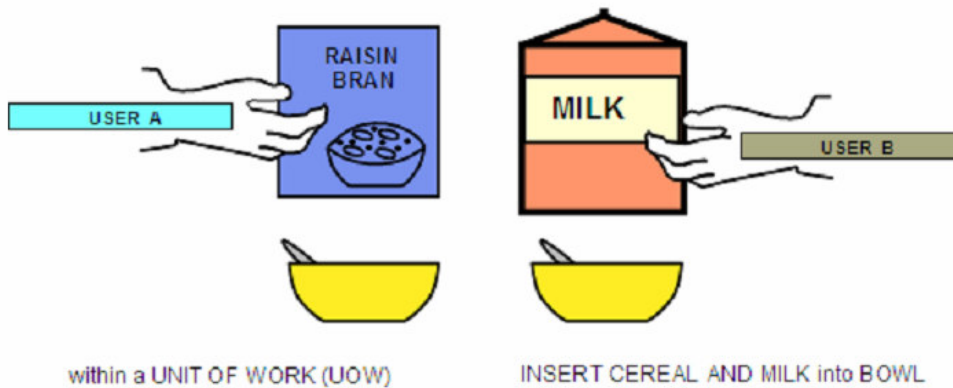
Ví dụ:

```
SET LOCK TIMEOUT=WAIT n
```

### 13.8 Vấn đề khóa phụ thuộc và phát hiện

Một sự bế tắc (deadlock) xuất hiện khi hai hoặc nhiều các ứng dụng được nối tới cùng cơ sở dữ liệu. Sự chờ đợi chưa bao giờ được giải quyết bởi vì mỗi ứng dụng đang giữ một tài nguyên mà nó phục vụ cho những nhu cầu khác nhau. Trong thiết kế ứng dụng, những sự bế tắc luôn là một vấn đề chiếm nhiều thời gian.

Hình 13.10: Minh họa một tình huống bế tắc.



**Hình 13.10 - Kịch bản sự bế tắc**

Trong hình 13.10, người A giữ được bột nho khô và giả sử sẽ không cho phép người khác dùng cho đến khi ông ta có sữa. Mặt khác, người B giữ được sữa, và sẽ không cho phép người khác cho đến khi ông ta có bột nho khô. Bởi vậy, chúng ta có một tình huống bế tắc.

Để mô phỏng một tình huống bế tắc ở DB2, theo những bước sau đây:

1. Mở hai cửa sổ soạn thảo lệnh của DB2 (chúng tôi sẽ gọi là "CLP1" và "CLP2" tương ứng). Chúng đại diện cho hai ứng dụng khác nhau kết nối tới cơ sở dữ liệu.
2. Từ CLP1 thực hiện các lệnh sau:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Mary' where empno = '000050'
```

Đầu tiên, chúng ta đang kết nối tới cơ sở dữ liệu có tên SAMPLE, và sau đó thực hiện một lệnh cập nhật hàng trên bảng *employee* làm với "empno= 50000". Lựa

chọn "+c" trong câu lệnh chỉ ra rằng, chúng tôi không muốn cửa sổ lệnh của DB2 tự động cam kết lệnh này. Chúng ta đang cố ý làm điều này sao cho chúng ta giữ được khóa.

3. Từ CLP2 thực hiện những lệnh sau:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Tom' where empno =
'000030'
```

Ở cửa sổ của CLP2, nó đại diện cho ứng dụng thứ hai, chúng ta cũng nối vào cơ sở dữ liệu SAMPLE, nhưng lại cập nhật một hàng khác ở bảng *employee*.

4. Lệnh từ CLP1:

```
db2 +c select firstnme from employee where empno = '000030'
```

Sau khi nhấn Enter thực hiện câu lệnh SELECT ở trên, câu lệnh SELECT này có vẻ sẽ treo. Nó thật sự là không phải là treo, mà là đợi khóa của hàng này trả về do CLP2 đã giữ trong bước 3. Tại điểm này, nếu CLOKETIMEOUT đã được để lại với giá trị ngầm định của nó là -1, ứng dụng CLP1 đợi mãi mãi.

5. Lệnh từ CLP2:

```
db2 +c select firstnme from employee where empno = '000050'
```

Bằng việc đánh lệnh SELECT ở trên, bây giờ chúng ta đang tạo ra một sự bế tắc. Phát biểu SELECT này cũng có vẻ treo, thực ra nó đang đợi khóa CLP1 đang giữ trả về.

Trong kịch bản bế tắc trên, DB2 sẽ kiểm tra tham số cấu hình cơ sở dữ liệu DLCHKTIME. Tham số này sẽ đặt khoảng thời gian kiểm tra cho những sự bế tắc. Chẳng hạn, giá trị của tham số này được đặt tới 10 giây, DB2 sẽ sử dụng một giải thuật bên trong để xác định giao dịch nào cần phải quay lui (hồi xử), và một giao dịch nào sẽ được tiếp tục.

Nếu bạn đang gặp nhiều tình huống bế tắc, bạn cần phải tái kiểm tra những giao dịch hiện hữu xem xem có thể tổ chức lại được không.

### 13.9 Truy cập đồng thời và thực tiễn khóa tốt nhất

Sau đây là một số mẹo nhỏ để truy cập đồng thời và thực tiễn khóa tốt nhất:

1. Giữ những giao dịch càng ngắn càng tốt. Điều này có thể đạt được bằng cách phát hành thường xuyên những lệnh COMMIT (thậm chí cho những giao dịch chỉ đọc) khi ứng dụng của bạn cho phép.
2. Chỉ ghi nhật ký thông tin giao dịch khi có yêu cầu.
3. Tẩy dữ liệu đã sử dụng nhanh chóng:

```
ALTER TABLE ACTIVATE NOT LOGGED INITIALLY WITH EMPTY TABLE
```

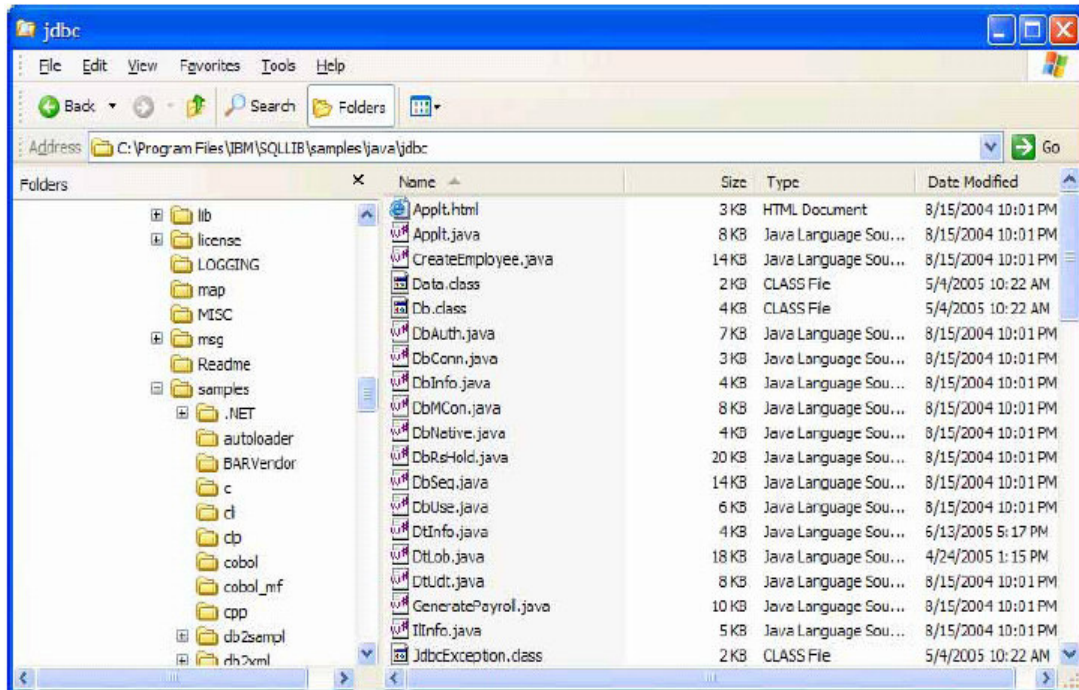
4. Thực hiện những thay đổi dữ liệu theo lô/nhóm. Chẳng hạn

```
DELETE FROM (
  SELECT *FROM tedwas.t1 WHERE c1 = ... FETCH FIRST s3000 ROWS
  ONLY)
```

5. Sử dụng những đặc tính tương tranh của những công cụ chuyển đổi dữ liệu trong DB2.
6. Đặt tham số ở mức LOCKTIMEOUT cho cơ sở dữ liệu (khoảng giữa 30-120 giây). Bạn đừng nên cho giá trị ngầm định của nó là -1. Bạn có thể cũng sử dụng khóa giới hạn thời gian (timeout) trên trên từng phiên làm việc.
7. Không khôi phục nhiều dữ liệu hơn cần thiết. Chẳng hạn, sử dụng mệnh đề FETCH FIRST n ROWS ONLY trong câu lệnh SELECT.

## PHẦN 3: HỌC DB2 – PHÁT TRIỂN ỨNG DỤNG

Trong phần này, chúng ta sẽ thảo luận về các đối tượng của cơ sở dữ liệu nằm ở tầng dưới của ứng dụng như thủ tục lưu (stored procedure), hàm do người dùng định nghĩa và bất sự kiện (trigger). Rằng bạn có thể học được cách lập trình sử dụng DB2 như một máy chủ dữ liệu bằng nhiều ngôn ngữ khác nhau bằng cách khảo sát các ứng dụng mẫu đi kèm trong thư mục SQLLIB\samples khi chúng ta cài đặt máy chủ DB2. Hình dưới đây là một số chương trình mẫu viết bằng Java, sử dụng DB2 trên nền Windows.



Một số chương trình Java mẫu đi kèm DB2.

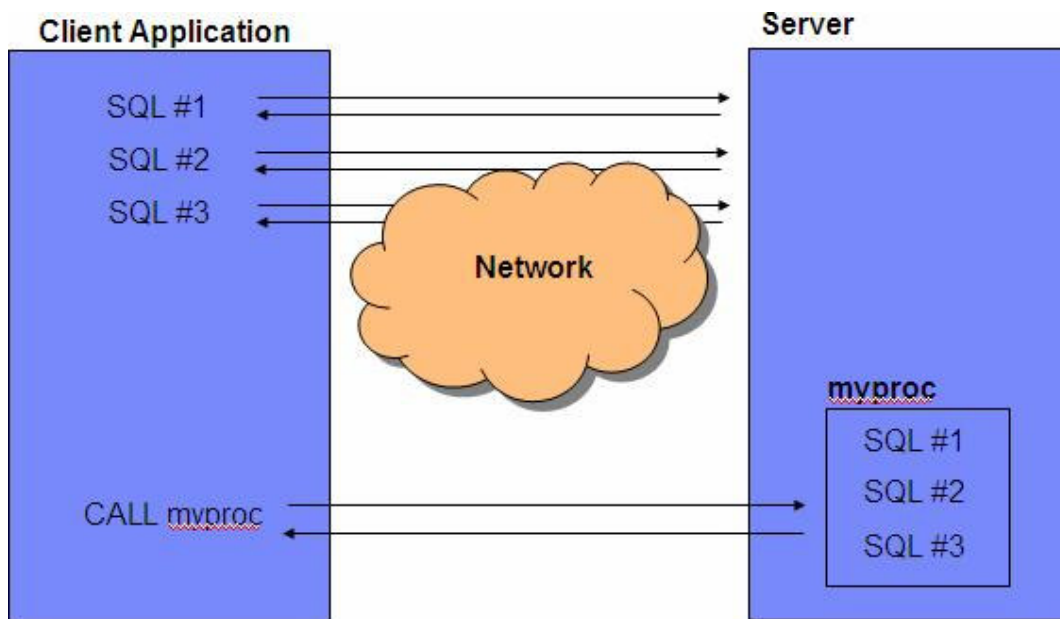
# 14

## Chương 14 – Các thủ tục SQL PL

Trong chương này chúng ta sẽ thảo luận về “thủ tục lưu” (stored procedure). Thủ tục lưu là một đối tượng ứng dụng cơ sở dữ liệu, có thể đóng gói các câu lệnh SQL và các nghiệp vụ. Lưu trữ hợp lí một phần ứng dụng trong cơ sở dữ liệu cho phép cải tiến hiệu suất thực hiện như giảm khả năng kẹt đường truyền mạng. Hơn nữa, các thủ tục cho phép lưu trữ tập trung tại một nơi nào đó để các ứng dụng khác cũng có thể sử dụng được.

Thủ tục lưu trong DB2 có thể viết bằng SQL PL, C/C++, Java, Cobol, các ngôn ngữ được hỗ trợ CLR (Common Language Runtime) và OLE. Trong chương này, chúng ta sẽ tập trung vào SQL PL vì tính phổ biến và đơn giản của nó.

Hình 14.1 mô tả cách hoạt động của stored procedure.



Hình 14.1 - Giảm kẹt đường truyền mạng bằng thủ tục.

Ở góc trên bên trái của hình, ta thấy một vài câu lệnh SQL được thực thi lần lượt hết cái này đến cái kia. Mỗi câu SQL được gửi từ máy khách đến máy chủ, và máy chủ trả kết quả về lại máy khách. Nếu nhiều câu SQL được thực thi theo kiểu đó thì lượng dữ liệu truyền trên mạng sẽ tăng lên. Tuy nhiên, ở góc dưới phải, ta thấy thủ tục lưu “myproc” được lưu trên máy chủ chứa những câu SQL tương tự trên; và sau đó tại máy khách (bên phía trái), lệnh CALL được dùng để gọi thực hiện thủ tục. Cách thứ hai gọi thực

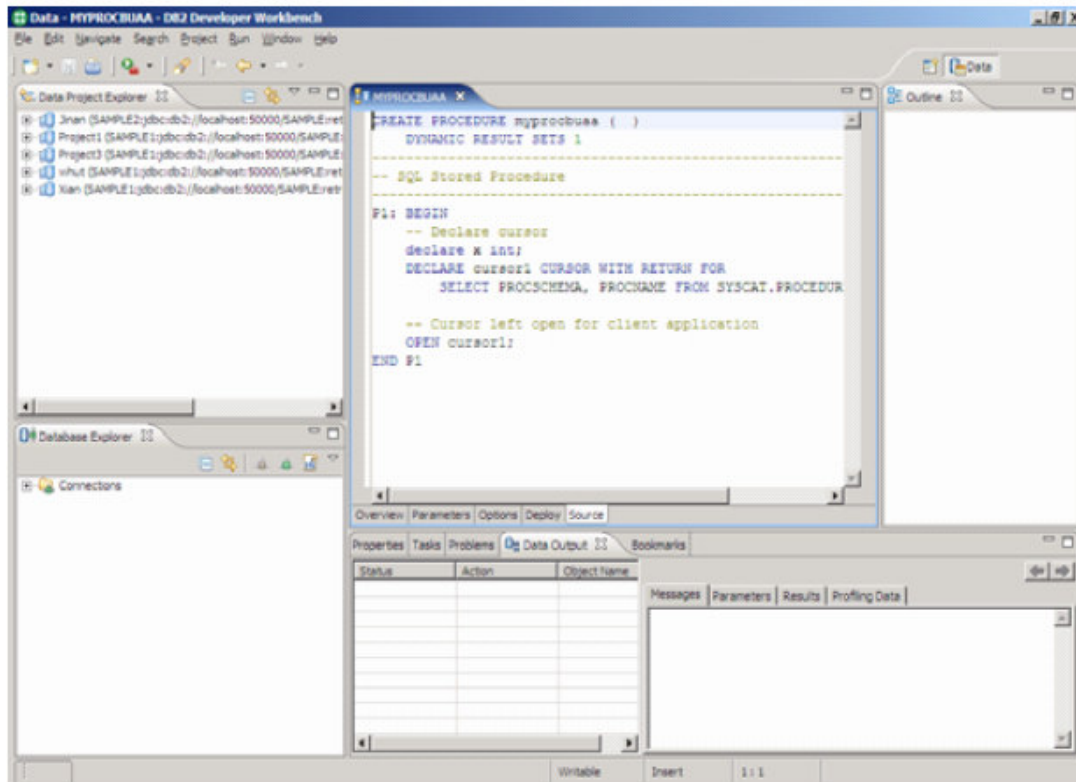
hiện một thủ tục hiệu quả hơn vì chỉ có một lệnh gọi được truyền trên mạng và một kết quả trả về cho máy khách.

Thủ tục lưu cũng rất hữu ích trong mục đích bảo mật cơ sở dữ liệu vì người sử dụng sẽ được phân cấp các quyền để thực thi hoặc chỉ được xem các thủ tục; điều đó giúp cho hệ thống được che chắn và không cho người dùng lấy thông tin nếu họ không có quyền. Khả năng này là hoàn toàn khả thi vì người dùng không đòi hỏi những quyền tương minh trên các bảng hoặc khung nhìn (view) trong thủ tục mà họ thực thi; họ chỉ cần được cấp đủ quyền để thực hiện các thủ tục đó.

### 14.1 DB2 Developer Workbench

DB2 Developer Workbench (DWB) là công cụ được xây dựng trên nền tảng Eclipse, phục vụ cho việc phát triển các thủ tục, hàm, truy vấn XML, ứng dụng SQLJ,... DWB đến như một hình ảnh riêng rẽ (nó không phải là một phần trong gói cài đặt DB2) nhưng hoàn toàn miễn phí. Có thể tải DWB về từ trang “Download” trên địa chỉ <http://ibm.com/db2/express>.

Hình 14.2 Giao diện của DWB.



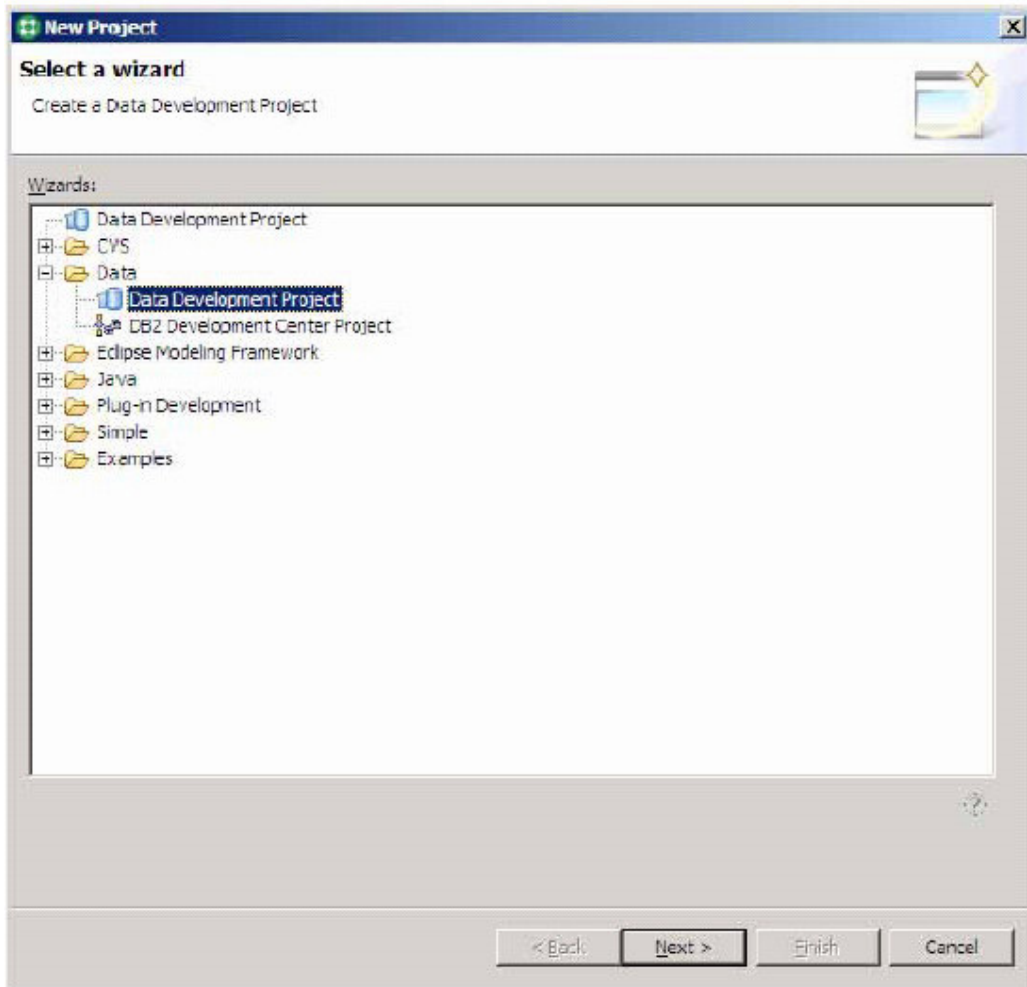
Hình 14.2 – DB2 Developer Workbench

#### 14.1.2 Tạo một thủ tục lưu trong DWB

Để tạo một thủ tục bằng Java hay SQL PL, hãy làm theo các bước sau. Lưu ý rằng không thể tạo được các thủ tục được viết bằng các ngôn ngữ khác trong DWB.

##### Bước 1: Tạo một dự án DWB (project)

Từ menu, chọn *File -> New -> Project* và chọn *Data Development Project*. Xem hình 14.3 dưới đây.



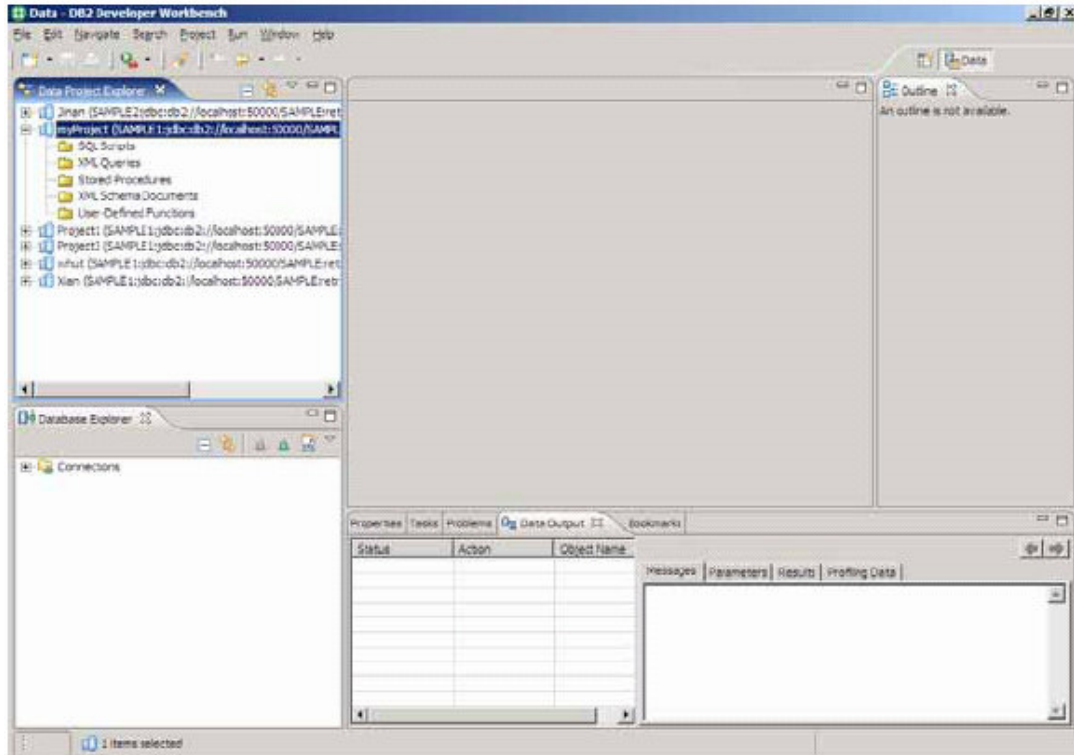
**Hình 14.3 – Data development project**

Từng bước làm theo hướng dẫn để nhập tên dự án, chọn cơ sở dữ liệu bạn muốn sử dụng và xác định rõ thư mục JDK (cái nào mặc định thường là đúng).

**Bước 2: Tạo một thủ tục**

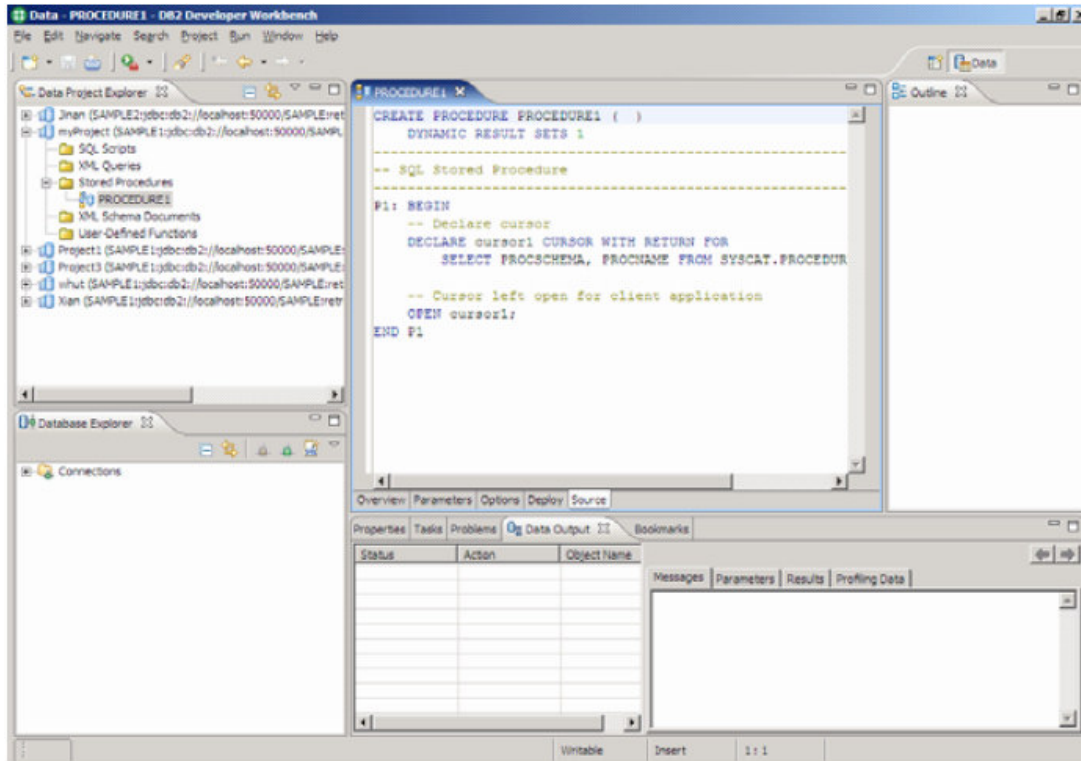
Sau khi tạo xong dự án, phía bên phải của cửa sổ (data perspective) sẽ xuất hiện dự án mới vừa tạo. Trong hình 14.4 bạn sẽ thấy dự án mới tạo trong tình trạng mở rộng.





**Hình 14.4 – Dự án “myProject”**

Hình 14.4 Cho thấy các thư mục khác nhau trong dự án của bạn. Khi muốn tạo một thủ tục, bạn nhấn chuột phải vào thư mục thủ tục và chọn *New -> Stored Procedure*. Hoàn thành các thông tin được yêu cầu trong từng bước tạo mới một thủ tục như: thủ tục đó kết hợp với dự án nào, tên và ngôn ngữ để viết thủ tục (lưu ý rằng chỉ có SQL PL và Java được hỗ trợ trong DWB) và các câu lệnh SQL dùng trong thủ tục. Mặc định, DWB sẽ cung cấp cho bạn câu lệnh SQL mẫu ví dụ. Tới đây bạn có thể nhấn Finish và thủ tục sẽ được tạo ra và trong đó có sử dụng đoạn mã mẫu và các câu lệnh SQL trong ví dụ được cung cấp ở trên. Xem hình 14.5

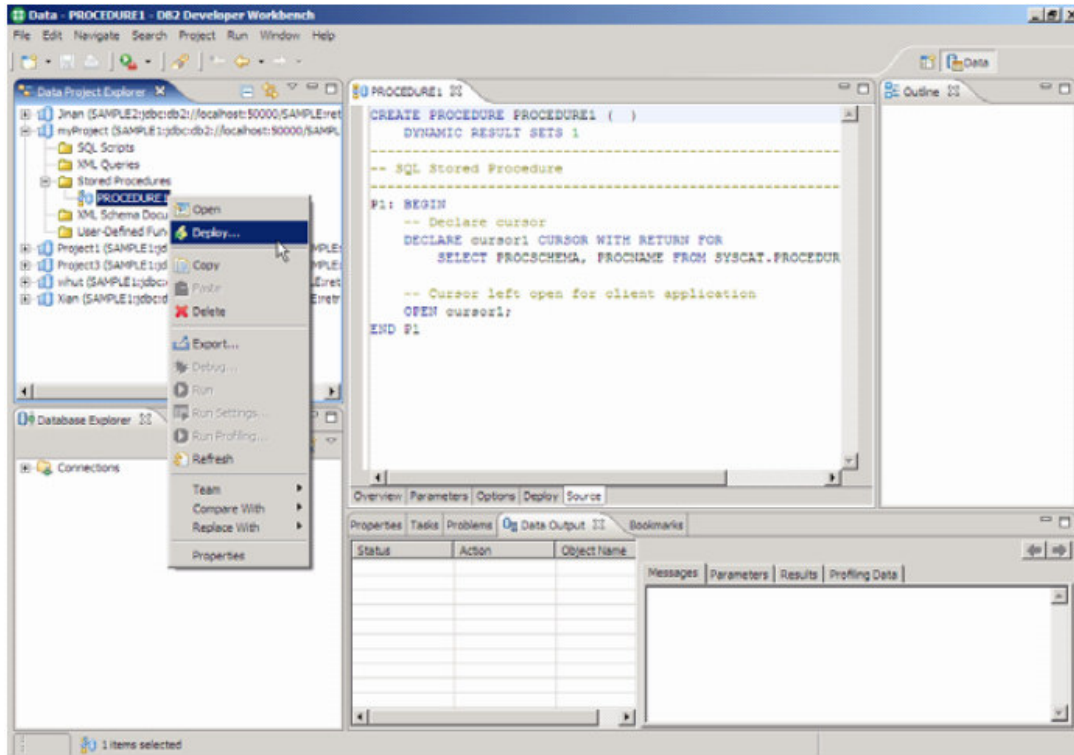


**Hình 14.5 – Một thủ tục mẫu**

Trong hình 14.5, đoạn mã mẫu cho thủ tục “PROCEDURE1” được phát sinh. Bạn có thể thay thế đoạn mã đó bằng đoạn mã riêng của bạn. Để đơn giản chúng ta sẽ dùng các đoạn mã mẫu trên để tiếp tục khảo sát quyển sách này.

**Bước 3: Biên dịch một thủ tục**

Khi một thủ tục đã được tạo, bạn biên dịch và triển khai bằng cách nhấp phải vào nó trong khung bên trái và chọn Deploy. Hình 14.6 minh họa cho bước này.



Hình 14.6 – Biên dịch một thủ tục

**Bước 4: Thực thi một thủ tục**

Khi một thủ tục đã được biên dịch, bạn có thể thực thi nó bằng cách nhấp phải chuột vào nó và chọn Run. Kết quả sẽ xuất hiện trong khung “Result” ở góc dưới phải

Để thực thi một thủ tục bằng dòng lệnh, bạn có thể sử dụng câu lệnh CALL <procedure name>

**14.2 Một số khái niệm về thủ tục SQL PL:**

Những thủ tục lưu với ngôn ngữ thủ tục rất dễ tạo và học. Trong DB2 thủ tục có hiệu suất tốt nhất. Thủ tục lưu SQL PL (hay nói tắt là “Thủ tục SQL”) là tiêu điểm của chương này.

**14.2.1 Cấu trúc của thủ tục**

Cú pháp của thủ tục

```
CREATE PROCEDURE proc_name [({tham số tùy chọn})]
    [thuộc tính tùy chọn của thủ tục] <câu lệnh>
```

Trong đó <câu lệnh> là một câu lệnh hoặc một tập các câu lệnh được nhóm bởi cấu trúc BEGIN [ATOMIC] ..... [END]

**14.2.2 Những thuộc tính tùy chọn của thủ tục**

Sau đây là 1 vài thuộc tính tùy chọn của thủ tục:

- LANGUAGE SQL
 

Thuộc tính này chỉ ra ngôn ngữ mà thủ tục sẽ sử dụng. LANGUAGE SQL là giá trị mặc định. Với những ngôn ngữ khác như JAVA hay C thì ta sử dụng tương ứng thuộc tính LANGUAGE JAVA hay LANGUAGE C

- `RESULT SETS <n>`  
Thuộc tính này cần thiết nếu thủ tục của bạn sẽ trả về tập hợp kết quả n
- `SPECIFIC my_unique_name`  
Đây là tên duy nhất được gán cho thủ tục. Một thủ tục có thể được nạp đề, nghĩa là, nhiều thủ tục lưu trữ có thể có cùng một tên, nhưng với số lượng tham số khác nhau. Bằng cách sử dụng từ khóa `SPECIFIC` bạn có thể gán một tên duy nhất với những thủ tục này, và điều này tạo ra sự dễ dàng cho việc quản lý các thủ tục lưu trữ. Ví dụ, để xóa một thủ tục sử dụng từ khóa `SPECIFIC` bạn có thể dùng câu lệnh: `DROP SPECIFIC PROCEDURE`. Nếu từ khóa `SPECIFIC` không được sử dụng thì với câu lệnh xóa thủ tục lưu trữ như trên, để DB2 biết được thủ tục nào cần được xóa (vì có rất nhiều thủ tục có tên giống nhau được ghi đề) thì bạn phải chỉ ra danh sách tham số đi kèm của thủ tục đó.

### 14.2.3 Các loại tham số

Có 3 loại tham số trong thủ tục lưu trữ SQL PL:

- `IN` - Tham số vào
- `OUT` - Tham số ra
- `INOUT` - Tham số vừa vào và ra

Ví dụ:

```
CREATE PROCEDURE proc(IN p1 INT, OUT p2 INT, INOUT p3 INT)
```

Khi gọi thủ tục, tất cả các tham số cần phải được đưa vào trong câu lệnh `CALL`. Ví dụ, để gọi thủ tục trên bạn phải gọi như sau:

```
CALL proc (10, ?, 4)
```

Dấu hỏi (?) được dùng cho các tham số trả về của câu lệnh `CALL`.

Đây là một ví dụ khác của thủ tục lưu mà bạn có thể thử:

```
CREATE PROCEDURE P2 ( IN      v_p1 INT,
                    INOUT  v_p2 INT,
                    OUT    v_p3 INT)
LANGUAGE SQL
SPECIFIC myP2
BEGIN
    -- my second SQL procedure
    SET v_p2 = v_p2 + v_p1;
    SET v_p3 = v_p1;
END
```

Để gọi thủ tục này từ dòng lệnh, ta dùng câu lệnh:

```
CALL P2 (3, 4, ?)
```

### 14.2.4 Chú thích trong thủ tục SQL PL

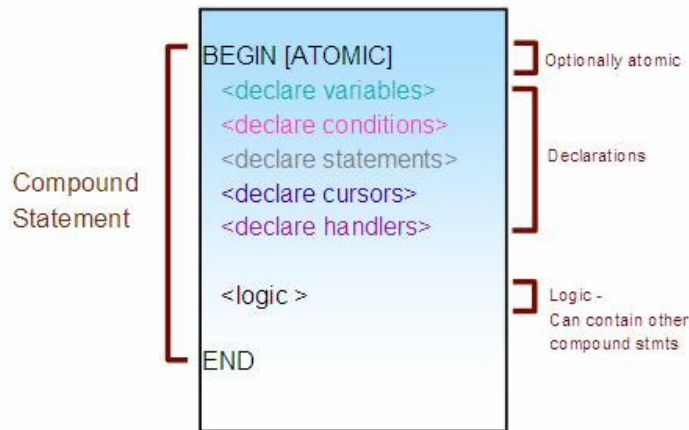
Có hai cách để chú thích trong thủ tục SQL PL:

- Sử dụng 2 dấu gạch ngang (--), ví dụ:  
-- Đây là 1 chú thích theo kiểu SQL

- Sử dụng định dạng như ngôn ngữ lập trình C, ví dụ:  
/\* Đây là 1 chú thích dạng C \*/

### 14.2.5 Câu lệnh ghép

Một câu lệnh ghép trong thủ tục là một câu lệnh trong đó chứa rất nhiều lệnh thủ tục và các câu lệnh SQL được bao bởi từ khóa BEGIN và END. Khi từ khóa ATOMIC được sử dụng theo sau từ khóa BEGIN, câu lệnh ghép được xem như là một đơn vị, nghĩa là, tất cả các chỉ thị hay câu lệnh trong câu lệnh ghép phải hoàn thành để câu lệnh ghép được hoàn thành. Nếu có một câu lệnh thất bại, tất cả sẽ được trả về trạng thái ban đầu (roll back). Hình 14.7 cho ta 1 minh họa về câu lệnh ghép.



Hình 14.7 – Câu lệnh ghép

### 14.2.6 Khai báo biến

Để khai báo biến, sử dụng câu lệnh DECLARE như sau:

```
DECLARE var_name <data type> [DEFAULT value];
```

Đây là một vài ví dụ:

```
DECLARE temp1 SMALLINT DEFAULT 0;
DECLARE temp2 INTEGER DEFAULT 10;
DECLARE temp3 DECIMAL(10,2) DEFAULT 100.10;
DECLARE temp4 REAL DEFAULT 10.1;
DECLARE temp5 DOUBLE DEFAULT 10000.1001;
DECLARE temp6 BIGINT DEFAULT 10000;
DECLARE temp7 CHAR(10) DEFAULT 'yes';
DECLARE temp8 VARCHAR(10) DEFAULT 'hello';
DECLARE temp9 DATE DEFAULT '1998-12-25';
DECLARE temp10 TIME DEFAULT '1:50 PM';
DECLARE temp11 TIMESTAMP DEFAULT '2001-01-05-12.00.00';
DECLARE temp12 CLOB(2G);
DECLARE temp13 BLOB(2G);
```

### 14.2.7 Câu lệnh gán

Để gán một giá trị cho một biến, sử dụng câu lệnh SET. Ví dụ:

```
SET total = 100;
```

Câu lệnh trên cũng tương đương với câu lệnh:

```
VALUES(100) INTO total;
```

Ngoài ra, bất kì biến nào cũng có thể được gán với giá trị NULL

```
SET total = NULL
```

Một biểu thức điều kiện sẽ được thiết lập nếu có nhiều hơn một dòng thì chỉ lấy dòng đầu tiên trong bảng

```
SET total = (select sum(c1) from T1);
SET first_val = (select c1 from T1 fetch first 1 row only)
```

Bạn cũng có thể gán biến với các thuộc tính ngoài của cơ sở dữ liệu

```
SET sch = CURRENT_SCHEMA;
```

### 14.3 Các con trỏ (cursor)

Một con trỏ là một tập hợp chứa kết quả của câu lệnh SELECT. Cú pháp để khai báo, mở, chuyển giá trị (fetch) và đóng con trỏ như sau:

```
DECLARE <tên con trỏ> CURSOR [WITH RETURN <đích trả về>]
    <câu lệnh SELECT>;
OPEN <tên con trỏ>;
FETCH < tên con trỏ > INTO <các biến>;
CLOSE < tên con trỏ>;
```

Khi một con trỏ được khai báo, mệnh đề WITH RETURN có thể được sử dụng với những giá trị sau:

- CLIENT: Kết quả sẽ được trả về cho ứng dụng máy khách
- CALLER: Kết quả sẽ được trả về cho máy khách hoặc thủ tục lưu đã gọi nó.

Đây là một ví dụ của thủ tục sử dụng con trỏ:

```
CREATE PROCEDURE set()
DYNAMIC RESULT SETS 1
LANGUAGE SQL
BEGIN
DECLARE cur CURSOR WITH RETURN TO CLIENT
FOR SELECT name, dept, job
FROM staff
WHERE salary > 20000;
OPEN cur;
END
```

### 14.4 Điều khiển luồng

Như trong rất nhiều ngôn ngữ khác, SQL PL có rất nhiều câu lệnh có thể được dùng để điều khiển luồng thực hiện mức logic. Dưới đây là một vài câu lệnh được hỗ trợ:

CASE (chọn ra một nhánh thực thi (tìm kiếm đơn giản))  
 IF  
 FOR (thực thi mỗi dòng trong bảng)

WHILE  
 ITERATE (buộc phải thực hiện vòng lặp tiếp theo. Tương tự với câu lệnh CONTINUE trong C)  
 LEAVE (rời khỏi một khối hay vòng lặp)  
 LOOP (lặp vô hạn định)  
 REPEAT  
 GOTO  
 RETURN  
 CALL (gọi thủ tục)

### 14.5 Gọi thủ tục

Những đoạn mã sau đây sẽ chỉ ra cách gọi một thủ tục bằng các ngôn ngữ lập trình khác nhau.

#### Gọi thủ tục từ ứng dụng CLI/ODBC

```
SQLCHAR *stmt = (SQLCHAR *)
"CALL MEDIAN_RESULT _SET( ? )" ;
SQLDOUBLE sal = 20000.0; /*Chỉ ra parameter marker trong stmt */
SQLINTEGER salind = 0; /*Chỉ định giá trị (variable) cho sal */

sqlrc = SQLPrepare(hstmt, stmt, SQL_NTS);
sqlrc = SQLBindParameter(hstmt, 1, SQL_PARAM_OUTPUT,
SQL_C_DOUBLE, SQL_DOUBLE, 0, 0, &sal, 0, &salind);
SQLExecute(hstmt);
if (salind == SQL_NULL_DATA)
    printf("Median Salary = NULL\n");
else
    printf("Median Salary = %.2f\n\n", sal);
    /* Lấy tập kết quả đầu tiên */
    sqlrc = StmtResultPrint(hstmt);
    if (sqlrc == SQL_SUCCESS) {
        /* Có 1 kết quả khác */
        sqlrc = SQLMoreResult(hstmt);
    }
}
```

Xem chi tiết ở tệp ví dụ của DB2: `sqllib/samples/sqlproc/resultset.c`

#### Gọi thủ tục từ ứng dụng VB.NET

```
Try
    ` Tạo một DB2 Command để chạy thủ tục
    Dim procName As String = "TRUNC_DEMO"
    Dim cmd As DB2Command = conn.CreateCommand()
    Dim parm As DB2Parameter

    cmd.CommandType = CommandType.StoredProcedure
    cmd.CommandText = procName

    ` Đăng ký các tham số đầu ra cho DB2Command
    parm = cmd.Parameters.Add("v_lastname", DB2Type.VarChar)
    parm.Direction = ParameterDirection.Output
    parm = cmd.Parameters.Add("v_msg", DB2Type.VarChar)
    parm.Direction = ParameterDirection.Output

    ` Gọi lại thủ tục
```

```

Dim reader As DB2DataReader = cmd.ExecuteReader

Catch myException As DB2Exception
    DB2ExceptionHandler(myException)
Catch
    UnhandledExceptionHandler()
End Try

```

### Gọi thủ tục từ ứng dụng Java

```

try
{
    // Kết nối CSDL mẫu
    String url = "jdbc:db2:sample";
    con = DriverManager.getConnection(url);
    CallableStatement cs = con.prepareCall("CALL
        trunc_demo(?, ?)");

    // Đăng ký tham số đầu ra
    callStmt.registerOutParameter(1, Types.VARCHAR);
    callStmt.registerOutParameter(2, Types.VARCHAR);

    cs.execute();
    con.close();
}
catch (Exception e)
{
    /* xử lý ngoại lệ ở đây */
}

```

### 14.6 Lỗi và xử lý điều kiện

Trong DB2, các từ khóa `SQLCODE` và `SQLSTATE` được dùng để xác định sự thành công hay thất bại trong việc thực thi một câu lệnh SQL. Những từ khóa này cần được khai báo tường minh trong phần ngoài cùng của thủ tục như sau:

```

DECLARE SQLSTATE CHAR[5];
DECLARE SQLCODE INT;

```

DB2 sẽ thiết lập giá trị một cách tự động cho những từ khóa trên sau mỗi thao tác trên SQL (SQL operation). Đối với `SQLCODE`, giá trị được thiết lập như sau:

- = 0, thành công
- > 0, thành công nhưng có cảnh báo
- < 0, không thành công
- = 100, không tìm thấy dữ liệu (ví dụ như: câu lệnh `FETCH` không trả về dữ liệu)

Đối với `SQLSTATE`, các giá trị của nó như sau:

- Thành công: `SQLSTATE '00000'`
- Không thấy: `SQLSTATE '02000'`
- Cảnh báo: `SQLSTATE '01XXXX'`
- Trường hợp đặc biệt: các giá trị khác

`SQLCODE` là một đặc tả cho cơ sở dữ liệu, chi tiết hơn cả `SQLSTATE`. `SQLSTATE` là tiêu chuẩn trong các hệ quản trị cho cơ sở dữ liệu nhưng tổng quát hơn. Nhiều



SQLCODE có thể trùng với SQLSTATE. SQLCODE và SQLSTATE sẽ được nói rõ hơn ở Phụ lục A, Xử lý sự cố.

Một điều kiện có thể được tạo ra bằng bất kì câu lệnh SQL nào và sẽ trùng với một SQLSTATE. Ví dụ, một điều kiện cụ thể như SQLSTATE '01004' được tạo ra khi có một giá trị được làm tròn số trong quá trình thi hành câu lệnh SQL. Thay vì dùng SQLSTATE '01004' để kiểm tra điều kiện này, một tên gọi có thể được gán. Trong ví dụ cụ thể này, tên "trunc" có thể được gán cho điều kiện SQLSTATE '01004' như sau.

```
DECLARE trunc CONDITION FOR SQLSTATE '01004'
```

Các điều kiện khác đã được định nghĩa trước:

- SQLWARNING
- SQLEXCEPTION
- NOT FOUND

### Xử lý điều kiện

Để xử lý một điều kiện, bạn có thể tạo ra một trình xử lý điều kiện, và phải chỉ rõ các yếu tố sau:

- Điều kiện mà nó phải xử lý
- Nơi phục hồi lại quá trình thực thi (dựa vào kiểu của trình xử lý: CONTINUE, EXIT or UNDO)
- Hành động được thi hành để xử lý điều kiện. Những hành động này có thể là bất cứ câu lệnh nào, bao gồm cả các cấu trúc điều khiển.

Nếu một điều kiện SQLEXCEPTION được tạo ra, và không có trình xử lý nào, thủ tục sẽ kết thúc và trả về cho máy khách một thông báo lỗi.

### Các loại trình xử lý

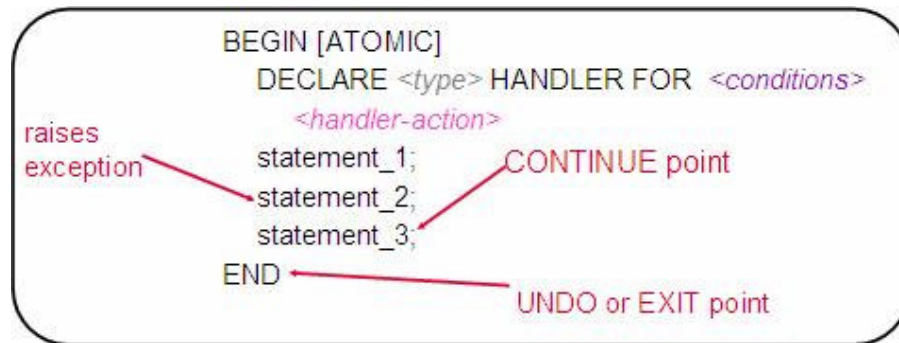
Có 3 loại trình xử lý:

**CONTINUE** – Xử lý này dùng để chỉ ra rằng, sau khi ngoại lệ xảy ra, và trình xử lý đã xử lý điều kiện, dòng xử lý sẽ TIẾP TỤC đi đến câu lệnh tiếp theo sau câu lệnh vừa được tạo ra.

**EXIT** – xử lý này dùng để chỉ ra rằng, sau khi ngoại lệ xảy ra, và trình xử lý đã xử lý điều kiện, dòng xử lý sẽ đi đến cuối thủ tục.

**UNDO** – xử lý này được dùng để chỉ ra rằng, sau khi ngoại lệ xảy ra, và trình xử lý đã xử lý điều kiện, dòng xử lý sẽ đi đến cuối thủ tục, và sẽ không làm gì ại hoặc khôi phục bất kì câu lệnh nào đã được thực thi (roll back).

Hình 14.8 minh họa sự khác nhau giữa các trình xử lý điều kiện và hành vi của nó



Hình 14.8 – Kiểu của các trình xử lý điều kiện

### 14.7 SQL động

Trong SQL động, trái ngược với SQL tĩnh, toàn bộ câu lệnh SQL không được nhận biết trong thời gian chạy. Ví dụ nếu `col1` và `tablename` là các biến trong câu lệnh này, chúng ta sẽ thực hiện với SQL động như sau:

```
`SELECT ` || col1 || `FROM ` || tablename;
```

SQL động cũng được đề nghị dùng cho DDL để tránh các vấn đề phụ thuộc và sai sót khi đóng gói. Nó cũng được yêu cầu khi thực hiện đệ quy.

SQL động có thể được thực thi theo hai cách:

- Dùng lệnh EXECUTE IMMEDIATE – đây là ý tưởng cho việc thực thi thủ tục SQL đơn.
- Dùng lệnh PREPARE song song với lệnh EXECUTE – ý tưởng này được dùng để thực thi đa thủ tục SQL.

Đoạn mã sau đưa ra một ví dụ về SQL động sử dụng cả hai cách trên. Ví dụ này giả sử rằng có một bảng T2 đã được tạo như sau:

```

CREATE TABLE T2 (c1 INT, c2 INT)

CREATE PROCEDURE dyn1 (IN value1 INT, INT value2 INT)
SPECIFIC dyn1
BEGIN
  DECLARE stmt varchar(255);
  DECLARE st STATEMENT;

  SET stmt = `INSERT INTO T2 VALUE (?, ?)`;

  PREPARE st FROM stmt;

  EXECUTE st USING value1, value1;
  EXECUTE st USING value2, value2;

  SET stmt = INSERT INTO T2 VALUES (9, 9)';
  EXECUTE IMMEDIATE stmt;
END
  
```

# 15

## Chương 15 – Hàm nội tuyến, hàm người dùng định nghĩa, các hàm xử lý các sự kiện bất (Inline SQL PL, UDF, Triggers)

Trong chương này, chúng ta sẽ xem xét về các hàm nội tuyến SQL các đối tượng khác của CSDL như hàm do người dùng định nghĩa (UDF) và các bất sự kiện.

### 15.1 Hàm nội tuyến SQL

Trong chương 14, chúng ta đã thảo luận về việc tạo các thủ tục lưu sử dụng ngôn ngữ SQL PL. Ngôn ngữ SQL PL này cũng có thể được sử dụng nội tuyến, nghĩa là có thể đứng một mình, không nằm trong một thủ tục nào cả. Các lệnh SQL PL được sử dụng trong các UDF và hàm bất sự kiện cũng là nội tuyến vì các mã lệnh đã được thêm vào mã lệnh của các UDF và hàm bất sự kiện, bản thân nó đã là các SQL động. SQL PL nội tuyến chỉ hỗ trợ một tập con trong toàn bộ các lệnh SQL PL, sau đây là các từ khóa của SQL PL nội tuyến:

```
DECLARE <variable>
SET
CASE
FOR
GET DIAGNOSTICS
GOTO
IF
RETURN
SIGNAL
WHILE
ITERATE
LEAVE
```

Đây là những từ khóa không nằm trong SQL PL nội tuyến:

```
ALLOCATE CURSOR
ASSOCIATE LOCATORS
DECLARE <cursor>
DECLARE ...HANDLER
PREPARE
EXECUTE
EXECUTE IMMEDIATE
LOOP
REPEAT
RESIGNAL
CALL
COMMIT/ROLLBACK
```

Dưới đây là ví dụ của SQL động sử dụng SQL PL nội tuyến. Nếu các bạn muốn thử nó, bản các thể lấy nó làm một phần của cửa tệp kịch bản, và bạn phải chắc chắn là đã tạo được các bảng như sau:

```
CREATE TABLE T1 (c1 INT)
CREATE TABLE T3 (c1 INT)

BEGIN ATOMIC
  DECLARE cnt          INT DEFAULT 0;
  DECLARE sumevens    INT DEFAULT 0;
  DECLARE err_msg     VARCHAR(1000) DEFAULT '';
  WHILE (cnt < 100) DO
    IF mod(cnt,2) = 0 THEN
      SET sumevens = sumevens + cnt;
    END IF;
    SET cnt=cnt+1;
  END WHILE;
  INSERT INTO T3 values (sumevens);
  SET cnt = (SELECT 0 FROM SYSIBM.SYSDUMMY1);
  FOR curl AS SELECT * FROM T1 DO
    IF curl.c1 > 100 THEN
      SET cnt = cnt + 1;
    END IF;
  END FOR;

  SET err_msg = 'Rows with values > 100 is: ' ||
char(cnt);
  SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT = err_msg;
END!
```

Nếu bạn lưu đoạn chương trình nội tuyến trên vào tệp kịch bản tên là “myScript.txt” thì bạn có thể thực hiện nó bằng lệnh sau;

```
db2 -td! -vf myScript.txt
```

## 15.2 Bẫy sự kiện (Triggers)

Bẫy sự kiện là các đối tượng cơ sở dữ liệu kết hợp với một bảng được định nghĩa bởi các thao tác được thực hiện khi có một trong các thao tác INSERT, UPDATE, hoặc DELETE được thực hiện trên bảng đó. Chúng được thi hành một cách tự động. Các thao tác là nguyên nhân để các bẫy này được thực thi, gọi là *triggering*.

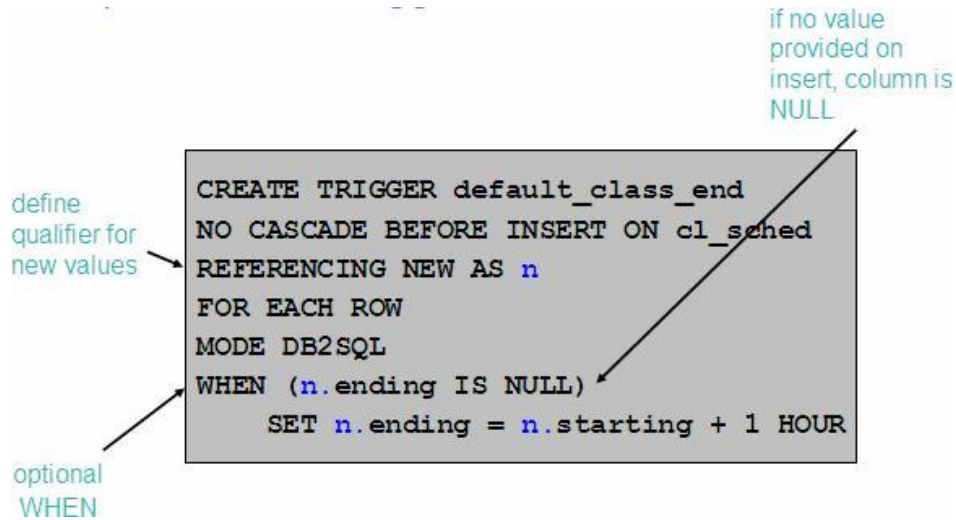
### 15.2.1 Các loại bẫy sự kiện

Có 3 loại bẫy đó là: bẫy trước (before), bẫy sau (after) và bẫy thay thế (instead of)

#### Bẫy trước

Các bẫy trước được kích hoạt trước khi một dòng được thêm, sửa hay xóa đi. Các thao tác được thực hiện bởi bẫy này thì không thể kích hoạt được các bẫy khác (vì thế các thao tác INSERT, UPDATE và DELETE không được phép có trong bẫy).

Một ví dụ đơn giản về bẫy trước được chỉ ra trên hình 15.1



**Hình 15.1 – Ví dụ về bẫy trước**

Trong hình 15.1, bẫy “default\_class\_end” sẽ bẫy trước lệnh khi INSERT SQL được thực hiện trong bảng cl\_sched. Bảng này là một phần của cơ sở dữ liệu SAMPLE, bạn có thể tạo và thử kiểm tra bẫy này khi kết nối với cơ sở dữ liệu này. Biến “n” trong định nghĩa của bẫy sẽ đại diện cho giá trị mới của lệnh INSERT, giá trị sẽ chèn vào. Bẫy này sẽ kiểm tra sự hợp lệ của các giá trị chèn vào. Nếu cột “ending” không có giá trị gì khi chèn, thì bẫy này sẽ đảm bảo chắc chắn là có giá trị của cột “starting” cộng 1 được chèn vào.

Các lệnh sau giúp ta kiểm tra bẫy này

```

C:\Program Files\IBM\SQLLIB\BIN>db2 insert into cl_sched
(class_code, day, starting) values ('abc',1,current time)
DB20000I The SQL command completed successfully.

C:\Program Files\IBM\SQLLIB\BIN>db2 select * from cl_sched

CLASS_CODE DAY    STARTING ENDING
-----
042:BF      4 12:10:00 14:00:00
553:MJA     1 10:30:00 11:00:00
543:CWM     3 09:10:00 10:30:00
778:RES     2 12:10:00 14:00:00
044:HD      3 17:12:30 18:00:00
abc         1 11:06:53 12:06:53

6 record(s) selected.
    
```

Thủ tục bẫy “validate\_sched” dưới đây cho phép mở rộng hàm bẫy “default\_class\_end” đã mô tả trước đây. Xin nhắc lại rằng, bạn có thể tạo ra nó và kiểm tra nó dựa vào cơ sở dữ liệu SAMPLE.

```

CREATE TRIGGER validate_sched
NO CASCADE BEFORE INSERT ON cl_sched
REFERENCING NEW AS n
FOR EACH ROW
MODE DB2SQL
BEGIN ATOMIC
-- supply default value for ending time if null
    
```

```

IF (n.ending IS NULL) THEN
    SET n.ending = n.starting + 1 HOUR;
END IF;

-- ensure that class does not end beyond 9pm
IF (n.ending > '21:00') THEN
    SIGNAL SQLSTATE '80000'
    SET MESSAGE_TEXT='class ending time is beyond 9pm';
ELSEIF (n.DAY=1 or n.DAY=7) THEN
    SIGNAL SQLSTATE '80001'
    SET MESSAGE_TEXT='class cannot be scheduled on a
weekend';
END IF;
END

```

### Bẫy sau

Bẫy sau được kích hoạt sau khi các *SQL triggering* thực hiện thành công. Các thao tác trong bẫy này có thể kích hoạt các bẫy khác (sự nổi tầng được cho phép đến 16 mức). Bẫy sau hỗ trợ các thao tác INSERT, UPDATE và DELETE. Dưới đây là ví dụ của một bẫy sau

```

CREATE TRIGGER audit_emp_sal
AFTER UPDATE OF salary ON employee
REFERENCING OLD AS o NEW AS n
FOR EACH ROW
MODE DB2SQL
INSERT INTO audit VALUES (
    CURRENT TIMESTAMP, ' Employee ' || o.empno || ' sal-
ary changed from ' || CHAR(o.salary) || ' to ' ||
CHAR(n.salary) || ' by ' || USER)

```

Trong ví dụ này thì thủ tục bẫy `audit_emp_sal` được sử dụng để thực hiện việc kiểm tra trên cột `salary` của bảng `employee`. Khi một ai đó tạo sự thay đổi trong cột này thì đoạn mã này sẽ thực hiện việc ghi thông tin về việc thay đổi liên quan đến bảng `Salary` vào trong bảng khác gọi là `audit`. Tại dòng “OLD as o NEW as n” chỉ ra tiền tố “o” này sẽ được dùng để đại diện giá trị cũ hoặc giá trị tồn tại trong bảng, và tiền tố “n” sẽ được sử dụng để đại diện giá trị mới sau đó từ câu lệnh UPDATE. Như vậy, “o.salary” miêu tả giá trị cũ hoặc giá trị tồn tại trong bảng `Salary` còn “n.salary” miêu tả giá trị được cập nhật mới vào cột dữ liệu `Salary`.

### Bẫy sự kiện thay thế

Bẫy thay thế được định nghĩa trên các khung nhìn (view). Định nghĩa một cách logic bẫy này sẽ thực hiện việc cập nhật cho các lệnh *SQL triggering*. Ví dụ như, nếu bạn thực hiện một thao tác cập nhật trên một khung nhìn, bẫy thay thế sẽ được thực thi để thực hiện việc cập nhật các bảng gốc mà các khung nhìn này thể hiện.

Các thủ tục bẫy không được tạo từ DB2 Developer Workbench. Chúng có thể được tạo từ trung tâm điều khiển hoặc từ các công cụ dòng lệnh khác (Command Window, Command Line Processor, hoặc Command Editor)

## Bài tập nhanh #10 - Khởi tạo một bẫy bằng Trung tâm điều khiển.

### Mục tiêu

Bẫy là một đối tượng cơ sở dữ liệu được sử dụng để hoàn thành công việc kinh doanh một cách chặt chẽ khi xảy ra thay đổi dữ liệu trên bảng (chèn, sửa, xóa). Trong bài tập nhanh này, bạn sẽ tạo một bẫy bằng việc sử dụng Trung tâm điều khiển. Cái bẫy này sẽ lưu nhật ký bán hàng trong bảng *SALES* (bán hàng) cho mục đích kiểm tra sổ sách. Bạn sẽ ghi vào nhật ký bán hàng mã của người bán hàng (*user ID*) cùng với thời điểm bán.

### Thủ tục:

1. Mở Trung tâm điều khiển.
2. Với bài tập nhanh này, bạn sẽ cần khởi tạo một bảng phụ được sử dụng để ghi nhật ký bán hàng. Khởi tạo một bảng theo các thuộc tính sau:

Tên bảng: saleslog (nhật ký bán hàng)

Cột đầu tiên:

Tên: userid (mã)

Kiểu dữ liệu: VARCHAR(128)

Các thuộc tính khác: NOT NULL

Cột thứ hai:

Tên: daytime (thời điểm)

Kiểu dữ liệu: TIMESTAMP

Các thuộc tính khác: NOT NULL

Gợi ý: Khởi tạo bảng này sử dụng CREATE TABLE có trong cửa sổ nhập lệnh, hoặc sử dụng *Create table wizard* từ Trung tâm điều khiển.

3. Từ Trung tâm điều khiển, mở rộng tài liệu cơ sở dữ liệu *EXPRESS*. Nhấp chuột phải trong thư mục *Trigger*, và chọn mục "Create". Cửa sổ đối thoại khởi tạo bẫy mở ra.
4. Điền đầy đủ theo thông tin trong cửa sổ đối thoại:

Sơ đồ bẫy (Trigger schema): User ID (mã) của người bán đã được nhập vào (nên được cài đặt mặc định)

Tên bẫy (Trigger name): *audit\_sales* (Kiểm tra bán hàng)

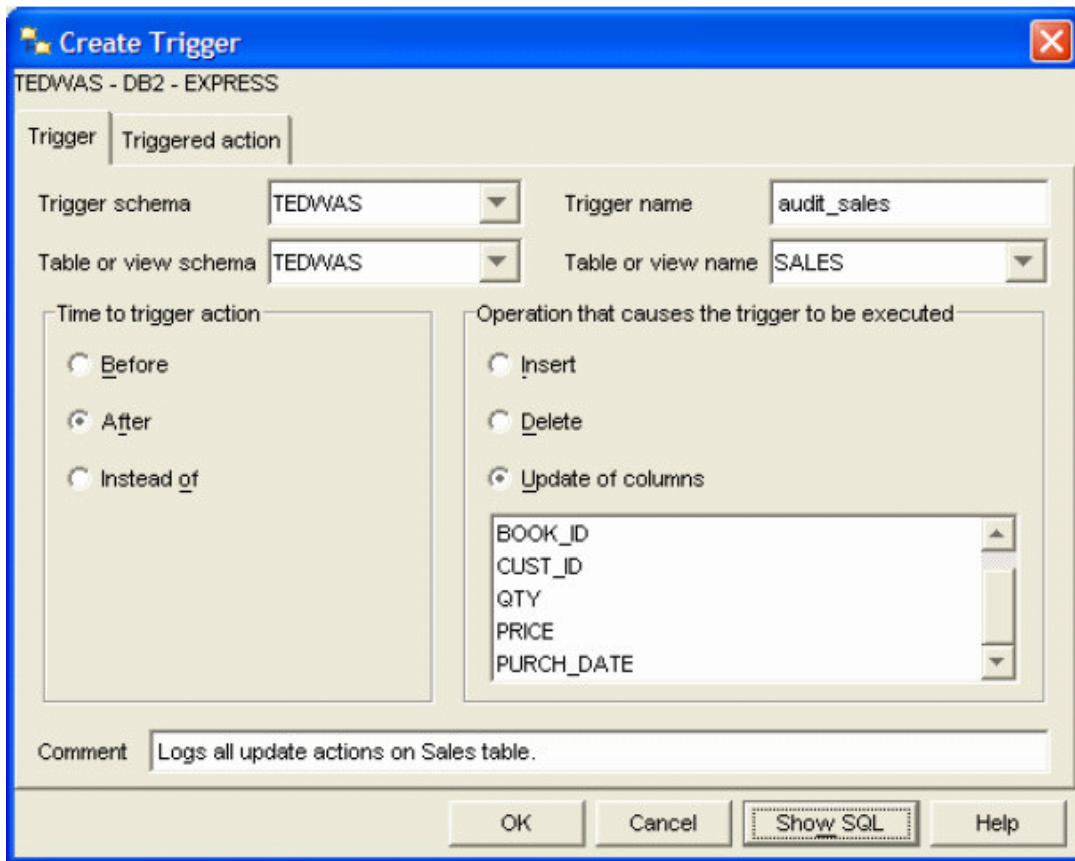
Bảng hay sơ đồ hiển thị: User ID (mã) của người bán đã được nhập vào (nên được cài đặt mặc định)

Bảng hay tên hiển thị (Table or view name): SALES (bán hàng)

Thời gian để bẫy hoạt động (Time to trigger action): After (sau)

Khi nào thì bẫy được thực hiện (Operation that causes the trigger to be executed): *Update of columns* (không liệt kê bất cứ cột nào vì chúng ta muốn bẫy thực hiện khi bất cứ cột nào trong số các cột được cập nhật)

Chú thích (Comment): Logs all update actions on Sales table (Ghi nhật ký tất cả các hoạt động cập nhật trong bảng Sales)

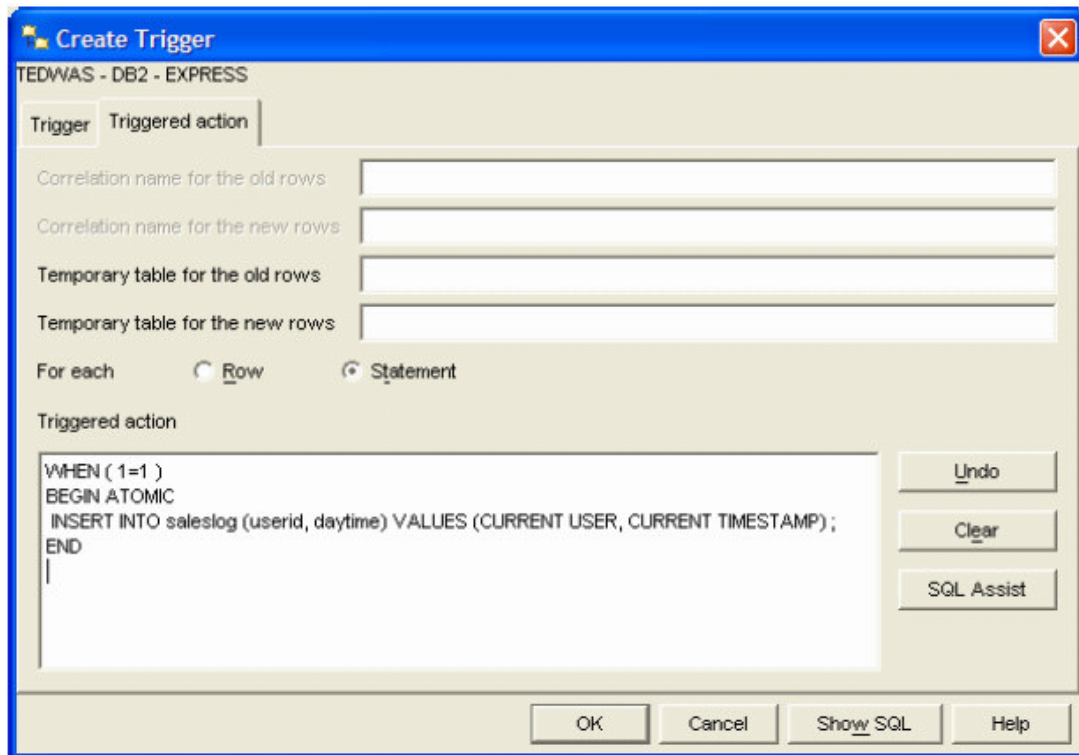


5. Trong bảng *Triggerred action*, chọn **STATEMENT** trong mục tùy chọn *For Each*. Nhập đoạn mã nguồn sau vào mục hành động của bẫy:

```
WHEN (1=1)
BEGIN ATOMIC
INSERT INTO saleslog (userid, daytime) VALUES (CURRENT
USER, CURRENT TIMESTAMP);
END
```

(Ghi chú: đoạn bẫy sẽ được thực hiện một lần sau khi bẫy được kích hoạt và kết thúc hoàn toàn. Bẫy theo dòng nghĩa là hoạt động của bẫy sẽ thực hiện một khi *SQL triggering* tác động vào một dòng.)





Kích chọn nút OK để khởi tạo bẫy.

6. Bây giờ bạn nên xem lại bẫy đã được khởi tạo trong tài liệu *Triggers* ở thư mục Trung tâm điều khiển.
7. Truy vấn bảng *Saleslog* để đảm bảo chưa có dữ liệu trong bảng này. Loại bỏ những hàng có dữ liệu trong bảng bằng lệnh `DELETE FROM saleslog`.
8. Thử cập nhật một bản ghi trong bảng *sales* (Gợi ý: sử dụng Command Editor hoặc SQL Assist Wizard).
9. Kiểm tra lại một lần nữa nội dung của bảng *saleslog*. Xem có bao nhiêu hàng trong đó.

### 15.3 Hàm người dùng định nghĩa (UDF)

Hàm người dùng định nghĩa là một đối tượng ứng dụng cơ sở dữ liệu, nó được sắp xếp thành bộ dữ liệu nhập và dữ liệu xuất. Ví dụ: một hàm có thể lấy kích thước dữ liệu vào bằng `inch`, và trả lại kết quả bằng `centimetes`.

DB2 hỗ trợ tạo những hàm sử dụng SQL PL, C/C++, Java, CLR (Common Language Runtime), và OLE (Object Linking and Embedding). Trong cuốn sách này, chúng tôi nói chủ yếu về hàm SQL PL bởi tính đơn giản, phổ biến và hiệu suất của chúng.

Có là 4 loại hàm: hàm vô hướng (Scalar Functions), hàm bảng (Table Functions), hàm dòng (Row Functions), và hàm cột (Column Functions). Trong bài này chúng tôi chỉ tập trung vào hàm vô hướng và hàm bảng.

#### 15.3.1 Hàm vô hướng

Hàm vô hướng trả lại một giá trị đơn. Nó không thể chứa các câu lệnh SQL, các câu lệnh `INSERT`, `UPDATE` và `DELETE` sẽ làm thay đổi trạng thái của cơ sở dữ liệu thì không được chấp nhận. Một vài hàm vô hướng được cài đặt sẵn là `SUM()`, `AVG()`, `DIGITS()`, `COALESCE()`, và `SUBSTR()`.

DB2 cho phép bạn tùy biến xây dựng hàm người dùng định nghĩa, từ đó bạn có thể thường xuyên sử dụng nó theo ý nghĩa logic. Ví dụ, tính đến sự chuyển đổi ứng dụng của bạn từ Oracle sang DB2. Trong ứng dụng này, bạn gọi hàm NVL() của Oracle ở khắp nơi. Hàm tương đương cài đặt sẵn trong DB2 được gọi là COALESCE. Thay vì bạn phải đổi tên tất cả các hàm NVL trong ứng dụng thành COALESCE, bạn có thể tạo một UDF trong DB2 gọi nó là NVL, và nó sẽ kích hoạt hàm COALESCE khi được gọi; ta sẽ viết hàm NLV như dưới đây:

```
CREATE FUNCTION NVL (p_var1 VARCHAR(30) ,
                    p_var2  VARCHAR(30))
SPECIFIC nvlvarchar30
RETURNS VARCHAR(30)
RETURN COALESCE (p_var1 , p_var2)
```

Hàm COALESCE trong trường hợp này trả lại đối số đầu tiên là không rỗng (not NULL)

Dưới đây là một ví dụ khác của hàm vô hướng. Hàm được gọi là “deptname”, và nó trả lại mã số của phòng/ban (department) của một nhân viên dựa trên các mã (ID) của họ:

```
CREATE FUNCTION deptname(p_empid VARCHAR(6))
RETURNS VARCHAR(30)
SPECIFIC deptname
BEGIN AUTOMIC
DECLARE v_department_name VARCHAR(30);
DECLARE v_err VARCHAR(70);
SET v_department_name =(
  SELECT d.deptname FROM department d,employee e
  WHERE e.workdept = d.deptno AND e.empno = p_empid);
SET v_err = 'Error:employee' || p_empid || 'was not found';
IF v_department_name IS NULL THEN
  SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT = v_err;
END IF;
RETURN v_department_name;
END
```

Để kiểm tra lại hàm, thử lại trong Command Window hoặc Linux/UNIX với lệnh:

```
db2 "values (deptname('000300'))"
```

### Gọi hàm vô hướng

Hàm vô hướng của người dùng có thể được gọi trong câu lệnh SQL, ở bất kỳ đâu giá trị vô hướng được sử dụng, hoặc trong mệnh đề liên quan đến một “Giá trị”. Đây là hai ví dụ biểu diễn lời gọi hàm vô hướng COALESCE:

```
SELECT DEPTNAME, COALESCE(MGRNO, 'ABSENT') FROM DEPARTMENT
VALUES COALESCE('A', 'B')
```

### 15.3.2 Hàm bảng

Hàm bảng trả lại một bảng của các bản ghi. Bạn có thể gọi chúng trong mệnh đề FROM của một truy vấn. Hàm bảng ngược lại với hàm vô hướng là có thể thay đổi trạng thái cơ sở dữ liệu, bởi vậy, các câu lệnh INSERT, UPDATE và DELETE được chấp nhận. Một vài hàm bảng được cài đặt sẵn là SNAPSHOT\_DYN\_SQL() và MQREADALL(). Hàm bảng cũng giống như khung nhìn (view), nhưng ở đây nó cho phép các lệnh làm thay đổi

dữ liệu (chèn, sửa, xóa) một cách mạnh hơn. Điển hình là chúng thường trả lại một bảng hoặc giữ lại một bản ghi kiểm tra.

Dưới đây là một ví dụ của hàm bảng để liệt kê một nhóm nhân viên trong phòng:

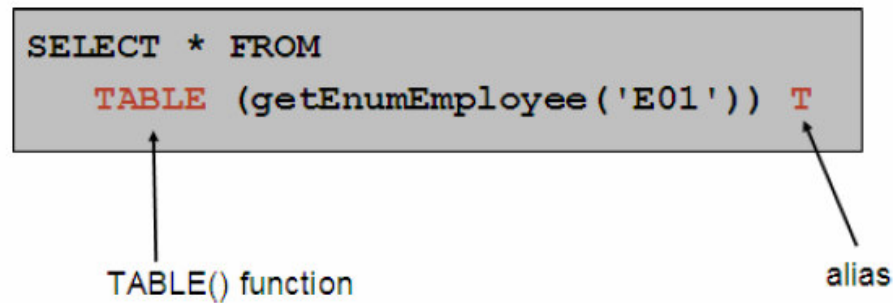
```
CREATE FUNCTION getEnumEmployee (p_dept VARCHAR(3))
RETURNS TABLE
    (empno CHAR(6),
     lastname VARCHAR(15),
     firstnme VARCHAR(12))
SPECIFIC getEnumEmployee
RETURN
    SELECT e.empno, e.lastname, e.firstnme
    FROM employee e
    WHERE e.workdept = p_dept
```

Để kiểm tra hàm trên, thử lại với câu lệnh sau:

```
db2 "SELECT * FROM table(getEnumEmployee('D11')) AS t"
```

### Gọi hàm bảng

Một hàm bảng người dùng định nghĩa được gọi trong mệnh đề *FROM* của câu lệnh *SQL*. Hàm *TABLE()* cần sử dụng và phải có bí danh. Hình 15.2 minh họa một ví dụ của cách gọi hàm “getEnumEmployee” mà chúng tôi đã kiểm tra.



Hình 15.3 - Lời gọi một hàm bảng.

## Bài tập nhanh #11 - Khởi tạo UDF sử dụng DB2 Developer Workbench

### Mục đích:

*DB2 Developer Workbench (DWB)* là một công cụ phát triển (IDE) để viết các thủ tục và các hàm. Trong bài tập nhanh này, các bạn sẽ phải tạo một hàm vô hướng do người dùng định nghĩa trong *DWB*. Việc này sẽ đem lại cho bạn nhiều kinh nghiệm hơn với *DBW*, tốt cho việc trau dồi kiến thức với ngôn ngữ *SQL PL* để xây dựng các hàm người dùng định nghĩa.

### Quy trình thực hiện:

1. Mở *DBW* ( Gợi ý: có sẵn trong bảng chọn Start)
2. Từ cửa sổ *Data Project Explorer* chọn dự án bạn đã tạo ở bài tập nhanh trước và chọn *Open Project*.
3. Kích chuột phải vào thư mục *User-Defined Functions*. Chọn mục *New*. Sau đó chọn mục *SQL User-Defined Function*. Bạn có thể chọn mục *User-Defined Function using Wizard* nếu cần được hướng dẫn trong suốt quá trình sử dụng.
4. Trình soạn thảo mở với hàm chính. Rồi sửa đổi mã lệnh như dưới đây:

```
CREATE FUNCTION booktitle(p_bid INTEGER)
RETURNS VARCHAR(300)
-----
SQL UDF (Scalar)
-----
SPECIFIC booktitle
F1: BEGIN ATOMIC
DECLARE v_book_title VARCHAR(300);
DECLARE v_err VARCHAR(70);
SET v_book_title = (SELECT title FROM books WHERE
    p_bid = book_id);
SET v_err = 'Error: The book with ID ' || CHAR(p_bid) ||
    'was Not found.';
IF v_book_title IS NULL THEN SIGNAL SQLSTATE '80000'
    SET MESSAGE_TEXT = v_err;
END IF;
RETURN v_book_title;
END
```

5. Xây dựng hàm bằng cách kích chuột phải trên hàm và chọn *Deploy*.
6. Cho chạy hàm bằng cách kích chuột vào nút *Run* trên thanh công cụ.
7. Sau khi hàm chấp nhận nhập một tham số vào, một cửa sổ thông báo hiện ra yêu cầu bạn điền giá trị cho tham số.

Nhập giá trị: 80002

Kết quả là gì?

Thử lại với giá trị: 1002

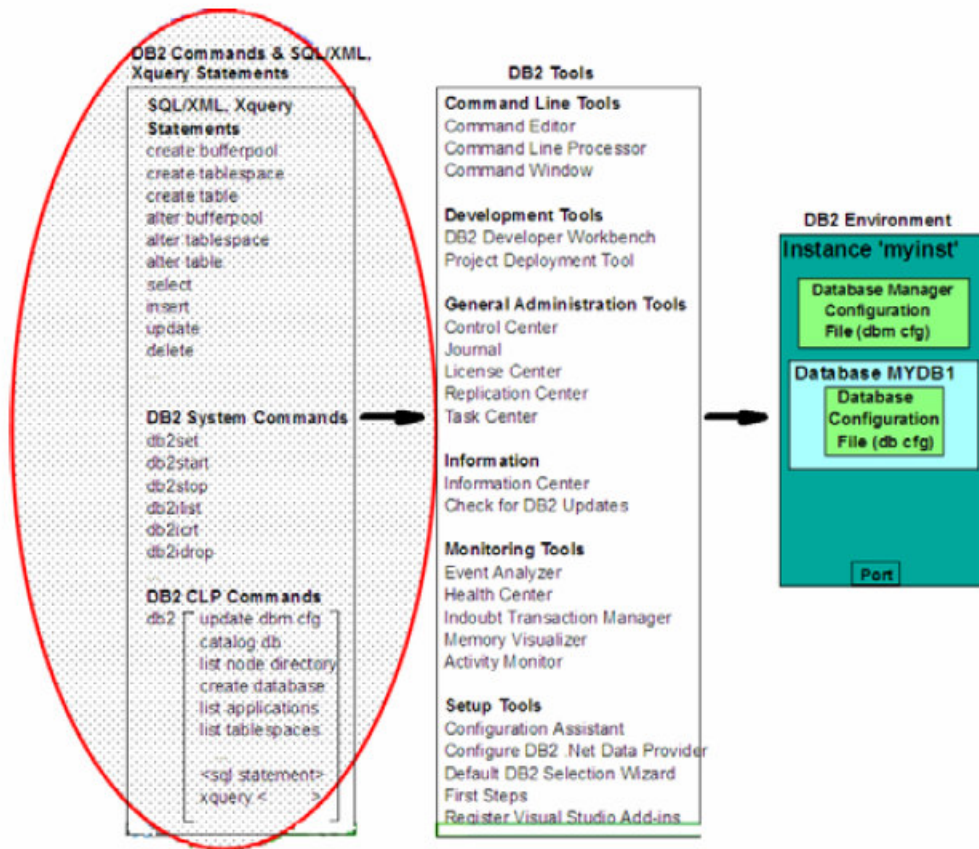
Điều gì xảy ra lúc này? (gợi ý: Nhìn vào đoạn *Messages* của *Out-put view*)

8. Đóng *DBW* khi bạn đã hoàn thành.

# 16

## Chương 16 – SQL/XML và XQuery

Trong chương này chúng ta bàn về pureXML, một công nghệ mới được cung cấp trong DB2 phiên bản 9 để hỗ trợ việc lưu trữ thuần XML. Nhiều ví dụ và khái niệm được bàn luận trong chương này được lấy từ sách đồ IBM: *DB2 9: Tổng quan và Bắt đầu nhanh với pureXML (IBM Redbook: DB2 9: pureXML overview and fast start)*. Xem phần tài nguyên để biết thêm thông tin. Hình 16.1 thể hiện phần kiến thức trong bức tranh tổng quát của DB2 sẽ được đề cập trong chương này.



Hình 16.1 – Bức tranh tổng quát DB2: các lệnh DB2, SQL/XML và XQuery

### 16.1 Sử dụng XML với các cơ sở dữ liệu

Các tài liệu XML được lưu dưới dạng tập tin văn bản, kho XML, hay cơ sở dữ liệu. Có hai nguyên nhân chính là tại sao nhiều công ty đề xuất lưu trữ XML bằng cơ sở dữ liệu:

- Quản lý lượng lớn hồ sơ XML là một vấn đề về cơ sở dữ liệu. XML là dữ liệu giống như các dữ liệu khác, chỉ khác phần định dạng. Những nguyên nhân tương tự để lưu trữ dữ liệu quan hệ trên cơ sở dữ liệu XML: các cơ sở dữ liệu cung cấp chức năng tìm kiếm và rút trích hiệu quả, hỗ trợ tính bền vững của dữ liệu một cách tin cậy, sao lưu và phục hồi, hỗ trợ giao dịch, tính hiệu quả và khả năng mở rộng.
- Tính tích hợp: bằng cách lưu trữ các tài liệu quan hệ và XML cùng với nhau, bạn có thể tích hợp dữ liệu XML mới cùng với dữ liệu quan hệ đã có, và kết hợp SQL với XPath hay XQuery trong một câu truy vấn. Hơn nữa, dữ liệu quan hệ có thể được chuyển thành XML và ngược lại. Thông qua việc tích hợp, các cơ sở dữ liệu có thể hỗ trợ các ứng dụng web (web applications), SOA (Service Oriented Architecture) và các dịch vụ web (web services).

**16.2 Cơ sở dữ liệu XML**

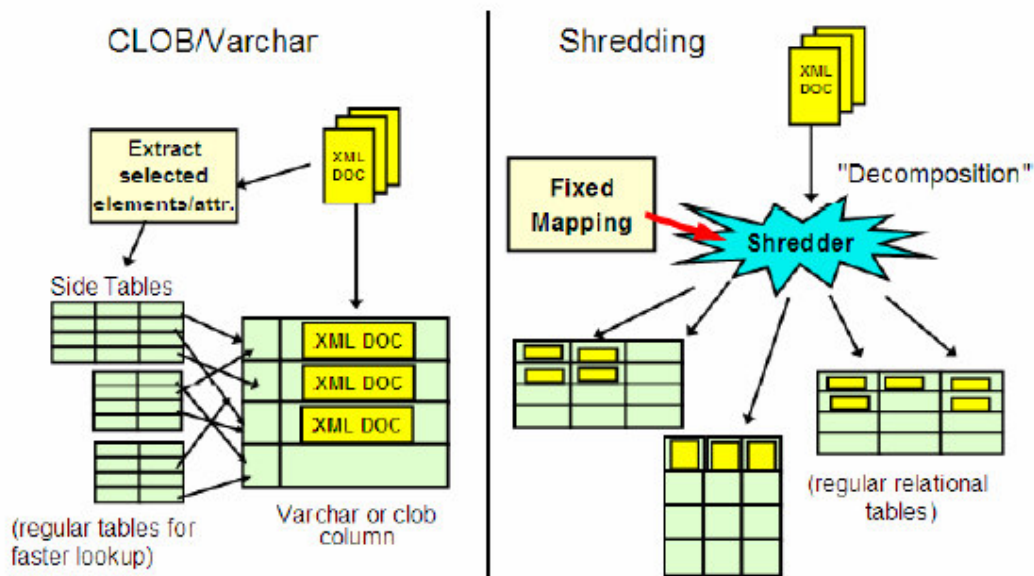
Có hai loại cơ sở dữ liệu để lưu trữ dữ liệu, hồ sơ XML:

Cơ sở dữ liệu hỗ trợ XML

Cơ sở dữ liệu thuần XML

**16.2.1 cơ sở dữ liệu hỗ trợ XML**

Một cơ sở dữ liệu hỗ trợ XML sẽ sử dụng mô hình quan hệ là mô hình chính để lưu trữ dữ liệu. Điều này cần sự ánh xạ giữa mô hình dữ liệu XML và mô hình dữ liệu quan hệ, hoặc là lưu trữ dữ liệu XML như là đối tượng lớn có thuộc tính. Trong khi xem xét việc này như là công nghệ cũ thì nhiều nhà cung cấp cơ sở dữ liệu vẫn sử dụng cách như trên. Hình 16.2 giải thích chi tiết hơn hai lựa chọn cho cơ sở dữ liệu hỗ trợ XML.



**Hình 16.2 – Hai lựa chọn để lưu trữ dữ liệu XML trong cơ sở dữ liệu hỗ trợ XML**

Bên trái hình 16.2 cho thấy phương thức lưu trữ tài liệu XML trong cơ sở dữ liệu “CLOB và Varchar”. Tài liệu XML dùng phương thức này được lưu bằng cả ảnh CLOB và cột kiểu Varchar trong cơ sở dữ liệu. Phương thức này không linh hoạt bởi vì khó tìm được các phần tử bên trong ảnh. Hiệu năng của nó cũng không cao vì các ảnh XML thường rất lớn do đó tốn bộ nhớ. Nhiều máy chủ dữ liệu, bao gồm máy chủ DB2, không tải hình ảnh vào bộ nhớ, mà xử lý trực tiếp trên đĩa. Sử dụng hình ảnh hiển nhiên sẽ không đạt hiệu năng tốt.

Lựa chọn còn lại cho cơ sở dữ liệu hỗ trợ XML là phương thức phân rã, phương thức này được minh họa bên phải của hình 16.2. Toàn bộ tài liệu XML dùng phương thức này bị phân rã thành nhiều phần (phần tử) rồi lưu trong các bảng. Sử dụng phương pháp này mô hình phân cấp của tài liệu XML bị chuyển thành mô hình quan hệ. Phương pháp này cũng không linh hoạt: một thay đổi trong tài liệu XML không dễ gì lan truyền đến các bảng tương ứng và có thể phải tạo nhiều bảng khác khi cần thiết. Phương thức này cũng không hiệu quả: nếu bạn cần lấy lại tài liệu XML ban đầu, bạn phải thực hiện các tác vụ SQL với chi phí cao, việc này còn tốn chi phí khi số lượng bảng được kết nối càng nhiều.

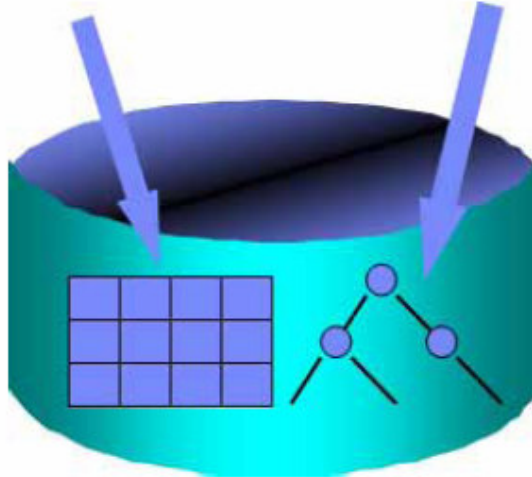
### 16.2.2 Cơ sở dữ liệu thuần XML

Cơ sở dữ liệu thuần XML sử dụng mô hình phân cấp của dữ liệu XML để lưu trữ và xử lý XML bên trong. Định dạng lưu trữ thì giống như định dạng xử lý: không có sự ánh xạ đến mô hình quan hệ, và không lưu các tài liệu XML thành dạng ảnh. Khi sử dụng các câu lệnh XPath hay XQuery, một cơ chế (engine) sẽ xử lý các câu lệnh đó chứ không chuyển sang SQL. Đây là nguyên nhân tại sao gọi các cơ sở dữ liệu này là cơ sở dữ liệu **thuần XML**. Hiện tại, DB2 9 là máy chủ thương mại duy nhất cung cấp tính năng này.

### 16.3 XML trong DB2

Hình 16.3 bên dưới phác họa cách thức lưu trữ dữ liệu quan hệ và dữ liệu phân cấp (các tài liệu XML) trong DB2 phiên bản 9. Trong hình, giả sử định nghĩa bảng *dept* như sau :

```
CREATE TABLE dept (deptID char(8) ,..., deptdoc XML);
```

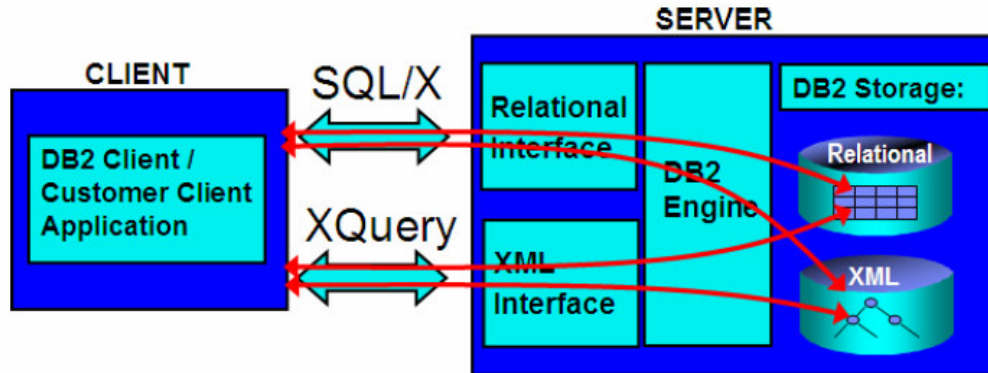


Hình 16.3 – XML trong DB2

Chú ý rằng khi định nghĩa bảng *dept* có sử dụng kiểu dữ liệu mới XML cho cột *deptdoc*. Mũi tên bên trái của hình trên cho thấy cột *deptID* được lưu trữ bằng định dạng quan hệ (dạng bảng), trong khi cột *deptdoc* được lưu bằng định dạng phân cấp đã được phân rã. Hình 16.4 minh họa bốn cách truy xuất dữ liệu trong DB2 phiên bản 9:

- Sử dụng SQL truy xuất dữ liệu quan hệ.
- Sử dụng SQL với XML mở rộng để truy xuất dữ liệu XML
- Sử dụng XQuery để truy xuất dữ liệu XML
- Sử dụng XQuery để truy xuất dữ liệu quan hệ





Hình 16.4 – Bốn cách truy xuất dữ liệu trong DB2

Vì vậy, trong khi một người sử dụng SQL xem DB2 như là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS – Relational Database Management System) cấp cao cũng có hỗ trợ XML, còn một người sử dụng XML thì sẽ xem DB2 như là kho trữ XML cấp cao có hỗ trợ SQL.

Chú ý rằng IBM dùng thuật ngữ **pureXML** thay vì “native XML” để mô tả công nghệ này. Trong khi các hãng khác vẫn dùng công nghệ cũ CLOB/Varchar hay phân rã để lưu trữ các tài liệu XML, họ gọi những công nghệ cũ đó là “native XML”. IBM quyết định dùng thuật ngữ mới **pureXML** để tránh sự nhầm lẫn và đặt thương hiệu tên này để các hãng khác về cơ sở dữ liệu hay XML không sử dụng trùng thuật ngữ biểu thị cho các công nghệ khác. Chú ý, sự hỗ trợ **pureXML** chỉ được cung cấp khi tạo ra cơ sở dữ liệu Unicode.

### 16.3.1 Ưu điểm của công nghệ pureXML trong DB2 9

Sử dụng công nghệ pureXML đem lại nhiều ưu điểm sau :

1. Bạn có thể nâng cao khả năng dùng dữ liệu quan hệ, bằng cách lưu trữ các tài liệu XML trong các cột của bảng có sử dụng kiểu dữ liệu mới XML.
2. Bạn có thể giảm độ phức tạp của mã. Ví dụ, trong hình 16.5 minh họa một đoạn mã PHP được viết vừa sử dụng và không sử dụng pureXML. Sử dụng pureXML (hộp màu xanh) giúp giảm các dòng mã lệnh. Điều này không chỉ có nghĩa là mã lệnh ít phức tạp mà hiệu năng toàn thể được cải thiện vì số dòng để phân tích và bảo trì trong mã lệnh ít hơn.



```

<?php
$conn = db2_connect($dbname, $dbuser, $dbpass);

/* Insert Customer Documents */

$stmt = db2_prepare($conn, "VALUES (NEXT VALUE FOR
Cid)");
db2_execute($stmt);
list($Cid) = db2_fetch_array($stmt);

$fileContents = file_get_contents
("customers/c1.xml");

$stmt = db2_prepare($conn, "INSERT INTO xmlcustomer
(Cid, Info) VALUES (?, ?)");
if(!db2_execute($stmt, array($Cid, $fileContents)))

echo db2_stmt_errormsg($stmt);
}

/* Insert Product Documents */

$fileContents = file_get_contents
("products/p1.xml");
$dom = simplexml_load_string($fileContents);

$prodID = (string) $dom["pid"];

$stmt = db2_prepare($conn, "INSERT INTO xmlproduct
(pid, Description) VALUES (?, ?)");
if(!db2_execute($stmt, array($prodID,

```

```

db2_execute($stmt);
list($Cid) = db2_fetch_array($stmt);

$fileContents = file_get_contents
("customers/c1.xml");
$dom = simplexml_load_string($fileContents);

$custName = (string) $dom->name;
$custCountry = (string) $dom->addr["country"];
$custStreet = (string) $dom->addr->street;
$custCity = (string) $dom->addr->city;
$custProvince = (string) $dom->addr->("prov-state");
$custZip = (string) $dom->addr->("code-zip");
$custPhone = (string) $dom->phone;

$stmt = db2_prepare($conn, "INSERT INTO sqlcustomer
(Cid, Name, Country, Street, City, Province, Zip,
Phone, Info) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
if(!db2_execute($stmt, array($Cid, $custName,
$custCountry, $custStreet, $custCity, $custProvince,
$custZip, $custPhone, $fileContents) )) {
echo db2_stmt_errormsg($stmt);
}

/* Insert Product Documents */

$fileContents = file_get_contents
("products/p1.xml");
$dom = simplexml_load_string($fileContents);

$prodID = (string) $dom["pid"];

```

Hình 16.5 - Độ phức tạp khi sử dụng hay không sử dụng pureXML

- Sử dụng XML và công nghệ pureXML thay đổi lược đồ (schema) của bạn dễ dàng hơn. Hình 16.6 minh họa một ví dụ về sự tăng tính linh động. Trong hình, giả sử rằng bạn đã có cơ sở dữ liệu gồm các bảng *Employee* và *Department*. Mặc định là cơ sở dữ liệu không có XML, nếu nhà quản lý của bạn yêu cầu bạn lưu trữ không chỉ mỗi nhân viên một số điện thoại (số điện thoại nhà) mà còn lưu thêm số thứ hai (số di động) thì bạn có thể thêm một cột trong bảng *Employee* và lưu số điện thoại di động trong cột mới đó. Tuy nhiên, cách làm này không đúng với các luật chuẩn hóa trong cơ sở dữ liệu quan hệ. Nếu bạn muốn giữ đúng các luật này thì bạn tạo một bảng *Phone* mới và chuyển toàn bộ thông tin số điện thoại qua bảng này. Bạn cũng có thể thêm nhiều số điện thoại di động. Tạo ra bảng *Phone* mới thì tốn chi phí, bởi vì không những lượng dữ liệu đã có cần được chuyển lớn mà còn tất cả các câu SQL trong các ứng dụng sẽ phải thay đổi điểm đến là bảng mới.

Thay vào đó, chúng tôi đưa ra cách thực hiện việc này sử dụng XML ở bên trái hình bên dưới. Nếu nhân viên “Christine” cũng có một số điện thoại di động, thì thêm một thẻ (tag) mới để lưu thông tin này. Nếu nhân viên “Michael” không có số điện thoại di động thì để trống.

```

<DEPARTMENT deptid="15" deptname="Sales">
  <EMPLOYEE>
    <EMPNO>10</EMPNO>
    <FIRSTNAME>CHRISTINE</FIRSTNAME>
    <LASTNAME>SMITH</LASTNAME>
    <PHONE>406-463-4963</PHONE>
    <PHONE>415-010-1234</PHONE>
    <SALARY>52750.00</SALARY>
  </EMPLOYEE>
  <EMPLOYEE>
    <EMPNO>27</EMPNO>
    <FIRSTNAME>MICHAEL</FIRSTNAME>
    <LASTNAME>THOMPSON</LASTNAME>
    <PHONE>406-463-1234</PHONE>
    <SALARY>41250.00</SALARY>
  </EMPLOYEE>
</DEPARTMENT>
    
```

**Requires:**

- Normalization of existing data !
- Modification of the mapping
- Change of applications

EMPNO	PHONE
27	406-463-1234
10	415-010-1234
10	408-463-4963

DEPTID	DEPTNAME
15	Sales

DEPTID	EMPNO	FIRSTNAME	LASTNAME	PHONE	SALARY
15	27	MICHAEL	THOMPSON	406-463-1234	41250
15	10	CHRISTINE	SMITH	408-463-4963	52750

Costly!

Hình 16.6 – Tăng sự mềm dẻo khi sử dụng XML

4. Bạn có thể cải thiện hiệu năng ứng dụng XML của mình. Thực hiện các kiểm tra dùng công nghệ pureXML cho thấy sự cải thiện đáng kể cho nhiều ứng dụng XML. Hình 16.7 cho thấy các kết quả đánh giá cho việc một công ty chuyển sang sử dụng công nghệ pureXML. Cột ở giữa cho thấy kết quả làm việc với XML bằng phương thức cũ, và cột thứ ba là kết quả sử dụng pureXML trong DB2 9.

Task	Other relational DB	DB2 9
Development of search and retrieval business processes	CLOB: 8 hrs Shred: 2 hrs	30 min.
Relative lines of I/O code	100	35 (65% reduction)
Add field to schema	1 week	5 min.
Queries	24 - 36 hrs	20 sec - 10 min

Hình 16.7 – Sự tăng hiệu năng khi sử dụng công nghệ pureXML

### 16.3.2 Khái niệm cơ bản về XPath

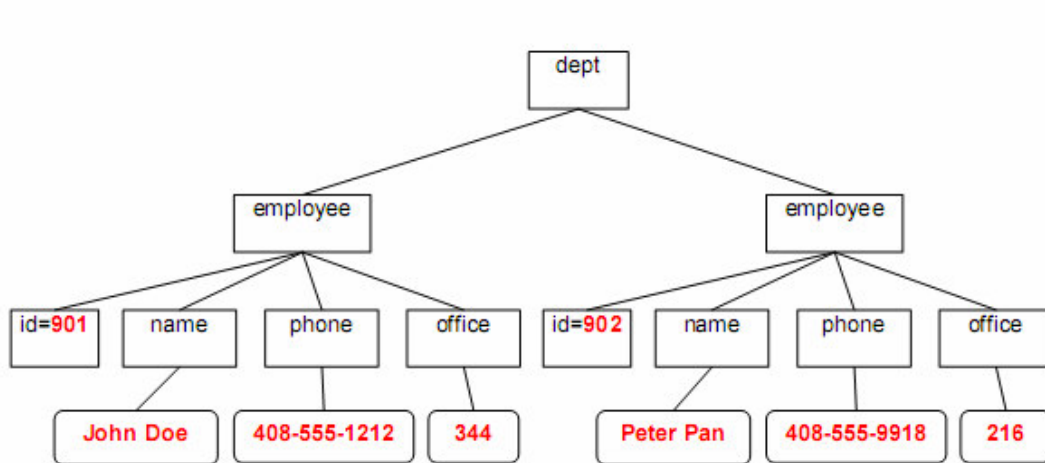
XPath là một ngôn ngữ mà ta có thể dùng nó để truy vấn các tài liệu XML. Hình 16.8 mô tả một tài liệu XML và hình 16.9 cũng minh họa tài liệu đó được trình bày với dạng "phân tích phân cấp" (cũng được gọi là nút (node) hay lá (leaf)). Chúng ta sẽ xử dụng định dạng "phân tích phân cấp" để giải thích XPath hoạt động như thế nào.

```

<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>

```

Hình 16.8 - Một tài liệu XML



Hình 16.9 - Trình bày tài liệu XML ở hình 16.8 theo dạng “phân tích phân cấp”

Một cách nhanh chóng để học XPath đó là so sánh nó với câu lệnh đổi thư mục (CD) trong MS-DOS hay trong Linux/UNIX. Sử dụng câu lệnh CD bạn đi qua một cây thư mục như sau:

```
cd /directory1/directory2/....
```

Tương tự, trong XPath bạn sử dụng những dấu gạch chéo (/) để chuyển từ một thành phần này đến một thành phần khác trong tài liệu XML. Ví dụ, bằng cách sử dụng tài liệu trong hình 16.9 với XPath bạn có thể lấy ra tên của tất cả các nhân viên thông qua câu truy vấn sau:

```
/dept/employee/name
```

**Các biểu thức XPath**

Các biểu thức XPath sử dụng đường dẫn đầy đủ để định ra các thành phần và/hay các thuộc tính. Kí hiệu "@" dùng để chỉ thuộc tính. Để chỉ lấy giá trị (text node) của một thành phần, ta sử dụng hàm "text()". Bảng 16.1 minh họa một vài câu truy vấn XPath và kết quả tương ứng khi áp dụng trên tài liệu XML đã cho ở hình 16.9

XPath	Kết quả
/dept/@bldg	101
/dept/employee/@id	901 902
/dept/employee/name	<name>Peter Pan</name> <name>John Doe</name>
/dept/employee/name/text()	Peter Pan John Doe

**Bảng 16.1 – Các ví dụ về biểu thức XPath**

**Các kí tự đại diện**

Có 2 kí tự đại diện chính trong XPath:

- "\*" thay thế cho bất kì tên thẻ nào
- "/" là kí tự đại diện cho "từ đây trở xuống" (descendent-or-self)

Bảng 16.2 cho ta thêm vài ví dụ dựa trên tài liệu XML ở hình 16.9

XPath	Kết quả
/dept/employee/*/text()	John Doe 408 555 1212 344 Peter Pan 408 555 9918 216
/dept/employee/@id	901 902
/dept/employee/name	<name>Peter Pan</name> <name>John Doe</name>
/dept/employee/name/text()	Peter Pan John Doe
/dept//phone	<phone>408 555 1212</phone> <phone>408 555 9918</phone>

**Bảng 16.2 - Các ví dụ về kí tự đại diện trong XPath**

**Các vị từ của XPath**

Các vị từ của XPath được đặt trong cặp dấu ngoặc vuông []. Một cách tương tự, bạn có thể nghĩ chúng như là một thành phần tương đương với mệnh đề WHERE trong SQL. Ví dụ [@id="902"] có thể được hiểu là "với điều kiện thuộc tính id có giá trị bằng 902". Trong một biểu thức XPath có thể có nhiều vị từ. Để chỉ ra một vị từ có vị trí, ta sử dụng [n] với ý nghĩa là thành phần con thứ n sẽ được chọn. Ví dụ employee[2] nghĩa là người công nhân thứ 2 sẽ được chọn. Bảng 16.3 sẽ cho ta nhiều ví dụ hơn:

XPath	Kết quả
/dept/employee[@id="902"]/name	<name>Peter Pan</name>
/dept[@bldg="101"]/employee[office > "300"]/name	<name>John Doe</name>
//employee[office="304" OR office="216"]/@id	901 902
/dept/employee[2]/@id	902

**Bảng 16.3 - Các ví dụ về vị từ XPath**

**Trục cha XPath:**

Tương tự với MS-DOS hay Linux/UNIX, bạn có thể dùng một dấu chấm (.) để định ra trong biểu thức rằng bạn tham chiếu đến vị trí hiện tại, và hai dấu chấm (..) để tham chiếu đến vị trí trên nó một cấp (mức cha).

Bảng 16.4 cho ta thêm vài ví dụ

XPath	Kết quả
/dept/employee/name[../@id="902"]	<name>Peter Pan</name>
/dept/employee/office[.>"300"]	<office>344</office>
/dept/employee[office > "300"]/office	<office>344</office>
/dept/employee[name = "John Doe"]/../../@bldg	101
/dept/employee/name[.="John Doe"]/../../@bldg	101

**Bảng 16.4 – Trục cha Xpath**

**16.3.3 Định nghĩa XQuery**

XQuery là một ngôn ngữ truy vấn được tạo ra cho XML. XQuery hỗ trợ các biểu thức đường dẫn để định vị trí trong cấu trúc phân cấp XML. Thực tế XPath là một tập con của XQuery, do đó, tất cả những gì chúng ta đã học trước đây về XPath cũng được áp dụng với XQuery. XQuery hỗ trợ cả dữ liệu có kiểu và dữ liệu không có kiểu. XQuery không chứa giá trị null vì tài liệu XML bỏ đi những dữ liệu thiếu hay không xác định. XQuery trả về một chuỗi các dữ liệu XML.

Một chú ý quan trọng là các biểu thức XQuery và XPath phân biệt ký tự viết hoa và viết thường.

XQuery hỗ trợ biểu thức FLWOR (đọc như flower). Trong SQL nó sẽ tương ứng với biểu thức SELECT-FROM-WHERE. Phần tiếp theo sẽ giải thích rõ hơn về FLWOR

**XQuery: Biểu thức FLWOR**

FLWOR viết tắt cho:

- FOR: lặp qua tất cả các phần tử của một dãy các thành phần và gán một biến cho các thành phần.
- LET: Gán một biến với một dãy.
- WHERE: loại ra các thành phần trong quá trình lặp
- ORDER: sắp xếp lại các thành phần trong khi lặp
- RETURN: tạo ra giá trị trả về cho câu truy vấn.

Đây là một biểu thức cho phép ta thao tác với các tài liệu XML, cho phép bạn trả về những biểu thức khác. Ví dụ, giả sử bạn đã có 1 bảng với định nghĩa như sau:

```
CREATE TABLE dept (deptID CHAR(8), deptdoc XML);
```

Và tài liệu XML sau được thêm vào trong các cột của deptdoc:

```
<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
```

```

        <phone>408 555 1212</phone>
        <office>344</office>
    </employee>
    <employee id="902">
        <name>Peter Pan</name>
        <phone>408 555 9918</phone>
        <office>216</office>
    </employee>
</dept>

```

Thì khi đó câu lệnh XQuery sau sử dụng FLWOR có thể được thực hiện:

```

xquery
for $d in db2-fn:xmlcolumn('dept.deptdoc')/dept
let $emp := $d//employee/name
where $d/@bldg > 95
order by $d/@bldg
return
    <EmpList>
    { $d/@bldg, @emp }
    </EmpList>

```

Và sẽ cho ta kết quả sau:

```

<EmpList bldg="101">
  <name>
    John Doe
  </name>
  <name>
    Peter Pan
  </name>
</EmpList>

```

### 16.3.4 Chèn các tài liệu XML

Việc chèn các tài liệu XML vào cơ sở dữ liệu DB2 có thể thực hiện bằng câu lệnh INSERT SQL, hay công cụ IMPORT. Không thể dùng XQuery để chèn tài liệu XML khi nó chưa được định nghĩa trong chuẩn.

Kiểm tra tệp kịch bản sau, nó có thể chạy trong DB2 Command Window hoặc Linux shell dùng lệnh này.

```
db2 -tvf table_creation.txt
```

#### **table\_creation.txt**

```

-- (1)
drop database mydb
;
-- (2)
create database mydb using codeset UTF-8 territory US
;
-- (3)
connect to mydb
;
-- (4)
create table items (

```

```

id      int primary key not null,
brandname  varchar(30),
itemname   varchar(30),
sku        int,
srp        decimal(7,2),
comments   xml
);
-- (5)
create table clients(
id      int primary key not null,
name    varchar(50),
status  varchar(10),
contactxml
);
-- (6)
insert into clients values (77, 'John Smith', 'Gold',
'<addr>111 Main St., Dallas, TX, 00112</addr>')
;
-- (7)
IMPORT FROM "D:\Raul\clients.del" of del xml from "D:\Raul" INSERT INTO CLIENTS
(ID, NAME, STATUS, CONTACT)
;
-- (8)
IMPORT FROM "D:\Raul\items.del" of del xml from "D:\Raul" INSERT INTO ITEMS (ID,
BRANDNAME, ITEMNAME, SKU, SRP, COMMENTS)

```

Lưu ý rằng tệp kịch bản này và những tệp liên quan được cung cấp trong tệp nén **expressc\_book\_quicklabs.zip** đi kèm với sách. Bên dưới chúng ta mô tả từng dòng lệnh trong tệp kịch bản.

1. Xóa cơ sở dữ liệu “mydb”. Thao tác này thường được thực hiện trong các tệp kịch bản để làm nhiệm vụ dọn dẹp. Nếu như trước đó chưa có “mydb”, bạn sẽ nhận được một thông báo lỗi, nhưng nó không gây ảnh hưởng đến việc ta đang làm.
2. Tạo cơ sở dữ liệu “mydb” sử dụng bộ mã UTF-8. Một cơ sở dữ liệu UNICODE được đòi hỏi hỗ trợ pureXML, vì vậy bước này là cần thiết để tạo một cơ sở dữ liệu thành một cơ sở dữ liệu UNICODE.
3. Kết nối đến cơ sở dữ liệu “mydb” vừa mới được tạo. Lệnh này cần thiết để tạo ra các đối tượng trong cơ sở dữ liệu.
4. Tạo bảng “items”. Để ý thấy cột cuối cùng trong bảng (cột “comments”) được định nghĩa như là một cột XML sử dụng kiểu dữ liệu XML mới.
5. Tạo bảng “clients”. Cột cuối trong bảng (cột “contact”) cũng được định nghĩa với kiểu dữ liệu XML mới.
6. Dùng lệnh SQL INSERT này, để chèn một tài liệu XML vào một cột XML. Trong câu lệnh INSERT bạn truyền tài liệu XML vào như là một chuỗi bao bởi dấu nháy đơn.
7. Dùng lệnh IMPORT, để chèn hoặc nhập nhiều tài liệu XML cùng với dữ liệu quan hệ vào cơ sở dữ liệu. Lệnh 7, bạn nhập dữ liệu từ tệp *clients.del* (một tệp ascii

được dấu phân cách), và bạn cũng có thể chỉ ra nơi mà dữ liệu XML được tham chiếu bởi tệp *clients.del* đó được định vị (trong ví dụ này là thư mục D:\Raul).

Chúng ta sẽ tìm hiểu kỹ hơn tệp *clients.del*, nhưng trước hết, hãy xem nội dung của thư mục D:\Raul (Hình 16.10)

Name	Size	Type
Client3227.xml	1 KB	XML Document
Client4309.xml	1 KB	XML Document
Client5681.xml	1 KB	XML Document
Client8877.xml	1 KB	XML Document
Client9077.xml	1 KB	XML Document
Client9177.xml	1 KB	XML Document
ClientInfo.xsd	2 KB	XML Schema
clients.del	1 KB	DEL File
Comment3926.xml	1 KB	XML Document
Comment4023.xml	1 KB	XML Document
Comment4272.xml	1 KB	XML Document
items.del	1 KB	DEL File

**Hình 16.10 – Nội dung của thư mục D:\Raul với các tài liệu XML**

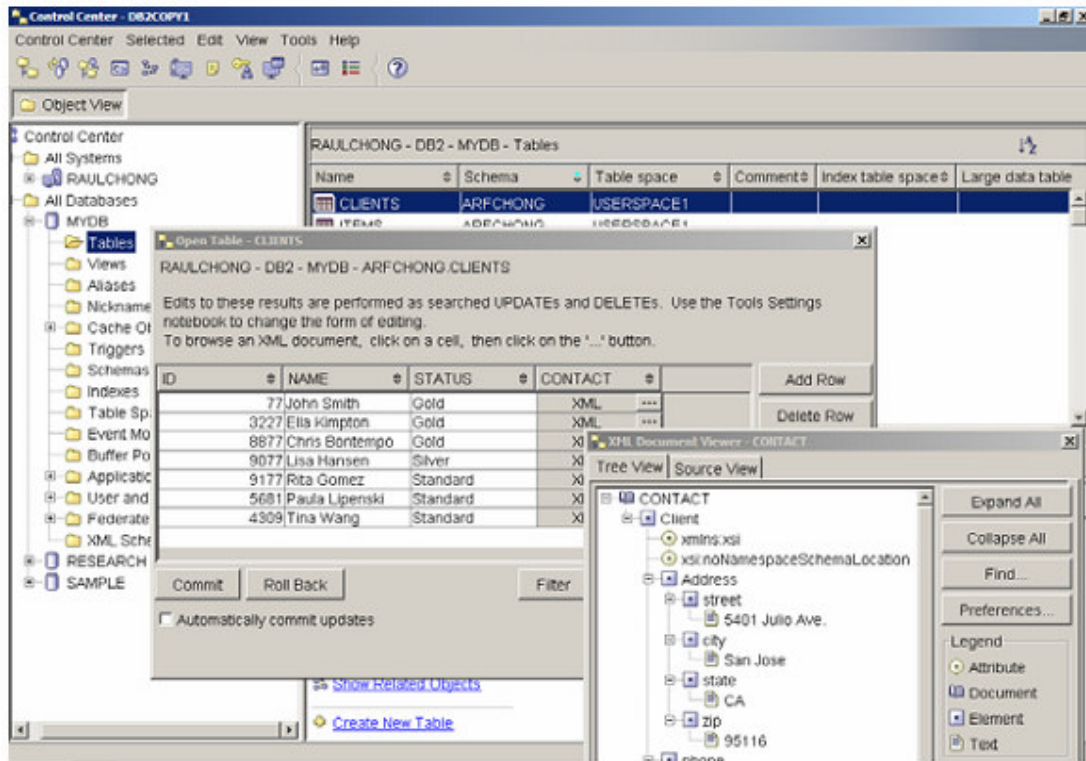
Đây là nội dung của tệp *clients.del*

**clients.del**

```
3227,Ella Kimpton,Gold,<XDS FIL='Client3227.xml' />,
8877,Chris Bontempo,Gold,<XDS FIL='Client8877.xml' />,
9077,Lisa Hansen,Silver,<XDS FIL='Client9077.xml' />,
9177,Rita Gomez,Standard,<XDS FIL='Client9177.xml' />,
5681,Paula Lipenski,Standard,<XDS FIL='Client5681.xml' />,
4309,Tina Wang,Standard,<XDS FIL='Client4309.xml' />
```

Trong tệp *clients.del*, “XDS FIL=” được dùng để trỏ đến một tệp tài liệu XML cụ thể. Hình 16.11 cho thấy trung tâm điều khiển sau khi chạy tệp kịch bản ở trên.





Hình 16.11 – Trung tâm điều khiển sau khi thực hiện *table\_creation.txt*

Trong hình là nội dung của bảng CLIENTS. Cột cuối cùng “Contact” là một cột XML. Khi bạn nhấp vào nút có ba dấu chấm, một cửa sổ khác mở ra cho bạn thấy nội dung tài liệu XML. Cửa sổ này nằm ở góc dưới bên phải của hình 16.11

### 16.3.5 Truy vấn dữ liệu XML

Trong DB2, có 2 cách để truy vấn dữ liệu XML:

- Sử dụng SQL với các lệnh mở rộng cho XML (SQL/XML)
- Sử dụng XQuery

Trong cả 2 cách trên, DB2 đều tuân theo chuẩn XML quốc tế.

#### Truy vấn dữ liệu XML với SQL/XML

Sử dụng những câu lệnh SQL thuần túy cho phép bạn làm việc với cả dòng và cột. Một câu lệnh SQL làm việc được với nguyên cả tài liệu XML; tuy nhiên nó lại không có khả năng làm việc với chỉ một phần của tài liệu XML. Trong trường hợp này, bạn cần sử dụng SQL với các lệnh mở rộng cho XML (SQL/XML).

Bảng 16.5 bên dưới mô tả một số hàm SQL/XML có sẵn với chuẩn SQL 2006

Tên hàm	Mô tả
XMLPARSE	Tách (phân tích) thuộc tính hoặc dữ liệu nhị phân trong 1 đối tượng lớn, lấy ra giá trị XML
XMLSERIALIZE	Chuyển giá trị XML sang thuộc tính hoặc dữ liệu nhị phân trong 1 đối tượng lớn
XMLVALIDATE	Đánh giá giá trị của XML với sơ đồ XML (XML Schema) và ghi chú kiểu các giá trị của XML
XMLEXISTS	Xác định (kiểm tra) câu truy vấn XQuery có giá trị trả về không (ví dụ một dãy giá trị hay nhiều hơn)

XMLQUERY	Thực thi câu truy vấn XQuery và trả về dãy kết quả
XMLTABLE	Thực thi câu truy vấn XQuery và kết quả trả về (nếu có) sẽ là một bảng quan hệ
XMLCAST	Ép kiểu XML

**Bảng 16.5 - Các hàm SQL/XML**

Những ví dụ minh họa sau sử dụng cơ sở dữ liệu “mydb” mà ta đã tạo ra từ trước.

Ví dụ 1:

Giả sử bạn cần xác định tên tất cả khách hàng sống ở một vùng nào đó (có cùng mã vùng – zip). Bảng “clients” lưu địa chỉ khách hàng, bao gồm mã vùng, trong một cột XML. Sử dụng XMLEXISTS, bạn tìm thấy cột dữ liệu XML phù hợp với mã vùng, sau đó giới hạn lại kết quả trả về.

```
SELECT name FROM clients
WHERE xmlexists (
    '$c/Client/Address[zip="95116"]'
    passing clients.contact as "c"
)
```

Dòng đầu tiên của câu lệnh SQL nói rằng bạn muốn lấy thông tin trong cột “name” của bảng “clients”.

Mệnh đề WHERE gọi hàm XMLEXISTS, chỉ định rõ biểu thức XPath yêu cầu DB2 tìm phần tử “zip” có giá trị 95116.

Mệnh đề “\$c/Client/Address” xác định đường dẫn bên trong cây cấu trúc của tài liệu XML để DB2 có thể tìm ra phần tử “zip”. Ký tự \$ được dùng để khai báo “c” là một biến. Biến “c” này sau đó được định nghĩa bằng câu lệnh *passing clients.contact as “c”*. Trong đó “clients” là tên bảng và “contact” là tên cột có kiểu dữ liệu XML. Nói cách khác, chúng ta đang đưa tài liệu XML vào biến “c”.

DB2 kiểm tra dữ liệu XML trong cột “contact”, duyệt từ nút “Client” xuống nút “Address”, cuối cùng là nút “zip” và xác định xem khách hàng có sống ở vùng có mã vùng đó không. Nếu thấy, hàm XMLEXISTS trả về “true” và DB2 lấy ra tên khách hàng tương ứng.

Ví dụ 2:

Chúng ta suy nghĩ xem làm cách nào tạo ra được mẫu báo cáo gồm danh sách địa chỉ thư điện tử của những khách hàng “Vàng” (có trạng thái “Gold”). Câu truy vấn dưới đây có thể giải quyết vấn đề này:

```
SELECT xmlquery('$c/Client/email' passing contact as "c")
FROM clients
WHERE status = "Gold"
```

Dòng đầu tiên nói rằng ta muốn lấy ra phần tử là địa chỉ thư điện tử của tài liệu XML chứ không muốn lấy ra cột dữ liệu. Tương tự ví dụ trước, tài liệu XML được lưu trong biến “\$c”. Trong ví dụ này, hàm XMLQUERY có thể được sử dụng sau mệnh đề SELECT, trong khi hàm XMLEXISTS được sử dụng sau mệnh đề WHERE.

Ví dụ 3:

Có trường hợp bạn muốn trình bày dữ liệu XML theo dạng bảng, ta có thể làm được điều này bằng cách sử dụng hàm XMLTABLE.

```

SELECT t.comment#, i.itemname, t.customerID, Message
FROM items I,
xmltable('$c/Comments/Comment' passing i.comments as "c"
columns Comment# integer path 'CommentID',
CustomerID integer path 'CustomerID',
Message varchar(100) path 'Message') AS t

```

Dòng đầu tiên cho biết những cột nào sẽ xuất hiện trong tập hợp kết quả (results set). Những cột bắt đầu bằng "t" là cơ sở trên các giá trị của phần tử XML.

Dòng tiếp theo gọi hàm XMLTABLE để xác định rõ cột XML của DB2 sẽ chứa dữ liệu mà ta cần ("i.comments") và đường dẫn đến phần tử đó trong tài liệu XML, là những phần tử mong muốn.

Từ dòng 4 đến 6 là mệnh đề "columns", xác định những phần tử XML cụ thể nào sẽ được ánh xạ tương ứng với những cột trong bảng kết quả của câu lệnh SQL ở dòng 1. Phần ánh xạ này có sự tham gia của kiểu dữ liệu mà các giá trị của các phần tử XML sẽ chuyển đổi thành. Trong ví dụ này, tất cả dữ liệu XML đều được đổi thành kiểu dữ liệu vốn có của SQL.

#### Ví dụ 4:

Bây giờ ta sẽ thực hiện một ví dụ đơn giản có sử dụng biểu thức XQuery FLWOR bên trong hàm XMLQUERY SQL/XML.

```

SELECT name, xmlquery(
    'for $e in $c/Client/email[1] return $e'
    passing contact as "c"
)
FROM clients
WHERE status = 'Gold'

```

Dòng thứ nhất xác định rằng bảng kết quả của câu truy vấn bao gồm tên khách hàng và kết quả trả về từ hàm XMLQUERY. Dòng thứ hai cho biết phần tử "email" đầu tiên của phần tử "Client" sẽ được trả về. Dòng thứ ba chỉ nguồn dữ liệu XML (cột "contact"). Dòng thứ tư cho ta biết cột này ở bảng "client"; và dòng năm chỉ ra khách hàng "vàng" là quan tâm của chúng ta.

#### Ví dụ 5:

Ví dụ này minh họa lại hàm XMLQUERY có sử dụng biểu thức XQuery FLWOR, nhưng lần này ta không chỉ trả về dữ liệu XML mà còn cả HTML.

```

SELECT xmlquery ('for $e in $c/Client/email[1]/text()
return <p>{$e}</p>'
passing contact as "c"
)
FROM clients
WHERE status = 'Gold'

```

Mệnh đề `return` của XQuery cho phép biến đổi dữ liệu XML đầu ra theo yêu cầu. Sử dụng hàm `text()` ở dòng đầu tiên chỉ ra rằng ta chỉ quan tâm đến biểu diễn văn bản địa chỉ e-mail đầu tiên của nhóm khách thỏa mãn yêu cầu. Dòng thứ hai cho biết thông tin này sẽ được bọc lại trong các thẻ HTML (trong trường hợp này là thẻ `<p>` và `</p>`).

#### Ví dụ 6:

Ví dụ dưới đây minh họa cách sử dụng hàm XMLELEMENT để tạo ra một dãy các phần

tử, mỗi phần tử này lại chứa trong nó các phần tử con (sub-element) có nội dung về ID, tên hàng hóa và chỉ số hàng hóa tối thiểu trong kho (SKU) tương ứng với các cột trong bảng "items". Về cơ bản, ta có thể dùng hàm XMLEMENT khi muốn chuyển từ dữ liệu quan hệ sang dữ liệu XML.

```
SELECT
  xmlelement (name "item", itemname),
  xmlelement (name "id", id),
  xmlelement (name "brand", brandname),
  xmlelement (name "sku", sku)
FROM items
WHERE srp < 100
```

Kết quả xuất ra của câu truy vấn trên như sau:

```
<item>
  <id>4272</id>
  <brand>Classy</brand>
  <sku>981140</sku>
</item>
...
<item>
  <id>1193</id>
  <brand>Natural</brand>
  <sku>557813</sku>
</item>
```

### Truy vấn dữ liệu XML với XQuery

Trong phần trước, ta đã thấy được cách truy vấn dữ liệu XML với câu SQL có phần mở rộng cho XML. Trường hợp này, SQL là phương pháp truy vấn chính, và XPath được nhúng vào bên trong câu lệnh SQL. Còn trong tiếp theo này, ta nói về cách sử dụng XQuery để truy vấn dữ liệu XML. Lúc này, XQuery là phương pháp chính, và trong một số trường hợp, ta sẽ nhúng SQL vào bên trong XQuery (bằng cách sử dụng hàm "db2-fn:sqlquery"). Khi thực hành với XQuery, ta sẽ gọi vài hàm, sử dụng cả biểu thức FLWOR.

#### Ví dụ 1:

Làm một truy vấn XQuery đơn giản là lấy dữ liệu liên lạc của khách hàng.

```
xquery db2-fn:xmlcolumn('CLIENTS.CONTACT')
```

Nhớ luôn luôn thêm lệnh "xquery" vào đầu biểu thức XQuery để cho DB2 biết rằng nó cần sử dụng bộ phân tích (parser) XQuery. Nếu không, DB2 sẽ cho rằng bạn đang cố gắng thực thi biểu thức SQL. Hàm db2-fn:xmlcolumn lấy cột ta chỉ định trong tài liệu XML làm tham số. Câu lệnh trên tương ứng với câu lệnh SQL sau đây:

```
SELECT contact FROM clients
```

#### Ví dụ 2:

Ta sử dụng biểu thức FLWOR để lấy thông tin về số fax của khách hàng.

```
xquery
  for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/fax
  return $y
```

Dòng đầu tiên gọi bộ phân tích XQuery. Dòng thứ hai yêu cầu DB2 lặp hết các phần tử con `fax` trong cột `CLIENTS.CONTACT`. Mỗi phần tử `fax` được lưu trong biến `$y`. Dòng cuối cùng nói rằng giá trị của biến `$y` sẽ được trả về trong mỗi lần lặp.

Kết quả xuất ra của câu truy vấn trông giống như sau (mặc định có thể gồm địa chỉ của không gian tên (namespace) nhưng ở đây tôi không thể hiện ra, mặt khác thông tin xuất có thể trải ra thành nhiều dòng nên hơi khó đọc):

```
<fax>4081112222</fax>
<fax>5559998888</fax>
```

### Ví dụ 3:

Ví dụ này tiếp tục truy vấn dữ liệu XML và trả về kết quả là HTML

```
xquery
  <ul> {
    for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/Address
      order by $y/zip
      return <li>{$y}</li>
  }
</ul>
```

Kết quả HTML trả về trông như sau:

```
<ul>
<li>
<address>
  <street>9407 Los Gatos Blvd.</street>
  <city>Los Gatos</city>
  <state>ca</state>
  <zip>95302</zip>
</address>
</li>
<li>
<address>
  <street>4209 El Camino Real</street>
  <city>Mountain View</city>
  <state>CA</state>
  <zip>95302</zip>
</address>
</li>
...
</ul>
```

### Ví dụ 4:

Tiếp theo là cách nhúng SQL vào trong XQuery bằng việc sử dụng hàm `db2-fn:sqlquery`. Hàm `db2-fn:sqlquery` thực thi câu truy vấn SQL và chỉ trả về những dữ liệu XML được chọn. Chỉ những câu truy vấn SQL trả về dữ liệu XML mới được truyền vào hàm `db2-fn:sqlquery`.

```
xquery
  for $y in
    db2-fn:sqlquery(
      'select comments from items where srp > 100'
    )/Comments/Comment
  where $y/ResponseRequested = 'Yes'
```

```

return (
  <action>
    {$y/ProductID
     $y/CustomerID
     $y/Message}
  </action>
)

```

Trong ví dụ trên, câu truy vấn SQL lọc những dòng dữ liệu thỏa điều kiện giá trị tại cột “srp” phải lớn hơn 100. Từ các dòng đã lọc này lấy ra cột “comments”, là cột dữ liệu XML. Câu lệnh XQuery (hoặc XPath) tiếp theo được áp dụng cho những phần tử con.

**Ghi chú:** DB2 không phân biệt chữ in hoa và chữ thường và giả định tất cả tên bảng và tên cột đều là chữ in hoa, trong khi đó XQuery lại phân biệt chữ in hoa và chữ thường. Các hàm trên là hàm giao diện của XQuery (XQuery interface function) do đó tên bảng và tên cột truyền vào các hàm này nên là chữ hoa. Truyền tên đối tượng ở dạng chữ thường có thể gây ra lỗi “tên đối tượng không xác định” (undefined object name)

### 16.3.6 Nối (join) với SQL/XML

Phần này sẽ mô tả cách thức thực hiện lệnh JOIN giữa hai cột XML của hai bảng khác nhau, hoặc giữa một cột XML và một cột quan hệ. Giả sử ta đã tạo hai bảng với các lệnh sau:

```

CREATE TABLE dept (unitID CHAR (8), deptdoc XML)

CREATE TABLE unit (unitID CHAR(8) primary key not null,
                   name CHAR(20),
                   manager VARCHAR(20),
                   . . .
)

```

Ta có thể thực hiện lệnh JOIN theo một trong hai cách

#### Phương pháp 1:

```

SELECT u.unitID
FROM dept d, unit u
WHERE XMLEXISTS(
  '$e//employee[name = $n]'
  passing d.deptdoc as "e", u.manager as "m")

```

Trên dòng 3 của lệnh này chỉ ra rằng lệnh JOIN xuất hiện giữa phần tử “name” là phần tử con của cột XML deptdoc của bảng “dept”, và cột quan hệ “manager” của bảng “unit”.

#### Phương pháp 2:

```

SELECT u.unitID
FROM dept d, unit u
WHERE u.manager = XMLCAST(
  XMLQUERY('$e//employee/name'
  passing d.deptdoc as "e")
  AS char(20))

```

Trong lựa chọn này, cột quan hệ thì ở bên trái của lệnh JOIN. Nếu cột có quan hệ ở bên trái dấu bằng, một chỉ mục quan hệ có thể được dùng thay cho một chỉ mục XML.

### 16.3.7 Nối (Join) với XQuery

Giả sử có các bảng sau được tạo ra:

```
CREATE TABLE dept(unitID CHAR(8), deptdoc XML)
CREATE TABLE project(projectdoc XML)
```

Nếu ta sử dụng SQL/XML, câu truy vấn sẽ như sau:

```
SELECT XMLQUERY (
    'd/de/p/employee' passing d.deptdoc as "d")
FROM dept d, project p
WHERE XMLEXISTS (
    'e/dept[@deptID=$p/project/deptID]'
    passing d.deptdoc as "e", p.project as "p")
```

Lệnh tương đương sử dụng XQuery sẽ là:

```
xquery
  for $dept in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept
  for $proj in db2-
fn:xmlcolumn("PROJECT.PROJECTDOC")/project
  where $dept/@deptID = $proj/deptID
  return $dept/employee
```

Phương pháp thứ 2 này dễ phiên dịch hơn – biến "\$dept" mang tài liệu XML của cột XML "deptdoc" của bảng "dept". Biến "\$proj" mang tài liệu XML của cột XML "projectdoc" của bảng "project". Như vậy dòng 4 thực hiện lệnh JOIN giữa một thuộc tính của tài liệu XML đầu tiên và một phần tử của tài liệu XML thứ hai.

### 16.3.8 Các lệnh cập nhật và xóa

Các lệnh cập nhật và xóa dữ liệu XML có thể được thực hiện bằng một trong hai cách:

- Sử dụng câu lệnh SQL UPDATE và DELETE
- Gọi thủ tục DB2XMLFUNCTION.XMLUPDATE

Trong cả hai trường hợp, cập nhật hoặc xóa tác động ở mức tài liệu; thật vậy, toàn bộ tài liệu XML được thay thế với một lần cập nhật. Ví dụ, nếu trong ví dụ dưới đây ta muốn thay thế phần tử <state>, toàn bộ tài liệu XML trên thực tế bị thay thế.

```
UPDATE clients SET contact=(
  xmlparse (document
    '<Client>
      <address>
        <street>5401 Julio ave.</street>
        <city>San Jose</city>
        <state>CA</state>
        <zip>95116</zip>
      </address>
      <phone>
        <work>4084633000</work>
        <home>4081111111</home>
        <cell>4082222222</cell>
      </phone>
```

```

    </Client>' )
)
WHERE id = 3227

```

### 16.3.9 Chỉ mục XML

Trong một tài liệu XML, các chỉ mục có thể được tạo ra theo các phần tử, thuộc tính hoặc giá trị (văn bản). Sau đây là một số ví dụ, giả sử bảng sau đã được tạo:

```
CREATE TABLE customer(info XML)
```

Và giả sử hồ sơ sau đã được lưu trong bảng này:

```

<customerinfo Cid="1004">
  <name>Matt Foreman</name>
  <addr country="Canada">
    <street>1596 Baseline</street>
    <city>Toronto</city>
    <state>Ontario</state>
    <zip>M3Z-5H9</zip>
  </addr>
  <phone type="work">905-555-4789</phone>
  <phone type="home">416-555-3376</phone>
  <assistant>
    <name>Peter Smith</name>
    <phone type="home">416-555-3426</phone>
  </assistant>
</customerinfo>

```

#### 1) Lệnh này tạo một chỉ mục trên thuộc tính "Cid"

```

CREATE UNIQUE INDEX idx1 ON customer(info)
GENERATE KEY USING
xmlpattern '/customerinfo/@Cid'
AS sql DOUBLE

```

#### 2) Lệnh này tạo một chỉ mục trên phần tử "name"

```

CREATE UNIQUE INDEX idx2 ON customer(info)
GENERATE KEY USING
xmlpattern '/customerinfo/name'
AS sql VARCHAR(40)

```

#### 3) Lệnh này tạo một chỉ mục trên mọi phần tử "name"

```

CREATE UNIQUE INDEX idx3 ON customer(info)
GENERATE KEY USING
xmlpattern '//name'
AS sql VARCHAR(40)

```

#### 4) Lệnh này tạo một chỉ mục trên toàn bộ văn bản (tất cả các giá trị). Đây là lệnh không nên dùng, nó quá tốn kém để duy trì chỉ mục khi thực hiện các lệnh cập nhật, xóa hoặc chèn, và chỉ mục sẽ rất lớn



```
CREATE UNIQUE INDEX idx4 ON customer(info)
GENERATE KEY USING
xmlpattern '//text()'
AS sql VARCHAR(40)
```

## Bài tập nhanh #12 – SQL/XML và XQuery

### Mục đích

Bạn đã thấy nhiều ví dụ về cú pháp của SQL/XML và XQuery và đã được giới thiệu trong DB2 Command Editor và công cụ DB2 Developer Workbench. Trong bài tập này, bạn hãy kiểm tra sự hiểu biết của bạn về SQL/XML và XQuery bằng việc thực hành với các công cụ này. Ta sẽ sử dụng cơ sở dữ liệu “mydb” được tạo ra bằng việc sử dụng tệp kịch bản **table\_creation.txt** mà chúng ta đã giải thích trong các chương trước.

### Thủ tục

1. Tạo cơ sở dữ liệu “mydb” và nạp dữ liệu XML, như đã trình bày ở chương trước
2. Sử dụng hoặc Cửa sổ lệnh hoặc DWB:
  - a. Tìm ra tất cả các nội dung chú thích (comments) trong tài liệu XML trong bảng ITEMS bằng hai cách, chỉ sử dụng XQuery
  - b. Tại sao khi ta sử dụng lệnh SQL sau lại không đưa ra cùng kết quả?

```
SELECT comments FROM items
```

- c. Tìm ra ID và BRANDNAME của các bản ghi mà tài liệu XML của nó có giá trị của trường ResponseRequested là “No”

### Lời giải

2a)

```
xquery db2-fn:xmlcolumn('ITEMS.COMMENTS')
xquery db2-fn:sqlquery("select comments from items")
```

2b)

Kết quả khác nhau là vì SQL trả lại giá trị NULL khi không có giá trị, còn XQuery không trả lại gì cả.

2c)

```
SELECT id, brandname FROM items WHERE
XMLEXISTS('$c/Comments/Comment [ResponseRequested="No"
passing ITEMS.COMMENTS as "c"
```

# 17

## Chương 17 – Phát triển với Java, PHP và Ruby

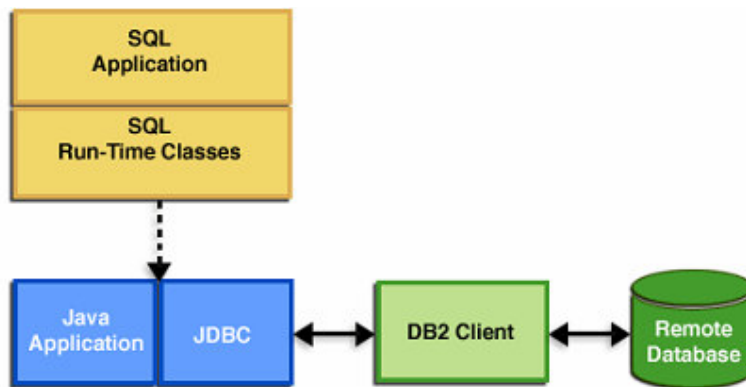
Chương này sẽ thảo luận những vấn đề cơ bản của việc phát triển ứng dụng trong Java, PHP và Ruby trên nền Rails sử dụng một máy chủ DB2. Mục đích của chương này không nhằm hướng dẫn về các ngôn ngữ trên, nhưng nó cung cấp thông tin thích hợp trong việc sử dụng chúng với DB2.

### 17.1 Phát triển ứng dụng bằng Java

Trình điều khiển IBM DB2 cho JDBC (được biết đến như trình điều khiển JCC) được xem là tối ưu đối với các máy chủ DB2 trên tất cả các nền tảng. Tập db2jcc.jar (com.ibm.db2.jcc) bao gồm trình điều khiển kiểu 2 và kiểu 4. Tập db2jcc.jar cũng bao gồm bất cứ máy khách DB2 nào, hoặc nó có thể được thấy (trình điều khiển IBM DB2 cho JDBC và SQLJ) từ trang web DB2 Express-C ([ibm.com/db2/express](http://ibm.com/db2/express))

#### 17.1.1 Trình điều khiển JDBC kiểu 2 (type 2)

Trình điều khiển JDBC kiểu 2 yêu cầu một máy khách DB2 đã được cài đặt nơi mà ứng dụng JDBC sẽ thực thi. Hình 17.1 minh họa một ứng dụng JDBC sử dụng trình điều khiển kiểu 2.



Hình 17.1 – Trình điều khiển JDBC kiểu 2

Hình 17.2 đưa ra một đoạn mã lệnh cho ta thấy làm thế nào để thiết lập một kết nối sử dụng trình điều khiển JDBC kiểu 2. Lưu ý là URL không bao gồm hostname hoặc thông tin về cổng bởi vì điều này được thực hiện từ máy khách DB2.

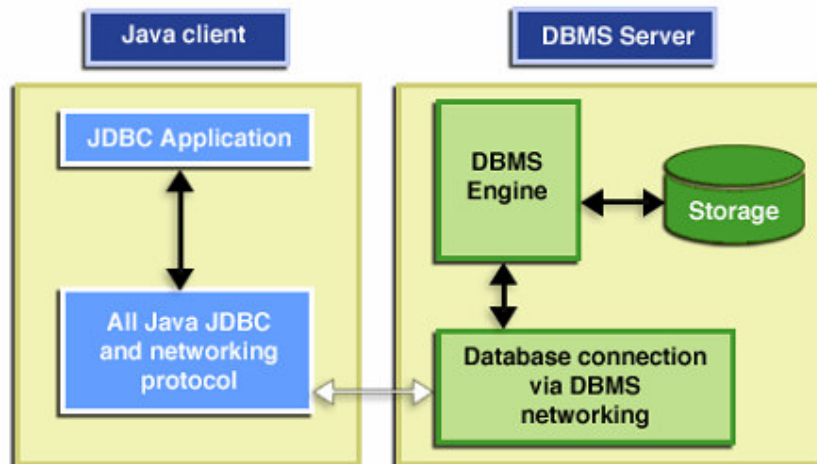
```

...
public static final String DB_URL = "jdbc:db2:sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url, connectProperties)
}
catch (Exception e)
throw e;
}
...
    
```

Hình 17.2 – Thiết lập một kết nối sử dụng trình điều khiển JDBC kiểu 2

17.1.2 Trình điều khiển JDBC kiểu 4

Trình điều khiển JDBC kiểu 4 không yêu cầu một máy khách phải kết nối đến một máy chủ DB2. Hình 17.3 minh họa một ứng dụng JDBC sử dụng trình điều khiển kiểu 4.



Hình 17.3 – Trình điều khiển JDBC kiểu 4

Hình 17.4 đưa ra một đoạn mã lệnh cho ta thấy làm thế nào để thiết lập một kết nối sử dụng trình điều khiển JDBC kiểu 4. Lưu ý là URL bao gồm cả hostname hoặc thông tin về cổng.

```

...
public static final String DB_URL = "jdbc:db2://server1:50000/sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url, connectProperties)
}
catch (Exception e)
throw e;
}
...
    
```

## Hình 17.4 – Thiết lập một kết nối sử dụng trình điều khiển JDBC kiểu 4

### 17.2 Phát triển ứng dụng bằng PHP

PHP (PHP Hypertext Preprocessor) là một nguồn mở, là ngôn ngữ kịch bản đọc lập về nền tảng được thiết kế cho việc phát triển ứng dụng Web. Đó là một trong những ngôn ngữ web được triển khai rộng rãi nhất trên thế giới hiện nay. Tính phổ biến của PHP dựa trên các đặc trưng của ngôn ngữ:

- Nhanh, dễ học từ thấp đến cao
- Mạnh, tính thực thi cao và khả chuyển
- Ổn định và an toàn
- Một sự lựa chọn đối với J2EE và .NET trên nền Web.
- Dễ dàng tích hợp vào các môi trường/hệ thống không thuần nhất.
- Được chứng tỏ thông qua triển khai rộng rãi
- Tạo nên một cộng đồng đầy sức sống

PHP là bộ phận của LAMP (viết tắt của các từ Linux, Apache HTTP Server, MySQL, PHP / Perl / Python). Đây là một nhóm công nghệ web mã nguồn mở, thường sẵn có trên các nhà cung cấp (ISP) với giá hợp lý.

#### 17.2.1 Lựa chọn kết nối DB2 cho PHP

IBM hỗ trợ truy xuất đến cơ sở dữ liệu DB2 từ các ứng dụng PHP thông qua hai phần mở rộng.

##### **ibm\_db2:**

Phần mở rộng `ibm_db2` đưa ra một giao diện lập trình thủ tục ứng dụng cho phép tạo, đọc, cập nhật và ghi các thao tác cơ sở dữ liệu thêm vào đó mở rộng truy xuất đến siêu cơ sở dữ liệu. Nó có thể được biên dịch để làm việc với PHP 4 hoặc PHP 5. Phần mở rộng sẵn có từ kho thư viện PECL dưới sự cho phép của Apache 2.0. Nó được IBM phát triển và hỗ trợ. Có đầy đủ các đặc tính hỗ trợ cho các thủ tục lưu trữ và LOBs, nhanh, và được tối ưu cho DB2.

##### **PDO\_ODBC:**

PDO\_ODBC là trình điều khiển cho phần mở rộng của Các đối tượng dữ liệu PHP (PHP Data Objects - PDO), và sẵn sàng truy cập đến cơ sở dữ liệu DB2 thông qua một giao diện cơ sở dữ liệu hướng đối tượng đã được thủ tục hóa trong PHP 5.1. Nó có thể tương thích ngược với các thư viện của DB2 một cách tự động. Nó cung cấp một giao diện truy xuất dữ liệu tiêu chuẩn cho PHP. Nhanh chóng, nhẹ nhàng, và hướng đối tượng. Phần mở rộng PDO\_ODBC sử dụng các thư viện DB2 để truy xuất một cách tự nhiên, và nó đã được xây dựng trong PHP 5.1. Có thể tham khảo thêm tại:

- <http://pecl.php.net/package/pdo>
- [http://pecl.php.net/package/PDO\\_ODBC](http://pecl.php.net/package/PDO_ODBC)

#### **Kết nối đến một cơ sở dữ liệu DB2 chưa phân loại (uncatalogued)**

Đoạn mã 17.1 cho thấy làm thế nào để kết nối đến cơ sở dữ liệu DB2 sử dụng một trong hai phần mở rộng đã được mô tả ở trên.

```

$host = 'localhost';
$port = 50000;
$DSN = "DRIVER={IBM DB2 ODBC DRIVER}; PORT=$port;
        HOSTNAME=$host; DATABASE=$database; PROTOCOL=TCPIP;
        USER=$user; PWD=$password";

-- If using the ibm_db2 extension --
$uconn = db2_connect($DSN, null, null);

-- If using the PDO_ODBC extension --
try {
    $uconn = new PDO("odbc:$DSN", null, null);
}
catch (PDOException $e) { print $e->errormsg(); }

```

### Đoạn mã 17.1 – Kết nối đến một cơ sở dữ liệu bên ngoài

Đoạn mã 17.2 cung cấp một ví dụ cho một ứng dụng PHP đơn giản sử dụng phần mở rộng `ibm_db2`

```

<?php
$sql = "SELECT name, breed FROM ANIMALS WHERE weight < ?";
$conn = db2_connect($database, $user, $password);
$stmt = db2_prepare($conn, $sql);
$res = db2_execute($stmt, array(10));
while ($row = db2_fetch_assoc($stmt)) {
    print "{$row['NAME']} is a {$row['BREED']}. \n";
}
?>

```

### Đoạn mã 17.2 – Một ứng dụng PHP đơn giản sử dụng phần mở rộng `ibm_db2`

#### Cấu hình PHP cho `ibm_db2`

Trong Linux hoặc UNIX bạn cần thay đổi tệp `php.ini` như sau:

```

extension=ibm_db2.iso
ibm_db2.instance_name=<instance name>

```

Đối với Windows, thay đổi tệp `php.ini` như sau:

```

extension=php_ibm_db2.dll

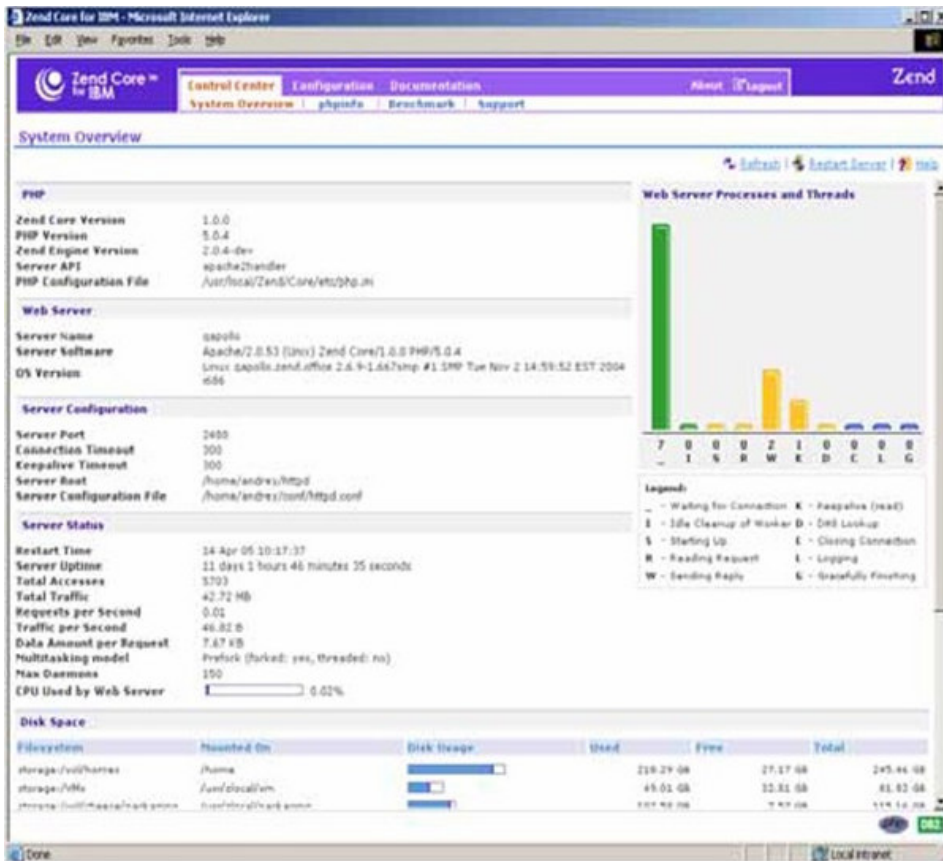
```

Tiếp theo, bạn có thể tải và cài đặt bộ ứng dụng **Zend Core for IBM** sẽ được mô tả trong phần tiếp theo, và đừng ngại về các vướng mắc ở cấu hình này.

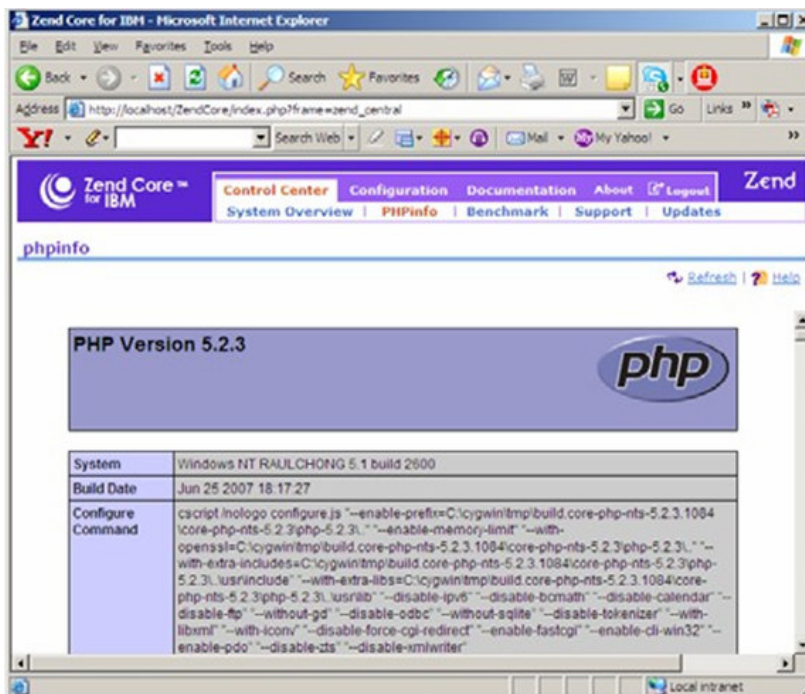
#### 17.2.2 Ứng dụng công nghệ Zend Core dành cho IBM

Zend [Core] là bộ công cụ PHP hoàn chỉnh cho môi trường phát triển và sản xuất các ứng dụng web tối quan trọng cho doanh nghiệp. Zend [Core] thể hiện sự tin cậy, hiệu quả và sự linh hoạt cần thiết cho việc vận hành những ứng dụng PHP. Zend [Core] có thể được tải về tại địa chỉ sau: <http://ibm.com/software/data/info/zendcore>

Zend [Core] cho IBM bao gồm cài đặt DB2 và máy khách IDS, một máy chủ HTTP của Apache (tùy chọn), PHP5 và một số thư viện phổ biến kèm theo bao gồm: `ibm_db2`, `PDO_INFOMIX`. Zend [Core] cho IBM cũng có thể cài đặt máy chủ DB2 Express-C, máy chủ IBM Cloudscape™, bộ hướng dẫn sử dụng PHP, và một số ứng dụng ví dụ trên DB2. Zend [Core] thể hiện sự tiện dụng và cài đặt môi trường PHP một cách dễ dàng, xem hình minh họa 17.5, 17.6 và 17.7.

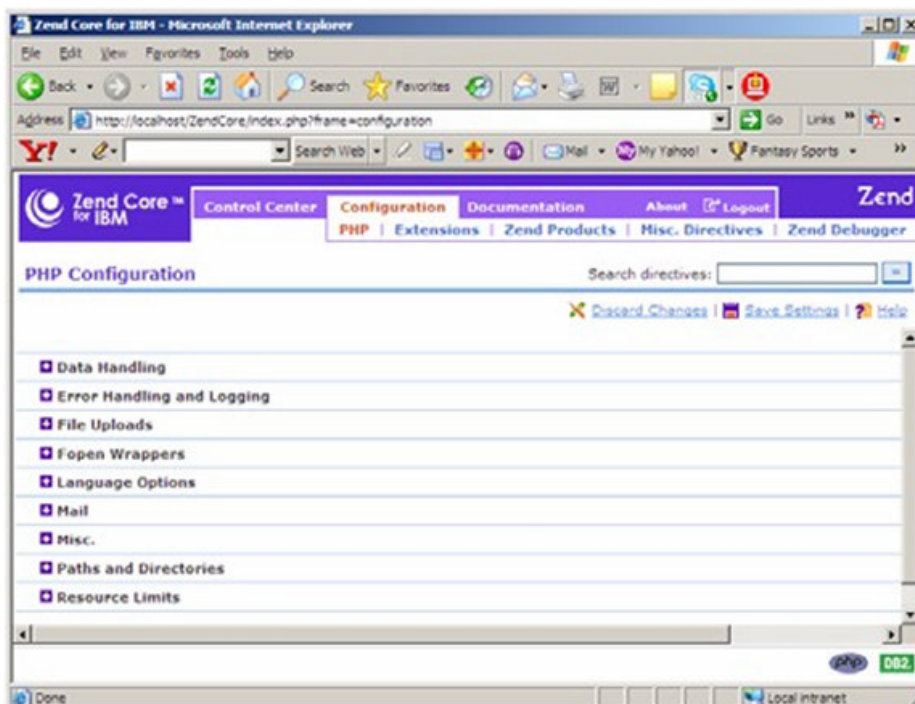


Hình 17.5 – Cửa sổ quản lý và điều khiển của Zend [Core]



Hình 17.6 – Cửa sổ cấu hình PHP của Zend [Core]





Hình 17.7 – Cửa sổ cấu hình PHP của Zend

### 17.3 Phát triển ứng dụng Ruby trên nền Rails

Ruby là một ngôn ngữ kịch bản hướng đối tượng, động và đa nền dựa trên mã nguồn mở. Ruby giúp cho phát triển các ứng dụng một cách dễ dàng, nhanh chóng và còn bao gồm tập hợp các thư viện rất phong phú. Ruby là ngôn ngữ lập trình đơn giản, gọn nhẹ được phát triển bởi Yukihiro Matsumoto (“Matz”) vào năm 1995.

Rails là một bộ khung hoàn chỉnh cho việc phát triển các ứng dụng web dựa trên cơ sở dữ liệu do Ruby phát triển. Rails cài đặt kiến trúc model-view-controller (MVC). Rails nổi lên như là một trong những khung phát triển ứng dụng web mạnh nhất kể từ năm 2004 do David Heinemeier Hansson phát triển.

#### 17.3.1 Bộ công cụ phát triển DB2 trên nền Rails

IBM nhận ra được tầm quan trọng của Ruby trên nền Rails trong cộng đồng phát triển ứng dụng web. Chính vì lý do đó mà IBM đã tạo ra bộ công cụ phát triển DB2 trên nền Rails (Startup Toolkit for DB2 on Rails). Đây là bộ cài đặt được tích hợp lại để tạo nên một môi trường phát triển DB2 bằng Ruby trên nền Rails. Bộ công cụ này có thể download tại địa chỉ: <http://www.alphaworks.ibm.com/tech/db2onrails>

Bộ công cụ này bao gồm:

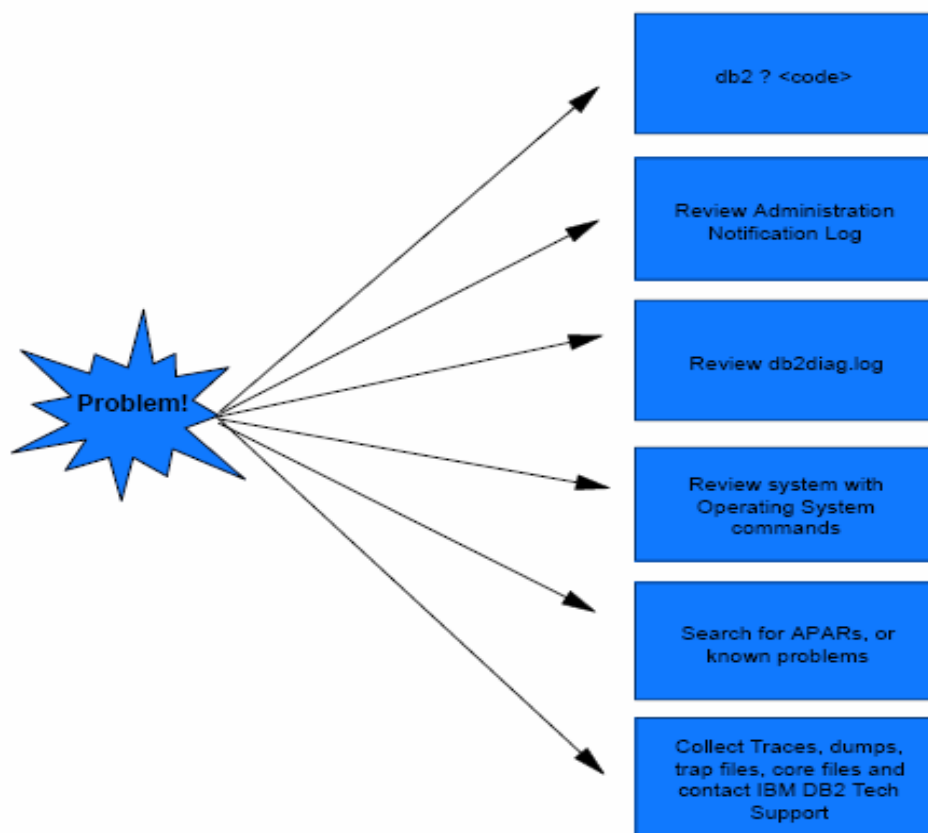
- Bộ cài đặt
- Hướng dẫn cài đặt và cấu hình Ruby và Rails
- Bộ cài đặt DB2 – Express C 9 và bộ công cụ kèm theo
- Các trình điều khiển (driver) cho DB2 Ruby và bộ thích hợp cho DB2 Rails
- Ví dụ và bài tập



# A

## Phụ lục A – Gỡ rối

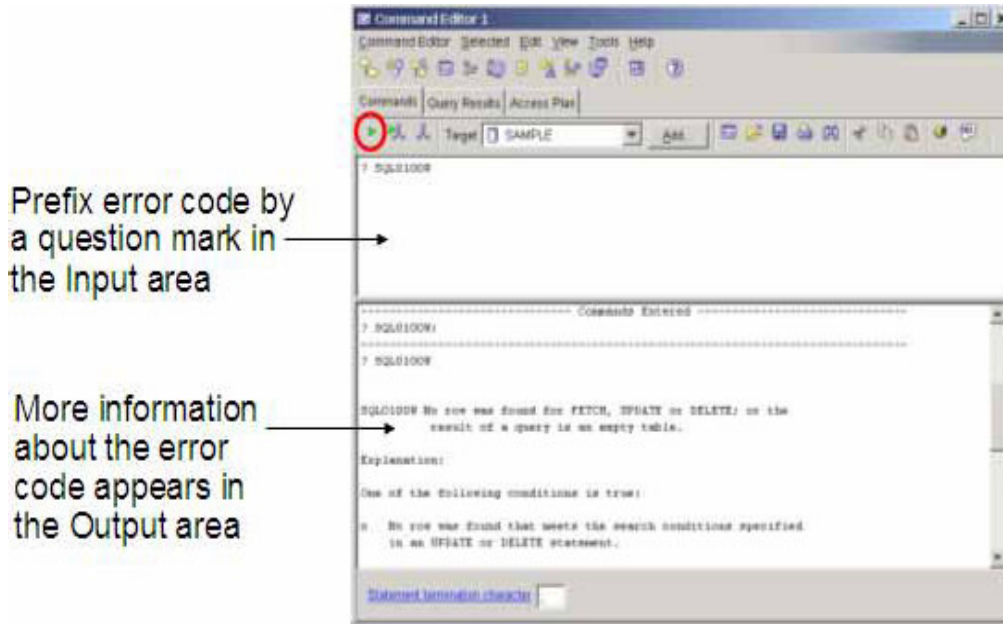
Trong phần này, chúng ta sẽ nói về cách làm thế nào để giải quyết (gỡ rối) những vấn đề có thể gặp phải khi làm việc với DB2. Hình A.1 cung cấp một sơ đồ tổng quan về những việc chúng ta nên làm khi gặp sự cố.



**Hình A.1 - Tổng quan về việc gỡ rối**

### A.1 Thêm thông tin về mã các lỗi

Để thu thập được nhiều thông tin hơn từ đoạn mã lỗi mà chúng ta nhận được hãy nhập mã lỗi mà bạn nhận được với dấu chấm hỏi (?) vào vùng nhập lệnh và nhấn vào nút *Execute*, như trong hình A.2 dưới đây



**Hình A.2 - Tìm kiếm thông tin về đoạn mã bị lỗi**

Dấu chấm hỏi (?) sẽ gọi đến trình trợ giúp (Help) của DB2. Dưới đây là một số ví dụ về kích hoạt trình help, ví dụ nhận được mã lỗi SQL là "-104". Tất cả những ví dụ dưới đây là tương đương

```
db2 ? SQL0104N
db2 ? SQL104N
db2 ? SQL - 0104
db2 ? SQL - 104
db2 ? SQL - 104N
```

**A.2 SQLCODE and SQLSTATE**

Một SQLCODE là mã nhận được sau khi mọi câu lệnh SQL đã được chạy. Ý nghĩa của các giá trị như sau:

- SQLCODE = 0; lệnh thành công
- SQLCODE > 0; lệnh thành công nhưng chương trình có cảnh báo
- SQLCODE < 0; lệnh không thành công và trả về một lỗi

SQLSTATE gồm 5 ký tự, chuỗi này tuân theo chuẩn ISO/ANSI SQL92. Hai ký tự đầu tiên cho ta biết mã lớp SQLSTATE

- 00: lệnh thành công
- 01: có cảnh báo
- 02: không tìm thấy điều kiện

Tất cả các mã lớp khác được xem như có lỗi

**A.3 Nhật ký khai báo quản trị DB2**

Nhật ký khai báo quản trị cung cấp công cụ dùng để chẩn đoán thông tin về các lỗi tại các điểm phát sinh lỗi. Ở hệ điều hành Linux/Unix thì nhật ký thông báo quản trị (Administration Notification) là một tệp văn bản <tên thể hiện>.nfy (ví dụ "db2inst.nfy"). Ở hệ điều hành Windows, tất cả các khai báo quản trị đều được ghi vào Windows Event Log.

Tham số cấu hình DBM `notifylevel` cho phép người quản trị chỉ ra mức độ thông tin được ghi lại

- 0 - không có khai báo quản trị nào được giữ lại (không khuyến khích)
- 1 - lỗi nặng hoặc không thể phục hồi
- 2 - hành động cần được thực hiện ngay lập tức
- 3 - thông tin quan trọng, không cần hành động ngay lập tức (mặc định)
- 4 - thông điệp

#### A.4 db2diag.log

Db2diag.log cung cấp thông tin chi tiết hơn về nhật ký khai báo quản trị DB2. Nó thường chỉ được sử dụng bởi nhân viên hỗ trợ kỹ thuật DB2 hay người quản trị có kinh nghiệm. Thông tin ở trong db2diag.log bao gồm:

- Vị trí của mã DB2 phát sinh lỗi
- Một chương trình nhận dạng, cho phép bạn có thể so sánh những dòng của ứng dụng trên tệp db2diag.log trên máy chủ và máy khách.
- Một thông điệp chẩn đoán (bắt đầu bằng "DIA") giải thích nguyên nhân của lỗi
- Các dữ liệu hỗ trợ sẵn có, như là cấu trúc dữ liệu SQLCA và con trỏ đến vị trí của kết xuất hoặc tệp bẫy

Ở Windows, db2diag.log được chứa trong một thư mục mặc định:

```
C:\Program Files\IBM\sqlllig\\db2diag.log
```

Ở Linux/Unix db2diag.log được chứa trong thư mục mặc định:

```
/home/<instance_ owner>/sqlllid/db2trum/db2diag.log
```

Ý nghĩa của chẩn đoán được quyết định bởi tham số cấu hình dbm cfg `DIAGLEVEL`. Cấp độ từ 0 đến 4, 0 là ít ý nghĩa nhất và 4 là cao nhất, mặc định là 3

#### A.5 Theo vết CLI

Với CLI và các ứng dụng Java, bạn có thể bật tiện ích theo vết CLI để khắc phục những sự cố cho nó. Điều này có thể được làm bằng cách sửa đổi tệp `db2cli.ini` ở máy chủ, nơi mà ứng dụng của bạn đang chạy. Những trường đặc trưng của `db2cli.ini` được chỉ ra bên dưới:

```
[common]
trace=0
tracerefreshinterval=300
tracepathname=/path/to/writeable/directory
traceflush=1
```

Theo vết ở mức thấp cũng sẵn có (db2trc), nhưng nó thường được nhân viên hỗ trợ kỹ thuật sử dụng.

#### A.6 Khắc phục và sửa lỗi trong DB2

Đôi khi một lỗi mà bạn thỉnh thoảng gặp phải lại là nguyên nhân của một sự cố (lỗi) của DB2. IBM sẽ đều đặn phát hành những bản sửa lỗi (Fixpacks) để sửa chữa những lỗi đó (APARs). Trong tệp văn bản của bản sửa lỗi có chứa danh sách những lỗi mà nó sửa chữa được. Khi phát triển một ứng dụng mới chúng ta nên cập nhật những bản sửa lỗi mới nhất để tránh những vấn đề có thể xảy ra trong khi làm việc. Để xem phiên bản hiện

tại cửa bản sửa lỗi, chọn **About** từ trình đơn **Help**, từ cửa sổ lệnh gõ “**db2level**”. Lưu ý rằng, bản sửa lỗi và sự hỗ trợ từ IBM chỉ có hiệu lực trong DB2 Express–C khi bạn mua bản quyền 12 tháng

## Nguồn tài nguyên

### Các trang web

#### 1.1. Trang web chính thức của DB2 Express-C:

[www.ibm.com/db2/express](http://www.ibm.com/db2/express)

Website này cho phép bạn download các file cài đặt cho máy chủ DB2, máy khách DB2, các driver dành cho DB2, bạn cũng có thể tìm thấy các hướng dẫn sử dụng, hoặc tìm đọc blog của nhóm phát triển DB2, đăng ký để nhận email tại đây.

#### 1.2. Diễn đàn DB2 Express:

[http://www.ibm.com/developerworks/forums/dw\\_forum.jsp?forum=805&cat=19](http://www.ibm.com/developerworks/forums/dw_forum.jsp?forum=805&cat=19)

Bạn có thể sử dụng Diễn đàn này để đăng tải các vấn đề về kỹ thuật mỗi khi bạn không thể tự mình tìm thấy câu trả lời cho các vấn đề đó.

#### 1.3. Trung tâm dữ liệu DB2

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>

Trung tâm dữ liệu cung cấp nguồn tài liệu hướng dẫn trực tuyến. Đây là nguồn tài nguyên được cập nhật một cách thường xuyên nhất.

#### 1.4. developerWorks

<http://www-128.ibm.com/developerworks/db2>

Website này là một nguồn tài nguyên tuyệt vời cho những người phát triển và những người quản trị hệ thống cơ sở dữ liệu tìm thấy được nhiều bài viết, bài hướng dẫn, v.v... hoàn toàn miễn phí.

#### 1.5. alphaWorks

<http://www.alphaworks.ibm.com/>

Website này cho phép bạn truy xuất trực tiếp vào các công nghệ nổi trội của IBM. Đây là nơi bạn có thể tìm thấy những công nghệ mới nhất từ Phòng Nghiên cứu của IBM.

#### 1.6. planetDB2

<http://www.planetdb2.com/>

Đây là nơi tập hợp nhiều blog của những người phát triển DB2.

#### 1.7. Hỗ trợ kỹ thuật DB2

Nếu bạn đã đăng ký hỗ trợ kỹ thuật 12 tháng từ DB2 Express-C, bạn có thể download các bản vá lỗi từ Website này:

[http://www-306.ibm.com/software/data/db2/support/db2\\_9/](http://www-306.ibm.com/software/data/db2/support/db2_9/)

**Sách**

1. Free Redbook: DB2 Express-C: The Developer Handbook for XML, PHP, C/C++, Java, and .NET  
Whei-Jen Chen, John Chun, Naomi Ngan, Rakesh Ranjan, Manoj K. Sardana,  
Tháng 8 năm 2006 – SG24-7301-00  
<http://www.redbooks.ibm.com/abstracts/sg247301.html?Open>
2. Understanding DB2 – Learning Visually with Examples V8.2  
Raul F. Chong, Clara Liu, Sylvia Qi, Dwaine Snow.  
Tháng 1 năm 2005 – ISBN: 0-13-185916-1
3. DB2 9: pureXML overview and fast start  
Cynthia M. Saracco, Don Chamberlin, Rav Ahuja  
Tháng 1 năm 2006 – SG24-7298
4. DB2® SQL PL: Essential Guide for DB2® UDB on Linux™, UNIX®, Windows™, i5/OS™, and z/OS®, 2<sup>nd</sup> Edition  
Zamil Janmohamed, Clara Liu, Drew Bradstock, Raul Chong, Michael Gao, Fraser McArthur, Paul Yip  
ISBN: 0-13-100772-6
5. Free Redbook: DB2 pureXML Guide  
Whei-Jen Chen, Art Sammartino, Dobromir Goutev, Felicity Hendricks, Ippei Komi, Ming-Pang Wei, Rav Ahuja, Matthias Nicola.  
Tháng 8 năm 2007.  
<http://www.redbooks.ibm.com/abstracts/sg247315.html?Open>
6. Information on Demand – Introduction to DB2 9 New Features  
Paul Zikopoulos, George Baklarz, Chris Eaton, Leon Katsnelson  
ISBN-10: 0071487832  
ISBN-13: 987-0071487832
7. Redbook: Developing PHP Applications for IBM Data Servers.  
Whei-Jen Chen, Holger Kirstein, Daniel Krook, Kiran H Nair, Piotr Pietrzak  
Tháng 5 năm 2006 – SG24-7218-00  
<http://www.redbooks.ibm.com/abstracts/sg247218.html?Open>

**Email liên hệ**

Hộp thư DB2 Express-C: [db2x@ca.ibm.com](mailto:db2x@ca.ibm.com)

Hộp thư DB2 dành cho các trường Đại học: [db2univ@ca.ibm.com](mailto:db2univ@ca.ibm.com)

**Bắt đầu với DB2 không thể nào dễ dàng hơn. Đọc cuốn sách này để:**

- Tìm thấy tất cả những điều về DB2 Express-C.
- Hiểu về kiến trúc của DB2, các công cụ và bảo mật DB2.
- Học cách quản trị Cơ sở dữ liệu DB2.
- Viết mã SQL, XQuery.
- Phát triển chương trình ứng dụng sử dụng cơ sở dữ liệu DB2.
- Luyện tập bằng những bài tập thực hành.
- Chuẩn bị cho kỳ thi DB2 trong trường.

Sự phát triển nhanh chóng của XML, Web 2.0, và SOA đã dẫn đến những nhu cầu về một loại hệ quản trị dữ liệu lai mang tính cách tân. DB2 Express-C của IBM là một hệ quản trị dữ liệu miễn phí, không giới hạn, và là cơ sở dữ liệu LAI có khả năng quản trị cả XML và dữ liệu quan hệ thuần túy một cách dễ dàng. Miễn phí nghĩa là DB2 Express-C có thể được tự do tải về, tự do để bạn xây dựng ứng dụng, tự do triển khai sản phẩm, và tự do phân phối lại các giải pháp của bạn. Và, DB2 không đặt một giới hạn nào cho kích thước dữ liệu, số lượng dữ liệu, hoặc số lượng người sử dụng.

DB2 Express-C chạy trên hệ thống Windows và Linux và hỗ trợ nhiều ngôn ngữ lập trình khác nhau, bao gồm cả C/C++, Java, .NET, PHP, Perl, và Ruby. Những hỗ trợ trực tuyến cũng có thể được đăng ký với giá thấp. Nếu bạn muốn có nhiều chức năng cao cấp hơn nữa đối với một hệ quản trị cơ sở dữ liệu, bạn có thể dễ dàng triển khai những ứng dụng đã được xây dựng bằng DB2 Express-C lên những ấn bản DB2 khác như DB2 Enterprise.

Ấn bản DB2 miễn phí này rất lý tưởng cho những người phát triển, những người tư vấn, những người bán phần mềm độc lập, các quản trị viên cơ sở dữ liệu, sinh viên, hoặc bất kỳ ai có ý định phát triển, thử nghiệm, triển khai, hoặc phân phối các ứng dụng về cơ sở dữ liệu. Hãy tham gia vào sự phát triển của cộng đồng những người sử dụng DB2 Express-C ngay hôm nay. Bạn sẽ khám phá ra cách hoàn toàn mới để tạo ra chương trình ứng dụng, cũng như cách phát triển những giải pháp đột phá.