

Conociendo el

DB2 Express-C

Actualizado
para la versión 9.7

Ideal para desarrolladores y administradores



Raul F. Chong

CON

**Ian Hakes,
Rav Ahuja**

Prólogo por

Dr. Arvind Krishna

Traducido por
Jorge Martelanz
Adrian H. Tozzi

Un libro de la comunidad para la comunidad

CONOCIENDO AL

DB2 Express-C

Un libro de la comunidad para la comunidad

RAUL CHONG, IAN HAKES, RAV AHUJA

PRÓLOGO POR EL DR. ARVIND KRISHNA



TERCERA EDICIÓN

Tercera Edición (Junio 2009)

**Esta edición ha sido actualizada para IBM® DB2® Express-C Versión 9.7 for Linux®,
UNIX® and Windows®.**

© Copyright IBM Corporation, 2007, 2009. Todos los derechos reservados.

Contenido

Acerca de este libro	11
Avisos y marcas registradas.....	11
¿Quién debería leer este libro?	13
¿Cómo está estructurado este libro?.....	13
Un libro para la comunidad.....	14
Autores y Colaboradores.....	15
Trabajo de traducción.....	15
Agradecimientos.....	16
Prólogo.....	17
PARTE I – VISIÓN GENERAL Y CONFIGURACIÓN	19
Capítulo 1 – ¿Qué es el DB2 Express-C?	21
1.1 Desarrolle, implemente y distribuya ¡Gratuitamente y sin límites!.....	22
1.2 Asistencia al usuario y soporte técnico	22
1.3 Servidores DB2	23
1.4 Clientes y controladores DB2	24
1.5 Libertad para el desarrollo de aplicaciones	25
1.6 Versiones DB2 versus ediciones DB2	26
1.7 Avanzando a otra edición de DB2	27
1.8 Mantenimiento y actualizaciones para el DB2 Express-C.....	27
1.9 Otros programas y componentes DB2 relacionados que se ofrecen en forma gratuita	28
1.9.1 IBM Data Studio	28
1.9.2 DB2 Text Search	29
1.9.3 WebSphere Application Server – Community Edition.....	29
1.10 Resumen	29
Capítulo 2 – Características y productos relacionados	31
2.1 Características incluidas con la licencia de suscripción de DB2 Express-C	34
2.1.1 Fix Packs.....	34
2.1.2 Alta disponibilidad de Recuperación Crítica (HADR)	35
2.1.3 Réplica de Datos	36
2.2 Características no disponibles con DB2 Express-C.....	36
2.2.1 Particionamiento de la Base de Datos.....	37
2.2.2 Concentrador de conexión.....	37
2.2.3 Geodetic Extender	37
2.2.4 Control de Acceso basado en Etiqueta (Label-based Access Control) (LBAC).....	37
2.2.5 Workload Manager (WLM).....	38
2.2.6 Alta compresión.....	39
2.2.7 Compatibilidad SQL.....	40
2.3 Productos pagados relacionados con DB2	41
2.3.1 DB2 Connect	41
2.3.2 InfoSphere Federation Server	42
2.3.3 InfoSphere Replication Server.....	43
2.3.4 Optim Development Studio (ODS).....	44
2.3.5 Optim Database Administrator (ODA).....	44

6 Conociendo al DB2 Express-C

2.4 DB2 se ofrece a través de la nube de cómputo elástica de Amazon	44
2.5 Resumen	44
Capítulo 3 – Instalación de DB2	47
3.1 Requisitos previos para la instalación.....	47
3.2 Autoridad de instalación a nivel de sistema operativo	47
3.3 Asistente de instalación.....	48
3.4 Validación de la instalación	55
3.5 Instalación silenciosa.....	57
3.6 Resumen	58
3.7 Ejercicios	58
Capítulo 4 – El ambiente DB2.....	63
4.1 Configuración de DB2	72
4.1.1 Variables de ambiente.....	73
4.1.2 Archivo de configuración del manejador de la base de datos (dbm cfg)	73
4.1.3 El archivo de configuración de la base de datos (db cfg).....	75
4.1.4 Registro de perfiles DB2.....	76
4.2 El Servidor de Administración (obsoleto)	77
4.3 Resumen	78
4.4 Ejercicios	78
Capítulo 5 – Herramientas de DB2	85
5.1 IBM Data Studio	87
5.1 Centro de Control (Control Center) (obsoleto)	88
5.2 El Editor de Comandos DB2 (obsoleto)	92
5.3 SQL Assist Wizard (obsoleto).....	94
5.4 El botón Show SQL (obsoleto)	96
5.6 Centro de Tareas (Task Center) (obsoleto)	97
5.6.1 La base de datos del catálogo de herramientas (The Tools Catalog database) (obsoleto)	97
5.7 Journal (obsoleto).....	99
5.7.1 Invocando el Journal	100
5.8 Monitor de Salud (Health Monitor) (obsoleto)	101
5.8.1 Centro de Salud (Health Center) (obsoleto).....	102
5.9 Administrador de memoria auto-ajustable	104
5.10 Scripting	104
5.10.1 Scripts SQL	105
5.10.2 Scripts del sistema Operativo (shell scripts)	106
5.11 Consideraciones para Windows Vista	107
5.12 Resumen	107
5.13 Ejercicios	108
PARTE II – APRENDIENDO DB2: ADMINISTRACIÓN DE BASES DE DATOS	115
Capítulo 6 – Arquitectura DB2.....	117
6.1 El modelo de proceso DB2	117
6.2 Modelo de memoria DB2	119
6.3 Modelo de almacenamiento de DB2	121

6.3.1 Páginas y Extensiones	121
6.3.2 Agrupaciones de almacenamientos intermedios	121
6.3.3 Espacios de tablas	123
6.4 Resumen	129
6.5 Ejercicios	129
Capítulo 7 – Conectividad de los clientes DB2	133
7.1 Directorios DB2	133
7.1.1 Directorio de bases de datos del sistema	133
7.1.2 Directorio de bases de datos locales	134
7.1.3 Directorio de nodos	134
7.1.4 Directorio DCS.....	134
7.2 Asistente de configuración (obsoleto)	134
7.2.1 Requerimientos de preparación en el servidor	135
7.2.2 Configuración requerida en el cliente	137
7.2.3 Creación de perfiles de cliente y servidor	141
7.3 Resumen	144
7.4 Ejercicios	144
Capítulo 8 – Trabajando con los objetos de la Base de datos	147
8.1 Esquemas	147
8.2 Sinónimos públicos (o alias)	148
8.3 Tablas	149
8.3.1 Tipos de dato.....	149
8.3.2 Columnas de identidad.....	154
8.3.3 Objetos de secuencia	155
8.3.4 Las tablas del catálogo del sistema	156
8.3.5 Tablas globales temporales declaradas (<i>Declared global temporary tables - DGTTs</i>)	156
8.3.6 Tablas creadas globales temporales (<i>Create Global Temporary Tables - CGTTs</i>)	158
8.4 Vistas	159
8.5 Índices	160
8.5.1 Asistente de diseño	160
8.6 Integridad referencial.....	162
8.7 Evolución de los esquemas	163
8.8 Resumen	164
8.9 Ejercicios	165
Capítulo 9 – Utilitarios para el movimiento de datos.....	169
9.1 Utilitario EXPORT.....	170
9.2 Utilitario IMPORT.....	171
9.3 Utilitario LOAD.....	172
9.4 El utilitario db2move	174
9.5 El utilitario db2look	174
9.6 Resumen	177
9.7 Ejercicios	177

Capítulo 10 – Seguridad de la base de datos	181
10.1 Autenticación.....	182
10.2 Autorización.....	183
10.2.1 Privilegios	184
10.2.2 Autoridades	185
10.2.3 Roles	189
10.3 Consideraciones cuando se usan privilegios a grupos	190
10.4 El grupo PUBLIC	190
10.5 Las sentencias GRANT y REVOKE.....	191
10.6 Chequeo de las autorizaciones y privilegios.....	191
10.7 Seguridad extendida en Windows	193
10.8 Resumen.....	193
10.9 Ejercicios	194
Capítulo 11 – Respaldo y recuperación.....	199
11.1 Anotación cronológica (<i>logging</i>)	199
11.2 Tipos de archivos de anotación cronológica	200
11.3 Tipos de anotación cronológica	201
11.3.1 Anotación cronológica circular	201
11.3.2 Anotación cronológica con archivo	202
11.4 Manejo de la anotación cronológica desde el Centro de control.....	203
11.5 Parámetros para la anotación cronológica.....	205
11.6 Respaldo de base de datos	206
11.7 Recuperación de base de datos	208
11.7.1 Tipos de recuperación	208
11.7.2 Recuperación de la base de datos	209
11.8 Otras operaciones con BACKUP y RESTORE	209
11.9 Resumen	210
11.10 Ejercicios	210
Capítulo 12 – Tareas de Mantenimiento	213
12.1 REORG, RUNSTATS, REBIND.....	213
12.1.1 El comando REORG	214
12.1.2 El comando RUNSTATS	214
12.1.3 BIND / REBIND	215
12.1.4 Tareas de mantenimiento desde el Centro de control	216
12.2 Las opciones de Mantenimiento	218
12.3 Resumen	220
12.4 Ejercicios	220
Capítulo 13 – Conurrencia y Bloqueo	225
13.1 Transacciones	225
13.2 Conurrencia	226
13.3 Problemas sin control de concurrencia.....	227
13.3.1 Actualizaciones perdidas	227
13.3.2 Lecturas no confirmadas o Uncommitted read	228
13.3.3 Lecturas no repetidas	229

13.3.4 Lectura fantasma.....	230
13.4 Niveles de aislamiento.....	231
13.4.1 Lectura no confirmada o Uncommitted read	231
13.4.2 Estabilidad de cursor o Cursor stability.....	231
13.4.3 Estabilidad de lectura o Read stability	233
13.4.4 Lecturas repetidas	234
13.4.5 Comparando niveles de aislamiento.....	234
13.4.6 Fijando los niveles de aislamiento	235
13.5 Escalamiento de Bloqueos	236
13.6 Monitoreo de bloqueos	237
13.7 Espera de Bloqueo	239
13.8 Bloqueo Mortal o Deadlock, causas y detección	239
13.9 Mejores prácticas para concurrencia y bloqueo.....	241
13.10 Resumen	242
13.11 Ejercicios	242
PARTE III – APRENDIENDO DB2: DESARROLLO DE APLICACIONES	249
Capítulo 14 – Introducción al Desarrollo de Aplicaciones en DB2.....	251
14.1 Desarrollo de Aplicaciones DB2: Una visión global	251
14.2 Desarrollo del lado del Servidor.....	253
14.2.1 Procedimientos Almacenados	253
14.2.2 Funciones definidas por el Usuario.....	254
14.2.3 Disparadores (Triggers).....	254
14.3 Desarrollo del lado del Cliente	255
14.3.1 SQL embebido.....	255
14.3.2 SQL Estático vs. SQL Dinámico	256
14.3.3 CLI y ODBC.....	259
14.3.4 JDBC, SQLJ y pureQuery.....	262
14.3.5 OLE DB	264
14.3.6 ADO.NET	265
14.3.7 PHP	267
14.3.8 Ruby on Rails (Ruby sobre Ruedas)	267
14.3.9 Perl.....	268
14.3.10 Python	268
14.4 XML y DB2 pureXML.....	269
14.5 Servicios Web (Web Services)	269
14.6 APIs Administrativas.....	271
14.7 Otros desarrollos	271
14.7.1 Trabajando con Microsoft Access y Microsoft Excel.....	271
14.8 Herramientas de Desarrollo	273
14.9 Programas de Ejemplo	273
14.10 Resumen	274
Capítulo 15 – DB2 pureXML.....	275
15.1 Uso de XML en bases de datos.....	276
15.2 Bases de datos XML	276

10 Conociendo al DB2 Express-C

15.2.1 Bases de datos XML-enabled.....	276
15.2.2 Bases de datos Nativa XML	277
15.3 XML en DB2	278
15.3.1 Ventajas tecnológicas de pureXML	279
15.3.2 XPath básico	282
15.3.3 XQuery: Definición.....	285
15.3.4 Inserción de documentos XML	286
15.3.5 Consulta de datos XML	290
15.3.6 Unir con SQL/XML.....	297
15.3.7 Unir con XQuery	298
15.3.8 Operaciones de actualización y eliminación	298
15.3.9 Indexación XML.....	301
15.4 Trabajando con Esquemas XML	302
15.4.1 Registrando los Esquemas XML.....	302
15.4.2 Validación del Esquema XML.....	305
15.4.3 Otro soporte para XML	306
15.6 Resumen	307
15.7 Ejercicios	307
Apéndice A – Resolución de problemas	309
A.1 Como encontrar más información sobre los códigos de error	310
A.2 SQLCODE y SQLSTATE	310
A.3 El registro administrativo de notificaciones de DB2.....	311
A.4 db2diag.log	311
A.5 Trazadores CLI	312
A.6 Defectos y correcciones de DB2	312
Apéndice B – Referencias y recursos.....	313
B.1 Referencias	313
B.2 Sitios web:.....	313
B.3 Libros	314
B.4 emails de contacto	315

Acerca de este libro

Avisos y marcas registradas

© Copyright IBM Corporation 2007, 2009

Todos los derechos reservados.

IBM Canada

8200 Warden Avenue

Markham, ON

L6G 1C7

Canada

La presente documentación no puede copiarse ni reproducirse, total o parcialmente, de ningún modo ni a través de ningún medio así como tampoco traducirse a otro idioma sin el consentimiento previo de todos los propietarios del Copyright mencionados anteriormente.

IBM no ofrece garantías ni se manifiesta respecto a este contenido y renuncia específicamente a cualquier garantía implícita de comercialización o idoneidad para cualquier fin determinado. IBM no asume ninguna responsabilidad por los errores que puedan aparecer en el presente documento, incluidos, aunque sin limitarse a ellos, los errores de traducción. La información contenida en el presente documento está sujeta a cambios sin previo aviso. IBM se reserva el derecho a realizar cualquier cambio de este tipo sin obligación de notificar a persona alguna dichas revisiones o dichos cambios. IBM no se compromete a mantener actualizada la información aquí contenida.

La información contenida en el presente documento relacionada con los productos no-IBM se ha obtenido del proveedor o de los proveedores de dichos productos. IBM no ha probado dichos productos y no puede confirmar la precisión del rendimiento, la compatibilidad ni ninguna otra afirmación relacionada con los productos de IBM. Las consultas sobre las capacidades de los productos no-IBM se deben dirigir al proveedor o a los proveedores de dichos productos.

IBM, el logotipo de IBM, e ibm.com son marcas o marcas registradas de International Business Machines Corporation en los Estados Unidos, registradas en muchas jurisdicciones en todo el mundo. Otros productos o nombres de servicios pueden ser marcas de IBM o de otras compañías. La lista actualizada de las marcas IBM está disponible en la Web bajo el título "Copyright and trademark information" en www.ibm.com/legal/copytrade.shtml.

Java y todas las marcas registradas basadas en Java son marcas de Sun Microsystems, Inc. en los Estados Unidos, otros países, o ambos.

Microsoft y Windows son marcas registradas de Microsoft Corporation en los Estados Unidos, otros países, o ambos.

Linux es una marca registrada de Linus Torvalds en los Estados Unidos, otros países, o ambos.

12 Conociendo al DB2 Express-C

UNIX es una marca registrada de “The Open Group” en los Estados Unidos, otros países, o ambos.

Otras empresas, productos, o nombres de servicios pueden ser marcas registradas o marcas de servicio de otros.

Las referencias realizadas en esta publicación a productos o a servicios de IBM no implican que IBM tenga la intención de comercializarlos en todos los países en los que IBM realiza operaciones comerciales.

¿Quién debería leer este libro?

Este libro está orientado a todo aquel que trabaja, o pretende trabajar, con bases de datos, como ser administradores de bases de datos (DBAs), desarrolladores de aplicaciones, consultores, arquitectos de “software”, gerentes de producto, instructores y estudiantes.

¿Cómo está estructurado este libro?

En la Parte I, Visión general y Configuración, se explica de qué se trata la edición DB2 Express-C, se introduce la familia de productos y opciones DB2, se da asistencia para la instalación y creación de bases de datos y se exploran las herramientas disponibles con DB2.

La Parte II, Aprendiendo DB2: Administración de Bases de Datos, está diseñada para familiarizarle con el ambiente DB2, su arquitectura, conectividad remota, objetos de la base de datos, movimiento de datos (exportación, importación y carga), seguridad, respaldo y recuperación, concurrencia y bloqueo, y otras tareas habituales de mantenimiento.

La Parte III – Aprendiendo DB2: Desarrollo de Aplicaciones, describe el proceso de desarrollo de aplicaciones utilizando DB2, incluyendo tanto el desarrollo en los servidores como en los clientes. SQL/XML, XQuery, y pureXML® son otros temas expuestos en esta parte.

El apéndice contiene información útil sobre resolución de problemas.

En la mayoría de los capítulos podrá encontrar ejercicios y todos los archivos requeridos para estos laboratorios están en el archivo comprimido adjunto al libro:
[expressc_book_exercises_9.7.zip](#).

Los materiales en este libro son también usados en cursos que se ofrecen como componentes del **“DB2 on Campus Program”**, y siguen de cerca las presentaciones “e-learning” (en inglés) disponibles en [www.channelDB2.com/oncampus](#). Más información acerca del programa “DB2 on Campus” puede hallarse en el sitio web del DB2 Express-C: [www.ibm.com/db2/express/students.html](#).

Nota:

Para mayor información sobre el programa “DB2 on Campus”, vea el video (en inglés) en:
<http://www.channeldb2.com/video/video/show?id=807741:Video:3902>

En esta tercera edición hemos hecho numerosos cambios y adiciones. Aquellos quienes hayan leído la segunda edición del libro basada en DB2 9.5, encontrarán fácilmente los cambios en el libro que corresponden a nuevas opciones o actualizaciones en la versión 9.7 de DB2. Esos cambios han sido identificados con este icono:

nuevo en
V9.7

Un libro para la comunidad

Este libro fue creado por el equipo de DB2 Express-C y liberado para la comunidad de DB2 Express-C sin cargo alguno. Hasta este momento ha sido descargado más de 45,000 veces y traducido a 9 idiomas por voluntarios de todo el mundo. ¡Un verdadero esfuerzo comunitario! Si usted desea hacer comentarios, contribuir con material nuevo, mejorar el existente o ayudar con la traducción de este libro a otro lenguaje, por favor envíe un email con una descripción de la contribución que pretende hacer a db2x@ca.ibm.com con el asunto: "DB2 Express-C book changes."

El suceso de este libro ha sido la inspiración para desarrollar más de 25 nuevos libros *online* gratis sobre productos IBM y también sobre tecnologías no-IBM. Los libros serán parte de la "**DB2 on Campus Book Series**", que se lanzó en octubre de 2009.

Para mayor información sobre este libro o la "DB2 on Campus Book Series" visite el sitio web del IBM® DB2 Express-C en www.ibm.com/db2/express

Autores y Colaboradores

Las siguientes personas han provisto el contenido y han hecho otras contribuciones significativas a este libro.

Raul F. Chong – Autor principal

Raul es el Gerente del Programa “DB2 on Campus” en el Laboratorio IBM de Toronto.

Ian Hakes – Co-autor y Editor

Ian se desempeñó anteriormente como facilitador de la Comunidad DB2 Express-C y actualmente trabaja como experto en usabilidad en el Laboratorio IBM de Toronto.

Rav S. Ahuja – Co-author and Publishing

Rav es Gerente de Producto senior para DB2 en el Laboratorio IBM de Toronto.

Trabajo de traducción

Quisiéramos agradecer a los voluntarios de la comunidad de DB2 por el increíble trabajo de traducción. Ellos trabajaron por muchas horas y durante fines de semana para hacer de este trabajo una realidad. Esta es la lista de colaboradores voluntarios:

Jorge Martelanz

Profesional experimentado en Tecnología de la Información. Trabaja en el área desde 1971 y con DB2 desde la primera versión en mainframe (1983). Su perfil completo se encuentra en: <http://www.linkedin.com/in/martelanz>

Jorge trabaja actualmente como Advisory DB2 consultant. Jorge hizo la traducción integral de la portada, la introducción, el prólogo, y los capítulos 1, 3, 4, 6 al 11 y apéndices.

Si desea contactarse con Jorge, su correo electrónico es: jorge_martelanz@hotmail.com

Adrián H. Tozzi

Adrian es un especialista en bases de datos y BI. Trabaja en tecnología desde el año 1993. Adrian es colaborador de Calificaciones Académicas DB2 y de la Iniciativa Académica IBM desde sus comienzos (www.ibm.com/university). También trabaja como investigador académico y docente universitario (CAECE-UB-UP) y como instructor a empresas desde el año 1996.

Adrian es socio y gerente de IL Consulting (www.ilconsulting.com.ar), una empresa proveedora de servicios tecnológicos.

16 Conociendo al DB2 Express-C

Adrian hizo la traducción integral del Capítulo 14 y la traducción de actualizaciones para DB2 V9.7 y agregados (no incluidos en la traducción anterior de 9.5) en Capítulos 2, 5 , 12 , 13 y 15.

Si desea contactarse con Adrian, su correo electrónico es: atozzi@ilconsulting.com.ar

Agradecimientos

Agradecemos sinceramente a las siguientes personas por su ayuda en el desarrollo de materiales referenciados en el libro:

- Ted Wasserman, Clara Liu, and Paul Yip del Laboratorio IBM de Toronto Lab quienes desarrollaron materiales que han servido como estructura básica para este libro.
- Don Chamberlin y Cindy Saracco por sus artículos sobre XQuery en "IBM developerWorks, y Matthias Nicola por sus presentaciones sobre pureXML.
- Kevin Czap y Grant Hutchison por desarrollar material técnico de introducción a DB2.
- Katherine Boyachok y Natasha Tolub por el diseño de la portada.
- Susan Visser por revisar y brindar asistencia en la publicación de este libro.

Prólogo

La innovación es la piedra angular del progreso tecnológico. En IBM, la innovación ha sido parte integral de la evolución de nuestros servidores de datos. Habiendo sido pionera en técnicas de manejo de datos en los años sesentas y setentas, IBM continúa produciendo tecnologías innovadoras en el manejo de información, lo que se refleja en los miles de patentes cuyos autores son tecnólogos de IBM. Como resultado, algunas de las mayores organizaciones en el mundo confían en productos IBM, incluyendo DB2, como motor de sus soluciones de manejo de datos más críticas y demandantes.

Sin embargo, DB2 ya no es más solo para grandes empresas. Con el lanzamiento de DB2 Express-C, la tecnología DB2 que ha ganado premios, está ahora a disposición de compañías medianas y pequeñas, ¡y sin costo alguno! Aunque hay otros servidores de datos gratis o de código fuente abierto, DB2 Express-C ofrece ventajas únicas sobre esas alternativas.

Hay muchos avances tecnológicos presentes en DB2 Express-C. Estas innovaciones proveen nuevas capacidades, reducen cargas administrativas, mejoran el rendimiento y reducen el costo de la infraestructura.

La tecnología híbrida de DB2 Express-C es capaz de manejar tanto datos relacionales como XML en su formato nativo. Esto hace a DB2 ideal para dar poder a la nueva camada de aplicaciones SOA y Web 2.0, donde XML fluye en abundancia. A diferencia de otros servidores de datos “gratis”, DB2 Express-C no limita la cantidad de datos que usted puede almacenar en una base de datos o la cantidad de bases de datos que pueden crearse en un sistema. Y por supuesto, si usted requiere soporte o asistencia de IBM, la ayuda está a un clic de distancia.

Este libro sirve de guía para conocer y comenzar a usar DB2 Express-C. Él le ayudará a entender los conceptos de DB2 y le permitirá desarrollar habilidades en la administración de DB2 y en el desarrollo de aplicaciones. Las habilidades y conocimiento que usted adquiera serán igualmente válidas para otras ediciones más avanzadas de DB2 en Linux, UNIX, y Windows.

A pesar que el DB2 Express-C no es un producto de código fuente abierto, en IBM creemos firmemente en fomentar y dar soporte a iniciativas comunitarias. Estoy encantado con que este libro haya sido desarrollado por miembros de la comunidad de DB2 Express-C y que será puesto a disposición de todos los miembros en forma gratuita. Les aliento a enriquecer este libro con vuestro conocimiento y experiencias, y también a colaborar con la traducción a otros idiomas de modo que más lectores puedan beneficiarse.



Arvind Krishna
Vice President, Data Servers
Information Management, IBM Software Group

PARTE I – VISIÓN GENERAL Y CONFIGURACIÓN

1

Capítulo 1 – ¿Qué es el DB2 Express-C?

El servidor de datos DB2 Express-C ("DB2 Express-C") es un miembro de la familia IBM DB2 que comprende poderosos servidores de datos para manejar tanto datos relacionales como XML. El "DB2 Express-C" es una edición fácil de usar y sin límites que se ofrece en forma gratuita. La 'C' en "DB2 Express-C" significa Comunidad. Una comunidad de usuarios del producto que se unen para asistirse mutuamente, tanto en línea como fuera de línea. La comunidad "DB2 Express-C" está compuesta de toda clase de gentes y compañías que diseñan, desarrollan, implementan o utilizan soluciones de bases de datos. Entre los miembros de la comunidad se cuentan:

- Desarrolladores de aplicaciones quienes requieren de un servidor de bases de datos de estándares abiertos para construir aplicaciones tanto individuales como cliente-servidor, basadas en la Web o empresariales.
- Proveedores independientes de "software" (ISVs por su denominación en inglés – "Independent Software Vendors"), proveedores de "hardware", proveedores de infraestructuras apiladas ("infrastructure stack"), y cualquier otro tipo de proveedores de soluciones quienes requieran incorporar o integrar un servidor de datos con todo tipo de características, en dichas soluciones.
- Consultores, administradores de bases de datos y arquitectos de tecnologías informáticas que requieren de un servidor de datos robusto para capacitación, desarrollo de habilidades, evaluación y desarrollo de prototipos.
- Empresas nuevas, pequeñas o de mediano tamaño que requieran de un servidor de datos confiables para sus aplicaciones y operación.
- Aficionados a las bases de datos y entusiastas de la tecnología de punta quienes desean un servidor de datos fácil de usar para construir aplicaciones Web 2.0 y de próxima generación.
- Estudiantes, profesores y otros usuarios académicos que requieren un servidor de datos muy versátil para educación, desarrollo de cursos, proyectos e investigación.

El "DB2 Express-C" comparte la misma funcionalidad y código base que las otras versiones con cargo de DB2 en Linux, UNIX y Windows. Puede correr en sistemas de 32 y 64 bits con sistemas operativos Linux o Windows. Está además disponible para Solaris (X64) y como versión beta para Mac OS X (x64). Puede ejecutarse en sistemas con cualquier número de procesadores y tamaño de memoria y no tiene ningún requerimiento

especial de memoria o configuración del sistema. El “DB2 Express-C” incluye además sin cargo, “pureXML”. Ésta es una tecnología única de DB2 que le permite almacenar y procesar documentos XML en forma nativa.

1.1 Desarrolle, implemente y distribuya ¡Gratuitamente y sin límites!

Esta frase resume los ideales clave detrás del DB2 Express-C:

- **Desarrolle sin costo:** Si usted es un desarrollador de aplicaciones y necesita una base de datos para su aplicación, puede usar el DB2 Express-C.
- **Implemente sin costo:** Si trabaja en un ambiente de Producción y requiere de un sistema de administración de datos para almacenar sus registros vitales, puede utilizar el DB2 Express-C.
- **Distribuya sin costo:** Si está desarrollando una aplicación o herramienta que requiere de un servidor de datos incorporado, puede incluir el DB2 Express-C. Aún cuando el DB2 Express-C está incorporado a su aplicación y es distribuido a sus clientes cada vez que la vende, continúa siendo gratuito. Lo único que usted debe hacer para poder redistribuir el DB2 Express-C es registrarse con IBM, sin embargo, este proceso de registro es también gratuito.
- **Sin límites:** Mientras otras ofertas de bases de datos de la competencia fijan límites en el tamaño de las bases de datos, número de ellas o cantidad de usuarios, el DB2 Express-C no impone NINGÚN límite al tamaño de los datos. Su base de datos puede continuar creciendo sin violar los términos de la licencia. Tampoco hay límites en el número de conexiones o usuarios por servidor.

Nota:

Para aprender más acerca del DB2 Express-C y su rol en el mundo de la información por demanda (“information on-demand”) y en la Web 2.0, eche un vistazo a esta presentación en video (en inglés): <http://www.channeldb2.com/video/video/show?id=807741:Video:3922>

1.2 Asistencia al usuario y soporte técnico

Si usted tiene preguntas técnicas acerca del DB2 Express-C, puede hacerlas en el [DB2 Express-C forum](#). Este foro gratuito es supervisado por expertos en DB2 de IBM, aún cuando es la comunidad de usuarios quienes proveen la mayoría de respuestas en forma voluntaria.

IBM también le da a los usuarios la posibilidad de adquirir a bajo costo una suscripción anual al servidor de datos DB2 Express (lo que se conoce como una Licencia de Periodo Fijo o FTL por su denominación en inglés – “Fixed Term License”). Esta suscripción viene con el respaldo del soporte técnico IBM 24 x 7 y actualizaciones de programa.

Adicionalmente al soporte y el mantenimiento de los programas, pagando esta suscripción

de bajo costo (USD 2.995 por Servidor por Año en los Estados Unidos – puede variar en otros países) usted además obtiene la autorización para utilizar las siguientes opciones adicionales: HADR (agrupación de servidores para Alta Disponibilidad y Recuperación en caso de Desastres – “High Availability and Disaster Recovery”), replicación SQL (para replicar datos con otros servidores DB2), y Compresión de Respaldos (“Backup Compression” – para crear copias comprimidas de la base de datos a utilizarse como respaldo). Mayor información sobre esta opción de suscripción (en inglés) puede verse en: www.ibm.com/db2/express/support.html

1.3 Servidores DB2

Todas las ediciones de servidores DB2 contienen los mismos componentes centrales; están empaquetados de modo que los usuarios puedan elegir las opciones que necesitan a un precio adecuado. La *Figura 1.1* muestra las diferentes ediciones del producto DB2.

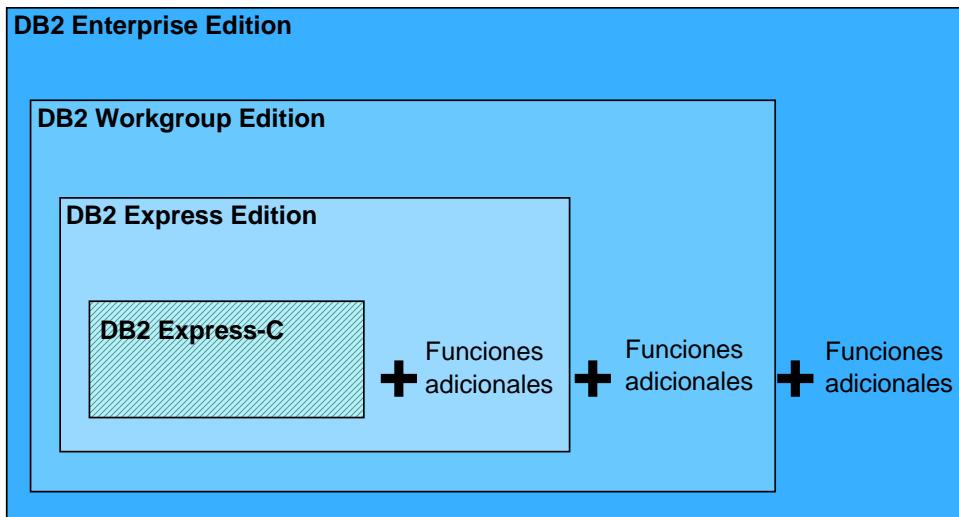


Figura 1.1 – Servidores DB2

Como se ve en la *Figura 1.1*, el DB2 Express-C es lo mismo que el DB2 Express con algunos componentes menos. El DB2 Express-C es gratis para la comunidad. Se puede obtener asistencia técnica a través de un foro en línea (también gratis), o bien se puede optar por recibir soporte técnico oficial de IBM 24 x 7 si usted decide adquirir la suscripción anual (“Fixed Term License”).

La *Figura 1.1* también describe el por qué es tan sencillo actualizar su servidor desde el DB2 Express-C a una edición superior. Si usted tomara la decisión de hacer esta actualización en el futuro a cualquiera de los otros servidores DB2, todos ellos tienen los mismos componentes centrales, lo cual significa que cualquier aplicación desarrollada para una edición, funcionará sin modificaciones en otras ediciones, y todos los

conocimientos y habilidades aprendidos en una edición, serán válidos para las otras ediciones.

1.4 Clientes y controladores DB2

Todos los clientes DB2 incluyen las funciones necesarias para conectarse con un servidor DB2; sin embargo no siempre es necesario instalar uno de estos clientes. Por ejemplo, una aplicación JDBC de Tipo 4, solo requiere que se instale un controlador JDBC para conectarse con un servidor DB2. Los clientes y controladores DB2 vienen en diferentes sabores:

- IBM Data Server Client: el más completo, incluye herramientas con interfaz gráfica (GUI) y controladores
- IBM Data Server Runtime Client: un cliente ligero con funcionalidad básica y controladores incluidos
- DB2 Runtime Client Merge Modules for Windows: principalmente usado para incorporar un cliente DB2 de tiempo de ejecución como parte de la instalación de una aplicación Windows
- IBM Data Server Driver for JDBC and SQLJ: permite a las aplicaciones Java conectarse con servidores DB2 sin la necesidad de instalar un cliente completo
- IBM Data Server Driver for ODBC and CLI: permite a las aplicaciones ODBC y CLI conectarse a un servidor DB2 sin la carga de instalar un cliente completo
- IBM Data Server Driver Package: Incluye un controlador específico para Windows con soporte para ambientes .NET adicionalmente a aquellos para ODBC, CLI y fuente de código abierta. Este controlador se conocía anteriormente como el IBM Data Server Driver for ODBC, CLI and .NET.

nuevo en
V9.7

Figura 1.2 muestra los diferentes clientes y controladores disponibles para DB2.

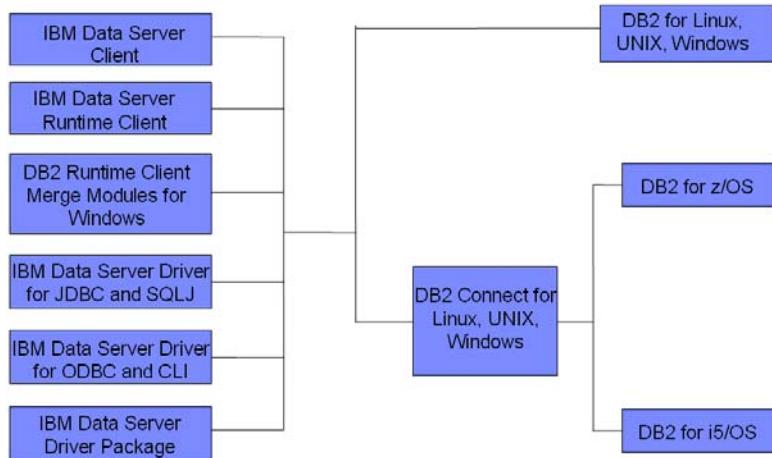


Figura 1.2 – Clientes y controladores para DB2

En el lado izquierdo de la *Figura 1.2*, puede usted ver todos los clientes y controladores de DB2. A pesar que todos los clientes DB2 incluyen los controladores requeridos, a partir de la versión 9 del servidor de datos DB2, estamos distribuyendo en forma individual los controladores. Todos los clientes y controladores se ofrecen en forma gratuita y están disponibles para ser descargados desde el sitio web del DB2 Express-C. Estos clientes y controladores pueden ser usados para conectarse a un servidor DB2 en Linux, UNIX o Windows. Para conectarse a un servidor DB2 for z/OS® o a un servidor DB2 for i5/OS®, usted necesita hacerlo a través de un servidor DB2 Connect™ (como se ve en la mitad de la *Figura 1.2*). Trataremos acerca del servidor DB2 Connect ("DB2 Connect") en el Capítulo 2.

Nota:

A pesar que este libro se enfoca en el servidor de datos DB2, los clientes del servidor de datos ("IBM Data Server clientes") pueden además conectarse con otros servidores de datos de la familia IBM, tal como Informix. Este es el motivo del nombre genérico "IBM Data Server client" en oposición con el más específico "DB2 client".

1.5 Libertad para el desarrollo de aplicaciones

DB2 ofrece un ambiente de desarrollo de aplicaciones basado en estándares y que es transparente a lo largo de toda la familia DB2. La estandarización de SQL a través de toda la línea de productos DB2 provee un conjunto común de interfaces de programación para el acceso a bases de datos.

Adicionalmente cada producto DB2 provee pre-compiladores SQL e interfaces de programación para las aplicaciones ("Application Programming Interfaces - APIs") que permiten a los desarrolladores incorporar sentencias SQL estáticas o dinámicas dentro de programas de aplicación transportables. DB2 incluso trae un proveedor nativo manejado por .NET e integración con herramientas para Microsoft® Visual Studio.

Los lenguajes y estándares que usted puede usar con DB2 incluyen:

- SQL, XQuery, XPath
- C/C++ (CLI, ODBC y SQL incorporado)
- Java (JDBC y SQLJ)
- COBOL
- PHP
- Perl
- Python

- Ruby on Rails
- Lenguajes .NET
- OLE-DB
- ADO
- MS Office: Excel, Access, Word
- Servicios Web

1.6 Versiones DB2 versus ediciones DB2

Si su relación con DB2 es reciente, es probable que la diferencia entre una versión y una edición DB2 no esté totalmente clara para usted.

Cada pocos años, IBM publica una nueva versión de DB2. Una versión incluye nuevas funciones y mejoras importantes al producto. Actualmente IBM da soporte a la Versión 9 de DB2 (DB2 Version 9). Una versión puede además tener algunas publicaciones ("releases"), las cuales son actualizaciones que pueden contener algunas nuevas funciones pero que no son tan importantes como para merecer una nueva versión. Por ejemplo, 9.5 y 9.7 son niveles de publicación de DB2 Versión 9. En los últimos años, IBM ha lanzado una nueva publicación de DB2 cada uno o dos años, en tanto las nuevas versiones están típicamente espaciadas a lo largo de tres o más años. La publicación más actual es V9.7, la cual fue hecha disponible para todos ("generally available" - GA) en junio de 2009. Cada publicación puede tener varios niveles de modificación, los cuales típicamente contienen correcciones o se corresponden con un nivel de paquetes de correcciones ("fix packs"), y difícilmente contengan funciones nuevas. Al momento de escribir este libro, el nivel de versión, publicación y modificación ("version", "release", "modification" - V,R,M) vigente para DB2 Express-C es 9.7.0 que corresponde al nivel de código con el paquete de correcciones 0, lo que significa que está a nivel de disponibilidad general (GA).¹

Por otro lado, las ediciones son ofertas seleccionadas o bien paquetes agrupados dentro de cada versión. Como ya se ha dicho, una edición es un paquete de diferentes funciones por una licencia a un precio dado. El DB2 Versión 9.7 (también conocido como DB2 9.7) tiene varias ediciones; por ejemplo, DB2 Express-C 9.7, DB2 Express 9.7, DB2 Workgroup 9.7, y DB2 Enterprise 9.7 (*Figura 1.1*).

¹ Al momento de la traducción de este libro (Mayo, 2011) el nivel de DB2 Express-C es 9.7.4 (9.5.2 para la versión beta para Mac OS X - N. del T.)

1.7 Avanzando a otra edición de DB2

A medida que su base de datos precisa crecer, usted puede requerir actualizar su instalación a una edición de DB2 que soporte computadores más grandes. Si este fuera el caso, es sencillo cambiarse a otra edición DB2:

- Si la actualización a la nueva edición de DB2 se realiza en el mismo computador, basta con desinstalar DB2 Express-C, y luego instalar la nueva edición DB2. Tenga en cuenta que cuando se desinstala DB2 Express-C, sus bases de datos no son eliminadas (pero siempre se recomienda tomar un respaldo de ellas antes del cambio).
- Si la nueva edición de DB2 será instalada en un computador diferente de mayor capacidad, con el mismo sistema operativo, entonces instale la nueva versión en el nuevo computador, tome un respaldo de sus bases de datos en el computador anterior, mueva las imágenes de respaldo al nuevo computador, y finalmente restaure dichas imágenes en el computador nuevo. También es necesario salvar los valores de configuración de la instancia (dbm cfg) del computador anterior, y aplicar esta configuración en el computador nuevo. Los comandos para respaldar (“*backup*”) y recuperar (“*restore*”) son tratados con mayor detalle en el *Capítulo 11, Respaldo y Recuperación*. La configuración de la instancia (dbm cfg) se ve con detalle en el *Capítulo 5, El Ambiente DB2*.
- En todos los casos, sus aplicaciones cliente no requieren modificación alguna.

1.8 Mantenimiento y actualizaciones para el DB2 Express-C

Las imágenes de instalación del DB2 Express-C son refrescadas periódicamente. Estas actualizaciones generalmente coinciden con la disponibilidad de nuevas publicaciones (“*releases*”) o versiones (“*versions*”) o cuando se ha acumulado un número significativo de correcciones de errores para el producto. Hasta el momento, las actualizaciones para el DB2 Express-C han estado disponibles una vez al año. Sin embargo, tenga en cuenta que siendo el DB2 Express-C una oferta sin costo ni garantías, no se produce para él ninguna publicación de mantenimiento (“*maintenance release*”) oficial ni existen paquetes de correcciones (“*fixpacks*”) programados (aquellos que se publican varias veces en el año). Una vez que una nueva versión o publicación de DB2 Express-C está disponible, todas las publicaciones anteriores dejan de tener mantenimiento.

Como ya se ha dicho, si usted requiere acceso a correcciones de seguridad y actualizaciones planificadas del producto, o bien paquetes con correcciones de fallas (“*bug fixes*”), IBM ofrece una licencia de suscripción anual (FTL). Una vez que usted compra esta suscripción, su instalación de DB2 Express-C puede ser actualizada con la clave para la licencia FTL, lo que le permite acceder al soporte técnico DB2 y a las actualizaciones y paquetes de corrección durante el periodo de validez de la licencia de suscripción. Esta licencia también lo autoriza para obtener gratuitamente actualizaciones de versión, o, si usted lo prefiere, tiene la flexibilidad de permanecer en una determinada versión o

publicación, y tan solo aplicar los paquetes de corrección y actualizaciones de seguridad por todo el tiempo que esa publicación siga siendo soportada y usted mantenga su suscripción anual vigente.

1.9 Otros programas y componentes DB2 relacionados que se ofrecen en forma gratuita

Todos los programas que están disponibles para descarga en la página de descargas de DB2 Express-C (www.ibm.com/db2/express/download.html) son libres de cargo. Además de los programas de DB2 Express-C, existen otros paquetes de programas muy útiles que también pueden descargarse y usarse sin costo:

- Visual Studio Add-ins
- DB2 Spatial Extender

También puede usted encontrar paquetes de herramientas preparados para que pueda comenzar a usar ciertos productos basados en DB2 Express-C. Ellos están disponibles para descarga en el sitio web IBM Alphaworks (www.alphaworks.ibm.com/datamgmt):

- Starter Toolkit for DB2 on Rails (www.alphaworks.ibm.com/tech/db2onrails/)
- Web 2.0 Starter Toolkit for DB2 (www.alphaworks.ibm.com/tech/web2db2)

Si usted está buscando un servidor de aplicaciones liviano (que también es gratis), IBM le ofrece:

- WebSphere® Application Server – Community Edition (WAS CE)

1.9.1 IBM Data Studio

IBM Data Studio es una herramienta basada en Eclipse que le permite administrar sus bases de datos y le ayuda a desarrollar consultas en XQuery y SQL, funciones definidas por el usuario (“*user-defined functions*” – UDFs), y procedimientos almacenados (“*stored procedures*”). También incluye un depurador de errores (“*debugger*”). Adicionalmente, el IBM Data Studio le permite trabajar con diagramas de modelamiento físico de datos (“*physical data modeling*” - PDM) y a través de ellos entender la relación de entidad entre las tablas. Además le ayuda en el desarrollo y publicación de datos como un servicio de Web usando el concepto de arrastre-y-suelte (“*drag-and-drop*”) que no requiere programación alguna. El IBM Data Studio reemplaza otras herramientas DB2 como ser el Centro de Control (“*Control Center*”) y el Editor de Comandos (“*Command Editor*”), los cuales están ya obsoletos (se incluyen con DB2, pero no están más bajo desarrollo). Mayores detalles acerca del IBM Data Studio los encontrará en el *Capítulo 5, Herramientas DB2*.

1.9.2 DB2 Text Search

El DB2 Text Search es un componente opcional integrado a DB2. Se basa en la tecnología IBM OmniFind™, y le permite realizar poderosas, rápidas y completas búsquedas de texto dentro de documentos de texto, incluyendo documentos XML almacenados en DB2 en forma nativa. Este componente usa procesamiento lingüístico para encontrar diferentes formas del término de búsqueda dentro del texto. Por ejemplo, si usted está buscando la palabra "estudio", DB2 Text Search también encontrará otras formas de la palabra, como ser "estudios" o "estudiado".

Para instalar el DB2 Text Search, al instalar DB2 Express-C, elija la opción de instalación personalizada y elija la opción de DB2 Text Search dentro de la categoría de soporte del servidor.

Nota:

Una funcionalidad similar está disponible en una extensión de DB2 llamada "Net Search Extender" (NSE)². Actualmente NSE se está dejando de usar y tornándose obsoleto en favor del DB2 Text Search,

1.9.3 WebSphere Application Server – Community Edition

El IBM WebSphere Application Server - Community Edition (WASCE) es una servidor de aplicaciones Java EE 5 ligero que está disponible libre de cargo. Basado en la tecnología *Apache Geronimo*, incorpora las últimas innovaciones provenientes de la comunidad de código fuente abierto ("open source") para entregar una base integrada, accesible y flexible donde desarrollar e implementar aplicaciones Java. Opcionalmente, a través de una suscripción anual, puede contratarse soporte técnico para el WASCE.

1.10 Resumen

La edición DB2 Express-C ofrece el mejor producto dentro de su clase sin costo alguno. Proporciona la libertad de desarrollar, instalar y distribuir sin ninguna limitación en el tamaño de la base de datos, manteniendo la misma funcionalidad base y la tecnología pureXML presentes en otras ediciones de DB2. El DB2 Express-C soporta una amplia

² Sin embargo el "Net Search Extender" sólo permite búsquedas lingüísticas en inglés. El "DB2 Text Search" permite realizar dichas búsquedas en todos los lenguajes soportados por DB2, incluyendo el español (N. del T.)

30 Conociendo al DB2 Express-C

gama de clientes, controladores y lenguajes de desarrollo y provee un camino fácil para actualizarse a otras ediciones DB2.

2

Capítulo 2 – Características y productos relacionados

Este capítulo describe características de DB2 incluidas con la compra de DB2 Express-C con licencia de suscripción de 12 meses. También incluye otras características no disponibles con DB2 Express-C, pero como parte de otras ediciones de DB2, en algunos casos, con pagos adicionales.

Las características incluidas en la versión gratuita de DB2 Express-C son:

- La funcionalidad base de DB2
- Control Center, Data Studio y otras herramientas de manejo
- pureXML
- Utilización de recursos hasta 2GB y 2 núcleos
- Disponible en Linux, Windows, y Solaris (x86)

Las capacidades disponibles con la licencia de 12 meses de DB2 Express-C son:

- Fix packs
- Alta disponibilidad y Recuperación Critica (HADR)
- Data Replication (Homogenous SQL)
- Utilización de recursos hasta 4GB y 4 núcleos (en 2 tomas)

La siguiente table lista características del producto y si están o no incluidas con las distintas ediciones de DB2. Características que pueden ser compradas separadamente están listadas por nombre para la edición correspondiente.

Funcionalidad	Express-C fixed term license	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
Replicación SQL Homogénea	Sí	Sí	Sí	Sí
Net Search	Sí	Sí	Sí	Sí

32 Conociendo al DB2 Express-C

Funcionalidad	Express-C fixed term license	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
Extender				
Spatial Extender	Sí	Sí	Sí	Sí
Tecnología pureXML™	Sí	pureXML	pureXML	pureXML
Alta disponibilidad y Recuperación Critica	Sí	Alta Disponibilidad	Sí	Sí
Tivoli® System Automation	Sí	Alta Disponibilidad	Sí	Sí
Advanced Copy Services	No	Alta Disponibilidad	Sí	Sí
Reorganización en linea	No	Alta Disponibilidad	Sí	Sí
Federación Homogénea	No	Federación Homogénea	Federación Homogénea	Federación Homogénea
MQT	No	No	Optimización de consultas	Sí
MDC	No	No	Optimización de consultas	Sí
Paralelismo de consultas	No	No	Optimización de consultas	Sí
Conexión concentrada	No	No	No	Sí

Funcionalidad	Express-C fixed term license	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
Partición de tablas	No	No	No	Sí
DB2 Governor	No	No	No	Sí
Compresión a nivel fila	No	No	No	Optimización de almacenamiento
Compresión: backup	No	No	No	Optimización de almacenamiento
Label-based access control (LBAC)	No	No	No	Advanced Access Control
Geodetic Extender	No	No	No	Geodetic Data Management
Query Patroller	No	No	No	Optimización de Performance
DB2 workload management	No	No	No	Optimización de Performance
Performance Expert	No	No	No	Optimización de Performance
Replicación Q Homogénea	No	No	No	Homogeneous Replication
Particionamiento de base de datos	No	No	No	No

Table 2.1: Ediciones de DB2 Versión 9.5: características y funcionalidad soportada

Características incluídas en otras ediciones de DB2:

Características pagadas en la edición DB2 Express

- pureXML
- Alta disponibilidad
- Federación Homogénea

Características incluidas sin costo en la edición DB2 Workgroup:

- Alta disponibilidad
- Disponible en AIX®, Solaris, y HP-UX además de Linux y Windows

Características pagadas en la edición DB2 Workgroup

- pureXML
- Optimización de Performance (MQT, MDC, Query Parallelism)
- Federación Homogénea

Características incluidas sin costo en la edición DB2 Enterprise:

- Particionamiento de Tablas (Range)
- Materialized Query Tables (MQT)
- Multi-dimensional Clustering (MDC)
- Alta disponibilidad y Recuperación Critica (HADR)
- Conexión Concentrada

Características pagadas en la edición DB2 Enterprise

- pureXML
- Optimización de almacenamiento. (incluido compresión)
- Control de acceso avanzado (seguridad avanzada)
- Optimización del Performance (Performance Experto, Query Patroller)
- Administración de datos Geodetic
- Federación Homogénea DB2

Productos pagados relacionados a DB2:

- DB2 Connect
- DB2 Warehouse Editions
- WebSphere® Federation Server
- WebSphere Replication Server

2.1 Características incluidas con la licencia de suscripción de DB2 Express-C

Esta sección explica los Fix packs de DB2, HADR y réplica de datos.

2.1.1 Fix Packs

Un fix pack de DB2 es un paquete de código corregido aplicado en un producto DB2 instalado, para corregir publicaciones diferentes después de que el producto fue liberado. Con una licencia de suscripción, la corrección de paquetes es gratis de descargar e instalar. Ellos están típicamente disponibles cada tres meses.

Para bajar los últimos fix packs, revisa esta página del soporte técnico de DB2 en
http://www.ibm.com/software/data/db2/support/db2_9/

2.1.2 Alta disponibilidad de Recuperación Crítica (HADR)

Alta disponibilidad de Recuperación Crítica (HADR) es una característica de fiabilidad de base de datos que proporciona una alta disponibilidad y la solución de recuperación de datos ante desastres de sitio completos así como parciales. Un ambiente HADR generalmente consiste en dos servidores de datos, el primario y el secundario (que puede estar en posiciones geográficamente distantes). El servidor primario es donde la fuente de base de datos es almacenada y accesada por aplicaciones de clientes. Como las transacciones son procesadas sobre la base de datos primaria, los registros de la base de datos automáticamente son transportados al servidor secundario a través de la red. El servidor secundario tiene una copia clonada de la base de datos primaria, por lo general creada por el back-up de la base de datos primaria y restaurándolo sobre la base de datos secundaria. Cuando los registros de la base de datos primaria son recibidos ellos son reproducidos de nuevo y aplicados a la base de datos secundaria. Por la repetición continua de los registros, la base de datos secundaria mantiene una réplica en sincronización de la base de datos primaria que puede asumir si la base de datos primaria falla.

Una solución completa de DB2-supported HADR nos da lo siguiente:

- Capacidad rápida de solución ante fallas, con transparencia completa para clientes y aplicación de cliente
- Atomicidad completa en las transacciones para prevenir perdida de datos
- Habilidad de actualizar sistemas o aplicaciones sin interrupción de servicios visibles
- Sistema remoto de fallas, proporcionando recuperación completa de desastre local que puede dañar el centro de datos
- Administración fácil con herramientas gráficas de DB2
- Todos esto con un impacto insignificante sobre funcionamiento total del sistema

Nota:

Para ver una demostración (en inglés) de cómo trabaja el HADR, puedes visitar:

<http://www-306.ibm.com/software/data/db2/express/demo.html>

nuevo en
V9.7

La nueva versión, DB2 9.7, brindará la posibilidad de permitir a los clientes realizar lecturas sobre el servidor secundario. Esta capacidad estará disponible con el Fixpack1 de DB2 9.7

2.1.3 Réplica de Datos

Esta característica permite la réplica de datos entre un servidor fuente donde los cambio de datos son capturados, y un servidor objetivo donde el cambio de datos son aplicados. Figura 2.1 proporciona una descripción de cómo trabaja la réplica.

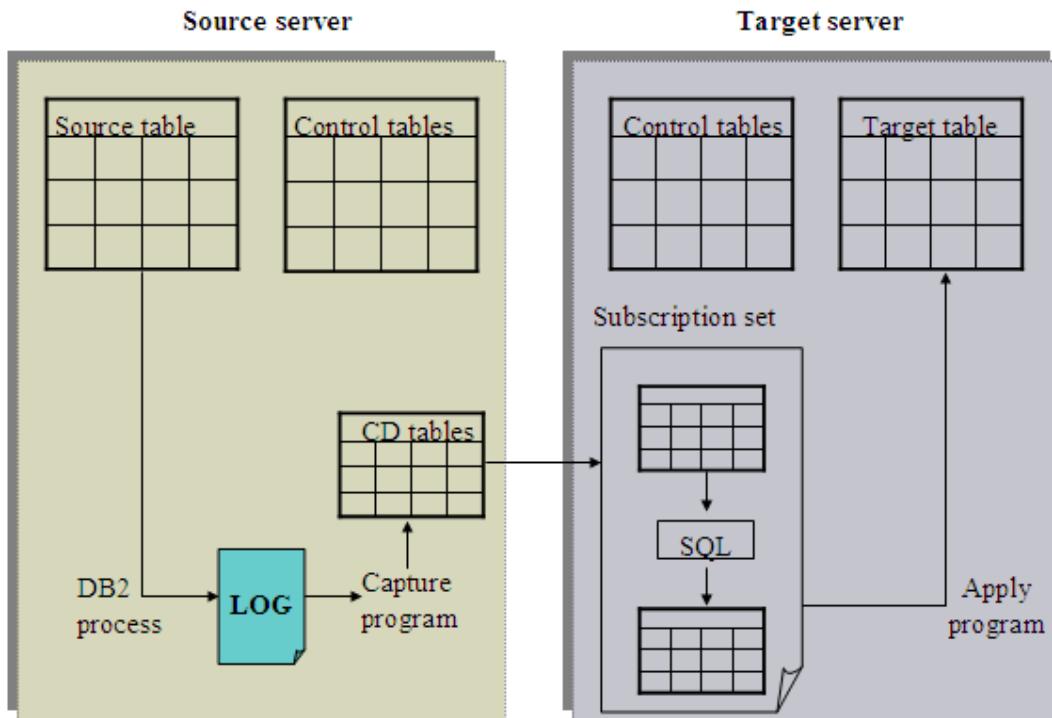


Figura 2.1 –Ráplica de SQL

En la Figura 2.1 hay dos servidores, un servidor fuente y un servidor objetivo. En el servidor fuente, el programa “Capture” captura el cambio hecho en la base de datos. En el servidor objetivo, un programa “Apply” aplica los cambios de la base de datos replica. La réplica es útil para una variedad de propósitos que requieren replica de datos, incluyendo alivio de capacidad, sosteniendo el data warehouse y el mercado de datos, revisando el historial de cambios. Usando las características de la réplica SQL puedes replicar datos entre DB2 Express-C y otros servidores DB2, incluyendo aquellos en otros sistemas Linux, UNIX, z/OS, y i5/OS.

2.2 Características no disponibles con DB2 Express-C

Esta sección describe alguna de las características disponibles en otras ediciones de DB2 pero no en DB2 Express-C.

2.2.1 Particionamiento de la Base de Datos

La característica de particionamiento de la base de datos (DPF) es solo disponible con DB2 Enterprise Edition con la licencia de pago adicional. Esto permite a la base de datos ser extendido a través de múltiples particiones las cuales pueden estar alojados en varias computadoras. DPF está basado en una arquitectura shared-nothing.

Cada computadora, como es añadido al grupo de partición, trae el poder adicional de procesamiento de datos con sus propias CPUs y memoria. DPF es en particular útil en ambientes de servidor de datos grandes como data warehouses donde las sentencias de los sistemas de apoyo de decisión (DSS) son controlados.

2.2.2 Concentrador de conexión

El concentrador de conexiones es una característica que permite el soporte de un largo número de usuarios conectados simultáneamente. Previamente, toda conexión a la base de datos requiere un “agente” de base de datos. El concentrador de conexiones introduce el concepto de un “agente lógico”, permitiendo un agente para mantener muchas conexiones. El concepto de “agentes” serán discutidos con más detalle en el Capítulo 6, Arquitectura DB2.

2.2.3 Geodetic Extender

DB2 Geodetic Extender esta disponible con pago adicional para DB2 Enterprise Edition. Este extender hace desarrollos para aplicaciones de inteligencia de negocios y e-government que requieren un análisis de localización geográfica mucho más sencillo. DB2 Geodetic Extender puede construir un globo mundial virtual a cualquier escala. La mayoría de información de localización es tomada usando sistemas worldwide, por ejemplo el sistema de satélites de posicionamiento global (GPS), y puede ser representado en coordenadas de latitud/longitud (geocódigo). Los datos de negocios, como direcciones, pueden ser convertidos a geocódigo con DB2 Geodetic Extender y aplicaciones a nivel empresarial trabajan mejor cuando mantienen los datos en esta forma sin proyección, saliendo las proyecciones del mapa (tierra a mapa plano) donde ellos pertenecen, en la capa de presentación, para mostrar e imprimir mapas.

2.2.4 Control de Acceso basado en Etiqueta (Label-based Access Control) (LBAC)

Provee seguridad granular a nivel de fila y columna. Utiliza una etiqueta asociada con las sesiones de usuario y las filas o columnas, para permitir el acceso a los datos de la tabla. La *Figura 2.2* muestra cómo trabaja LBAC.

No LBAC	SEC=254	SEC=100	SEC=50	ID	SALARY
				255	60000
SELECT * FROM EMP WHERE SALARY >= 50000				100	50000
				50	70000
				50	45000
				60	30000
				250	56000
				102	82000
				100	54000
				75	33000
				253	46000
				90	83000
				200	78000

Figura 2.2 – Ejemplo de cómo trabaja LBAC

En la figura 2.2, la tabla *EMP* tiene una columna, *SALARY*, y una columna interna, *ID*, que contiene la etiqueta para una fila determinada. Las otras columnas de la figura se utilizan solamente para fines ilustrativos. Si la consulta que se muestra en la figura se ejecuta, dependiendo de la etiqueta que el usuario tenga, le será posible ver diferentes filas. La columna de título ‘No LBAC’ representa las filas que se seleccionarían si LBAC no estuviese implementado. Como se puede apreciar, la consulta solicita todas las filas con salario mayor o igual a 50,000.

Ahora supongamos que el usuario que ejecuta la consulta tiene la etiqueta de seguridad 100. Como se puede ver en este caso, las filas seleccionadas son aquellas donde la columna ID indica un valor de 100 o menor. Es decir, se seleccionan las filas cuyo salario es mayor o igual a 50,000 y no se pueden obtener las filas de etiquetas con valores superiores a 100. La primer fila tiene como salario 60000 y un label ID de 255. Como el usuario tiene un valor de ID menor a 255, no tiene la posibilidad de ver esta fila, y por lo tanto, el resultado de la consulta no devolverá este registro, cuyo salario es 60000.

La seguridad LBAC tiene que ser implementada por un administrador con autoridad SECADM.

2.2.5 Workload Manager (WLM)

Maneja cargas de trabajo en una base de datos en base a prioridades de un usuario y de aplicaciones combinadas con la disponibilidad de recursos y límites. Permite regular la carga de trabajo de tu base de datos y consultas, de tal manera que consultas importantes y de alta prioridad pueden correr inmediatamente, y , and previene consultas espontáneas

que no tienen buen rendimiento de monopolizar los recursos del sistema de manera que tu sistema puede correr eficientemente. WLM is Nuevo en DB2 9.5 y prove mas poder que las características de las herramientas Query Patroller y del DB2 Governor que eran disponibles en versiones anteriores de DB2. WLM ha sido mejorado aún más con DB2 9.7 ya que provee capacidades más poderosas que las herramientas Query Patroller y DB2 Governor de las versiones anteriores de DB2.

nuevo en
V9.7

2.2.6 Alta compresión

DB2 soporta varios mecanismos de compresión:

- NULL y el valor de compresión por omisión.

Este tipo de compresión es aplicable a columnas cuyos valores sean generalmente NULL o los valores por omisión del sistema, tal como el 0, que no ocupa espacio en disco.

- Clustering Multidimensional

Tablas de Clustering Multidimensional (MDC) donde las páginas físicas de datos están agrupadas en dimensiones múltiples. Estas, usan bloques de índices, que es una forma de comprimir los índices porque apuntan a bloques de registros en vez de a un solo registro.

- Compresión del Resguardo de la Base de datos

Esto aplica a las imágenes de backup. Los índices y tablespaces LOB se comprimen.

- Compresión a nivel de fila

La compresión a nivel fila trabaja reemplazando las cadenas de caracteres que se repiten dentro de una fila por un símbolo mucho más pequeño. El mapeo de este símbolo y de la cadena de caracteres se guarda en el diccionario. La compresión de fila puede mejorar el rendimiento drásticamente en cargas de trabajo de fuerte E/S ya que una mayor cantidad de filas se pueden transferir entre el disco y la memoria (y viceversa) debido a que las filas son mucho más chicas. También se ahorra en espacio en disco, el cual, generalmente, ocupa una gran parte del presupuesto de TI dentro de las compañías. Los sistemas de gran consumo de CPU pueden recurrir en una exigencia mayor debido a que las filas comprimidas necesitan descomprimirse antes de ser procesadas. Para tener en cuenta también: los logs de datos comprimidos también están en formato comprimido.

Cuando se accede a columnas XML y LOB, generalmente DB2 no utiliza el bufferpool (memoria), sino que realiza operaciones directas de E/S de disco. Esto sucede porque los XMLs y los LOBs en general, son de gran tamaño; por lo tanto, traerlos a memoria causaría que las páginas necesiten limpiarse de la misma. Con DB2 9.5, sin embargo, se permite el XML inline para documentos XML pequeños (menores a 32K). Esto significa que documentos XML pequeños pueden ser almacenados en conjunto con las filas, y no

en un objeto interno separado que se denomina XDA. Esta alternativa tiene dos ventajas: Primero, los documentos XML pueden ser accedidos a través de la buffer pool, y segundo, los documentos XML también se pueden beneficiar de la compresión por fila

nuevo en
V9.7

DB2 9.7 incluye aún más mejoras de compresión de datos:

- Los objetos internos XDA (donde se almacena el XML) se pueden también comprimir.
- Los índices y las tablas temporales (tanto de sistema como de usuario) puede ser comprimidas también.
- Los LOBs pueden incluirse inline de misma forma que los XML mencionados anteriormente.

nuevo en
V9.7

2.2.7 Compatibilidad SQL

Si bien muchas compañías siguen los estándares SQL 92 y SQL/PSM, no todas las características de los estándares son soportadas. Además hay características fuera de los estándares que sí las soportan estas compañías. Con la compatibilidad SQL de DB2 9.7, ahora sí puede soportar la mayoría de la sintaxis PL/SQL, la cual está incluida por otras compañías de SGBD además del propio SQL PL de DB2. La Figura 2.3 resume cómo trabaja esta nueva compatibilidad.

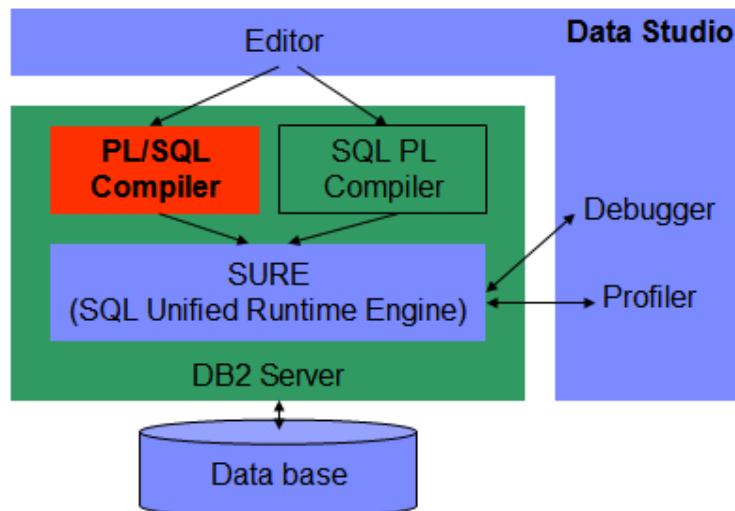


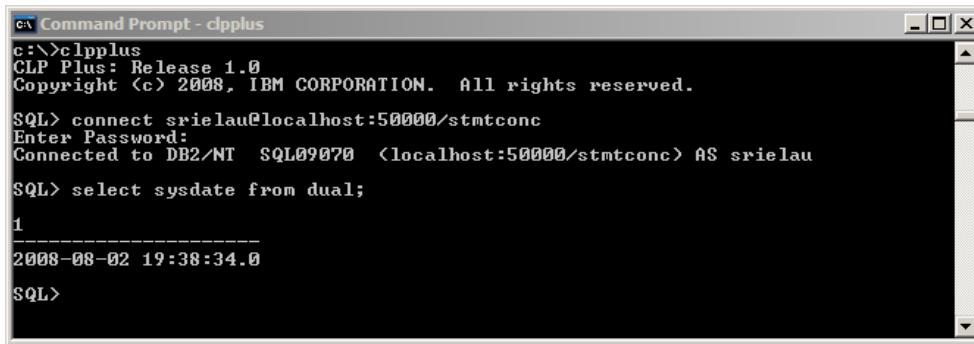
Figura 2.3 – Soporte PL/SQL en DB2

La figura muestra que un compilador PL/SQL ha sido desarrollado e incluído dentro del motor de DB2.

La característica de compatibilidad también incluye soporte para una herramienta llamada CLPPlus, la cual es una herramienta de línea de comando que permite ejecutar sentencias

SQL y otros comandos. Es una herramienta parecida al Procesador de Línea de Comando de DB2 (CLP).

En la Figura 2.4 se ilustra la herramienta CLPPlus.



```
c:\>clpplus
CLP Plus: Release 1.0
Copyright (c) 2008, IBM CORPORATION. All rights reserved.

SQL> connect srielau@localhost:50000/stmtconc
Enter Password:
Connected to DB2/NT SQL09070 <localhost:50000/stmtconc> AS srielau

SQL> select sysdate from dual;
1
-----
2008-08-02 19:38:34.0
SQL>
```

Figura 2.4 –CLPPlus

Se ha incorporado soporte para la mayoría de los tipos de datos de PL/SQL como BINARY_INTEGER, RAW, y demás. Otros tipos de datos como VARCHAR2 también se soportan sin la necesidad de la característica de Compatibilidad SQL, pero se necesita habilitar la misma por medio de la variable de registro DB2_COMPATIBILITY_VECTOR. Se explican más tipos de datos de Oracle y se comenta esta variable en subsiguientes partes del libro.

Las características de Compatibilidad SQL señaladas anteriormente están disponibles en las ediciones Workgroup y Enterprise de DB2 9.7. Se espera que las mismas se incluyan en la edición Express de DB2 (incluyendo la opción anual de Suscripción o FTL) en el futuro próximo.

Si bien las características CLPPlus y el soporte PL/SQL no están disponibles en DB2 Express-C 9.7, otras si lo están, con el propósito de facilitar el funcionamiento de aplicaciones Oracle sobre DB2. Estas incluyen nuevos tipos de datos, nuevas funciones escalares, soporte de módulo, y semánticas de actualmente confirmado (currently committed) (CC) para el nivel de aislamiento estabilidad de cursor (cursor stability) (CS). Estas características se discuten más adelante en el libro.

2.3 Productos pagados relacionados con DB2

2.3.1 DB2 Connect

DB2 Connect es un software pagado que permite al cliente DB2 para Linux, UNIX o Windows conectarse a un servidor DB2 para z/OS o para i5/OS, como se muestra en la Figura 2.2. DB2 Connect no es requerido cuando la conexión ocurre en la dirección opuesta: cuando conectas desde DB2 para z/OS o i5/OS hacia DB2 para Linux, UNIX o Windows. DB2 Connect viene en dos ediciones principales dependiendo de tus

necesidades de conexión: DB2 Connect Personal Edition y DB2 Connect Enterprise Edition.

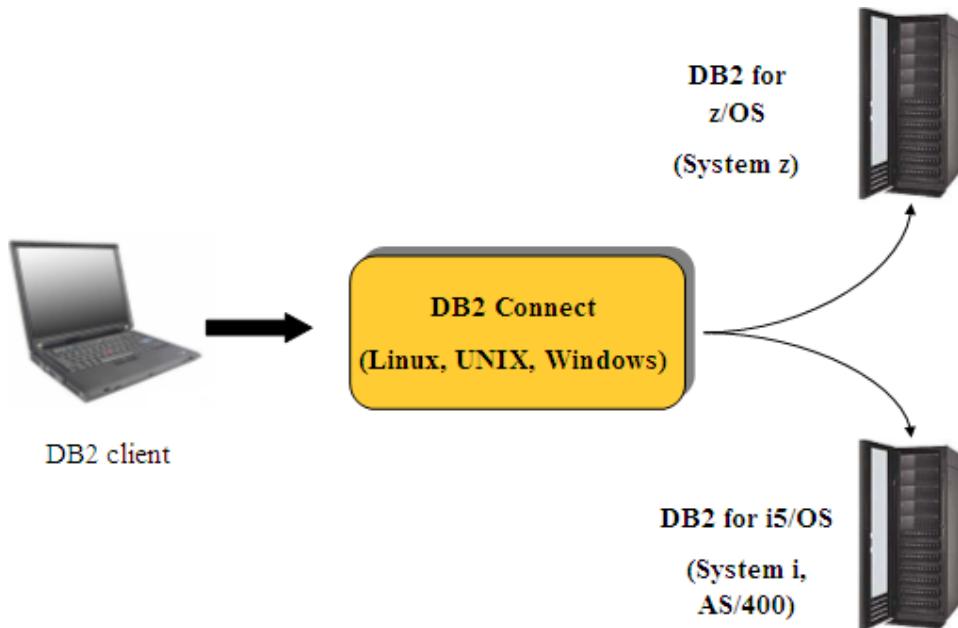


Figura 2.2 – DB2 Connect

2.3.2 InfoSphere Federation Server

Antiguamente conocido como WebSphere Information Integrator (para soporte de federación), InfoSphere Federation Server permite una federación de base de datos; significando que puedes correr consultas de base de datos que puedan trabajar con objetos de diferentes sistemas de base de datos relacional. Por ejemplo, si compras InfoSphere Federation Server, puedes correr la siguiente consulta:

```
SELECT *  
FROM Oracle.Table1 A  
      DB2.Table2 B  
      SQLServer.Table3 C  
WHERE
```

```

A.col1 < 100
and B.col5 = 1000
and C.col2 = 'Test'

```

Figura 2.3 provee una ilustración donde InfoSphere Federation Server es usado.

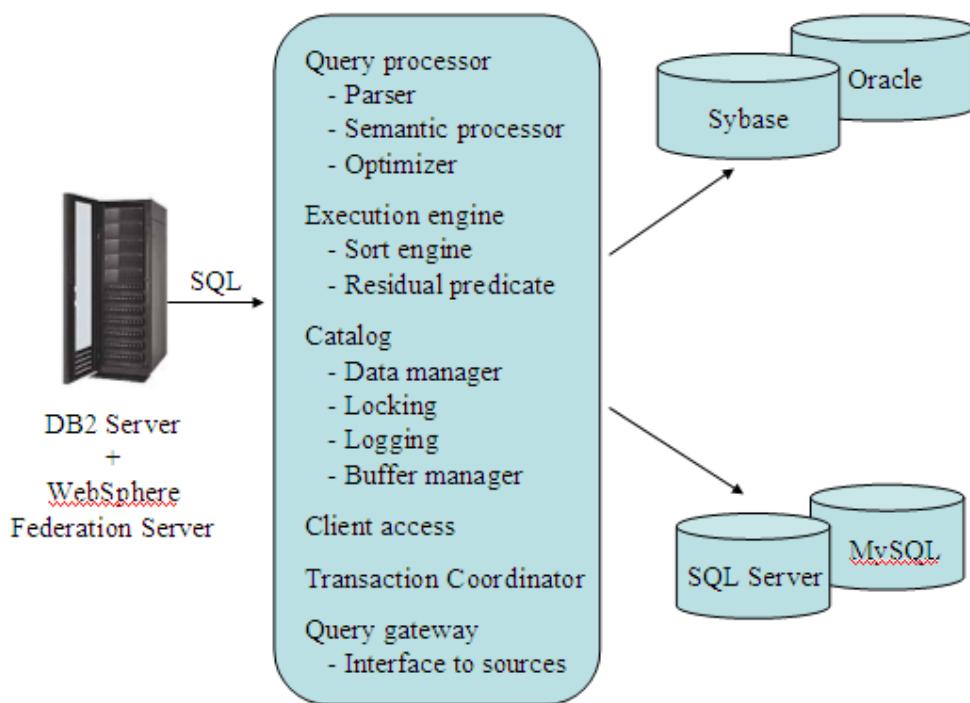


Figure 2.3 – InfoSphere Federation Server

Para sistemas de manejo de base de datos relacionales que son parte de la familia IBM, soporte de federaciones está incluido en DB2 Express-C. Esto significa que InfoSphere Federation Server no es requerido cuando, por ejemplo, puedes correr una consulta entre dos base de datos diferentes, o entre una base de datos DB2 y una base de datos Informix (Informix es parte de la familia IBM).

2.3.3 InfoSphere Replication Server

Antiguamente conocido como WebSphere Information Integrator (para soporte de replicación), InfoSphere Replication Server permite replicación SQL en registros de base de datos cuando servidores de datos que no son de IBM están involucrados. Esto también incluye una característica conocida como Q-Replication para replicar información usando cola de mensajes.

2.3.4 Optim Development Studio (ODS)

Anteriormente conocido como Data Studio Developer, ODS es una herramienta basada en Eclipse que puede ser fácilmente integrada con el Data Studio, y compartir el mismo Eclipse. ODS brinda la posibilidad de crear bases de datos de desarrollo a partir de bases de datos Oracle o DB2 ya existentes.

2.3.5 Optim Database Administrator (ODA)

Anteriormente conocido como Data Studio Administrator, ODA es una herramienta basada en Eclipse que puede ser fácilmente integrada con Data Studio y compartir el mismo Eclipse. ODA posibilita una capacidad de administración de cambio y la habilidad de automatizar cambios del esquema de manera sencilla.

2.4 DB2 se ofrece a través de la nube de cómputo elástica de Amazon

Vale la pena mencionar que IBM se asoció a los Servicios Web de Amazon (AWS) para poder ejecutar DB2 en la nube de cómputo elástica de Amazon (EC2). AWS entrega un conjunto de servicios integrados que forman una plataforma de cómputo “en la nube”, y está disponible bajo el modelo pague-lo que use (pay-as-you-go). Esto quiere decir que AWS permite ‘alquilar/rentar’ capacidad de cómputo (almacenamiento y servidores virtuales), y pagar solamente por la capacidad que se utilice. Por ejemplo, digamos que se provisiona de un servidor virtual EC2 para operaciones de bases de datos normales, y durante la hora pico o por necesidades puntuales se provisiona de un servidor de base de datos adicional por unas pocas horas. En este caso, solamente se paga AWS por el servidor de base de datos extra por las horas utilizadas.

IBM ofrece tres opciones de implementación de DB2 sobre la plataforma de nube de Amazon:

- DB2 Express-C AMIs para implantación y evaluación
- Pague-lo que use (pay-as-you-go) Listo para-producción (Production-ready) AMIs con DB2 Express y DB2 Workgroup
- Habilidad de crear propias AMIs utilizando licencias de DB2 ya adquiridas

Para más información y cómo comenzar con DB2 sobre Amazon EC2, por favor visitar:
ibm.com/db2/cloud

2.5 Resumen

DB2 Express-C posibilita sin-costo y fácil de usar, una base robusta para desarrollar aplicaciones de bases de datos, impactarlas en producción, e incluso embeber y distribuir las aplicaciones con soluciones de terceros. Es ideal si se está cómodo con la asistencia basada en la comunidad, de manera tal de no tener que necesitar los últimos fixes o características avanzadas. Si se requiere de un soporte técnico formal de parte de IBM, de actualizaciones de software regulares (fixpacks), o utilización de recurso adicional y

soporte de clustering para alta disponibilidad, IBM ofrece una licencia de suscripción (FTL) para DB2 Express a un bajo costo anual. Si se necesita de características más avanzadas para cargas masivas de datos y de misión crítica, IBM ofrece ediciones de DB2 más escalables en conjunto con productos relacionados. Esto permite comenzar de lo más básico con DB2 Express-C, y escalar hacia nuevas alturas en función de las demandas del negocio.

3

Capítulo 3 – Instalación de DB2

Instalar DB2 es bastante sencillo, y en una instalación típica, simplemente eligiendo las opciones por omisión se tendrá un servidor DB2 activo y en funcionamiento en poco tiempo.

3.1 Requisitos previos para la instalación

El DB2 Express-C está disponible en Linux®, Sun Solaris (x64), and Microsoft Windows® 2003, XP, y Vista³. También está disponible una versión beta para Mac OS X. Las arquitecturas de procesador disponibles son 32-bit, 64-bit y PowerPC (Linux). Si usted necesita ejecutar DB2 en otra plataforma (por ejemplo UNIX), es necesario que adquiera una de las diferentes ediciones del servidor de datos que se han descrito anteriormente en este libro. Los requerimientos a nivel de sistema operativo para todas las ediciones de DB2 se encuentran detalladas en este documento:

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

En términos de recursos de “hardware”, el DB2 Express-C puede ser instalado en sistemas con cualquier número de núcleos de procesadores y cantidad de memoria, sin embargo, solo utilizará hasta dos núcleos y 2GB de memoria para la versión gratuita sin garantía, y hasta 4 núcleos y 4 GB de memoria para la versión con suscripción paga de DB2 Express. Los sistemas pueden ser sistemas físicos, o sistemas virtuales creados por partición o por programas de máquina virtual. Por supuesto, si usted lo prefiere, puede ejecutar DB2 Express-C en sistemas más reducidos, por ejemplo con un procesador de núcleo único y 1 GB de memoria.

Para la última información disponible sobre los requerimientos de “hardware” para el DB2 Express-C, visite la página Web del DB2 Express-C en
<http://www.ibm.com/software/data/db2/express/about.html>

3.2 Autoridad de instalación a nivel de sistema operativo

Para instalar el DB2 Express-C en Linux o Windows, se requiere de un usuario del sistema operativo con la autoridad suficiente.

³ Al momento de esta traducción, Windows 7 (Ediciones Professional, Enterprise y Ultimate) y Windows Server 2008 también son soportados (N. del T.)

Para **Linux**, usted necesita ser root (el superusuario) para instalar el DB2 Express-C. También puede instalar el DB2 Express-C desde un usuario no-root, pero tendrá limitaciones en lo que pueda hacer con el producto. Por ejemplo, bajo una instalación no-root, no podrá crear más instancias fuera de aquella creada por omisión al momento de la instalación.

En **Windows**, el usuario debe pertenecer al grupo de administradores en la máquina donde se ejecutará la instalación. Alternativamente, en Windows 2008, Windows Vista, o superior un no-administrador puede ejecutar la instalación, pero el Asistente de instalación de DB2 le exigirá credenciales de administrador.

En caso que la instalación requiera de una cuenta de dominio, el usuario deberá pertenecer al grupo de Administradores de Dominio en el dominio dónde se instala.

También puede ejecutar la instalación utilizando la cuenta incorporada de Sistema Local, aun cuando esto no se recomienda. Una cuenta Local del Sistema no requiere de contraseña, pero no puede tener acceso a recursos de la red.

La cuenta de usuario debe además tener el derecho a "Accesar este computador desde la red".

Nota:

Vea una presentación sobre la instalación del DB2 Express-C (en inglés) siguiendo este enlace. A pesar que la presentación muestra una instalación de DB2 9.5, no hay mayor diferencia con la instalación del DB2 9.7, más allá del color de los paneles:

<http://www.channeldb2.com/video/video/show?id=807741;Video:4442>

3.3 Asistente de instalación

Aun cuando hay varios métodos para instalar el DB2 Express-C, la forma más fácil es utilizando la interfase gráfica del Asistente de instalación DB2. Luego de descargar y descomprimir la imagen del DB2 Express-C, el asistente se invoca como sigue:

- Windows: Ejecute el archivo **setup.exe** ubicado en el directorio EXP/image/
- Linux: Ejecute el comando **db2setup** ubicado en el directorio exp/disk1/

Es muy sencillo instalar el DB2 Express-C siguiendo las instrucciones del Asistente de Instalación DB2. En la mayoría de los casos, los valores por omisión son suficientes, por lo tanto basta con aceptar la licencia, dar clic en el botón *Adelante* hasta que el botón "Finalizar" se activa, y luego dar clic en el botón *Finalizar*. Luego de algunos minutos la instalación está terminada y ¡DB2 estará activo y ejecutándose!

La Figura 3.1 muestra el Área de ejecución para la instalación de DB2. Haga clic en *Instalar un Producto* y luego elija *Instalar nuevo* para instalar una nueva copia del DB2 Express-C en su sistema. Si se ha instalado anteriormente el DB2 Express-C o alguna otra edición de DB2 aparecerá además un botón titulado “*Trabajar con existente*”. Con DB2, se puede instalar el producto varias veces, y esas instalaciones pueden ser de diferentes versiones o niveles de publicación.

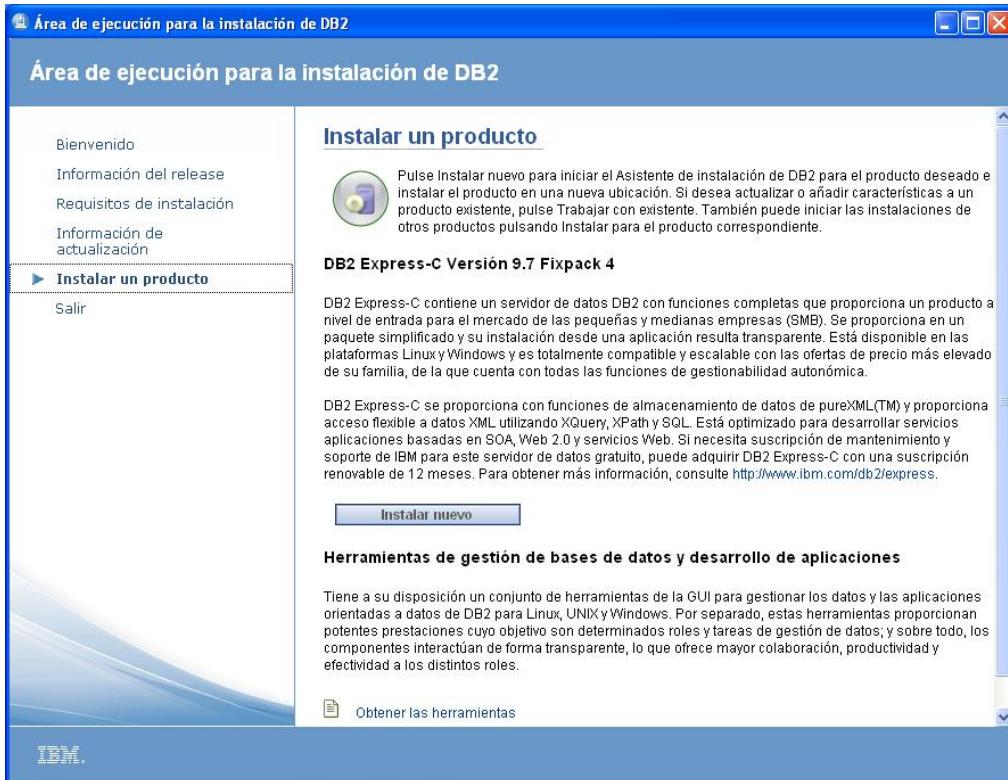


Figura 3.1 – Área de ejecución para la instalación de DB2⁴

⁴ Estos ejemplos en español, se han hecho con la última versión de DB2 Express-C disponible al momento de traducir el libro (V9.4), a diferencia de la versión en inglés del libro (N. del T.)

Después de aceptar la licencia, basta en la mayoría de los casos, elegir la instalación “*Típica*” (opción por omisión) tal como se ve en la *Figura 3.2*. Si quisiera incluir el componente “DB2 Text Search”, en ese caso elija “*Personalizada*”.

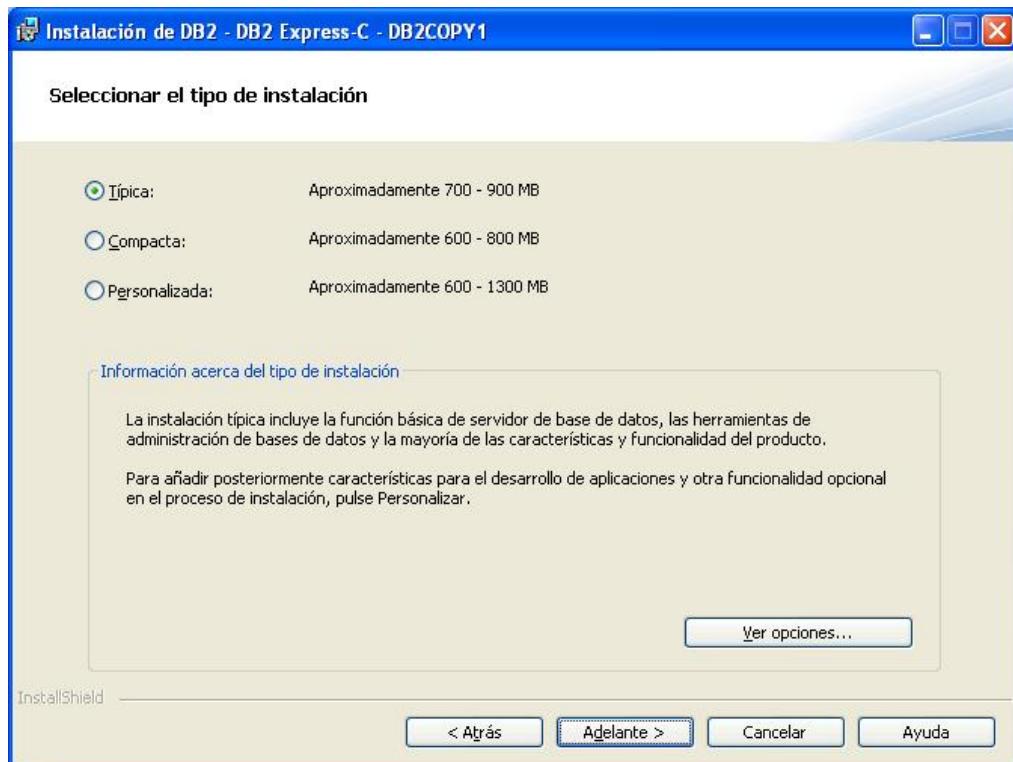


Figura 3.2 – Tipos de instalación

En el siguiente paso, mostrado en la *Figura 3.3*, se puede elegir si se desea instalar el producto, crear un archivo de respuestas o ambas cosas. Los archivos de respuesta se tratan en detalle en la sección 3.5, *Instalación Silenciosa*. La opción por omisión es “*Instalar DB2 Express-C en este sistema y guardar mis valores en un archivo de respuestas*”.

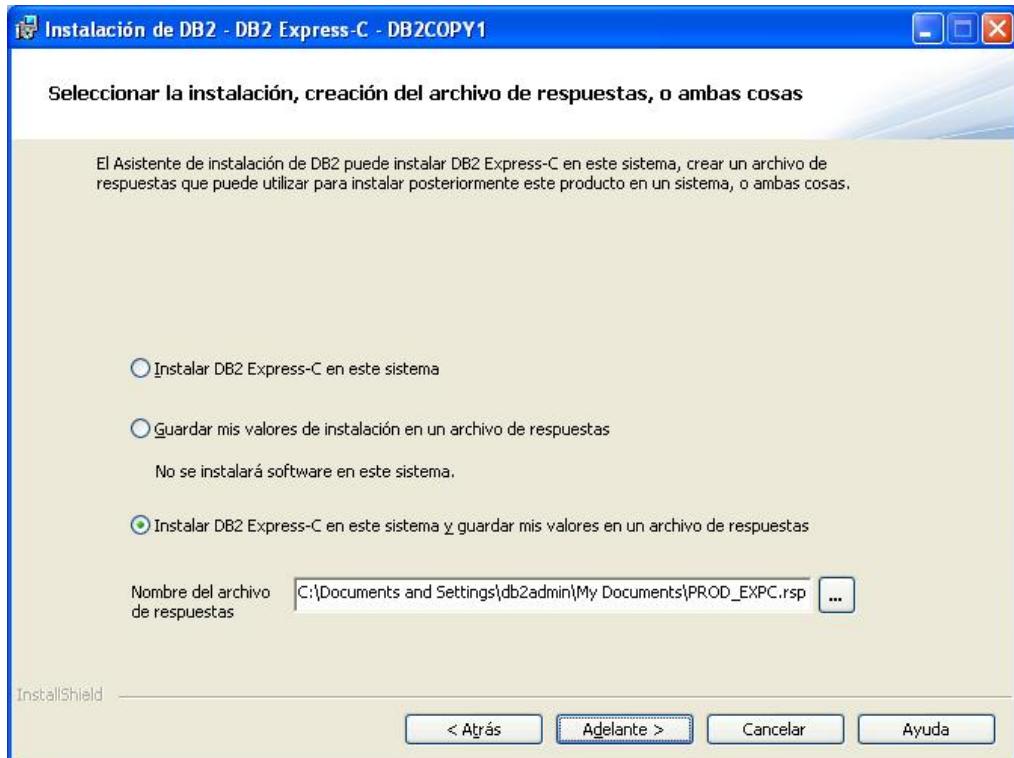


Figura 3.3 – Seleccionar la instalación

52 Conociendo al DB2 Express-C

En las siguientes pantallas, elija los valores por omisión, hasta que llegue al panel de la *Figura 3.4*, donde deberá ingresar el nombre de usuario que será utilizado para preparar y ejecutar la instancia, además de otros servicios.

Si se usa un nombre de usuario existente, éste debe ser miembro del grupo de Administradores Locales de Windows.

Si el nombre de usuario no existe en el sistema, será creado como Administrador Local. El campo de dominio puede dejarse en blanco si el usuario no pertenece a un dominio.

El nombre de usuario por omisión en Windows es **db2admin**. En el caso de Linux, se asigna por omisión **db2inst1**.

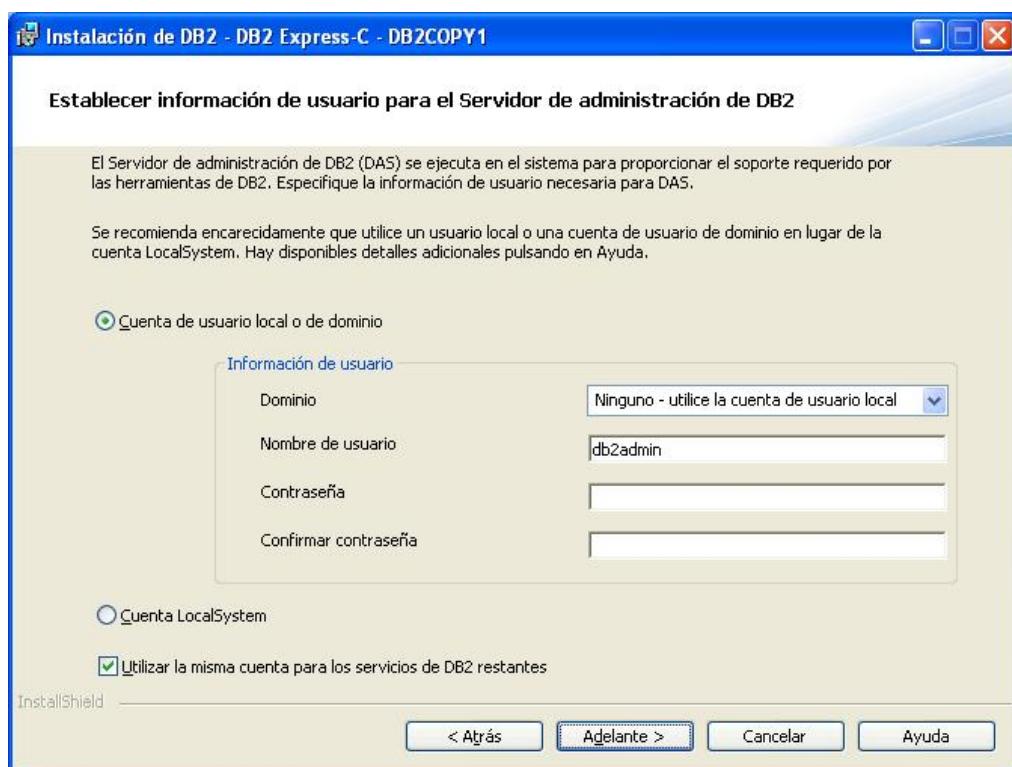


Figura 3.4 – Establecer información de usuario para el Servidor de administración de DB2

Finalmente, como se ve en la *Figura 3.5*, el asistente de instalación despliega un resumen de lo que será instalado y los valores de configuración que se han ingresado en los pasos anteriores. Al dar clic en el botón de “Finalizar”, comienza la instalación propiamente dicha y los archivos de programa son copiados al sistema.

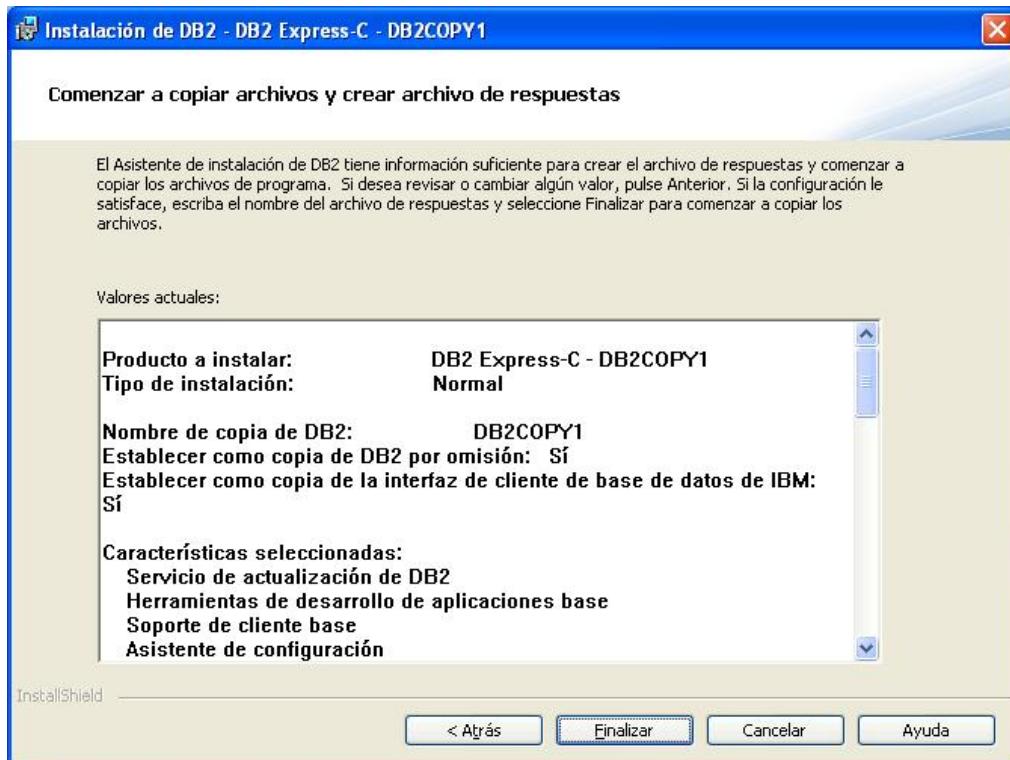


Figura 3.5 – Resumen de lo que será instalado

Al completarse la instalación, aparece una ventana similar a la que se ve en la *Figura 3.6*, informándole del resultado del asistente de instalación, como así también cualquier otro paso adicional que se requiera para completar la instalación.

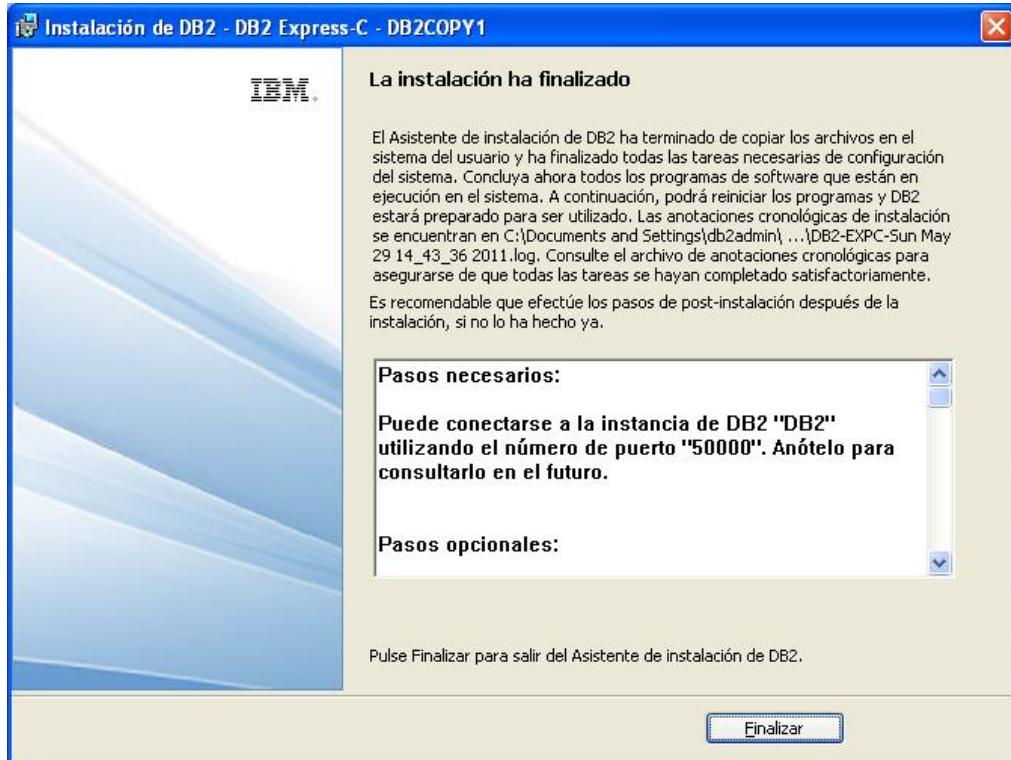


Figura 3.6 – La instalación ha finalizado

Una vez que se da clic en “Finalizar” en la ventana de resultado de la instalación (*Figura 3.6*), se ejecuta automáticamente la aplicación *DB2 Primeros Pasos*, como se ve en la *Figura 3.7*.

Esta pequeña aplicación le presenta varias opciones diferentes para permitirle comenzar a utilizar DB2, tal como crear una base de datos de prueba (llamada **SAMPLE**) o crear su propia base de datos. No es necesario que use estas opciones de *Primeros Pasos* en este momento si no lo desea, simplemente cierre la ventana. Podrá invocar la aplicación más tarde.

Para ejecutar *DB2 Primeros Pasos* en Windows, elija *Inicio -> Programas -> IBM DB2 -> DB2COPY1 (Valor por omisión) -> Herramientas de configuración -> Primeros Pasos* o bien ejecute el comando **db2fs** desde una ventana de mandatos.

En Linux, ejecute el comando **db2fs** desde una ventana de terminal.

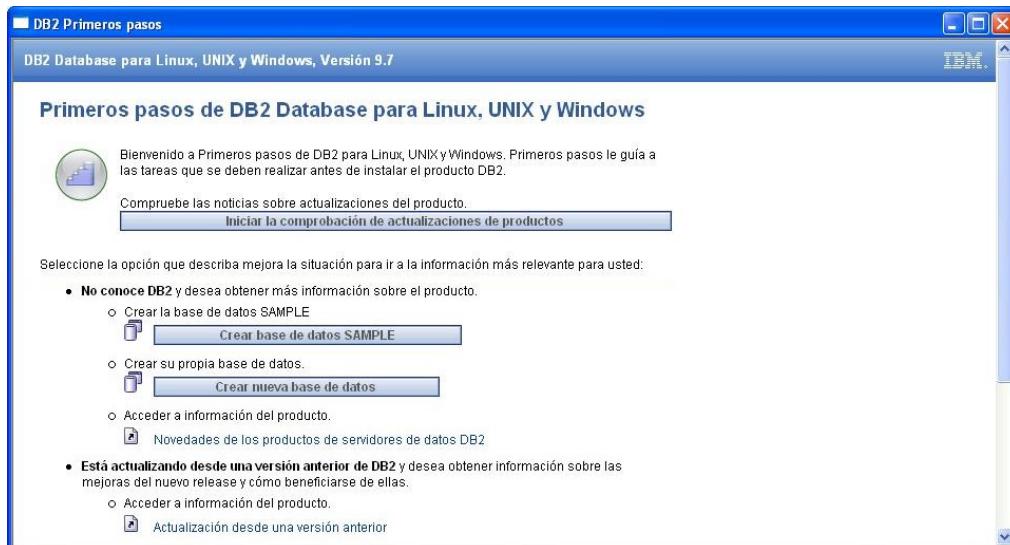


Figura 3.7 – Primeros Pasos

3.4 Validación de la instalación

Después de instalar DB2, se pueden ejecutar tres comandos desde la ventana de mandatos (en Windows) o desde la terminal (en Linux) para verificar que la instalación está correcta:

- **db2level**: Este comando despliega información acerca del producto DB2 instalado, nivel de paquete de correcciones y otros detalles.
- **db2licm -l**: Este comando lista toda la información de licencias relacionadas con los productos DB2 instalados.

nuevo en
V9.7

- **db2val:** Este es un comando nuevo disponible en DB2 9.7 que permite validar la instalación verificando la funcionalidad central de su copia de DB2. Le da la seguridad que sus instancias son consistentes y que tanto la creación de la base de datos y sus conexiones trabajan bien.

La Figura 3.8, más abajo, da un ejemplo del resultado de ejecutar estos tres comandos.

```

C:\Program Files\IBM\SQLLIB\BIN>db2level
DB21085I La instancia "DB2" utiliza "32" bits y el release de código de DB2
"SQL09074" con el identificador de nivel "08050107".
Las señales informativas son "DB2 v9.7.400.501", "s110330", "IP23237" y FixPack
"4".
El producto está instalado en "C:\PROGRA~1\IBM\SQLLIB" con nombre de copia de
DB2 "DB2COPY1".

C:\Program Files\IBM\SQLLIB\BIN>db2licm -l
Nombre del producto: "DB2 Express-C"
Tipo de licencia: "Injustificado"
Fecha de caducidad: "Permanente"
Identificador del producto: "db2expc"
Información de versión: "9.7"
Número máximo de CPU: "2"
Cantidad máxima de memoria (GB): "2"

C:\Program Files\IBM\SQLLIB\BIN>db2val
DBI1379I Se está ejecutando el mandato db2val. Esto puede tardar varios
minutos.
DBI1333I La validación de los archivos de instalación para la copia de
DB2 DB2COPY1 se ha realizado correctamente.
DBI1339I La validación de la instancia DB2 se ha
realizado correctamente.
DBI1343I El mandato db2val se ha completado correctamente. Para obtener
más información, consulte el archivo de anotaciones
C:\DOCUME~1\db2admin\MYDOCU~1\DB2LOG\db2val-Sun May 29 17_29_28 2011.log.
C:\Program Files\IBM\SQLLIB\BIN>

```

Figura 3.8 – Los comandos db2level, db2licm, y db2val para validar la instalación

En la figura, el comando **db2level** nos indica que estamos ejecutando DB2 9.7 (DB2 v9.7.400.501) al nivel de paquete de corrección de errores ("Fix Pack") 4⁵, lo que significa que el código DB2 está a nivel base (GA) más algunas correcciones. El comando **db2licm -l** nos indica que se ha instalado la edición DB2 Express-C que comprende una licencia permanente y sin garantía (Tipo de licencia "Injustificado"), la cual le permite usar hasta dos núcleos de procesador (CPU) y hasta 2GB de memoria. El resultado del comando **db2val** se explica por sí mismo.

⁵ Recuerde que el nivel de DB2 referenciado en esta traducción está más actualizado que el usado en la versión en inglés (N. del T.).

Note:

Si se desea validar la consistencia de una base de datos en cualquier momento, debe usarse el utilitario INSPECT.

3.5 Instalación silenciosa

Puede haber situaciones en las cuales se requiera instalar un cliente DB2 en varios computadores; o bien se precisa incluir un servidor de datos DB2 como parte de una aplicación, y se deseara instalarlo conjuntamente dentro del proceso de instalación de la aplicación. En estos casos la forma ideal de instalar DB2 es mediante una instalación silenciosa. La forma en que se instala silenciosamente DB2, es mediante el uso de archivos de respuesta, en los que se almacena la información de instalación en forma de simples opciones de texto. En el *Listado 3.1* se aprecia un extracto de un archivo de respuesta de muestra.

```
PROD=EXPRESS_C
LIC AGREEMENT=ACCEPT
FILE=C:\Program Files\IBM\SQLLIB\
INSTALL_TYPE=TYPICAL
LANG=EN
LANG=ES
INSTANCE=DB2
DB2 .NAME=DB2
DEFAULT_INSTANCE=DB2
DB2 .SVCENAME=db2c_DB2
DB2.DB2COMM=TCPIP...
```

Listado 3.1 – Ejemplo de archivo de respuesta

Hay diferentes formas de generar un archivo de respuestas:

- Instalar el DB2 Express-C una vez en un computador usando el Asistente de instalación DB2. Una de las primeras opciones del asistente (tal como se ve en la *Figura 3.3*) permite seleccionar que las respuestas a las opciones de instalación, se guarden en un archivo de respuestas. Al finalizar la instalación, el asistente genera dicho archivo de respuestas en el directorio y con el nombre especificados. Este archivo es un archivo de texto, por lo tanto puede ser editado manualmente después.
- Editar el archivo de respuestas ejemplo que viene con la distribución del DB2 Express-C. Puede encontrar este archivo ejemplo (con la extensión .rsp) en el directorio db2/platform/samples/
- En Windows, también puede usarse el comando generador del archivo de respuesta:

```
db2rspgn -d <directorio destino>
```

Finalmente, para instalar silenciosamente el DB2 usando un archivo de respuestas, se usa el siguiente comando en Windows:

```
setup -u <archivo de respuestas>
```

En Linux, se usa este comando:

```
db2setup -r <archivo de respuestas>
```

3.6 Resumen

En este capítulo se han visto los detalles de la instalación del DB2 Express-C. Esta edición de DB2 está disponible en Linux, Solaris y en casi todas las variedades de Windows, y puede correr en arquitecturas de 32-bit, 64-bit y en Power PC. Despues de haber tratado sobre las autorizaciones necesarias para instalar DB2 en un sistema, seguimos paso a paso una instalación sencilla utilizando el Asistente de instalación gráfico de DB2. Continuamos con una discusión de las actividades a realizarse luego de terminada la instalación, incluyendo la ejecución de los Primeros pasos de DB2 y la validación de la instalación. Por ultimo, dimos un vistazo a como generar y utilizar archivos de respuesta para una instalación silenciosa de DB2.

3.7 Ejercicios

En este ejercicio, usted instalará el DB2 Express-C y creará la base de datos SAMPLE

Objetivo

Antes de comenzar a explorar todas las características y herramientas que vienen con el DB2 Express-C, es necesario, en primer lugar, instalarlo en el sistema. En este ejercicio, usted realizará una instalación básica del DB2 Express-C en Windows. En Linux está disponible el mismo asistente de instalación; por lo tanto los pasos a seguir son muy similares en esa plataforma.

Procedimiento

1. Obtenga las imágenes del DB2 Express-C: Descargue la imagen apropiada del DB2 Express-C desde el sitio Web del DB2 Express-C (www.ibm.com/db2/express). Desempaque los archivos en el directorio que usted quiera.
2. Ubique los archivos: Navegue hasta el directorio (o disco) donde ha desempaquetado los archivos de instalación del producto DB2.
3. Active el Área de ejecución para la instalación de DB2: De doble-clic en el archivo `setup.exe` para activar el Área de ejecución para la instalación de DB2. En Linux, ejecute el comando `db2setup` con autoridad de “root”. Luego, en el Área de ejecución para la instalación de DB2, de clic en la opción *Instalar un producto* en el panel izquierdo de la ventana.

4. Ejecute el Asistente de instalación de DB2: El Asistente de instalación de DB2 verifica que se cumplan todos los requerimientos del sistema y ve si hay alguna instalación DB2 previa. Ejecute el Asistente de instalación dando clic en *Instalar nuevo*, luego dé clic en *Avanzar*.
5. Revise el convenio de licencia: Lea y acepte el convenio de licencia (marcando la opción “*Acepto...*”) y dé clic en el botón *Avanzar* para continuar.
6. Seleccione el tipo de instalación: Para este ejercicio, seleccione la opción *Típica* (ya está marcada por omisión). En los otros casos, la opción *Compacta* realiza una instalación básica, mientras que la opción *Personalizada* permite seleccionar componentes específicos que se deseen instalar. Dé clic en el botón *Avanzar* para continuar.
7. Seleccione instalación, creación de archivo de respuesta o ambos: Deje la opción por omisión seleccionada, de modo que se instale DB2 y además se cree un archivo de respuestas. Dé clic en el botón *Avanzar* para continuar.
8. Seleccione la carpeta de instalación: En esta pantalla se elige el disco y directorio dónde el código DB2 será instalado en el sistema. Asegúrese que haya espacio suficiente para la instalación. Para este ejemplo, use el disco y directorio indicados por omisión (como se muestra):

Disco: C:

Directorio: C:\Archivos de Programa\IBM\SQLLIB

Dé clic en el botón *Avanzar* para continuar.

9. Ingrese la información de usuario: Una vez que el DB2 Express-C está instalado, algunos de sus procesos ejecutan como servicios del sistema. Estos servicios requieren de una cuenta en el sistema operativo para poder ejecutar. En el ambiente Windows, se recomienda usar la cuenta de usuario por omisión **db2admin**. Si ésta cuenta no existe previamente, DB2 la creará en el sistema operativo. También puede especificarse una cuenta existente, pero en ese caso dicha cuenta deberá tener autoridad de administrador local. Recomendamos usar los valores por omisión sugeridos. Asegúrese de especificar una clave para la cuenta. En Linux use los valores por omisión: **db2inst1** para el propietario de la instancia, **db2fenc1** para el usuario acorralado (“fenced”) y **dasusr1** para el usuario del Servidor de administración de DB2. Dé clic en el botón *Avanzar* para continuar.
10. Configure la instancia DB2: Una instancia DB2 puede pensarse como un contenedor de bases de datos. Debe existir una instancia antes de poder crear una base de datos en su interior. Durante la instalación bajo Windows, se crea la instancia de nombre **DB2** en forma automática. En el ambiente Linux, la instancia por omisión es **db2inst1**. Hablaremos más acerca de las instancias posteriormente en el libro.

Por omisión, la instancia DB2 se configura para escuchar por conexiones TCP/IP en el puerto 50000. Se pueden cambiar tanto el protocolo por omisión como el puerto, dando clic al botón de *Configurar*. Recomendamos usar los valores por omisión para este ejercicio. Dé clic en el botón *Avanzar* para continuar.

11. Comience la instalación: Revise el resumen de las opciones de instalación que se han seleccionado anteriormente. De clic en el botón *Finalizar* para comenzar a copiar los archivos al disco o directorio de instalación. El DB2 también realizará algunos procesos iniciales de configuración.
12. Primeros Pasos: Una vez finalizada la instalación, se despliega otro utilitario llamado Primeros Pasos. Este utilitario puede también ser ejecutado más tarde con el comando `db2fs`.
13. La base de datos **SAMPLE** es una base de datos que puede ser usada para pruebas. Puede ser creada usando Primeros Pasos, dando clic en el botón *Crear base de datos SAMPLE*. Tras dar clic en este botón, aparece la ventana que se muestra más abajo. Elija la segunda opción (*Objetos y datos XML y SQL*). La base de datos SAMPLE puede ser también creada usando el comando `db2sampl -xml -sql`.



-
14. Luego de algunos minutos, puede verificarse que la base de datos ha sido creada. Para ello abra la herramienta de DB2 Centro de control seleccionando: *Inicio -> Programas -> IBM DB2 -> DB2COPY1 (Valor por omisión) -> Herramientas de administración general -> Centro de control.* El Centro de control también puede invocarse mediante el comando `db2cc`. La primera vez que el Centro de control se ejecuta, una ventana emergente le pedirá que elija cual vista del Centro de control desea. Deje el valor por omisión (Avanzada), y dé clic en *Bien*. En el panel de la izquierda, expanda la carpeta *Todas las bases de datos*. Si no ve listada la base de datos SAMPLE en esa carpeta, renueve la vista eligiendo *Ver -> Renovar*
 15. Reinicie el computador. Este paso es optativo. A pesar que no se lo menciona en la documentación oficial de instalación de DB2, le recomendamos reiniciar el sistema (si es posible, al menos en Windows) para asegurarse que todos los procesos parten satisfactoriamente y para liberar cualquier recurso de memoria que no se haya despejado correctamente.
 16. Valide la instalación de DB2 ejecutando los comandos: `db2level`, `db2licm`, y `db2val`. Desde el menú de Inicio de Windows, abra una Ventana de mandatos DB2 de esta forma: *Inicio -> Programas -> IBM DB2 -> DB2COPY1 (Valor por omisión) -> Herramientas de línea de mandatos -> Ventana de mandatos*. Dentro de la Ventana de mandatos (O del “shell” en Linux) escriba `db2level` y examine el resultado. Haga lo mismo con el comando `db2licm -l`, y por último ingrese el comando `db2val`. Si termina satisfactoriamente, ¡la instalación está en buena forma! Si hay errores, revise el archivo de anotaciones especificado en el mensaje para mayores detalles. El resultado de los tres comandos debería ser similar a lo ya visto en la *Figura 3.8* más arriba.

4

Capítulo 4 – El ambiente DB2

El ambiente DB2 comprende diferentes objetos de bases de datos y archivos de configuración. La *Figura 4.1* nos muestra un resumen de los diferentes comandos y herramientas disponibles para trabajar con DB2, como así también, a la derecha, destaca el ambiente DB2. Este capítulo se enfoca en estos temas. A la izquierda de la figura pueden verse los distintos comandos DB2 y las sentencias SQL, SQL/XML, y XQuery que pueden ser usadas para interactuar con un servidor de datos DB2. La lista de las diferentes herramientas usadas para trabajar con el servidor de datos DB2 aparece en el medio de la figura.

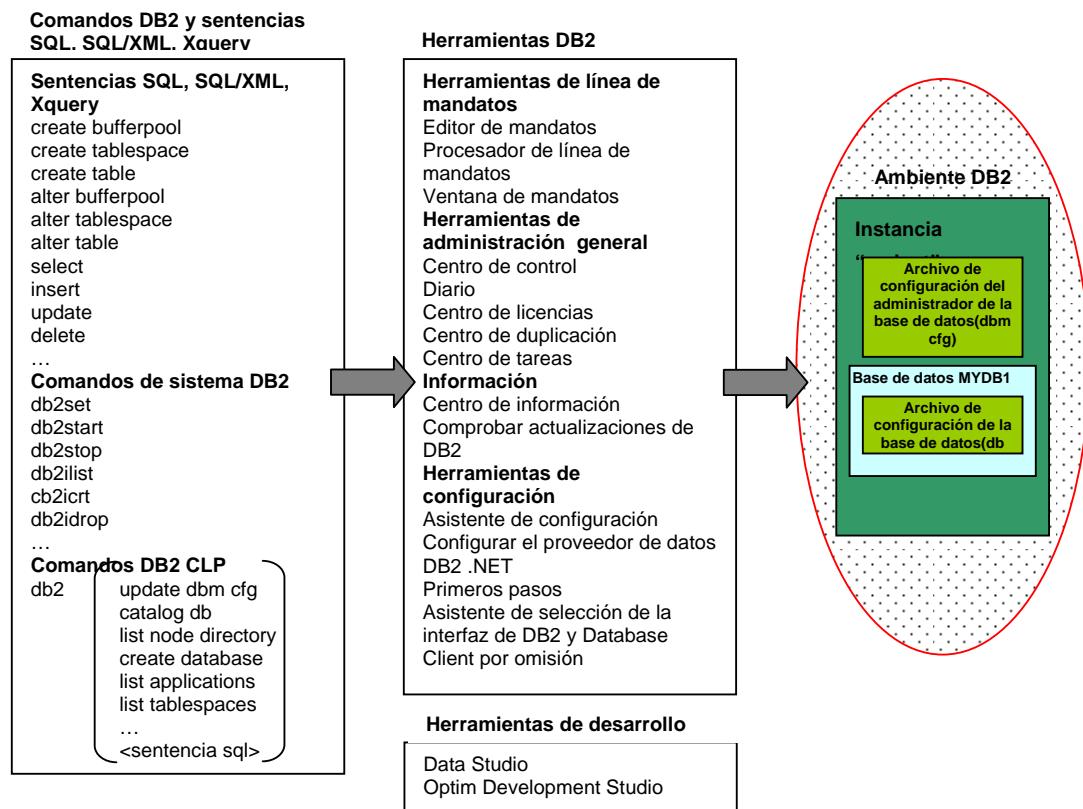


Figura 4.1 – Visión general de DB2: El ambiente DB2

Nota:

En estos enlaces podrá ver unas presentaciones acerca del Ambiente DB2 (en inglés):

<http://www.channeldb2.com/video/video/show?id=807741:Video:4029>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4042>

Para mejor entender el ambiente DB2, veamos cada uno de los elementos que lo componen paso a paso. En la *Figura 4.2* se ve una representación de un servidor de datos DB2 después de haber instalado el DB2 Express-C 9.7.

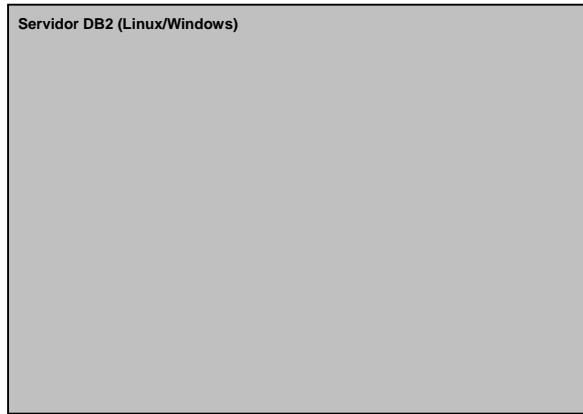


Figura 4.2 – Representación de un servidor DB2 después de haber instalado el DB2 Express-C 9.7

Dentro de la instalación en Windows, se crea una instancia por omisión bajo el nombre **DB2** (**db2inst1** en Linux). En la *Figura 4.3* esta instancia está representada por el rectángulo verde que se ve a la izquierda. Diremos que una instancia es simplemente un ambiente independiente donde pueden ejecutarse aplicaciones y pueden crearse bases de datos. Se pueden crear varias instancias en un servidor de datos y dar a cada una un uso diferente. Por ejemplo una instancia puede contener las bases de datos usadas en production, otra para las bases de datos del ambiente de pruebas y otra más para el ambiente de desarrollo. Todas estas instancias son independientes, o sea que cualquier operación que se realice en una de ellas, no afectará a las otras.

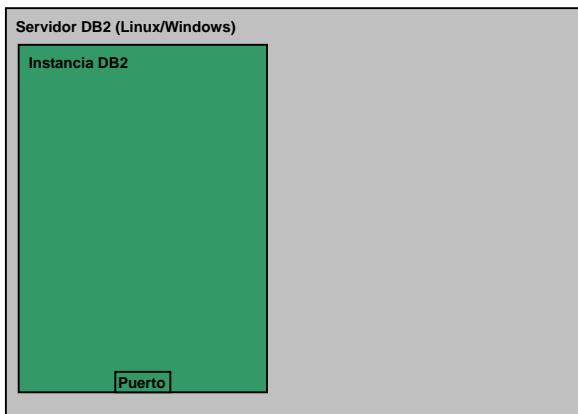


Figura 4.3 – Se ha creado la instancia DB2 por omisión

Para crear una nueva instancia DB2, se usa el comando **db2icrt <nombre de la instancia>**, donde **<nombre de la instancia>** se reemplaza por cualquier nombre de hasta 8 caracteres. Por ejemplo, para crear la instancia **myinst**, usamos este comando: **db2icrt myinst**.

La figura 4.4 muestra una instancia nueva de nombre **myinst** como un rectángulo separado a la derecha.

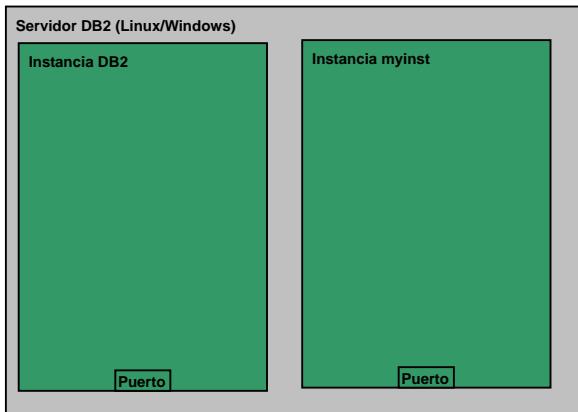


Figura 4.4 – Servidor DB2 con dos instancias

Es importante hacer notar que cada instancia tiene un número de puerto único. Esto es lo que permite distinguir entre distintas instancias al momento de conectarse a una base de datos en una determinada instancia usando TCP/IP desde un cliente remoto. Para hacer que una de las instancias de un servidor DB2 sea la instancia “activa”, desde una Ventana de mandatos DB2 se usa el siguiente comando del sistema operativo Windows:

```
set db2instance=myinst
```

Tenga en cuenta que no debe haber ningún espacio en blanco delante o detrás del signo igual (=). Siguiendo el ejemplo, si ahora usted creara una base de datos desde la Ventana de Mandatos, ésta sería creada en la instancia `myinst`.

Mediante el siguiente comando pueden listarse las instancias definidas en el sistema:

`db2ilist`

En el caso de Linux, cada instancia se corresponde con un usuario del sistema operativo; por lo tanto para cambiarse de una instancia a otra basta con cambiar de un usuario a otro. Estos usuarios son los propietarios de las instancias, y para cambiar de uno a otro, se puede desconectar de un usuario ("logoff") y conectarse con otro ("logon"), o bien usar el comando `su`.

La *Tabla 4.1* muestra algunos comandos útiles a nivel de instancia.

Comando	Descripción
<code>db2start</code>	Hace partir la instancia actual
<code>db2stop</code>	Detiene la instancia actual
<code>db2icrt</code>	Crea una nueva instancia
<code>db2idrop</code>	Da de baja a una instancia
<code>db2ilist</code>	Lista las instancias presentes en el sistema
<code>db2 get instance</code>	Lista la instancia actual activa

Tabla 4.1 – Comandos DB2 útiles a nivel de instancia

Algunos de estos comandos pueden ser también ejecutados desde el Centro de Control. Por ejemplo, si se expande la carpeta *Instancias* en el Centro de Control y se le da clic con el botón de la derecha del ratón sobre la instancia con la cual se desea trabajar, en el menú que se despliega se puede seleccionar *Start*, lo cual es lo mismo que ejecutar el comando `db2start` desde la Ventana de Mandatos DB2 Command Window, o bien seleccionar *Stop*, que es equivalente a dar el comando `db2stop` tal como se ve en la *Figura 4.5*.

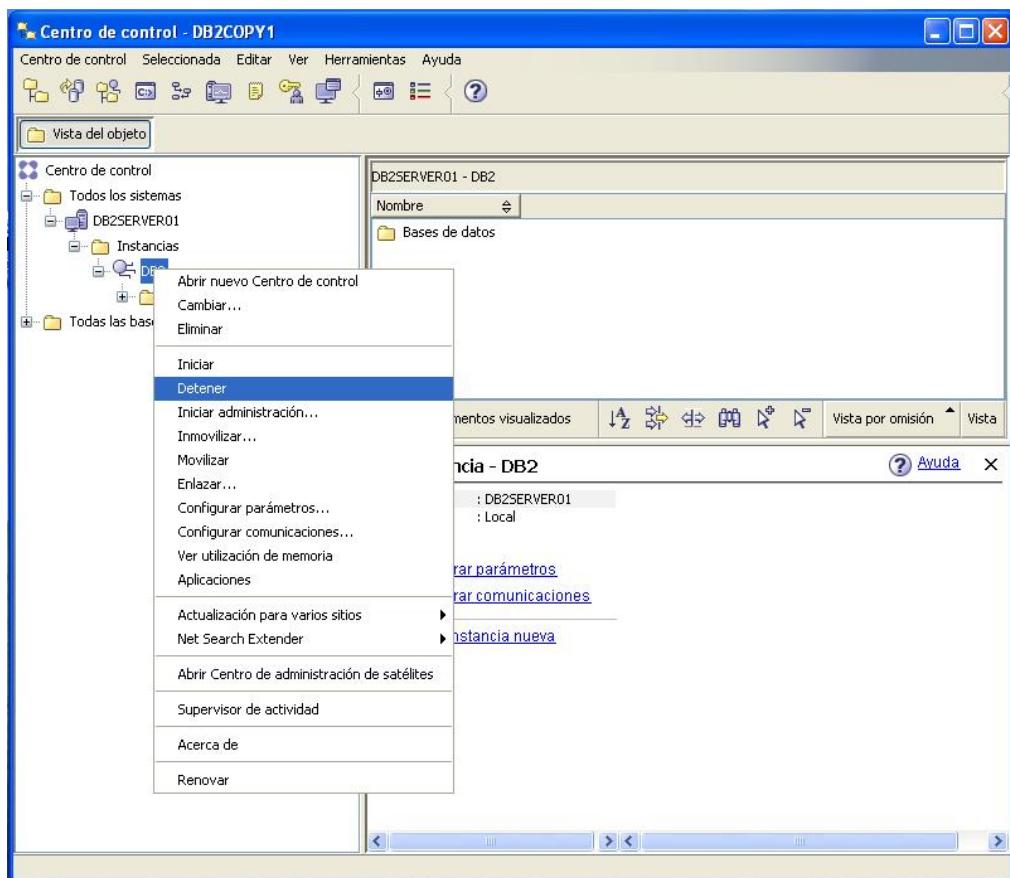


Figura 4.5 – Comandos de Instancia desde el Centro de Control

Para crear una base de datos en la instancia active, ejecute este comando desde la Ventana de Mandatos DB2:

```
db2 create database mydb1
```

Este comando lista todas las bases de datos creadas:

```
db2 list db directory
```

Dentro de cualquier instancia, se pueden crear muchas bases de datos. Una base de datos es un conjunto de objetos tales como tablas, vistas, índices, etc. Las bases de datos son unidades independientes, por lo tanto no comparten objetos con otras bases de datos. En la *Figura 4.6* se ve una representación de la base de datos **MYDB1** creada dentro de la instancia **DB2**.

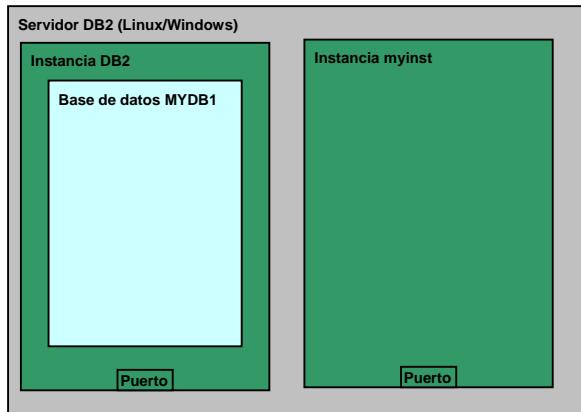


Figure 4.6 – Database MYDB1 created in instance DB2

La *Tabla 4.2* muestra algunos comandos que pueden ser usados a nivel de la base de datos.

Comando/sentencia SQL	Descripción
db2 create database	Crea una base de datos nueva
db2 drop database	Da de baja una base de datos
db2 connect to <nombre_de_la_base_de_datos>	Hace conexión con una base de datos
db2 create table/create view/create index	Sentencias SQL para crear tablas, vistas e índices respectivamente

Tabla 4.2 – Comandos y sentencias SQL a nivel de base de datos

Si deseáramos crear otra base de datos con el mismo nombre (*MYDB1*) pero en la instancia *myinst*, podríamos utilizar la siguiente serie de comandos desde la Ventana de Mandatos DB2:

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

La Figura 4.7 muestra la nueva base de datos **MYDB1** creada en la instancia **myinst**.

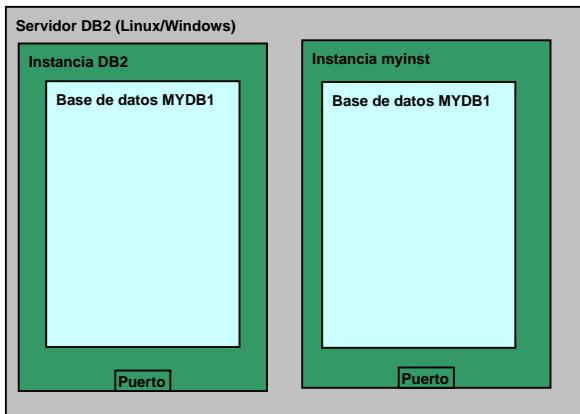


Figura 4.7 – Base de datos MYDB1 creada en la instancia myInst

Cuando se crea una base de datos, también se crean varios otros objetos por omisión: espacios de tablas (*table spaces*), tablas, una agrupación de almacenamientos intermedios (*buffer pool*) y archivos de anotación cronológica (*log files*). La creación de estos objetos toma algo de tiempo, es por ello que el comando **create database** requiere de algunos minutos para procesarse. La Figura 4.8 nos muestra tres espacios de tablas que han sido creados por omisión en el lado izquierdo. Trataremos con más detalle a los espacios de tablas en el Capítulo 6, Arquitectura DB2; por ahora considere a los espacios de tablas como un nivel lógico intermedio entre las tablas lógicas y los recursos físicos, como ser discos y memoria.

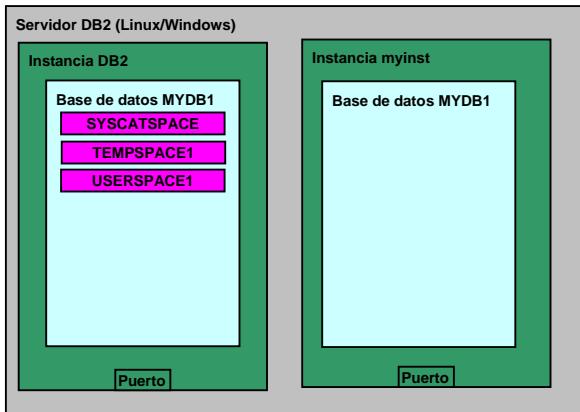


Figura 4.8 –Espacios de tablas que se crean por omisión al momento de crear una base de datos

El espacio de tablas **SYSCATSPACE** contiene las tablas del Catálogo del Sistema. En otros sistemas de bases de datos relacionales, este catálogo se conoce como *diccionario de datos*, y básicamente contiene información que no debe ser modificada o eliminada, pues

si no, la base de datos no funcionaría correctamente. El segundo espacio de tablas, **TEMPSPACE1**, es usado por DB2 cuando necesita espacio adicional para realizar operaciones tales como clasificación de datos (*sorts*). Por ultimo, el espacio de tablas **USERSPACE1** es utilizado normalmente para almacenar aquellas tablas a las cuales no se les ha asignado un espacio de tablas específico en el momento de la creación.

Uno puede crear sus propios espacios de tablas usando la sentencia **CREATE TABLESPACE**. En la *Figura 4.9* se ve al espacio de tablas **MYTBL1** que ha sido creado dentro de la base de datos **MYDB1** en la instancia **DB2**. En el momento de crear un espacio de tablas, se especifica que discos utilizar y cual agrupación de almacenamientos intermedios utilizar. Por lo tanto, en el caso de una tabla que se utiliza frecuentemente (también conocida como “*hot table*”); normalmente se la define dentro de un espacio de tablas el cual se ha definido a su vez con los discos más rápidos y asociado a una agrupación de almacenamientos intermedios de mayor tamaño.

En la *Figura 4.9*, podemos ver otros dos objetos que también fueron creados por omisión: Una agrupación de almacenamientos intermedios llamada **IBMDEFAULTBP**, y los archivos de anotación cronológica.

Una agrupación de almacenamientos intermedios es básicamente un área de memoria intermedia usada por la base de datos. Pueden crearse uno o más agrupaciones de almacenamientos intermedios, pero siempre debe existir una agrupación de almacenamientos intermedios con un tamaño de página que se corresponda con el tamaño de página de los espacios de tablas existentes. Hablaremos en mayor detalle acerca de las páginas y su tamaño en el *Capítulo 6, Arquitectura del DB2*.

Los archivos de anotación cronológica se utilizan para recuperación. A medida que se trabaja con una base de datos, la información no solo se almacena en los discos de la base de datos, sino que a medida que se va trabajando, en los archivos de anotación cronológica se guardan todos los cambios que se han ejecutado sobre los datos. Puede pensarse sobre estos archivos de anotación cronológica, como archivos temporales en los cuales se almacenan automáticamente los cambios (*autosave*). Veremos a los archivos de anotación cronológica con mayor detalle en el *Capítulo 11, Respaldo y Recuperación*.

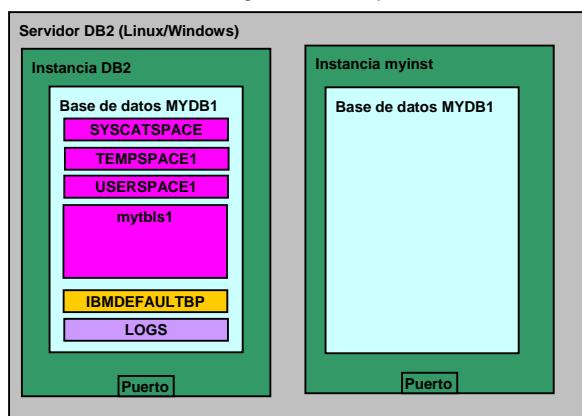


Figura 4.9 – Agrupación de almacenamientos intermedios y archivos de anotación cronológica creados por omisión

Anteriormente dijimos que las instancias eran ambientes independientes, y por lo tanto, una base de datos con el mismo nombre podía ser creada en varias instancias diferentes. Tal como las instancias, las bases de datos son unidades independientes; por ello, los objetos de una base de datos no tienen relación alguna con los objetos de otra base de datos. En la *Figura 4.10* se ve un espacio de tablas llamado *mytbls1* tanto dentro de la base de datos **MYDB1** como en la base de datos **SAMPLE**, ambas dentro de la instancia **DB2**. Esto es válido debido a que las bases de datos son unidades independientes. En la *Figura 4.10* no se han mostrado otros objetos creados por omisión dentro de la base de datos **SAMPLE** para no agrandar innecesariamente el tamaño del diagrama.

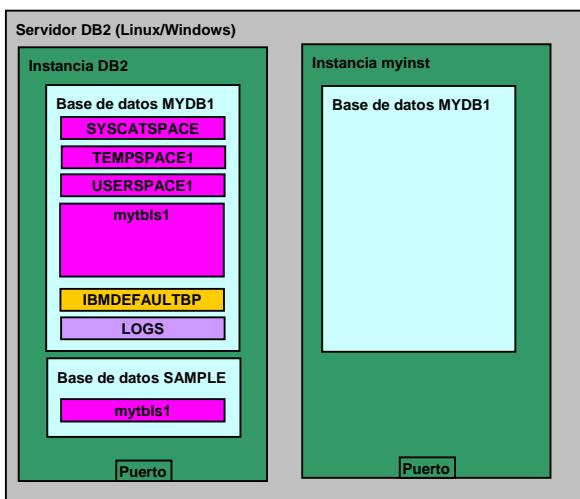


Figura 4.10 – Espacios de tablas con el mismo nombre en bases de datos distintas.

Una vez que se ha creado un espacio de tablas, otros objetos tales como tablas, vistas e índices pueden crearse dentro de él, tal como se ve en la *Figura 4.11*.

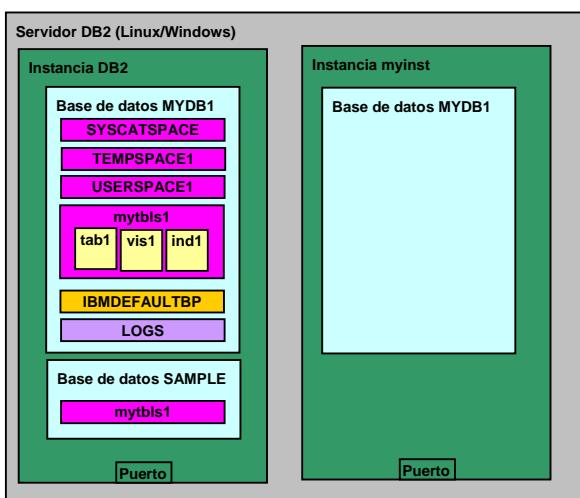


Figura 4.11 – Tablas, vistas e índices creados dentro del espacio de tablas

4.1 Configuración de DB2

Los parámetros de DB2 pueden ser configurados utilizando la Herramienta Asesor de Configuración. Para invocarla desde el Centro de Control, se da clic con el botón derecho del ratón sobre la base de datos que se desea configurar y se selecciona *Asesor de Configuración*. El asesor de configuración hará algunas preguntas acerca de los recursos del sistema y de la carga de trabajo proyectada y basándose en las respuestas que se le den, proveerá una lista de parámetros DB2 y valores sugeridos para los mismos. Continúe leyendo si es que está interesado en conocer más detalles acerca de la configuración de DB2, pero si no es así, ¡utilice el Asesor de Configuración e igualmente estará haciendo un buen trabajo en DB2!

Un servidor DB2 puede ser configurado en cuatro niveles diferentes:

- Variables de ambiente
- Archivo de configuración del manejador de la bases de datos (dbm cfg)
- Archivo de configuración de la base de datos (db cfg)
- Registro de perfil DB2 (*profile registry*)

Esto es lo que se ve en la *Figura 4.12*. Es importante notar allí donde se ubica cada rectángulo. Por ejemplo, las variables de ambiente son definidas al nivel del sistema operativo del servidor, mientras que los parámetros en el archivo de configuración del manejador de la base de datos se definen a nivel de la instancia. A nivel de la base de datos se definen sus parámetros de configuración y por ultimo los valores del registro de perfil de DB2 se asignan, ya sea a nivel del sistema operativo o bien a nivel de la instancia.

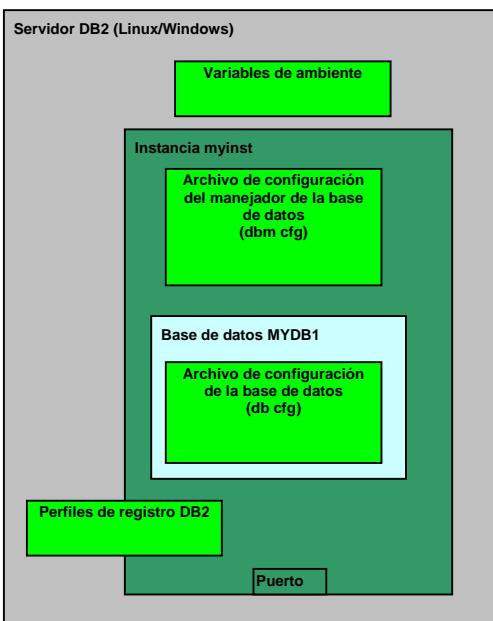


Figura 4.12 – Configuración de DB2

4.1.1 Variables de ambiente

Las *variables de ambiente* son variables que se especifican a nivel del sistema operativo. Una de estas variables que es fundamental es **DB2INSTANCE**. Esta variable indica cual es la instancia active con la que se está trabajando y a la cual los comandos DB2 que se ejecuten irán a afectar. Por ejemplo para hacer que la instancia **myinst** pase a ser la instancia active, en Windows se ejecuta el siguiente comando del sistema operativo:

```
set db2instance=myinst
```

4.1.2 Archivo de configuración del manejador de la base de datos (dbm cfg)

En el archivo de configuración del manejador de la base de datos (dbm cfg), se incluyen aquellos parámetros que afectan a la instancia y a todas las bases de datos contenidas en ella. Este archivo de configuración es consultado o modificado ya sea desde la línea de mandatos o mediante el Centro de Control DB2.

Para trabajar con el dbm cfg desde el Centro de Control, se selecciona la instancia en la carpeta de instancias, de clic con el botón derecho del ratón para desplegar el menú y allí se elige la opción *Configuración de Parámetros*, tal como se ve en la Figura 4.13.

74 Conociendo al DB2 Express-C

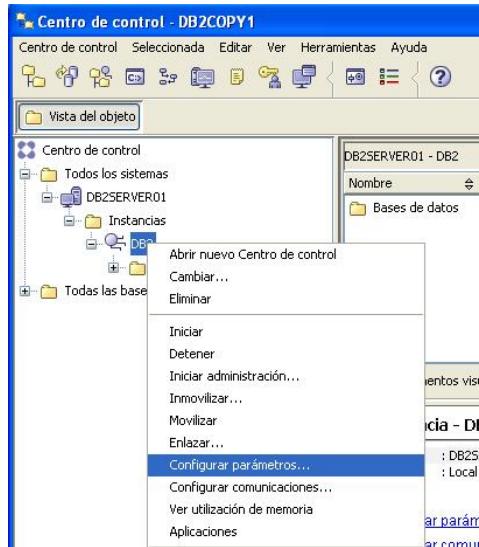


Figura 4.13 – Configuración del dbm cfg desde el Centro de Control.

Tras elegir *Configurar Parámetros*, se abre la ventana de la Figura 4.14 con la lista de los parámetros dbm cfg.

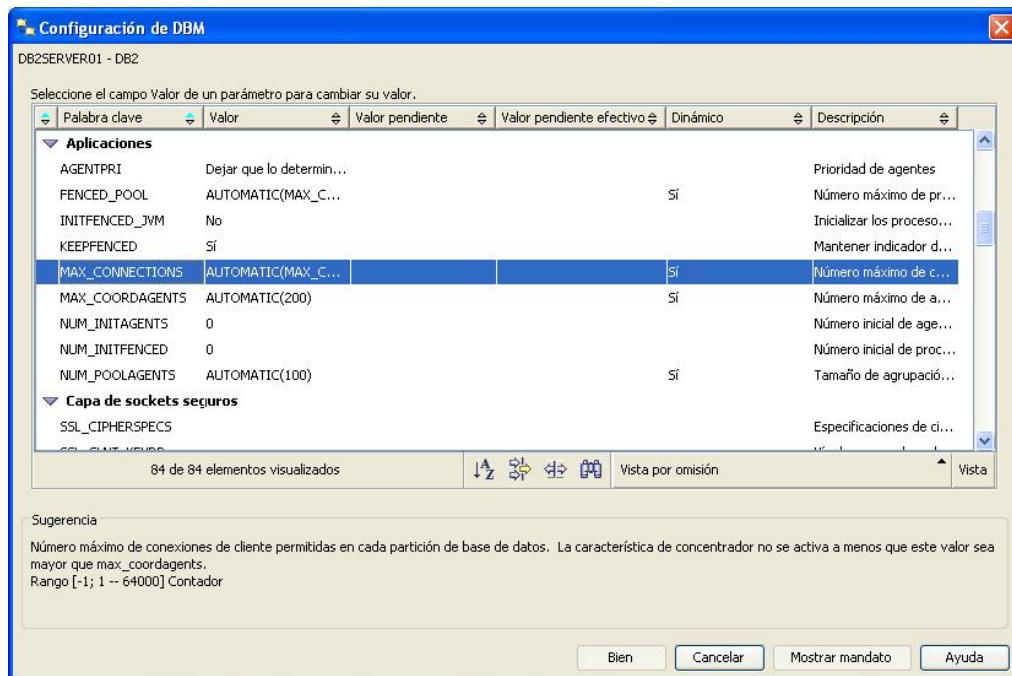


Figura 4.14 – El diálogo dbm cfg

Muchos de los parámetros son dinámicos, esto quiere decir que cuando su valor se altera este cambio produce un efecto inmediato; por otro lado al cambiar otros parámetros, se requiere detener y hacer partir nuevamente la instancia para que el cambio tenga efecto. Para detener la instancia desde la Línea de Mandatos se utiliza el comando **db2stop** y para hacerla partir, el comando **db2start**.

Para que una instancia pueda detenerse, todas las aplicaciones conectadas a ella deben desconectarse previamente. En caso que se desee forzar la detención de una instancia puede utilizarse el comando **db2stop force**.

Del mismo modo, una instancia puede ser detenida (o arrancada) desde el Centro de Control, dando clic sobre la instancia y seleccionando ya sea *Stop* o *Start*.

La *Tabla 4.3* muestra algunos comandos útiles para trabajar con el dbm cfg desde la Línea de Mandatos.

Comando	Descripción
<code>db2 get dbm cfg</code>	Obtiene información desde el dbm cfg
<code>db2 update dbm cfg using <nombre_del_parámetro> <valor></code>	Actualiza el valor de un parámetro en el dbm cfg

Tabla 4.3 - Comandos para manipular el dbm cfg

4.1.3 El archivo de configuración de la base de datos (db cfg)

El archivo de configuración de la base de datos (db cfg) contiene parámetros que afectan a una base de datos determinada. Este archivo puede ser consultado o modificado ya sea desde la línea de mandatos o bien mediante el Centro de Control DB2.

Para trabajar con el db cfg desde el Centro de Control, se selecciona la base de datos dentro de la carpeta correspondiente, se le da clic con el botón derecho del ratón. Luego, en el menú que se despliega se selecciona *Configurar Parámetros*, tal como se ve en la *Figura 4.15*.

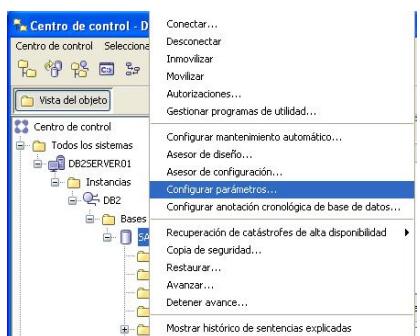


Figura 4.15 – Configuración del db cfg desde el Centro de Control.

76 Conociendo al DB2 Express-C

Tras seleccionar *Configurar Parámetros*, se abre la ventana que se muestra en la Figura 4.16 con la lista de los parámetros db cfg.

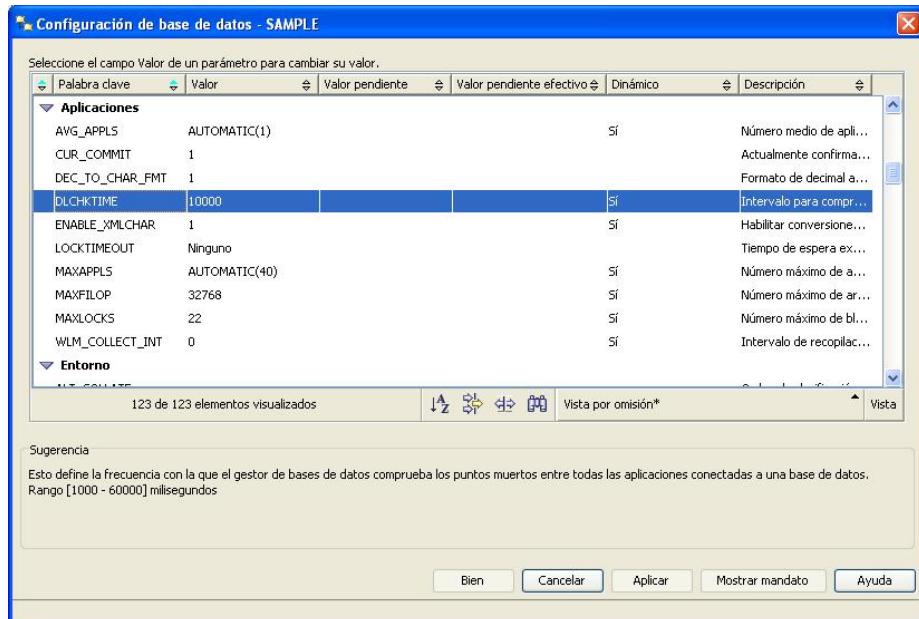


Figura 4.16 – El db cfg

En la Tabla 4.4 se ven algunos comandos útiles para manejar el db cfg desde la Línea de Mandatos.

Comando	Descripción
<code>get db cfg for <nombre_de_la_base_de_datos></code>	Muestra información desde el db cfg para una base de datos dada
<code>update db cfg for <nombre_de_la_base_de_datos> using <nombre_del_parámetro> <valor></code>	Actualiza el valor de un parámetro en el db cfg

Tabla 4.4 - Comandos para manipular el db cfg

4.1.4 Registro de perfiles DB2

El registro de perfiles DB2 contiene parámetros que pueden ser específicos de una plataforma determinada y a los que se les pueden asignar valores que pueden tener efecto global (o sea que afectan a todas las instancias) o bien a nivel de la instancia (afectando a una instancia determinada).

La Tabla 4.5 muestra algunos comandos útiles para manipular el registro de perfiles DB2.

Comando	Descripción
db2set -all	Despliega aquellas variables del registro de perfiles DB2 a los cuales se les ha asignado un valor
db2set -lr	Despliega todas las variables del registro de perfiles DB2
db2set <parámetro>=<valor>	Asigna un valor a un parámetro

Tabla 4.5 – Comandos para manipular el registro de perfiles DB2

La Tabla 4.6 muestra algunos de las variables más utilizadas en el registro DB2

Variable del registro	Descripción
DB2COMM	Especifica los manejadores de comunicaciones que deben activarse cuando el manejador de la base de datos arranca.
DB2_EXTSECURITY	En Windows, evita el acceso no autorizado a DB2 bloqueando los archivos del sistema DB2.
DB2_COPY_NAME	Contiene el nombre de la copia de DB2 que se esta utilizando. Si se desea cambiar a otra copia de DB2 instalada en el mismo equipo, debe usarse el comando <code>installpath\bin\db2envvar.bat</code> . No puede usarse esta variable para hacer el cambio.

Table 4.6 – Commonly used DB2 profile registry variables

Por ejemplo, para permitir establecer comunicación usando TCPIP, se debe asignar el valor TCPIP a la variable DB2COMM de esta forma:

```
db2set db2comm=tcPIP
```

4.2 El Servidor de Administración (obsoleto)

El Servidor de Administración DB2 (DAS) es un *proceso demonio* que ejecuta en el servidor DB2 y permite a clientes remotos administrar gráficamente al servidor DB2. El DAS solo se necesita cuando se utilizan herramientas DB2 gráficas, ya sea en forma local o remota. Hay un único DAS por computador físico, tal como se aprecia en la *Figura 4.16*.

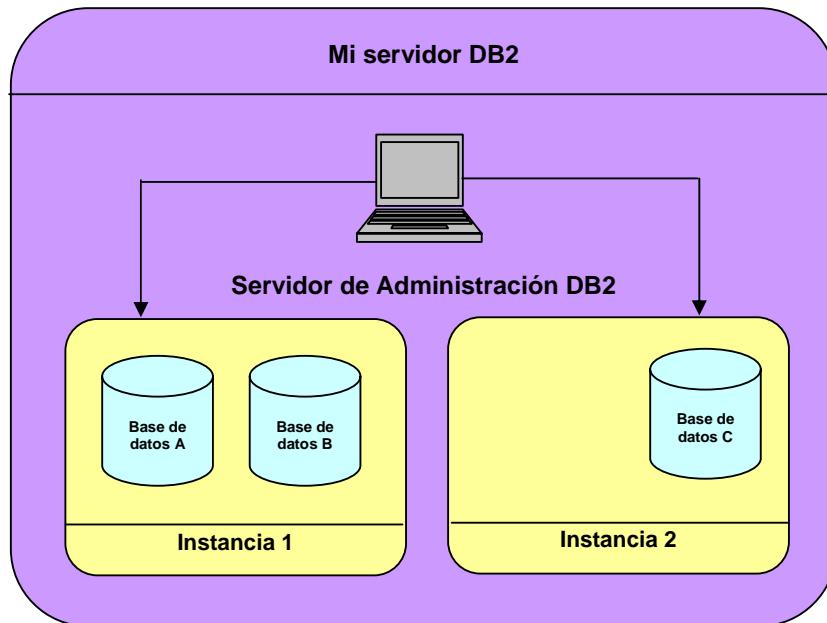


Figura 4.16 – El Servidor de Administración DB2 (DAS)

4.3 Resumen

En este capítulo hemos explorado el ambiente DB2, cubriendo los conceptos de creación de instancias y bases de datos, conjuntamente con sus comandos más usuales. A partir de allí, hemos visto otros aspectos clave de las instancias, incluyendo los espacios de tablas (con los tres tipos de espacios de tablas disponibles) y las tablas, vistas e índices que son creados dentro de los espacios de tablas, sin olvidar los agrupamientos de almacenamiento intermedios y los archivos de anotación cronológica.

Por último, hemos disertado acerca de la estructura de configuración de DB2 y como ésta puede ser alterada en cuatro diferentes lugares, mediante variables de ambiente, el archivo de configuración del manejador de la base de datos, el archivo de configuración de la base de datos y el registro de perfiles de DB2.

4.4 Ejercicios

Los ejercicios en esta sección le permitirán explorar los conceptos vertidos en este capítulo y le presentarán algunas de las herramientas DB2.

Primera parte: Crear una nueva base de datos mediante el Asistente de Creación de Bases de Datos

En esta parte crearemos una base de datos usando el Asistente de Creación de Bases de Datos en el Centro de Control.

Procedimiento

1. En el panel de Vista de Objetos al lado izquierdo del Centro de Control, de clic con el botón derecho del ratón en la carpeta de *Todas las Bases de Datos* y elija las opciones *Crear Base de Datos -> Con Mantenimiento Automático*. Esto lanza el Asistente de Creación de Bases de Datos.
2. Especifique el nombre y ubicación de la base de datos en la página *Nombre* del asistente. Use los siguientes valores:

Nombre de la base de datos:	EXPRESS
Vía de acceso por omisión: (Windows)	C:
Vía de acceso por omisión: (Linux)	/home/db2inst1
Alias:	Este campo, si es dejado en blanco tomará por omisión el valor EXPRESS
Comentario:	Este campo es opcional y puede dejarse en blanco.

Dé clic en el botón *Siguiente* para continuar con la siguiente página del asistente.

**nuevo en
V9.7**

Nota: En Windows, si se crea la base de datos por omisión, solo puede hacerse en un disco, no en un directorio (*path*). Para poder crear una base de datos en un directorio, entonces se debe cargar la variable DB2_CREATE_DB_ON_PATH en el registro de perfiles DB2.

3. En la página de *Almacenamiento*, no haga ningún cambio y dé clic en *Siguiente*.
4. En la página de *Mantenimiento*, deje la opción por omisión (*Sí, puedo especificar un periodo...*), y dé clic en *Siguiente*.
5. Especifique la ventana de tiempo en la cual se puede efectuar mantenimiento fuera de línea en la página de *Momento* del asistente. Configure la ventana de modo que comience a la una de la mañana de lunes a jueves con una duración de seis horas. Luego dé clic en el botón *Siguiente*.
6. Configure las notificaciones en la página del *Servidor SMTP* del asistente. DB2 puede automáticamente avisar a alguien o enviar un email si detecta algún problema o un estado de funcionamiento insatisfactorio. Para ello es en esta página donde debe indicarse el servidor SMTP que DB2 debe utilizar para enviar los emails. Para este ejercicio no disponemos de un servidor SMTP, por lo tanto deje el valor en blanco y de clic en *Siguiente*.

Revise las opciones que ha seleccionado previamente, en la página *Resumen* del asistente. Luego de clic en el botón *Finalizar* para así dar comienzo al proceso de creación de la base de datos. Este proceso toma algunos minutos y durante su ejecución verá un indicador de progreso desplegado en la pantalla.

Segunda Parte: Trabajando con instancias, bases de datos y configuración

En esta parte, crearemos una nueva instancia y una base de datos; además modificaremos parámetros de configuración en un servidor DB2 en Windows. Esto puede hacerse tanto desde el Centro de Control como desde la Ventana de Mandatos. A continuación veremos las instrucciones de como hacerlo desde la Ventana de Mandatos.

Procedimiento

1. Abra la Ventana de Mandatos DB2 de la siguiente forma: Elija *Arranque -> Programas -> IBM DB2 -> DB2COPY1 (Valor por omisión) -> Herramientas de línea de mandatos -> Ventana de mandatos*. Otro modo de hacerlo, más corto, es elegir *Arranque -> Ejecutar*, y tipiar **db2cmd**.

2. Desde la Ventana de mandatos DB2, cree una nueva instancia de nombre **newinst**

```
db2icrt newinst
```

3. Cámbiese a la instancia **newinst** y verifique que realmente sea su instancia válida. Luego de ello hágala partir.

```
set db2instance=newinst (Aviso: ¡No ponga espacios antes o después del signo "="!)
```

```
db2 get instance (Con ésto se verifica que newinst sea la instancia actual)  
db2start
```

4. Cree la base de datos **newdb** con valores por omisión en la instancia **newinst**. Tenga en cuenta que este proceso puede tomar algunos minutos ya que DB2 creará objetos dentro de la base de datos y proveerá alguna configuración inicial.

```
db2 create database newdb
```

5. Conéctese y desconéctese de la nueva base de datos **newdb**. Luego déla de baja.

```
db2 connect to newdb  
db2 terminate  
db2 drop db newdb
```

6. Detenga la instancia actual **newinst**

```
db2stop
```

7. Liste todas las instancias en el servidor

```
db2ilist
```

8. Cámbiese a la instancia DB2 y luego verifique que realmente se ha cambiado

```
set db2instance=db2
db2 get instance
```

9. Dé de baja a la instancia **newinst**

```
db2idrop newinst
```

10. Verifique cual es el valor actual del parámetro FEDERATED en el dbm cfg.

Debería ser NO por omisión

```
db2 get dbm cfg
```

Ayuda: En Linux se puede hacer: db2 get dbm cfg | grep FEDERATED

11. Cambie el valor del parámetro FEDERATED en el dbm cfg a YES y verifique que el cambio se haya efectuado.

```
db2 update dbm cfg using FEDERATED YES
```

Dado que FEDERATED no es un parámetro dinámico, los cambios no se hacen efectivos hasta que no se detiene y arranca la instancia. Sin embargo, para detener la instancia, primero hay que asegurarse que no hay conexiones. Una forma de hacerlo es mediante estos comandos:

```
db2 force applications all
db2 terminate
```

Ahora reciclamos la instancia y verificamos el nuevo valor de FEDERATED:

```
db2stop
db2start
db2 get dbm cfg
```

12. Conéctese con la base de datos SAMPLE usando el identificador de usuario y clave que está usando en el sistema operativo

```
db2 connect to sample user <identificación_de_usuario> using <clave>
```

13. Revise cuantas aplicaciones están ejecutándose en la instancia actual

```
db2 list applications
```

82 Conociendo al DB2 Express-C

14. Abra otra Ventana de Comandos DB2 y conéctese nuevamente a la base de datos SAMPLE sin especificar ni usuario ni clave. Luego vea cuantas conexiones hay.

```
db2 connect to sample  
db2 list applications
```

15. Finalice por la fuerza la conexión de una de las ventanas de mandatos DB2. Ésto le dará un ejemplo de como un administrador de bases de datos puede terminar por la fuerza el trabajo de un usuario dado (el cual probablemente está consumiendo los recursos del sistema)

```
db2 force application (<manejador de aplicación para db2bp.exe>)
```

El manejador de aplicación es el número de la aplicación que se desea terminar por la fuerza. Este número se obtiene del listado del comando **db2 list applications**. La aplicación db2bp.exe representa a la Ventana de Mandatos DB2.

16. Verifique que la conexión de una de las ventanas de comando DB2 ha sido terminada por la fuerza. En caso que no sepa en cual de las dos ventanas de comando DB2 la conexión con la base de datos ha terminado, ejecute esta sentencia en ambas ventanas:

```
db2 select * from staff
```

La Ventana de Mandatos cuya conexión fue terminada por la fuerza debería arrojar un mensaje con el código SQL1224N. La otra ventana debería entregar la respuesta a la consulta.

17. Dé de baja al DAS, créelo nuevamente y hágalo arrancar.

```
db2admin stop  
db2admin drop  
db2admin create  
db2admin start
```

18. Vea cual es el valor actual de la variable DB2COMM en el registro

```
db2set -all
```

19. Limpie la variable DB2COMM del registro y verifique que está limpia

```
db2set db2comm=  
db2stop
```

(Ayuda: Recibirá un error al ejecutar el comando db2stop si es que hay conexiones activas. ¿Qué se debe hacer? Vea uno de los pasos anteriores para resolver esto)

```
db2start  
db2set -all
```

20. Asigne a la variable DB2COMM del registro DB2, los valores **tcpip** y **npipe** en su instancia y verifique esos valores

```
db2set db2comm=tcpip,npipe  
db2stop  
db2start  
db2set -all
```

21. Verifique el valor actual del parámetro LOGSECOND en el db cfg, y luego cámbielo a un valor de 5. Posteriormente verifique el valor nuevo

```
db2 connect to sample  
db2 get db cfg  
db2 update db cfg using LOGSECOND 5  
db2 get db cfg
```


5

Capítulo 5 – Herramientas de DB2

nuevo en
V9.7

En este capítulo, describiremos algunas de las herramientas que puedes usar con DB2. A partir de DB2 9.7, la mayoría de estas herramientas son obsoletas, si bien seguirán funcionando, no se les hará mejoras, y podrían no estar incluidas con el producto en futuras versiones. IBM Data Studio se encarga de reemplazar a las herramientas anteriores a partir de DB2 9.7. La elipse roja en la figura 5.1 muestra el área en la que se centra este capítulo.

DB2 Commands & SQL/XML, XQuery Statements

SQL/XML, Xquery Statements
create bufferpool
create tablespace
create table
alter bufferpool
alter tablespace
alter table
select
insert
update
delete
...
DB2 System Commands
db2set
db2start
db2stop
db2ilist
db2icrt
db2idrop
...
DB2 CLP Commands
db2 update dbm cfg
catalog db
list node directory
create database
list applications
list tablespaces
...
<sql statement>
xquery < >

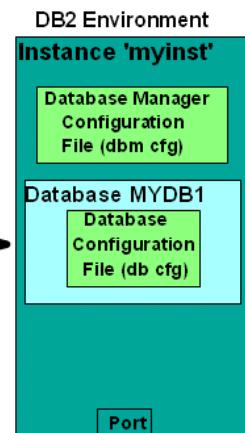


Figura 5.1 – La gran imagen de DB2: herramientas DB2

Nota:

Ver las presentaciones en video acerca de las herramientas DB2 y de scripting en estos links:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4202>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4182>

La figura 5.2 lista todas las herramientas disponibles en el menú de inicio de DB2. La mayoría de esas herramientas son las mismas en Linux y Windows.

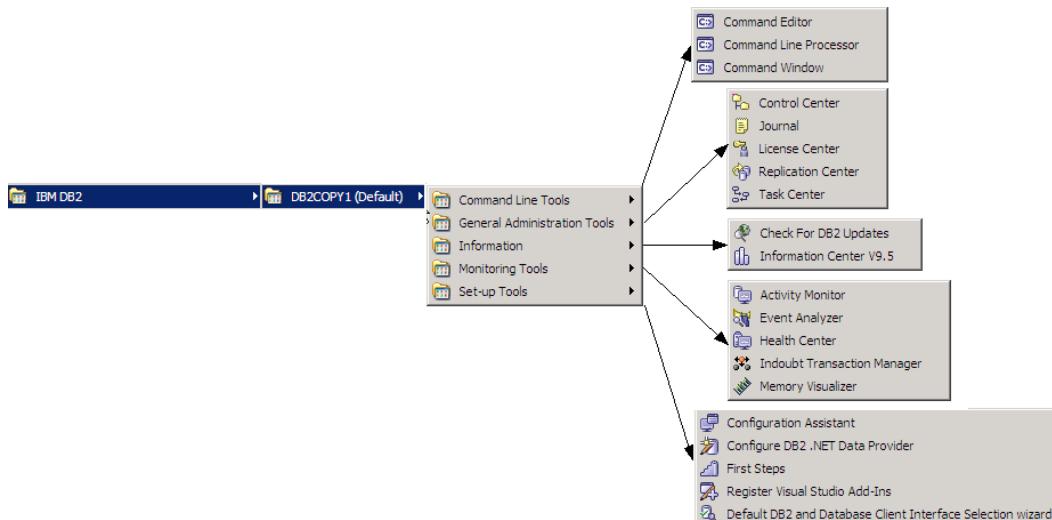


Figura 5.2 – Herramientas DB2 desde el menú de inicio de DB2

La tabla 5.1 provee una lista de comandos que pueden ser usados para iniciar algunas de las herramientas más populares tanto en Linux o Windows.

Nombre de la herramienta	Comando
Editor de Comandos	db2ce
Command Line processor	db2
Ventana de Comandos (Sólo disponible en Windows)	db2cmd
Control Center	db2cc

Task Center	db2tc
Health Center	db2hc
Configuration Assistant	db2ca
First Steps	db2fs

Tabla 5.1 – Comandos para iniciar algunas herramientas DB2

nuevo en
V9.7

5.1 IBM Data Studio

A partir de DB2 9.7, IBM Data Studio es la principal herramienta para la administración de bases de datos y desarrollos con DB2. IBM Data Studio no tiene costo. Puede ejecutarse en Linux y Windows, y es parte del [IBM Integrated Data Management portfolio](#). El desarrollo de Data Studio sigue una planificación que no está relacionada con las versiones de DB2; sin embargo, los productos tratan de estar con el mismo número de versión. Por ejemplo, IBM Data Studio 2.2 y DB2 9.7 fueron lanzados el mismo día en Junio de 2009.

La figura 5.3 muestra cómo luce la interface gráfica de IBM Data Studio.

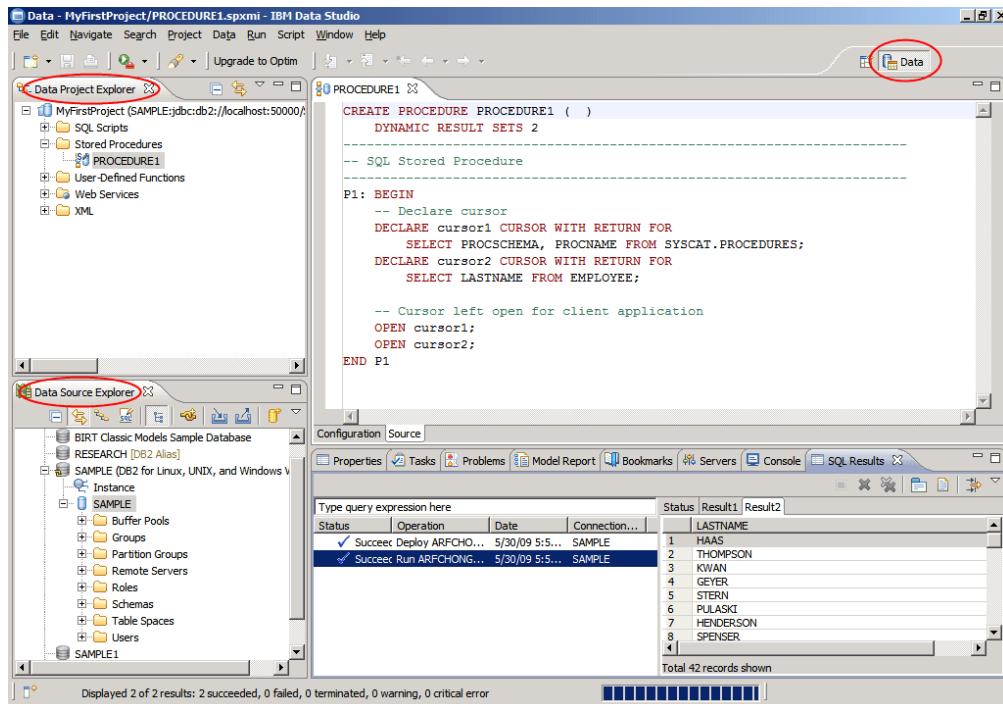


Figura 5.3 - IBM Data Studio

Si está familiarizado con Eclipse, notará que Data Studio se basa en la mencionada plataforma. Con Data Studio, generalmente se trabajará dentro de la ventana de perspectiva de Datos (resaltado en la figura en la esquina superior derecha). También se puede cambiar a la perspectiva de Java si es que se está desarrollando un programa Java. Hay dos vistas resaltadas en la figura:

- Data Project Explorer (izquierda superior)
- Data Source Explorer (izquierda inferior)

La vista Data Project Explorer se utiliza para desarrollar scripts SQL, XQuery, procedimientos almacenados, UDFs, y servicios de Data Web.

La vista Data Source Explorer es utilizada por los administradores de bases de datos para gestionar instancias y bases de datos DB2. Por medio de esta vista, se pueden realizar la mayoría de las funciones previamente disponibles en el Control Center.

En la figura, la vista con el título PROCEDURE1 es un editor para el procedimiento resaltado en el Data Project Explorer. Dependiendo de la tarea que se está ejecutando, editores u otras ventanas aparecerán, permitiendo codificar o realizar más configuraciones.

IBM Data Studio viene en dos variantes

- Stand-alone package
- IDE package

El stand-alone package es mucho más pequeño que el IDE package y no soporta el desarrollo de los servicios de Data Web, ni tampoco puede extenderse a otros productos de IBM basados en Eclipse como InfoSphere Data Architect. Fuera de eso, la interface y capacidades que tiene son las mismas.

Con IBM Data Studio, se puede trabajar con otros servidores de datos, como Informix. Las compañías que trabajan con varios servidores de datos y tienen un pequeño equipo de desarrolladores y DBAs, ahora tienen la posibilidad de trabajar y administrar los mismos desde una única herramienta.

Nota:

Para más información acerca de Data Studio, referirse al libro [Getting Started with IBM Data Studio for DB2](#) eBook sin cargo el cual es parte de esta serie de libros.

5.1 Centro de Control (Control Center) (obsoleto)

La herramienta primaria para la administración de DB2 es el Centro de Control, ilustrado en la figura 5.3.

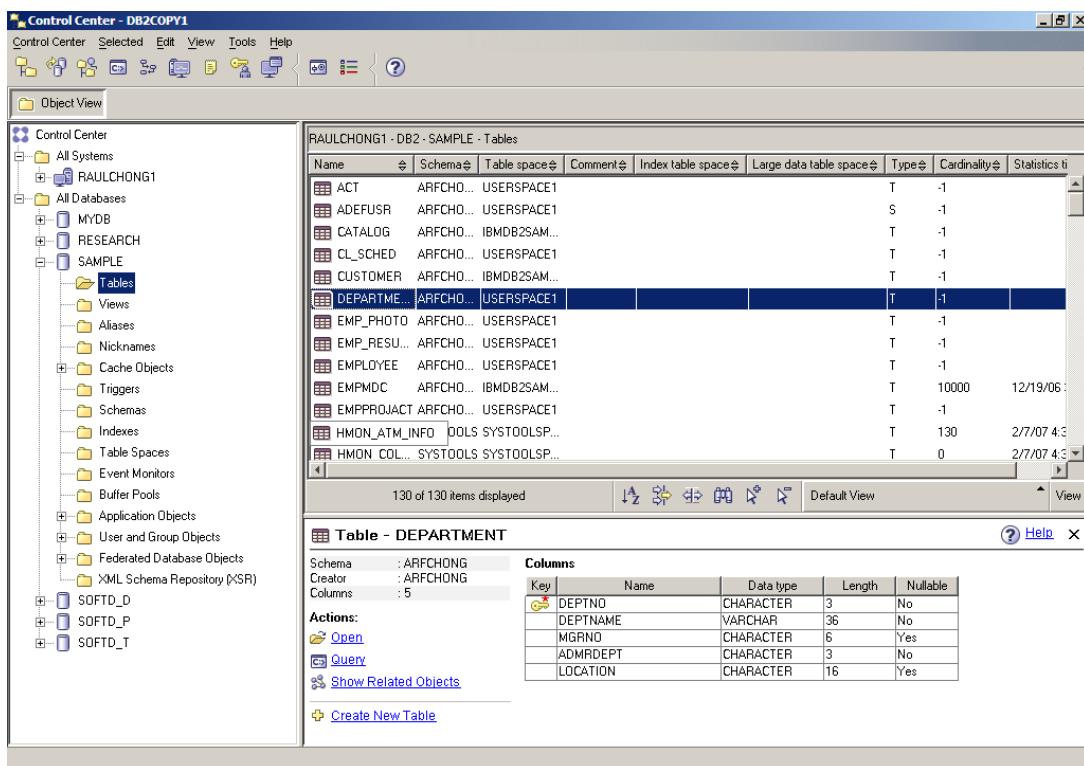


Figura 5.3 – El Centro de Control de DB2

El Centro de Control es una herramienta de administración centralizada que te permite:

- Ver el sistema, instancias, bases de datos y objetos de la base de datos;
- Crear, modificar y manejar objetos de las bases de datos;
- Lanzar otras herramientas gráficas de DB2

El panel del lado izquierdo provee una visión jerárquica de los objetos de la base de datos en el sistema, teniendo una carpeta para Tablas, Vistas, etc. Al dar doble clic sobre una carpeta (por ejemplo, como se muestra en la figura 5.3), del lado derecho listaremos todos los objetos relacionados, en este caso, todas las tablas asociadas con la base de datos SAMPLE. Si se selecciona una tabla en la parte alta del panel, en la parte baja del lado derecho del panel se proveerá información más específica acerca de esa tabla.

Hacer clic derecho sobre los diferentes objetos/carpetas en el árbol de objetos para ver el menú con las opciones aplicables a los objetos/carpetas. Por ejemplo, dar clic derecho sobre una instancia y elegir “Configure parameters” que te permite ver y actualizar el archivo de configuración del manejador de base de datos. De forma similar, si das clic derecho sobre una base de datos y se elige “Configure parameters”, se podrá ver y actualizar el archivo de configuración de la base de datos. Los parámetros de ambiente y configuración DB2 se discuten a mayor detalle en el capítulo 4, El Ambiente DB2.

La primera vez que se lanza el Centro de Control, preguntará qué vista se prefiere usar. La elección de la vista determina qué tipo de opciones y objetos de la base de datos son expuestos. La figura 5.4 muestra la caja de diálogo de la vista de el Centro de Control.

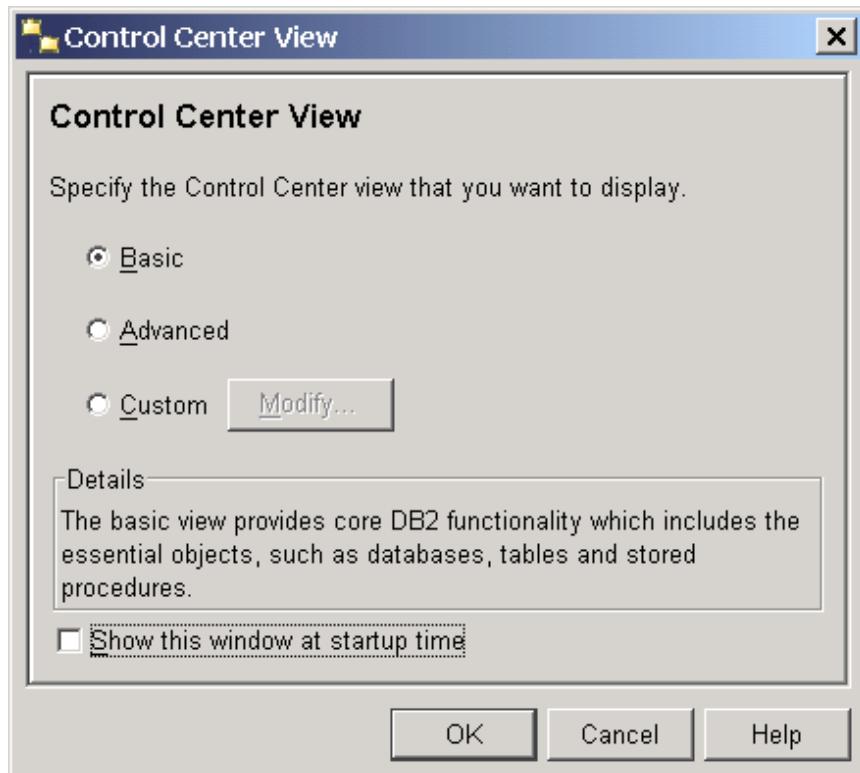


Figura 5.4 – La caja de diálogo de el Centro de Control DB2

La vista básica provee la funcionalidad elemental.

La vista avanzada muestra más opciones y mejoras.

La vista personalizada permite ajustar las mejoras específicas, opciones y objetos que se mostrarán.

Para volver a invocar el diálogo Control Center View, selecciona la opción "Customize Control Center" desde el menú Tools como se muestra en la figura 5.5.

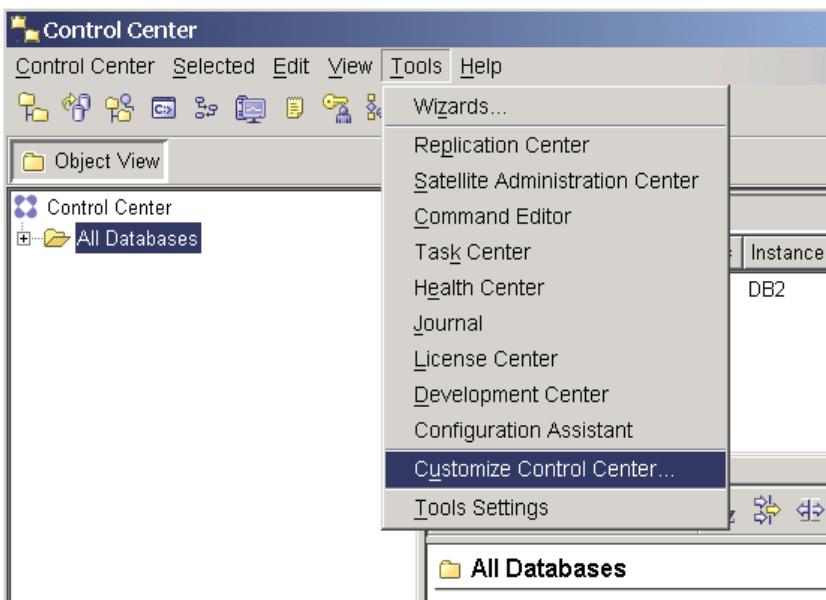


Figura 5.5 – Personalizando el Centro de Control

Lanzando el Centro de Control

Hay varias formas de lanzar el Centro de Control :

- Navegando a través del menú de inicio de Windows
- Ejecutando db2cc en la línea de comandos
- Dando clic en el ícono de el Centro de Control  en la barra de herramientas de cualquier otra herramienta gráfica de DB2
- Desde el ícono DB2 en Windows como se muestra en la figura 5.6 (Da clic sobre el ícono verde DB2 y selecciona la opción Central de Control de DB2)

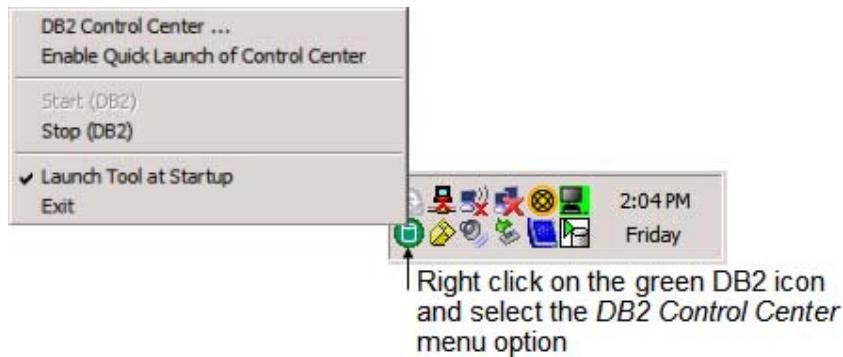


Figura 5.6 – Lanzando el Centro de Control de DB2 desde la bandeja de Sistemas de Windows

5.2 El Editor de Comandos DB2 (obsoleto)

Usando el Editor de Comandos DB2 (editor de comandos), se pueden ejecutar comandos DB2, sentencias SQL y XQuery, analizar el plan de ejecución de una sentencia, y ver o actualizar resultados de consultas.

La figura 5.7 muestra el Editor de Comandos con una descripción de sus elementos.

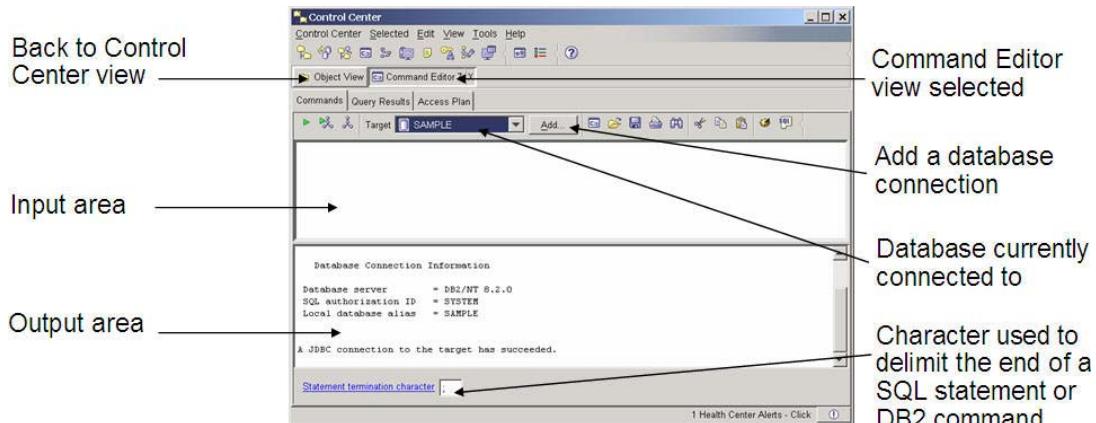


Figura 5.7 – El Editor de Comandos DB2

En el área de entrada, puedes ingresar varias sentencias, tan largas hasta que se presente el carácter de finalización. Si se presiona el botón ejecutar (ver figura 5.8), las sentencias serán ejecutadas una después de otra. Si tú seleccionas explícitamente una sentencia, sólo esa sentencia será ejecutada. Debe existir una conexión a la base de datos para devolver sentencias SQL, sin embargo, una de esas sentencias puede ser una sentencia de conexión.

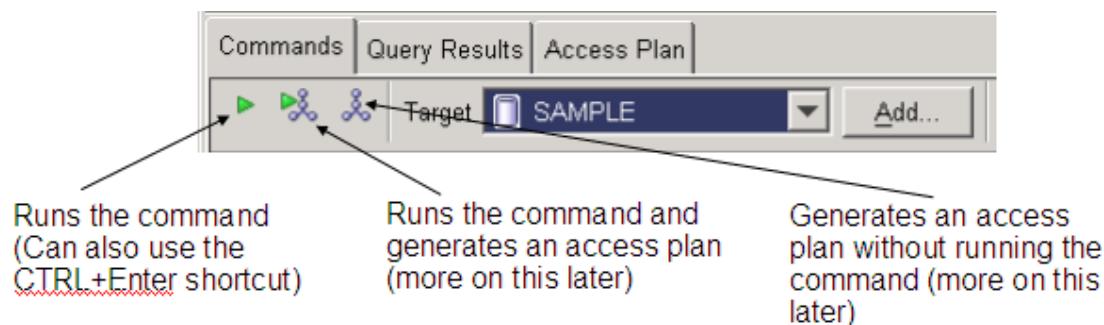


Figura 5.8 – El Editor de Comandos – Pestaña de comandos

Lanzando el Editor de Comandos

Se podrá lanzar el Editor de Comandos de varias formas:

- ▶ Desde el menú de inicio de Windows:
Start -> Programs -> IBM DB2 -> DB2COPY1 (Default) -> Command Line Tools -> Editor de Comandos
- ▶ Desde la línea de comandos, escribe db2ce
- ▶ Desde el menú herramientas en El Centro de Control
- ▶ Dentro de el Centro de Control
 - Dar clic derecho en el ícono de la base de datos SAMPLE en el árbol de objetos del panel de el Centro de Control y selecciona el elemento del menu *Query*
 - Siempre que un objeto consultable esté seleccionado (base de datos, tabla, etc.), se puede lanzar el Editor de Comandos dando clic en la liga de *Query* que está en el panel de detalle de objetos de el Centro de Control
- ▶ Desde el Centro de Control , dar clic en el ícono  de Editor de Comandos en la barra de herramientas como muestra la figura 5.9



Figura 5.9 – El ícono Editor de Comandos en El Centro de Control

Agregando una conexión a una base de datos

Para agregar una conexión a una base de datos, dar clic en el botón Add (agregar). Un diálogo como en la figura 5.10 aparecerá.

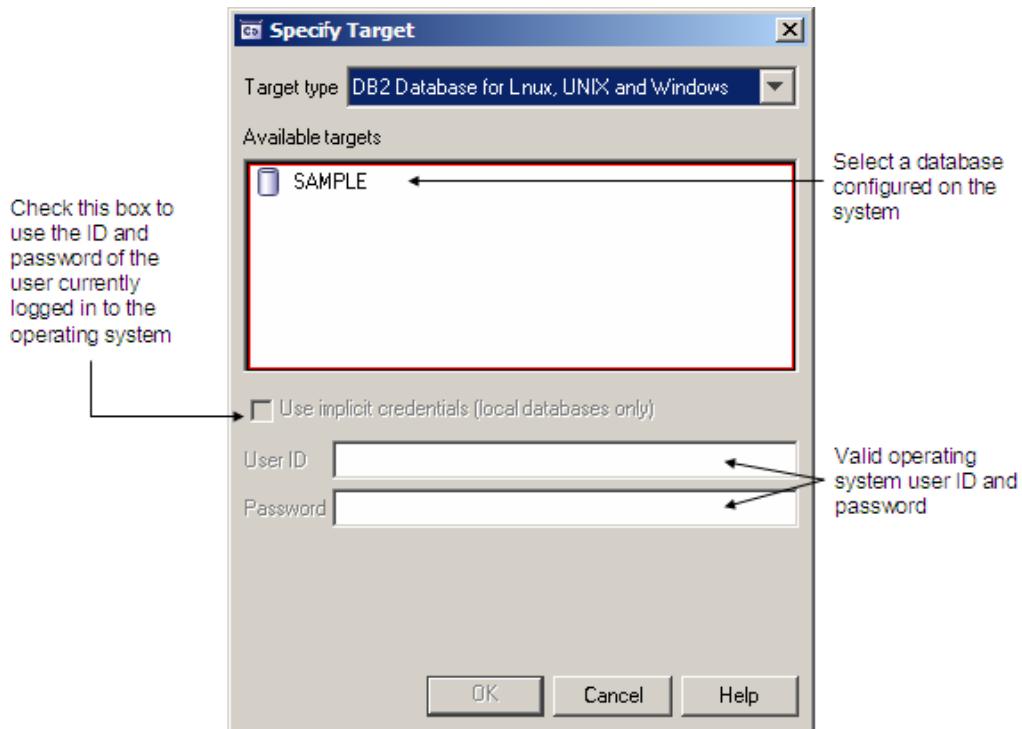


Figure 5.10 – Agregar una conexión a la base de datos

5.3 SQL Assist Wizard (obsoleto)

Si no está familiarizado con el lenguaje SQL se podría usar un asistente para generar código SQL, el SQL Assist Wizard está disponible en el Editor de Comandos para ayudarte. Como se muestra en la figura 5.11, invócalo desde el Editor de Comandos dando clic sobre le último ícono con el símbolo SQL.

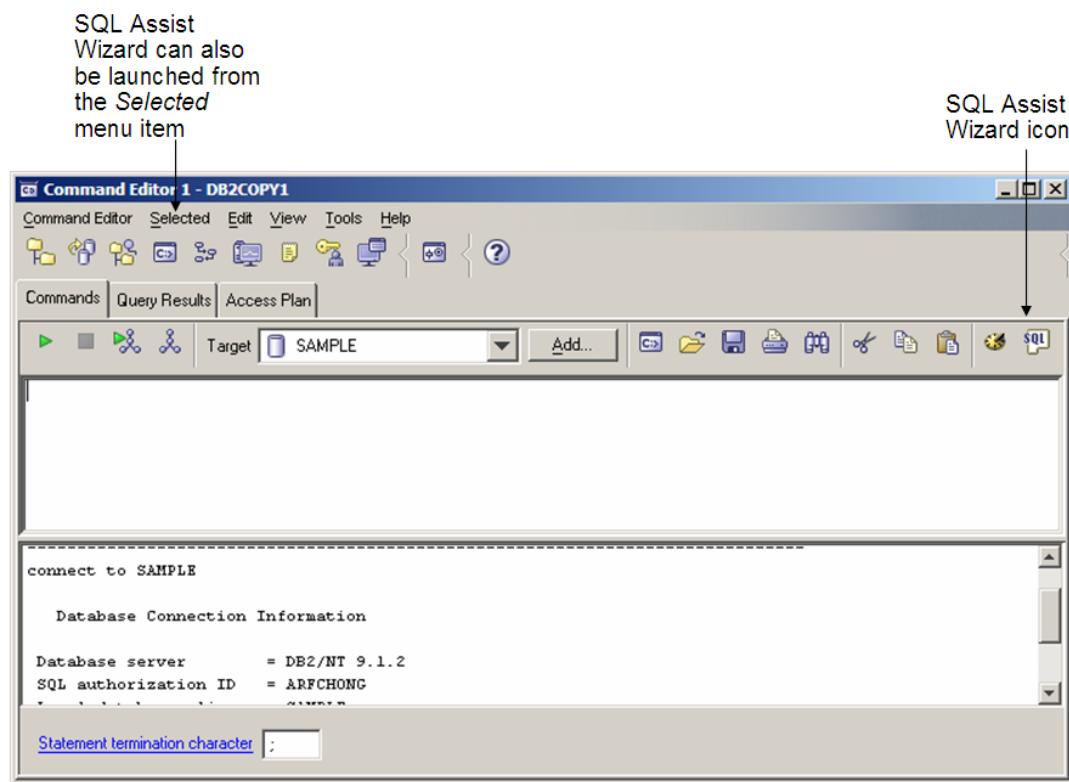


Figure 5.11 – Invocando el SQL Assist Wizard

La figura 5.12 muestra el SQL Assist Wizard. Es bastante claro su uso. Primero indica el tipo de sentencia SQL para la cual necesitas asistencia (SELECT, INSERT, UPDATE, DELETE). Dependiendo de que sentencia se haya elegido, diferentes opciones aparecerán. En la parte baja de la ventana verás cómo la sentencia SQL es construida a partir de diferentes elecciones en el asistente.

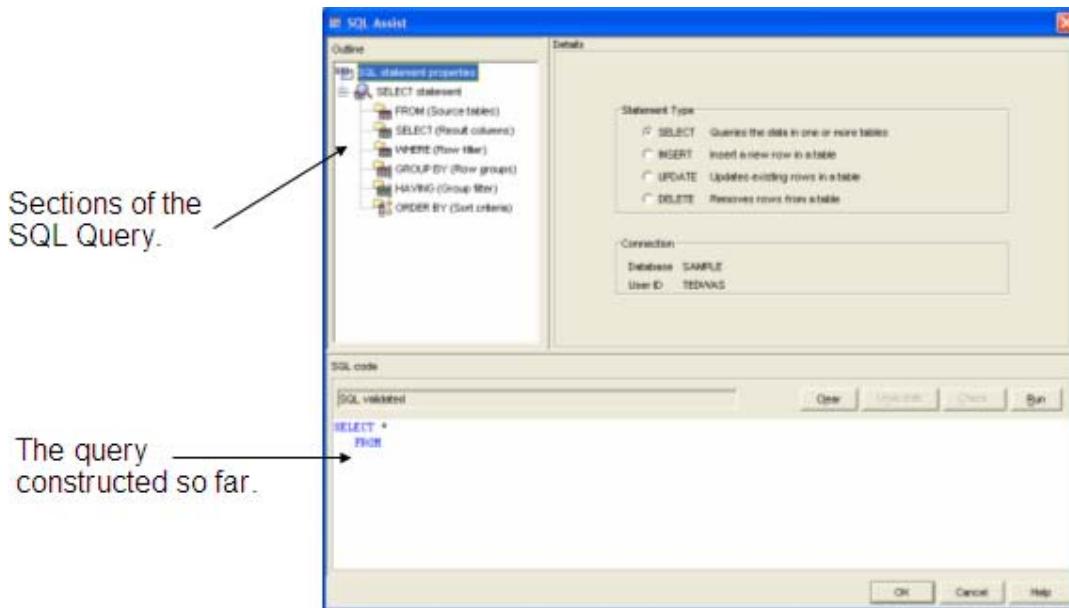


Figura 5.12 – El SQL Assist wizard

5.4 El botón Show SQL (obsoleto)

La mayoría de las herramientas gráficas y los asistentes en DB2 permiten revisar el comando actual o la sentencia SQL que es creada como resultado de usar la herramienta para realizar tal acción. Para ver esto, da clic en el botón Show SQL en la herramienta en la que estés trabajando, como lo muestra la figura 5.13 y figura 5.14



Figura 5.13 – El botón Show SQL

```

CONNECT TO SAMPLE;
CREATE view TEDWAS.VIEW1 [(columns, ...)]
AS [WITH (common_table_expression, ...)]
SELECT <columns>
FROM <tables>
WHERE <search_conditions>;
CONNECT RESET;

```

Figura 5.14 – La salida de un botón Show SQL

La posibilidad de revisar las sentencias SQL y comandos es muy útil para aprender sintaxis SQL, y para guardar comandos o sentencias para su posterior uso. También se puede construir scripts re usando esos comandos y sentencias generados.

5.6 Centro de Tareas (Task Center) (obsoleto)

La herramienta visual Centro de Tareas permite la creación de tareas como: grupo de operaciones para ejecutar comandos DB2, comandos del sistema operativo o scripts. En caso de que las mismas resulten exitosas o fallen, se pueden invocar subsiguientes tareas para complementar todo el proceso. Por ejemplo, si una tarea de backup de una base de datos se ejecuta a las 3:00 AM y finaliza exitosamente, se puede enviar un e-mail al administrador de la base de datos informándole de este resultado. Por otro lado, en caso de falla, el Centro de Tareas puede enviarle una alarma telefónica o alerta debido a ser esta, una situación que requiere de atención inmediata. La *Figura 5.16* muestra el Centro de Tareas.

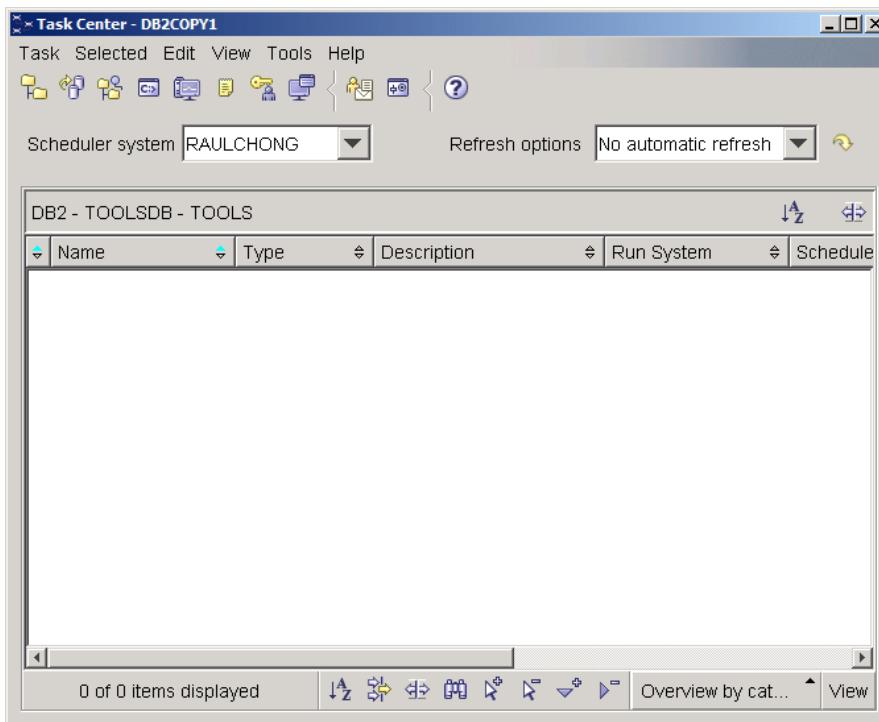


Figura 5.16 – El Centro de Tareas

5.6.1 La base de datos del catálogo de herramientas (The Tools Catalog database) (obsoleto)

Los detalles de las tareas y la planificación de las mismas se almacenan en una base de datos de DB2 llamada base de datos del catálogo de herramientas. Esta base de datos

debe existir previo a la planificación de tareas. Para crear esta base de datos, se puede utilizar este comando:

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

El ejemplo anterior, **systools** es el esquema de todas las tablas de la base de datos y el nombre de la base de datos es **toolsdb**. Se comentará más acerca de los esquemas en el *Capítulo 8, Trabajando con objetos de bases de datos*.

5.6.1.1 Invocando al Centro de Tareas

Se puede invocar al Centro de Tareas desde el Centro de Control, seleccionando *Tools -> Task Center*, como se muestra en la *Figura 5.17*. Otra forma de invocar la misma es desde el menú de Inicio de Windows: *Start -> Programs -> IBM DB2 -> DB2COPY1(Default) -> General Administration Tools -> Task Center*

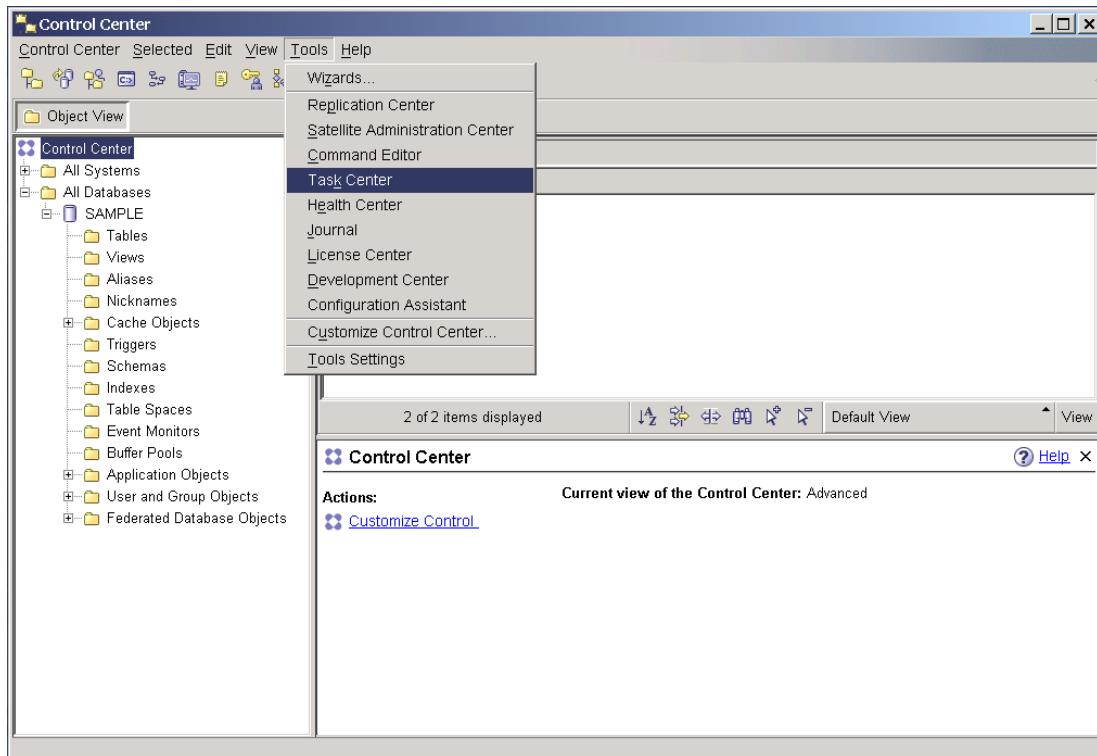


Figure 5.17 – Invocando el Centro de Tareas

5.6.1.2 Planificando con el Centro de Tareas

Cualquier tipo de script, sea creado o no con una herramienta visual de DB2, se puede planificar utilizando el Centro de Tareas. Las tareas se ejecutan al horario establecido desde el sistema donde se creo la base de datos del catálogo de herramientas.

Incentivamos al lector a explorar el Centro de Tareas, donde se verá lo sencillo que es crear una tarea.

5.7 Journal (obsoleto)

La herramienta gráfica DB2 Journal le brinda al administrador una lista de tareas visualizadas en el momento. La *Figura 5.18* muestra el Journal y la *Tabla 5.2* muestra la información que se puede obtener de esta herramienta.

The screenshot shows the DB2 Journal application window. The title bar reads "Journal". The menu bar includes "Journal", "Selected", "Edit", "View", "Tools", and "Help". Below the menu is a toolbar with various icons. The main area has tabs: "Task History" (selected), "Database History", "Messages", and "Notification Log". A sub-tab "Database" is shown above the list. The list table has columns: "Object Type", "Name", "Operation", and "Start Date". The data shows several entries for "TAB1" and "SYSTOOLSPACE", mostly "Drop" operations on Sep 19, 2004, with one "Create" entry on Sep 18, 2004. The bottom of the window shows "7 of 7 items displayed" and a set of navigation buttons.

Object Type	Name	Operation	Start Date
Table	TAB1	Drop	Sep 19, 2004
Table	TAB1	Drop	Sep 19, 2004
Table Space	SYSTOOLSPACE	Create	Sep 18, 2004
Table Space	SYSTOOLSPACE	Create	Sep 18, 2004
Table	TAB1	Drop	Sep 19, 2004
Table Space	SYSTOOLSPACE	Create	Sep 18, 2004
Table	TAB1	Drop	Sep 19, 2004

Figure 5.18 –El Journal

Tipo de Información	Descripción
Task History	Todas las tareas planificadas que se ejecutaron y su correspondiente estado de finalización
Database History	Un registro de las actividades de mantenimiento realizadas de la base de datos (resguardo/backup, rerestore, REORGs, etc.)
Message	Historia de los mensajes de las herramientas de DB2. Es de gran utilidad para recordar situaciones y comparar antiguos mensajes o en caso de que se haya cerrado una ventana de diálogo muy rápido o por error.
Notification Log	Contiene mensajes a nivel sistema. Los errores críticos se guardan aquí.

Tabla 5.2 - Información provista en el Journal

5.7.1 Invocando el Journal

Se puede ejecutar el Journal desde el Centro de Control seleccionando *Tools -> Journal*, tal como se muestra en la *Figura 5.19*. Otra forma es desde el menú Inicio de Windows: *Start -> Programs -> IBM DB2 -> DB2COPY1(Default) -> General Administration Tools -> Journal*.

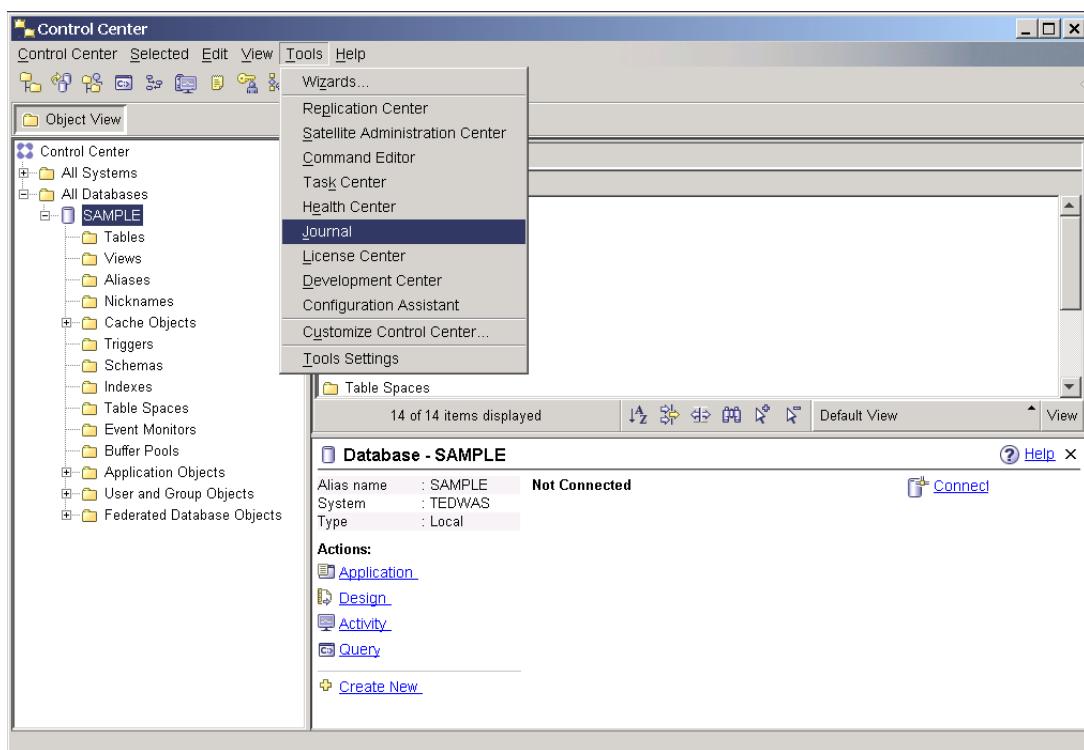


Figura 5.19 – Invocando el Journal

5.8 Monitor de Salud (Health Monitor) (obsoleto)

El Monitor de Salud es un agente que se ejecuta dentro del motor DB2, cuya función es la de revisar todos los aspectos que hacen a la salud de la base de datos (memoria, administración de espacio, actividades automáticas previamente definidas, etc.). Cuando alguna parte de DB2 opera fuera de los parámetros establecidos, una excepción se origina y requiere ser atendida por el administrador de la base de datos. Hay tres tipos de estados de alerta:

- Atención: Estado anormal
- Advertencia/Warning: Estado no crítico que no requiere atención inmediata pero puede indicar una situación sub-óptima
- Alarma: Condición crítica que requiere atención inmediata

El Monitor de Salud se puede activar o desactivar a través del parámetro de configuración del administrador de la base de datos: **HEALTH_MON**. (parámetro a nivel instancia)

5.8.1 Centro de Salud (Health Center) (obsoleto)

El Centro de Salud es una herramienta gráfica que interactúa con el Monitor de Salud. El Centro de Salud divide las alertas de salud a nivel instancia, base de datos y espacio de tablas dentro de un sistema. *La Figura 5.20 muestra el Centro de Salud.*

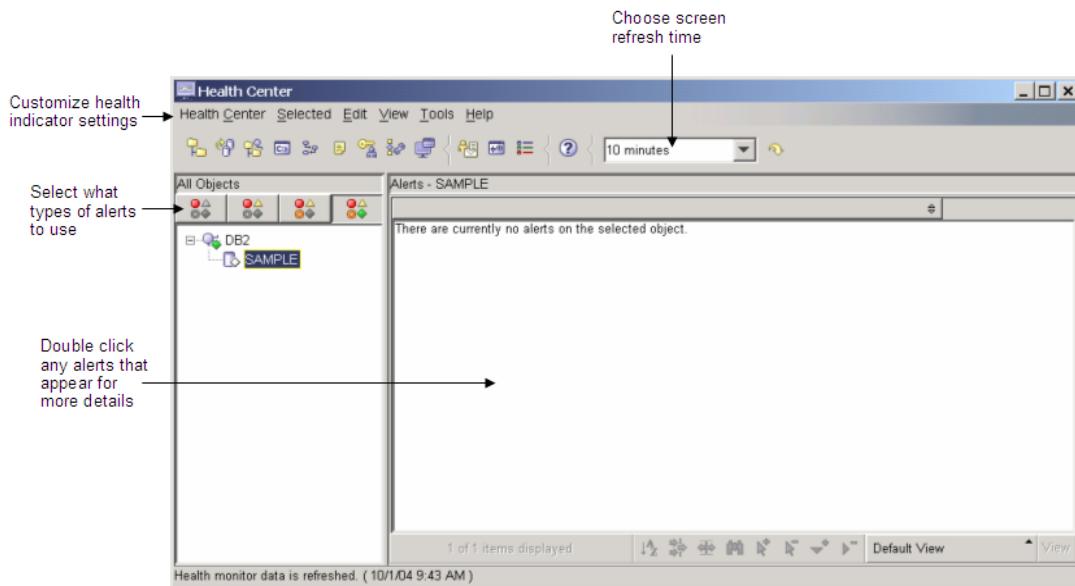


Figura 5.20 – El Centro de Salud

5.8.1.1 Invocando el Centro de Salud

Puede invocar el Centro de Salud desde el Centro de Control seleccionando *Tools -> Health Center* como se muestra en la *Figura 5.21*. También se puede invocar esta herramienta desde el meú de Inicio de Windows. *Start -> Programs-> IBM DB2 -> DB2COPY1(Default) -> Monitoring Tools -> Health Center*

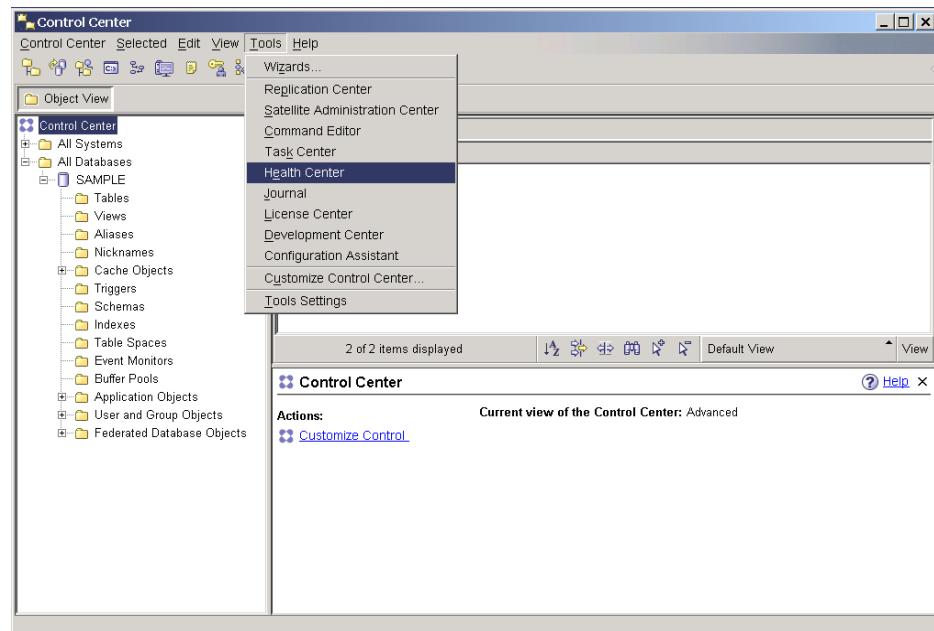


Figura 5.21 – Invocando el Centro de Salud

5.8.1.2 Configurando la Notificación de Alerta de Salud

Una vez que se ejecuta el Centro de Salud, se puede configurar la notificación de Alerta seleccionando el menú *Health Center -> Configure -> Alert Notification* como se muestra en la Figura 5.22. La notificación de Alerta permite ingresar nombre de contactos con direcciones de e-mail ó numeros de alertas (pager) de la gente a ser contactada en caso de que ocurra una situación que requiera atención.

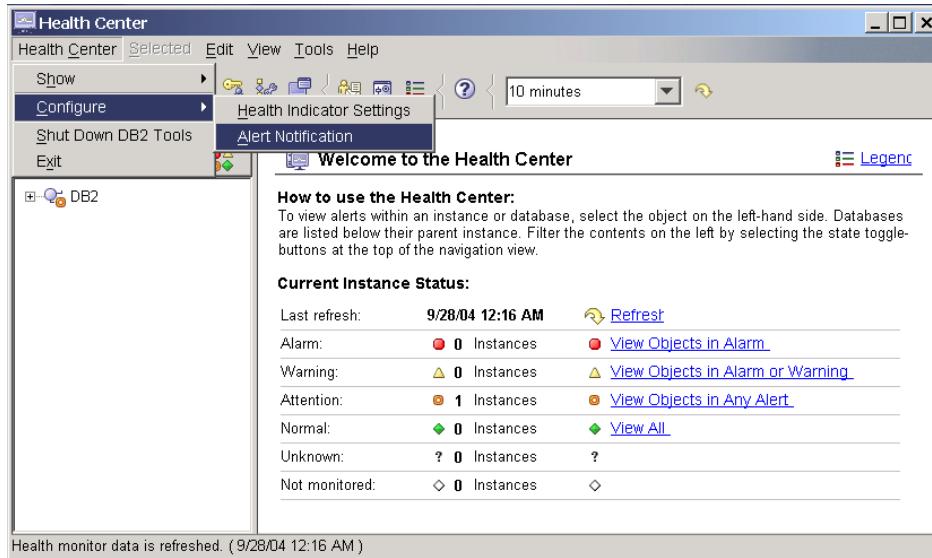


Figura 5.22 – Notificación de Alerta

5.9 Administrador de memoria auto-ajustable

El administrador de memoria auto-ajustable (STMM) presentado en la versión 9 de DB2 es una de las tantas características que simplifica la tarea de configuración de memoria estableciendo valores de manera automática a varios parámetros de memoria. Cuando está habilitado, el gestor automático dinámicamente distribuye los recursos de memoria disponibles entre varios consumidores para la base de datos. El administrador de memoria responde a los cambios de la carga de trabajo ajustando los valores de memoria a través de parámetros de configuración y los almacenes de memoria (buffer pools). El gestor de memoria responde a los cambios en las características de carga de trabajo, ajustando los valores de los parámetros de configuración de memoria y los tamaños de los buffer pools para optimizar el rendimiento. Para activar el STMM, se actualiza el parámetro SELF_TUNING_MEM al valor ON.

Otras características computacionales automáticas tales como el mantenimiento automático y el almacenamiento automático se discuten en capítulos posteriores.

5.10 Scripting

Siempre es útil poder crear scripts que realizan varios comandos DB2 o sentencias SQL de forma repetida. Por ejemplo, un administrador de base de datos puede necesitar ejecutar un script a diario para revisar la cantidad de filas que tienen las tablas más importantes. Hay dos formas generales de scripting:

- Scripts SQL
- Scripts del sistema operativo (shell scripts).

5.10.1 Scripts SQL

Los scripts SQL contienen sentencias de consultas y comandos de base de datos. Estos scripts son simples de entender e independientes de la plataforma. Sin embargo, no soportan variables o parámetros. Por ejemplo, los comandos que se muestran en el *Listado 5.1* debajo se guardan en un archivo denominado `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;
```

Listado 5.1 – Un Script SQL de ejemplo que se guarda en el archivo script1.db2

Este script contiene todas sentencias SQL y cada sentencia está separada por un delimitador, el punto y coma en este caso. El nombre del archivo no necesariamente tiene que tener la extensión “db2”. Cualquier extensión se puede utilizar para estos archivos.

5.9.1.1 Ejecutando Scripts SQL

Un script SQL puede ser ejecutado en el Editor de Comando, en la Ventana de Comando DB2 o en el shell de Linux. Para ejecutar el script del *Listado 5.1* desde la ventana de Comandos de DB2 o desde el shell de Linux, se puede utilizar el siguiente comando:

```
db2 -t -v -f script1.db2 -z script1.log
```

o el equivalente:

```
db2 -tvf script1.db2 -z script1.log
```

En este comando:

- t** indica que las sentencias utilizan el carácter de terminación por omisión (punto y coma)
- v** indica que tiene que mostrar en pantalla el resultado del comando que se está ejecutando
- f** indica el nombre del archivo a ejecutar
- z** indica que el nombre de archivo especificado se utiliza para posterior análisis (parámetro opcional, pero recomendado)

Cuando se utiliza **-t** y no se especifica ningún delimitador, se asume al punto y coma como delimitador de las sentencias. Puede haber situaciones donde otro delimitador se requiera. Por ejemplo, un script que contiene código SQL PL necesita utilizar un carácter de terminación diferente al punto y coma debido a que los punto y coma se utilizan dentro de las definiciones de objetos de SQL PL para indicar la finalización de esas definiciones.

Por ejemplo, en el archivo `functions.db2` que se muestra en el *Listado 5.2*, se puede visualizar la sentencia para crear una función, y se necesita un punto y coma al final de la sentencia `SELECT` porque es parte de la sintaxis requerida dentro de la función. Como carácter de finalización de la sentencia `CREATE FUNCTION`, utilizamos un signo de exclamación (!). Si hubiéramos utilizado un punto y coma para la finalización de dicha sentencia, se hubiera ocasionado un conflicto en el script con el carácter de finalización de la sentencia SQL, resultando en un error reportado por DB2.

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

Listado 5.2 - Contenidos del archivo functions.db2

Para informarle a DB2 que se va a utilizar un carácter de finalización diferente, se usa el indicador `-d`, seguido por el carácter de terminación deseado como se indica a continuación:

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

Para saber más de otros indicadores que pueden utilizarse desde la Ventana de Comandos o del shell de Linux, ejecutar lo siguiente:

```
db2 list command options
```

5.10.2 Scripts del sistema Operativo (shell scripts)

Los scripts del sistema operativo proveen más flexibilidad y poder que los scripts SQL, debido a que brindan la posibilidad de agregar lógica de programación adicional. Estos comandos dependen de la plataforma y soportan parámetros y variables. *El listado 5.3* muestra un ejemplo sencillo de un script del sistema operativo Windows (shell script).

```
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
```

Listado 5.3 – Contenido del script de sistema operativo del archivo create_database.bat

Para ejecutar este shell script desde la linea de comando, se invoca el siguiente comando desde Windows:

```
create_database.bat db2admin ibmdb2
```

donde **db2admin** es el userID y el primer parámetro del script, e **ibmdb2** es la palabra clave (paasword) y el segundo parámetro.

En Windows, al utilizar la extensión “bat”, se le está indicando al sistema operativo que es un archivo ejecutable de procesamiento por lotes. En Linux, es necesario cambiar el modo de ejecución del archivo para indicar que sí se pueda ejecutar. Esto se logra mediante el comando **chmod +x**. Luego de dicho cambio, se ejecuta el archivo de la misma forma que se indicó anteriormente.

5.11 Consideraciones para Windows Vista

La característica de Control de Acceso de Usuario (UAC) en Windows Vista causa que las aplicaciones sean ejecutadas con privilegio estándar, aunque el usuario sea administrador local. Esto quiere decir que cualquier herramienta DB2 o comando que se ejecute en Vista se puede ejecutar sin problemas a pesar de que se informen problemas relacionados con el acceso. Para evitar esta situación, hay que invocar el acceso directo “Command window – Administrator” específicamente creado al momento de instalación para los usuarios de Vista. Desde esta ventana, se pueden ejecutar otros comandos e invocar otras herramientas (utilizando los comandos que se muestran en la *Tabla 5.1 al comienzo del capítulo*). Como opción alternativa, desde el menú de Inicio de Windows Vista o desde un acceso directo de DB2, se elige la herramienta que se desea ejecutar, se le da click derecho y se elige la opción *Ejecutar como administrador*.

Si la seguridad extendida de DB2 se encuentra habilitada, lo cual es el valor por omisión (ver el *Capítulo 10, Seguridad de la Base de Datos*), hay que asegurarse que el userID es un miembro del grupo DB2ADMNS para que sea posible invocar las herramientas gráficas como por ejemplo el Centro de Control (Control Center).

5.12 Resumen

En este capítulo, hemos explorado una variedad de herramientas utilizadas para administrar, configurar, investigar y gestionar el servidor de datos de DB2.

La llegada de IBM Data Studio en DB2 9.7 como la principal herramienta de administración proporciona una nueva dimensión a la administración de base de datos y al desarrollo.

Se presentaron una serie de herramientas gráficas obsoletas: Centro de Control, Asistencia de SQL, Centro de Tareas y Journal, y el Agente y Monitor de Salud. Sin embargo, el procesador de linea de comando y las herramientas de comandos continuarán siendo parte de la aplicación en versiones posteriores a DB2 9.7. Además, la herramienta de administración de memoria auto adjustable sigue siendo una gran parte dentro del proceso de optimización de la base de datos.

Un componente clave dentro del paquete de herramientas para cualquier administrador de base de datos es el uso de scripts para ejecutar comandos DB2 y funciones. En este

capítulo, se detallaron los scripts tanto de SQL como de sistema operativo (shell scripts). Particularmente, se detalló su composición, cómo se guardan y ejecutan.

Se concluye con una nota de las instrucciones necesarias para que las herramientas de DB2 puedan ser ejecutadas correctamente sobre Windows Vista.

5.13 Ejercicios

Los ejercicios de esta sección permitirán trabajar con scripts en DB2.

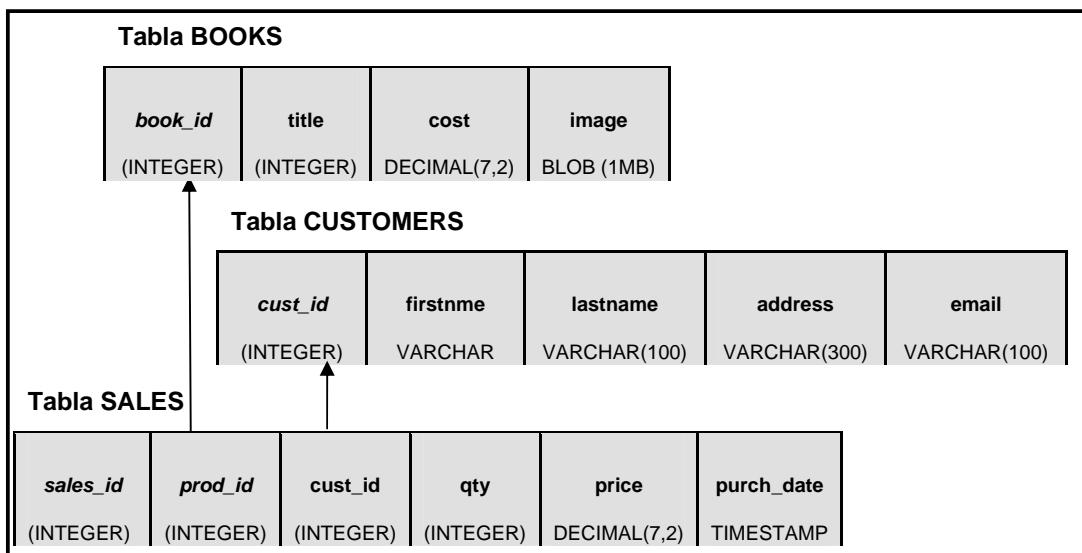
Parte 1: Cargando la base de datos EXPRESS utilizando scripts

En esta parte, se cargarán datos en la base EXPRESS (creada previamente) utilizando el Editor de Comandos y dos scripts adicionales.

Procedimiento

22. Cargar la base de datos EXPRESS con algunas tablas y datos. Para conveniencia, dos scripts, `Lab_Chpt5.db2` y `Lab_Chpt5.dat` han sido creados que realizan esta tarea de carga. El script `Lab_Chpt5.db2` contiene los comandos utilizados para crear las tablas, por tal motivo es que debe ser el primero en ejecutarse. El script `Lab_Chpt5.dat` contiene sentencias que agregan datos a las tablas. Ambos scripts se encuentran en el archivo `expressc_book_exercises_9.7.zip` que viene con este libro. Para ejecutar estos scripts, se invoca el Editor de Comandos. Asegurarse que la nueva base de datos que se creo (EXPRESS) es la que está seleccionada en la lista de la barra de herramientas. En caso que la nueva base de datos no aparezca en la lista, agregar una conexión utilizando el botón *Add*.
23. Seleccionar el menu *Selected → Open* desde el Editor de Comandos y navegar a la carpeta donde los scripts están almacenados. Elegir el archivo `Lab_Chpt5.db2` y seleccionar el botón *OK*. Los contenidos del archivo se deberían mostrar en el área de entrada del Editor de Comandos. Seleccionar el botón *Run* para ejecutar el script. Verificar que no hubo errores ocasionados por la ejecución del script.
24. Repetir el Paso (2) para el archivo `Lab_Chpt5.dat`.

La nueva base de datos que se creo es para la venta de libros por Internet. La tabla **BOOKS** contiene toda la información relacionada a los libros del negocio. La tabla **CUSTOMERS** contiene la información de cada cliente del local. Finalmente, la tabla **SALES** contiene datos de las ventas. Cada vez que un cliente adquiere un libro, se crea un registro en la tabla **SALES**. El diagrama siguiente muestra el diseño y las relaciones entre las mencionadas tablas.



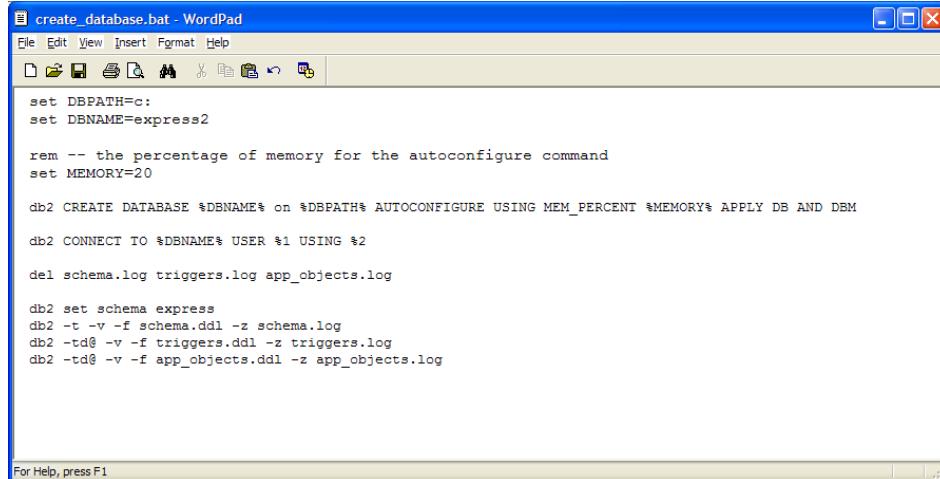
Parte 2: Crear un script de instalación para la base de datos EXPRESS

Los scripts son mecanismos poderosos para realizar tareas repetitivas tales como recolección de estadísticas, resguardos e implementaciones en la base de datos. Los scripts del sistema operativo tienen la ventaja de aceptar parámetros, motivo por el cual los hace muy flexibles. En esta parte, se creará un script de sistema operativo para recrear la base de datos **EXPRESS** como la base de datos **EXPRESS2**. El script invocará los scripts SQL previamente generados para los objetos de la base de datos. Con el objetivo de ahorrar espacio, este ejercicio muestra los scripts y comandos específicos a la plataforma Windows. En caso de trabajar con Linux, asegurarse de hacer los cambios apropiados para las correspondientes instrucciones.

Procedimiento

25. Abrir un editor de texto e ingresar la información que se muestra debajo. La mayoría de la gente se equivoca cuando escriben las líneas siguientes. Las mismas no se proveen en un script separado de forma tal de que aprenda a corregir los errores de tipo usted mismo!

Nota: Especificar el directorio correcto para los archivos `schema.ddl`, `triggers.ddl` y `app_objects.ddl` que están incluídos en el archivo `expressc_book_exercises_9.7.zip` que viene con este libro.



```
create_database.bat - WordPad
File Edit View Insert Format Help
New Open Save Print Find Replace Undo Redo
set DBPATH=c:
set DBNAME=express

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 DBM

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

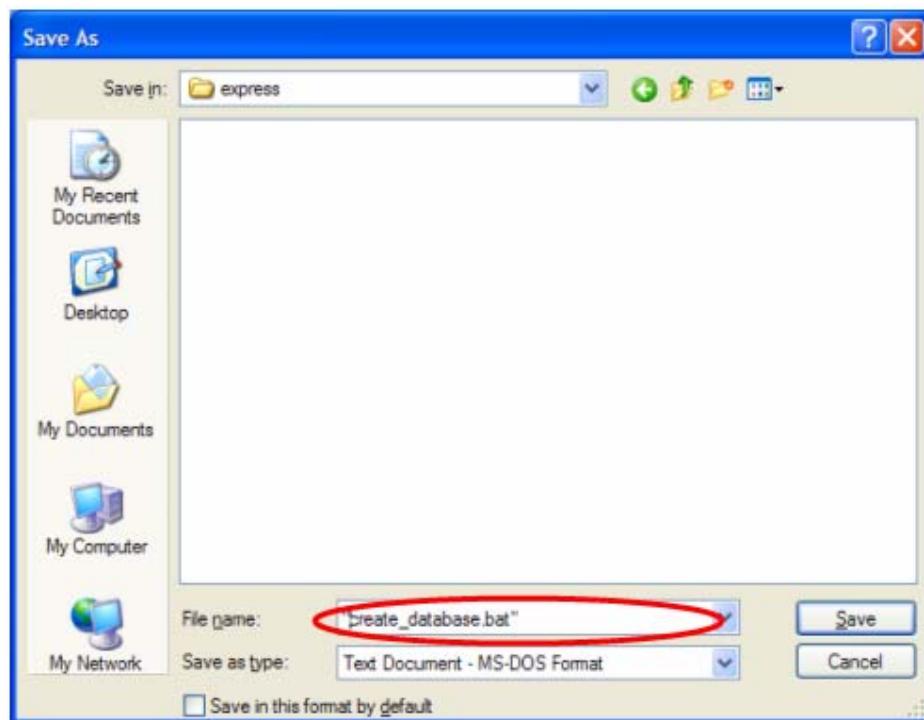
del schema.log triggers.log app_objects.log

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

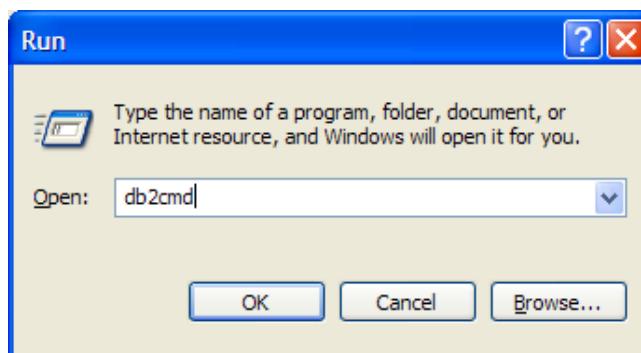
For Help, press F1
```

26. Guardar el script en el directorio C:\express por ejemplo, y nombrarlo create_database.bat.

Nota: En caso de utilizar Wordpad, asegurarse que en la ventana de diálogo de *Guardar Como* (Save As), se seleccione el *Formato MS-DOS*. Si se guarda el archivo con un formato diferente, Wordpad podría incluir caracteres invisibles que provocan que el script se ejecute con problemas y/o errores. Además, hay que colocarle comillas al nombre del archivo, como se muestra en la figura siguiente, para asegurarse que Windows no le agregue la extensión .TXT al mismo.



27. Para invocar los scripts que interactúan con DB2, debe disponer de un ambiente de linea de comandos de DB2. Para abrir una Ventana de Comandos de DB2, ir a *Start -> Program Files -> IBM DB2 -> DB2COPY1 (default) -> Command Line Tools -> Command Window*. Como otra opción, se puede utilizar *Start-> Run*, escribir **db2cmd**, y oprimir *enter* como se muestra en el siguiente cuadro.



28. Luego, para ejecutar el script, ingresar estos comandos en la Ventana de Comandos:

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


29. Tómese un tiempo para familiarizarse con el script recién creado. ¿ Comprende la instrucción de cada linea?

30. Intente responder las siguientes preguntas:

- A. ¿ Dónde se establece la conexión a la base de datos ?
- B. ¿ Qué significan %1 y %2 ?
- C. ¿ Qué indica la siguiente linea de código ? ¿ Dónde se utiliza ? ¿ Para qué ?

SET DBPATH=C:

- D. ¿ Qué acción realiza la siguiente linea de código ?

del schema.log, triggers.log, app_objects.log

- E. ¿ Qué sucede cuando se invoca al script sin parámetros ?

- F. ¿ Por qué los script SQL no tienen la instrucción CONNECT TO ? ¿ Cómo se conectan estos script SQL a la base de datos ?

PARTE II – APRENDIENDO DB2: ADMINISTRACIÓN DE BASES DE DATOS

6

Capítulo 6 – Arquitectura DB2

En este capítulo trataremos brevemente la arquitectura DB2 y podrá ganar conocimientos sobre:

- El modelo de proceso de DB2
- El modelo de memoria de DB2
- El modelo de almacenamiento de DB2

Nota:

Para mayor información sobre la arquitectura DB2, vea este video (en inglés):
<http://www.channeldb2.com/video/video/show?id=807741:Video:4482>

6.1 El modelo de proceso DB2

En la *Figura 6.1* está representado el Modelo de Proceso DB2. Allí los rectángulos representan proceso, mientras que las elipses corresponden a “hilos” de procesamiento (*threads*). El proceso principal de DB2 es db2sysc. Bajo este proceso hay varios hilos de procesamiento, de los cuales el principal también lleva el nombre de db2sysc. Este es el hilo principal que genera otros hilos. Cuando una aplicación remota trata de conectarse al servidor usando la sentencia SQL CONNECT, los oyentes remotos del protocolo de comunicación reciben este requerimiento y se comunican con un agente coordinador de DB2 (db2agent). Estos agentes son como pequeños trabajadores que realizan operaciones en nombre de DB2. Cuando la aplicación es local, o sea que corre en el mismo servidor que DB2, los pasos son bastante similares, salvo que un agente db2ipccm maneja el requerimiento en lugar del hilo db2tcpcm. En algunos casos, como cuando está activado el paralelismo, un db2agent puede generar otros agentes los cuales aparecerán como hilos db2agntp. En la figura se ven otros agentes, tales como db2pfchr, db2loggr, db2dlock los que pueden ser usados con diferentes propósitos. En la *Tabla 6.1* se ven los procesos más comunes, mientras que los hilos más comunes están en la *Tabla 6.2*.

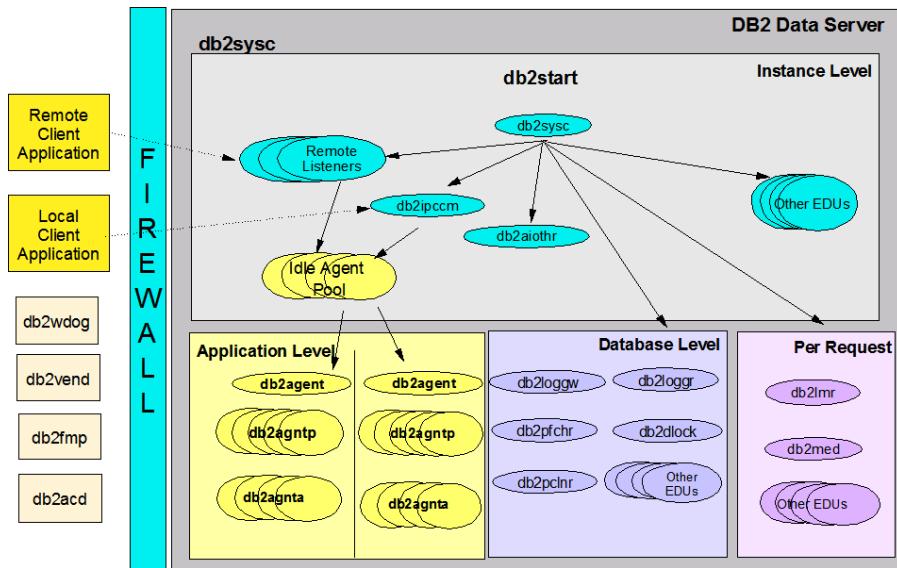


Figura 6.1 – El Modelo de Proceso DB2

Nombre del Proceso	Descripción
db2sysc (Linux) db2syscs (Win)	Este es el controlador principal o motor del sistema DB2. A partir de DB2 9.5, hay un único motor principal multi-hilo para toda la partición. Todas las Unidades que pueden Despacharse del Motor (<i>Engine Dispatchable Units - EDUs</i>) son hilos dentro de este proceso. El servidor de bases de datos no puede funcionar sin este proceso.
db2acd	Es el demonio de computación autonómica y se usa para realizar tareas automáticas del lado del cliente, tales como el monitor de salud, utilitarios automáticos para mantenimiento y el programador de tareas administrativas. Anteriormente este proceso se llamaba db2hmon.
db2wdog	El “perro guardián” (<i>watchdog</i>) de DB2. Está por encima de el proceso principal del motor DB2 y se encarga de liberar recursos en caso que el proceso db2sysc termine anormalmente.
db2vend	Este proceso para ejecutar dentro de un vallado procesos de terceras partes fué introducido en la versión 9.5 de DB2. Todos el código de proveedores de aplicaciones de terceras partes, ejecuta en este proceso fuera del motor. Las aplicaciones de terceras partes son programas no producidos por IBM que pueden interactuar con DB2;

	por ejemplo, el archivado de los archivos de anotación cronológica puede ser manejado por código provisto por una tercera parte especificando un parámetro de rutina de usuario externa que apunte a este código.
db2fmp	Proceso vallado que ejecuta código de usuario en el servidor por fuera del cortafuego tanto para procedimientos almacenados como para funciones definidas por el usuario. Este proceso reemplaza a los procesos db2udf y db2dari que se usaban en versiones anteriores de DB2.

Tabla 6.1 – Procesos DB2 habituales

Nombre del hilo	Descripción
db2sysc	El hilo controlador del sistema. Es responsable del arranque y la detención del sistema y el manejo de la instancia en ejecución.
db2tcpcm	Oyente de comunicaciones TCP/IP
db2agent	Agente coordinador que realiza operaciones de base de datos en nombre de las aplicaciones (hay al menos uno por conexión, dependiendo si el Concentrador de Conexiones está activado).
db2agntp	Es un subagente activo que es generado si INTRA_PARALLEL tiene el valor YES. Este hilo realiza operaciones de base de datos para la aplicación. El agente db2agent coordina el trabajo entre los diferentes subagentes db2agntp.
db2pfchr	Lee datos en forma anticipada realizando operaciones de E/S asincrónicas para DB2 (NUM_IOSERVERS)
db2pclnr	Escribe datos en forma asincrónica para DB2 realizando operaciones de E/S (NUM_IOCLEANERS)

Tabla 6.2 – Hilos DB2 habituales

6.2 Modelo de memoria DB2

El modelo de memoria DB2 consiste de diferentes áreas en memoria a nivel de la instancia, de la base de datos, de la aplicación y de los agentes, tal como se ve en la *Figura 6.2*. No veremos con detalle en este libro las diferentes áreas en memoria, tan solo brindaremos una visión general reducida.

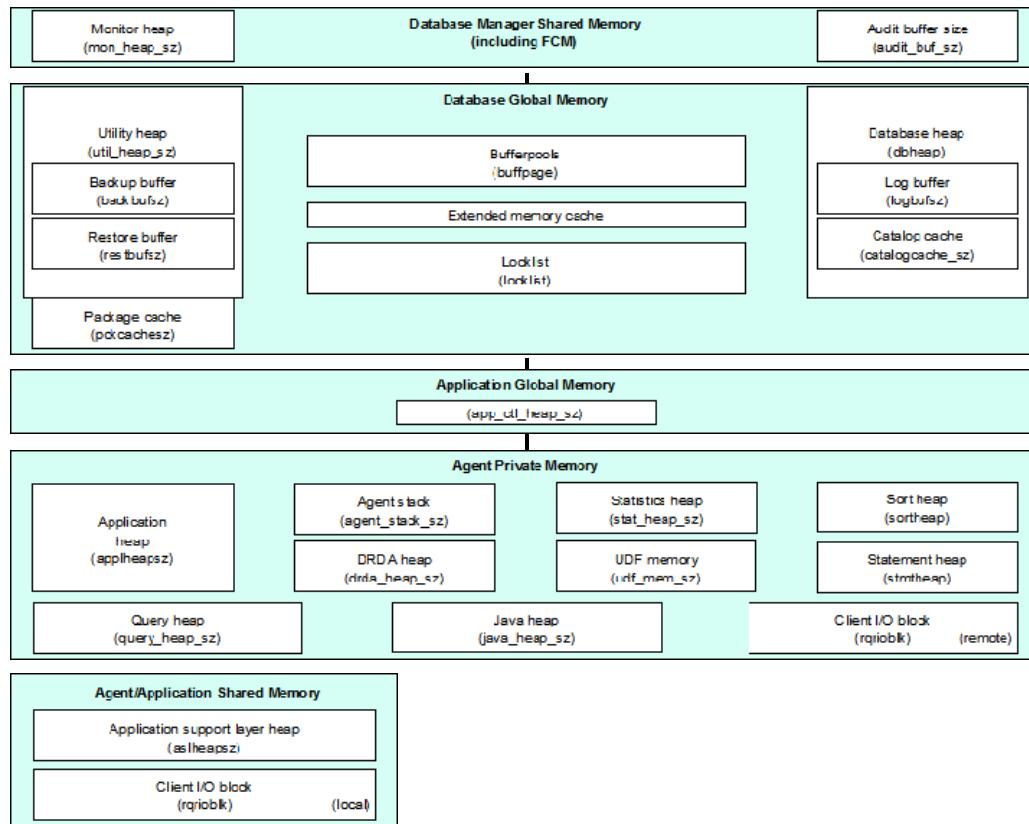


Figura 6.2 – El modelo de memoria DB2

La memoria compartida del manejador de bases de datos es asignada en el momento en que arranca la instancia, pero en general no ocupa mucho espacio. Por otro lado, la Memoria Global de la Base de Datos se asigna cuando ocurre la primera conexión con la base de datos, y en este bloque la agrupación de almacenamientos intermedios es una de las partes más importantes, especialmente en lo que se refiere al rendimiento de las consultas. Es el tamaño de las agrupaciones de almacenamientos intermedios lo que determinará el tamaño total de la Memoria Global de la Base de Datos.

Cada agente DB2 usa un área de memoria denominada Memoria Privada del Agente. Cada conexión requiere de un agente, a menos que se utilice el concentrador de conexiones. En un uso típico, cada agente utiliza entre 3 y 5MB. Con el concentrador de conexiones, un agente maneja varias conexiones, reduciendo de esta manera la necesidad de disponer de más memoria física.

6.3 Modelo de almacenamiento de DB2

En esta sección veremos los siguientes conceptos:

- Páginas y Extensiones
- Agrupación de almacenamientos intermedios
- Espacio de tablas

6.3.1 Páginas y Extensiones

Una página es la unidad mínima de almacenamiento en DB2. Los tamaños permitidos de páginas son: 4K, 8K, 16K y 32K. Una extensión es un grupo de páginas. Dado que trabajar de a una página por vez sería costoso desde el punto de vista del rendimiento, DB2 trabaja con extensiones a la vez. Cuando trabajemos con agrupaciones de almacenamientos intermedios y espacios de tablas deberemos definir el tamaño de la página y de la extensión. Esto lo veremos en las siguientes secciones.

6.3.2 Agrupaciones de almacenamientos intermedios

Una agrupación de almacenamientos intermedios es una sección de memoria real que se usa para contener datos de tablas e índices. Las agrupaciones de almacenamientos intermedios mejoran el rendimiento reduciendo la necesidad de ejecutar operaciones de E/S secuenciales directas y promoviendo lecturas (captación previa) y grabaciones asincrónicas. O sea que DB2 anticipa que páginas serán necesarias y las lee por anticipado desde el disco a la agrupación de almacenamientos intermedios de manera que estén listas para ser usadas.

Las agrupaciones de almacenamientos intermedios se crean en memoria en unidades de páginas de 4K, 8K, 16K, y 32K. Por cada base de datos debe haber al menos una agrupación de almacenamientos intermedios y una agrupación de almacenamientos intermedios del mismo tamaño de página de aquellos espacios de tablas definidos en la base de datos.

6.3.2.1 Creación de una Agrupación de almacenamientos intermedios

Para crear una agrupación de almacenamientos intermedios se utiliza la sentencia **CREATE BUFFERPOOL**. Otra forma de hacerlo es mediante el Centro de Control, dando clic con el botón derecho del ratón sobre la carpeta de las Agrupaciones de almacenamientos intermedios dentro de una base de datos determinada y seleccionando *Crear*, tal como se ve en la *Figura 6.3*.

122 Conociendo al DB2 Express-C

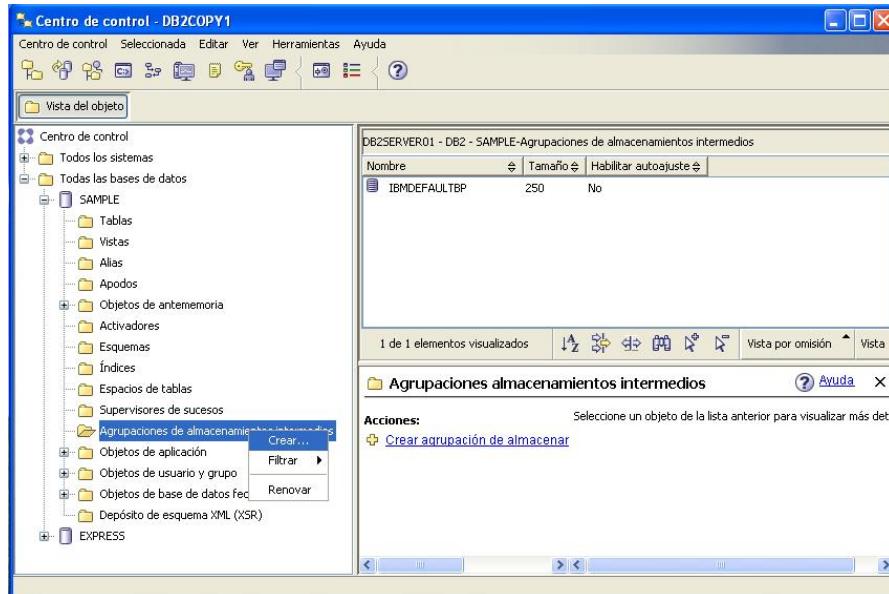


Figura 6.3 – Creación de una agrupación de almacenamientos intermedios

Después de dar clic en *Crear*, se abre la ventana de *Crear agrupación de almacenamientos intermedios*, tal como se ve en la Figura 6.4

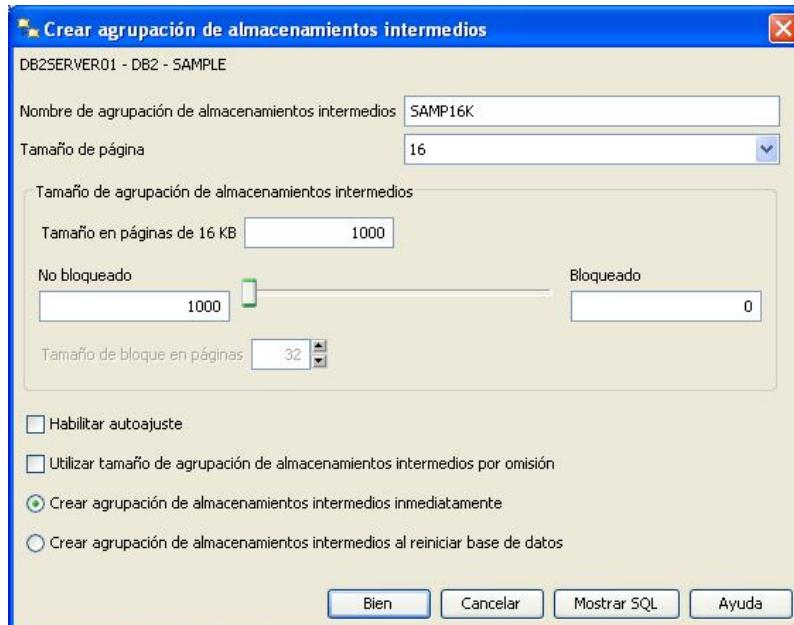


Figura 6.4 – Ventana de *Crear agrupación de almacenamientos intermedios*

La mayoría de los campos en la *Figura 6.4* se explican por sí mismos. Los campos *No bloqueado* y *Bloqueado* se refieren al número de páginas que deben existir, tanto no bloqueadas como bloqueadas. Las agrupaciones de almacenamientos intermedios bloqueadas aseguran que cuando un grupo de páginas contiguas en el disco son movidas a una agrupación de almacenamientos intermedios son almacenadas también en forma contigua en el área bloqueada; lo que puede mejorar el rendimiento. El número de páginas bloqueadas no debe ser mayor que el 98 por ciento del número total de páginas en la agrupación de almacenamientos intermedios. Si se especifica un valor de cero para el área bloqueada, entonces no se harán operaciones de E/S en bloque.

Una vez que se ha creado la agrupación de almacenamientos intermedios, ésta aparece en el Centro de control, tal como se ve en la *Figura 6.5*.

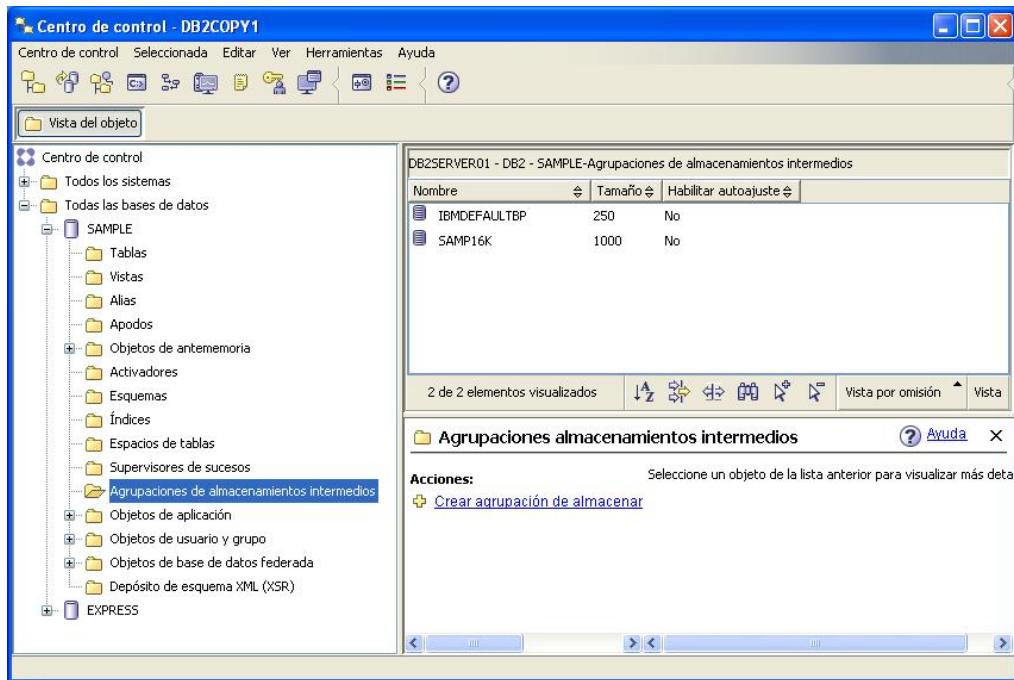


Figura 6.5 – El Centro de control después de la creación de la agrupación de almacenamientos intermedios SAMP16K

6.3.3 Espacios de tablas

Los Espacios de tablas son una interfase lógica entre las tablas lógicas y la memoria física del sistema (agrupación de almacenamientos intermedios), y los contendores (discos). Para crear un espacio de tablas se usa la sentencia **CREATE TABLESPACE** y en ella se puede especificar:

- El tamaño de página para el espacio de tabla (4KB, 8KB, 16KB, o 32KB). Este tamaño debe corresponderse con una agrupación de almacenamientos intermedios que tenga el mismo tamaño de página.
- El nombre de la agrupación de almacenamientos intermedios asociada con este espacio de tablas.
- El tamaño de la extensión
- El tamaño de captación previa

6.3.3.1 Tipos de Espacios de tablas

Existen tres tipos de espacios de tablas:

- Normal

Son para contener tablas del usuario. Por ejemplo el espacio de tablas USERSPACE1 creado por omisión es de tipo normal.

- Grande

Se usan para separar datos LOB dentro de su propio espacio de tablas. También son usados para almacenar datos XML en aquellas bases de datos creadas con soporte XML puro que utilizan el tipo de datos XML para las columnas. Estos son los espacios de tablas por omisión.

- Temporales

Hay dos tipos de espacios de tablas temporales:

- Sistema temporal

DB2 los utiliza para operaciones internas tales como ordenamiento de datos. Por ejemplo, el espacio de tablas TEMPSPACE1 creado por omisión cuando se crea una base de datos, es un espacio de tablas de sistema temporal. Siempre debe existir al menos un espacio de tablas de este tipo.

- Usuario temporal

En estos espacios de tabla es donde se crean las Tablas Globales Temporales Declaradas (*Declared Global Temporary Tables - DGTTs*) y las Tablas Globales Temporales Creadas (*Created Global Temporary Tables - CGTTs*) que son tablas en memoria temporales. A menudo se confunden con los espacios de tablas de sistema temporal. Antes que un usuario pueda crear DGTTs o CGTTs debe crear un espacio de tablas de este tipo.

6.3.3.2 Gestión de espacios de tablas

Los espacios de tablas pueden ser clasificados en base a como se gestionan, lo que se especifica en la sentencia **CREATE TABLESPACE**.

Gestionados por el sistema

Este tipo de espacio de tablas se denomina Almacenamiento Gestionado por el Sistema (*System Managed Storage - SMS*), lo que significa que el sistema operativo es quien gestiona el almacenamiento. Estos espacios son fáciles de gestionar y los contenedores son directorios de archivo del sistema operativo. El espacio no está preasignado, pero los archivos crecen dinámicamente. Una vez que los contenedores son definidos, quedan fijos al momento de ser creados y no hay modo de agregar otros contenedores posteriormente, a menos que se use una recuperación redirigida. Cuando se usan espacios de tabla SMS los datos de las tablas, índices y LOB no pueden ser distribuidos entre varios espacios de tablas diferentes.

Gestionados por la base de datos

A este tipo de espacio de tablas se le conoce como Almacenamiento Gestionado por la Base de Datos (*Database Managed Storage - DMS*), o sea que DB2 gestiona el almacenamiento. Esto requiere mayor intervención del Administrador de Bases de Datos en la gestión del espacio. Los contenedores pueden ser archivos preasignados o bien discos en bruto (*raw devices*). En estos discos los datos son grabados directamente sin almacenamiento intermedio provisto por el sistema operativo.

Los contenedores pueden agregarse, darse de baja o bien su tamaño ser cambiado mediante la sentencia ALTER TABLESPACE. Los espacios de tablas DMS son óptimos por su rendimiento y los datos de tablas, índices y LOB pueden ser distribuidos entre varios espacios de tablas separados, lo que mejora el rendimiento.

Gestionados por DB2 con almacenamiento automático

Este tipo de espacio de tablas es gestionado con almacenamiento automático y combina la facilidad de uso de los espacios de tabla SMS con el rendimiento óptimo y flexibilidad de los espacios de tabla DMS. Por lo tanto, a partir de DB2 9, éste es el tipo de espacio de tablas por omisión. En estos espacios de tablas, primero se especifica un directorio de almacenamiento y un grupo lógico de dispositivos de almacenamiento los cuales DB2 utilizará para gestionar el espacio. No se dan definiciones explícitas de los contenedores, éstos son creados automáticamente a través de los directorios de almacenamiento. Tanto el crecimiento de los contenedores existentes como así también el agregado de otros nuevos son gestionados por completo por DB2. Si en el comando **CREATE DATABASE** no se especifica un directorio de almacenamiento, el directorio de la base de datos es utilizado como directorio de almacenamiento. El directorio de la base de datos es aquel donde reside la definición principal de ésta. Si no se especifica este directorio, se obtiene del parámetro del manejador de la base de datos DFTDBPATH. En Windows, éste puede ser únicamente un disco, no un directorio.

Para poder utilizar almacenamiento automático es necesario primero crear una base de datos con la opción de almacenamiento automático activada (que por otro lado es la opción por defecto) y asociarle un conjunto de directorios de almacenamiento. Si es necesario redefinir estos directorios de almacenamiento después de haber creado la base de datos, esto puede hacerse con una operación de RESTORE de la base. Una vez

creada la base de datos con esta característica, se pueden crear en ella espacios de tabla que usen almacenamiento automático (que es también la opción por omisión).

Los espacios de tablas gestionados por almacenamiento automático son muy parecidos a los DMS, pero las operaciones están automatizadas y son gestionadas por DB2, incluyendo la asignación, definición y cambio automático de tamaño de los contenedores.

Veamos ahora un ejemplo de un espacio de tablas gestionado por almacenamiento automático. En primer lugar crearemos la base de datos con almacenamiento automático activado:

Recordemos que el almacenamiento automático se activa por omisión:

```
CREATE DATABASE DB1
```

O bien puede ser especificado explícitamente:

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

Si bien el almacenamiento automático se activa por omisión, los directorios de almacenamiento se deben indicar en el momento de la creación, y si alguno de estos directorios es un subdirectorio dentro de un disco, debe ser creado previamente en aquel disco:

Por ejemplo en Windows:

```
CREATE DATABASE DB1 ON C:/, C:/storagepath1, D:/storagepath2
```

Se debe tomar nota que el primer ítem en la lista es un disco, dado que en Windows, el directorio de una base de datos debe ser el directorio raíz de un disco y no un subdirectorio; y este directorio raíz será también usado como uno de los directorios de almacenamiento. Por lo tanto en este ejemplo el directorio de la base de datos es C: y los directorios de almacenamiento son C:, C:\storagepath1 y D:\storagepath2, siendo necesario crear previamente estos dos últimos.

Y por ejemplo en Linux:

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

Si se desea crear la base de datos sin almacenamiento automático, esto debe indicarse explícitamente:

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

Continuando con los ejemplos, el paso siguiente es crear el espacio de tablas con el almacenamiento automático activado:

Recordemos que si no lo especificamos, el almacenamiento automático se activa por omisión:

```
CREATE TEMPORARY TABLESPACE TEMPTS
```

Pero también lo podemos especificar explícitamente:

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

En este ejemplo, en que el almacenamiento automático se ha asignado por omisión, se ha especificado el tamaño inicial, las porciones de incremento y el valor máximo que puede alcanzar.

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

6.3.3.3 De que forma los datos son almacenados en los espacios de tablas

La forma por omisión en que DB2 almacena los datos en disco es grabando extensiones en forma simultánea y distribuida entre los contenedores. Por ejemplo, supongamos que tenemos un espacio de tablas DMS con páginas de 4K, un tamaño de extensión de 8 en tres contenedores en bruto. Esto significa que 32K de datos (4K x 8 páginas por extensión = 32K) se grabarán en un disco antes de continuar con el siguiente, tal como se ve en la Figura 6.6. Nótese que las tablas no comparten extensiones.

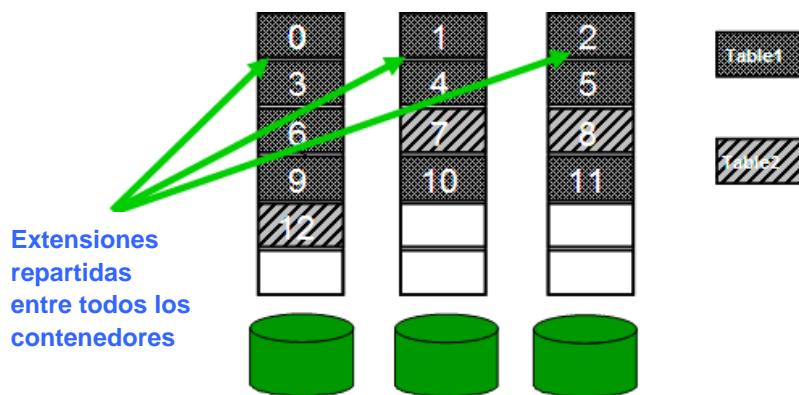


Figura 6.6 – Grabación de datos en los espacios de tablas

6.3.3.4 Creación de un espacio de tablas usando el Centro de control

Para crear un espacio de tablas usando el Centro de control, dé clic con el botón derecho del ratón sobre la carpeta de *Espacios de tablas* dentro de una base de datos dada y seleccione *Crear*, tal como se ve en la Figura 6.7. Esto hará que se abra el *Asistente para crear espacio de tablas* como se muestra en la Figura 6.8.

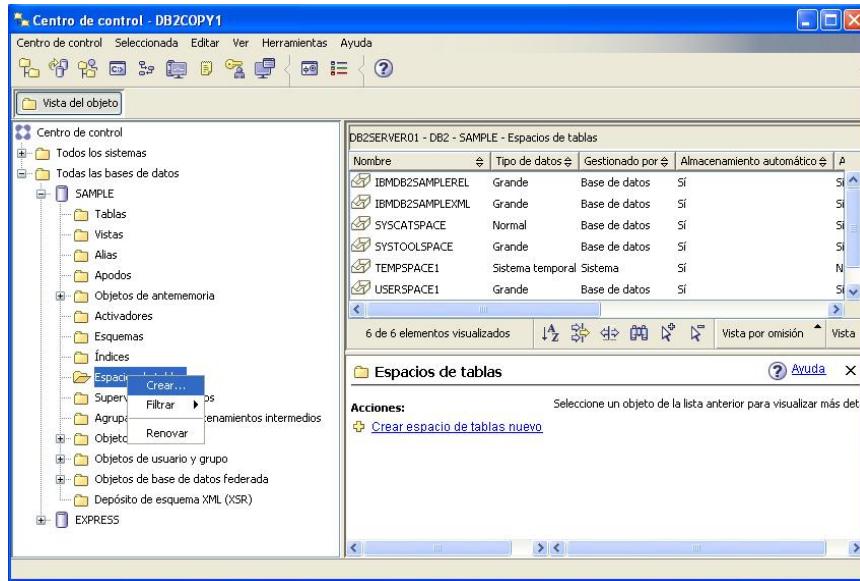


Figura 6.7 – Creación de un Espacio de tablas usando el Centro de control



Figura 6.8 – Asistente para crear espacio de tablas

El asistente de la Figura 6.8 le guiará a través de los pasos necesarios para crear un espacio de tablas.

6.4 Resumen

En este capítulo hemos visto los tres principales aspectos de la arquitectura DB2: el modelo de proceso, el modelo de memoria y el modelo de almacenamiento. Dentro del modelo de procesos se ha hablado de los procesos e hilos más comunes, incluyendo db2sysc, si el cual DB2 no puede ejecutarse.

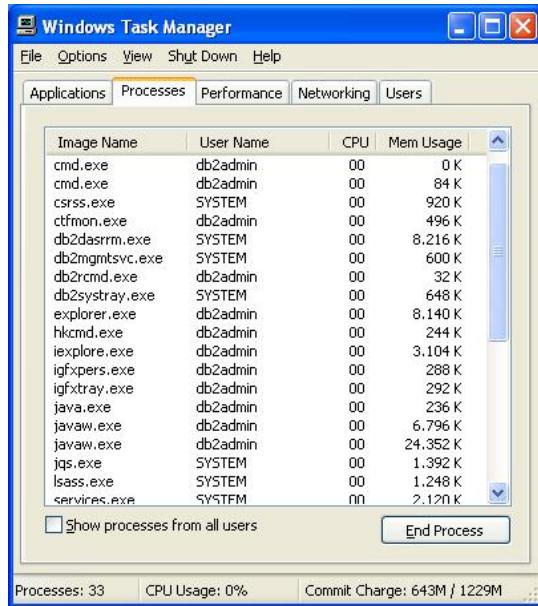
Se ha tratado el modelo de almacenamiento en detalle, cubriendo sus tres aspectos principales: páginas y extensiones, la agrupación de almacenamientos intermedios (incluyendo detalle de cómo crearla) y los espacios de tablas. Por último hemos visto los distintos tipos de espacios de tablas, tratando en detalle como son gestionados (SMS, DMS, Automáticos) y como crear un nuevo espacio de tablas usando el Centro de control.

6.5 Ejercicios

Este ejercicio le ayudará a comprender los modelos DB2 en Windows: proceso, memoria y almacenamiento. Podrá revisar algunos procesos e hilos, monitorear el uso de memoria y practicar creando una base de datos que usa almacenamiento automático y subdirectorios de almacenamiento en Windows. Lo ideal es que los directorios de almacenamiento estén en diferentes discos, pero dado que su computador puede no estar configurado con varios discos, este ejercicio solo utilizará el disco C:\.

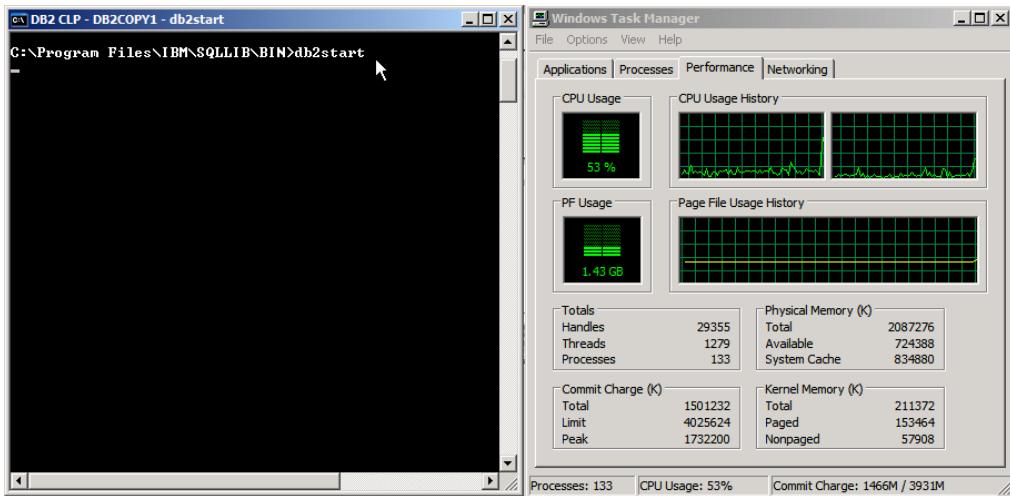
Procedimiento:

1. Veamos algunos procesos en Windows. En primer lugar, abra la Ventana de mandatos DB2 (Arranque -> ejecutar -> db2cmd) y asegúrese que su instancia se detenga usando este comando: **db2stop force**
2. Abra el Administrador de tareas de Windows Task Manager, seleccione la pestaña de *Procesos*, de clic en la columna de *Nombre de la imagen* para ordenar la lista por esa columna y busque el proceso db2sysc.exe tal como se ve en la siguiente figura.



No debería ser capaz de encontrar el proceso db2syscs.exe precisamente porque le hemos pedido en el primer paso que detuviera la instancia DB2.

3. Arranque la instancia DB2 con este comando: **db2start**, y repita el paso anterior. ¿Ha podido encontrar ahora el proceso db2syscs.exe?
4. Veamos ahora el consumo de CPU y memoria. Siga estos pasos:
 - A. Asegúrese que nada esté ejecutando en el sistema. Cierre todas las otras aplicaciones.
 - B. Abra una nueva Ventana de mandatos DB2 y ejecute: **db2stop force**
 - C. En el Administrador de tareas, seleccione la pestaña de Rendimiento.
 - D. Mantenga abiertas y lado a lado las ventanas del Administrador de tareas y de Mandatos de DB2, tal como se ve más abajo. Tome nota de la cantidad de Memoria física disponible y la utilización de la CPU.
 - E. Ejecute **db2start** y al mismo tiempo monitoree el uso de memoria y CPU tan pronto como el comando se haya ejecutado. Debería ver un salto limitado en el uso de CPU y la memoria disponible debería reducirse en unos 50MB a 70MB. Esto es lo que la instancia DB2 consume. Si ahora ejecuta nuevamente **db2stop force**, la memoria debería volver a su valor anterior.



- Repita el paso anterior, pero esta vez, vea que sucede después de conectarse con la base de datos SAMPLE. Emite estos comandos desde la Ventana de mandatos DB2:

```
db2start
db2 connect to sample
```

- Se verá una reducción de la memoria física disponible tan pronto como se conecte a la base de datos SAMPLE. Esto ocurre pues al momento de la conexión, se define la memoria global de la base de datos (Agrupación de almacenamientos intermedios, almacenamiento intermedio del catálogo, etc.).
- Repita el paso anterior, pero esta vez monitoree que sucede después de crear una agrupación de almacenamiento intermedio de cualquier tamaño. Asegúrese de no superar la memoria física instalada en su computador, pues si lo hace DB2 no asignará la agrupación de almacenamiento intermedio inmediatamente, sino que postergará esta acción hasta que la base de datos sea desactivada. Mientras tanto DB2 utilizará una pequeña agrupación de almacenamiento intermedio del sistema y continuará usándola hasta que haya suficiente memoria disponible. Por ejemplo, para crear una agrupación de almacenamiento intermedio de unos 160MB ejecute este comando mientras está conectado a la base de datos SAMPLE:

```
db2 create bufferpool mybp immediate size 5000 pagesize 32k
```

- Cree la base de datos *mydb1* de modo que use almacenamiento automático y asigne como directorio de la base al disco C:, y como directorios de almacenamiento C:, C:\mystorage1, C:\mystorage2. Ejecute el siguiente comando desde la Ventana de mandatos DB2:

```
db2 create database mydb1 on C:\, C:\mystorage1, C:\mystorage2
```

Lo más probable es que reciba un mensaje de error al ejecutar el comando dado que primero debería haber creado los subdirectorios C:\mystorage1, C:\mystorage2. ¡Créelos y trate nuevamente!

7

Capítulo 7 – Conectividad de los clientes DB2

En este capítulo veremos la preparación requerida para poder conectar un cliente DB2 con un servidor DB2 usando TCP/IP. Tenga en cuenta que los servidores DB2 vienen con un componente cliente incluido, lo que les permite comportarse como clientes y conectarse a otro servidor DB2. Hay varias formas de preparar la conectividad de un cliente DB2, sin embargo en este capítulo solo trataremos el método más sencillo, o sea usando el Asistente de configuración.

Nota:

Para mayor información acerca de la Conectividad del cliente DB2, vea este video (en inglés): <http://www.channeldb2.com/video/video/show?id=807741:Video:4222>

A partir de DB2 9.7, el Asistente de Configuración está ya obsoleto, lo cual no quita que pueda seguir siendo usado y además sea incluido con el producto.

7.1 Directorios DB2

Los directorios DB2 son archivos binaries que almacenan información acerca de a cuales bases de datos puede usted conectarse desde su computador. Hay cuatro directorios:

- Directorio de Bases de Datos del Sistema
- Directorio de bases de datos locales.
- Directorio de nodos
- Directorio DCS

Con la herramienta gráfica *Asistente de configuración* se puede revisar y actualizar el contenido de todos estos directorios.

7.1.1 Directorio de bases de datos del sistema

Este directorio es como la tabla de contenidos de un libro, en él están descritas todas las bases de datos a las que puede conectarse, ya sean éstas locales o remotas. En el caso de una base de datos local, tiene un puntero hacia el *Directorio de bases de datos locales*. Cuando la base de datos es remota, el puntero indica hacia una entrada en el *Directorio*

de nodos. Para listar el contenido del Directorio de bases de datos del sistema, se utiliza este comando:

```
list db directory
```

7.1.2 Directorio de bases de datos locales

Este directorio contiene información acerca de las bases de datos a las cuales puede conectarse y que residen en su computador. Para listar el contenido de este directorio se ejecuta este comando:

```
list db directory on <drive/path>
```

7.1.3 Directorio de nodos

Este directorio incluye la información necesaria para conectarse a una base de datos remota determinada. Por ejemplo en caso de usar protocolo TCP/IP, el nodo correspondiente incluiría la dirección IP del servidor donde reside la base de datos DB2 a la que se pretende conectar como asimismo el Puerto de la instancia correspondiente a esa base de datos. En este caso el comando utilizado para listar el contenido de este directorio es:

```
list node directory
```

7.1.4 Directorio DCS

Este directorio solo existe si se ha instalado DB2 Connect para conectarse a DB2 for z/OS (*mainframe*), o DB2 for i5/OS. Liste el contenido del directorio DCS con este comando:

```
list dcs directory
```

7.2 Asistente de configuración (obsoleto)

Mediante esta herramienta gráfica se puede fácilmente configurar la conectividad entre un cliente y un servidor DB2. Para ejecutar el Asistente de configuración en Windows, se puede seleccionar: *Arranque -> Programas -> IBM DB2 -> DB2COPY1 -> Herramientas de configuración -> Asistente de configuración*

Desde la Línea de mandatos la herramienta puede arrancarse usando el comando `db2ca`. La *Figura 7.1* deja ver el Asistente de configuración.

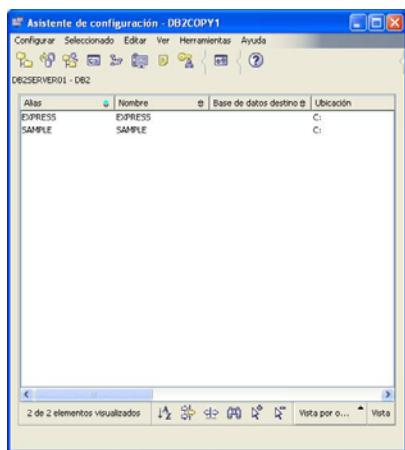


Figura 7.1 – El Asistente de configuración

7.2.1 Requerimientos de preparación en el servidor

Dos cosas precisan ser configuradas en el servidor:

1) DB2COMM

Esta variable del registro de DB2 es la que determina cuales oyentes de los protocolos de comunicación estarán activos esperando por requerimientos de conexión por parte de los clientes. El protocolo de comunicación más típico y que es usado más frecuentemente es TCP/IP. Cuando se cambia el valor de este parámetro, es necesario reciclar la instancia para que el cambio sea efectivo. Para desplegar y cambiar el valor de DB2COMM en el Asistente de configuración, elija *Configurar -> Registro DB2* tal como se ve en la Figura 7.2 y en la Figura 7.3.

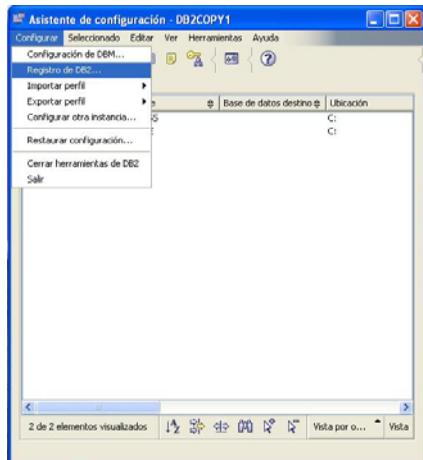


Figura 7.2 – Acceso al Registro DB2

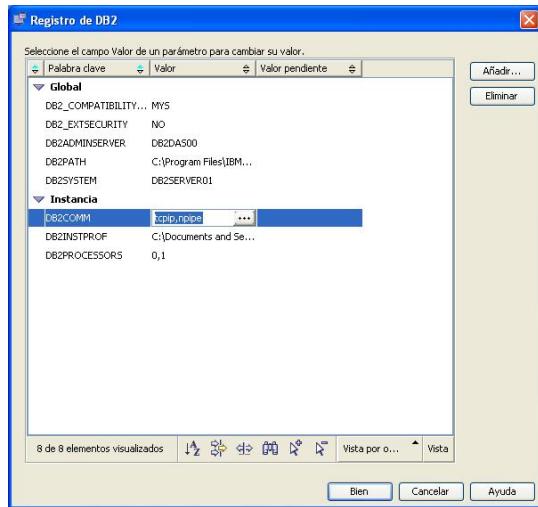


Figura 7.3 – Verificando el valor de la variable DB2COMM en el Registro DB2

2) SVCENAME

En este parámetro de configuración del manejador de la base de datos se debe colocar el valor del nombre del servicio tal como aparece en el archivo de servicios de TCP/IP, o bien el número del puerto a usar para tener acceso a las bases de datos en esta instancia. En la ventana del Asistente de configuración, seleccione *Configurar -> Configuración de DBM* como se ve en la *Figura 7.4*

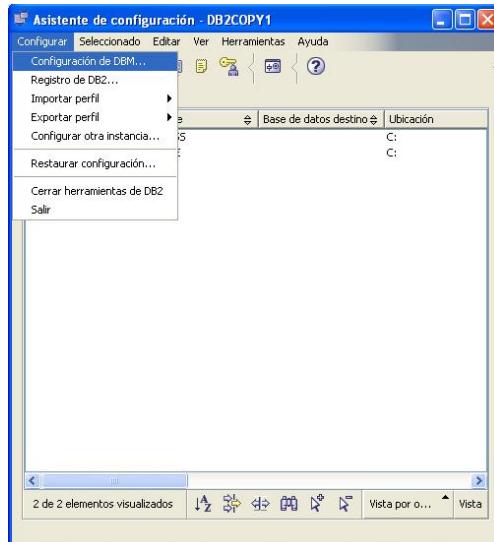


Figura 7.4 – Revisión de dbm cfg en el Asistente de configuración

Una vez que la ventana de Configuración de DBM se abra, vaya a la sección de Comunicaciones y busque SVCENAME. El valor puede entonces cambiarse por el nombre del servicio o el número de puerto según sea necesario, tal como se ve en la *Figura 7.5*.

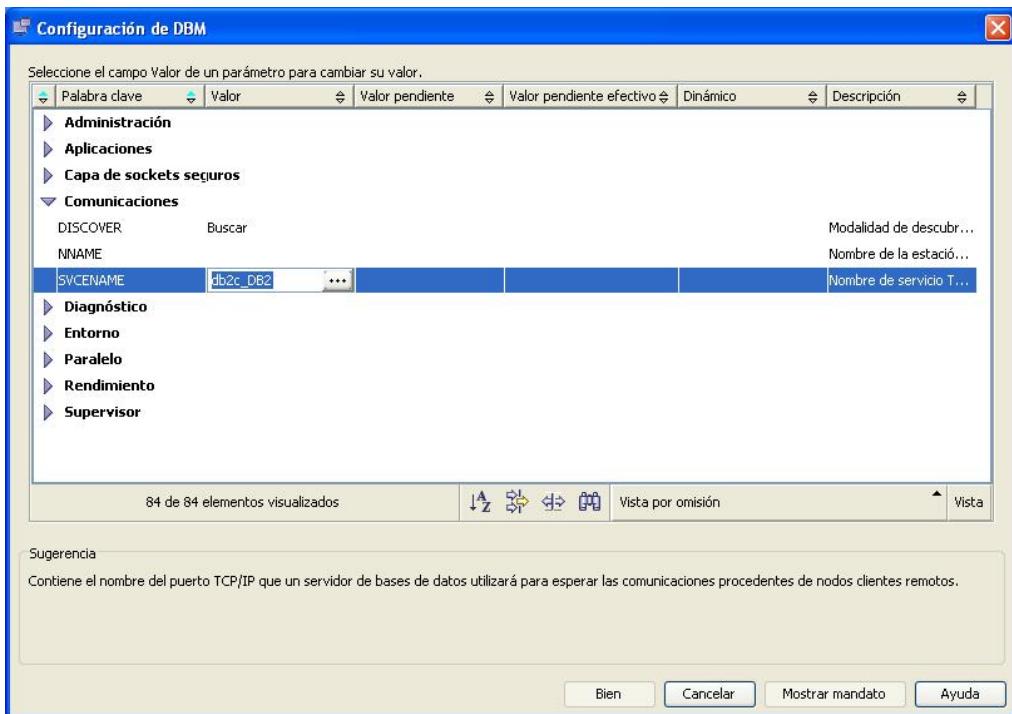


Figura 7.5 – Revisión del parámetro SVCENAME en el dbm cfg

7.2.2 Configuración requerida en el cliente

Del lado del cliente, se precisa conocer la siguiente información de antemano:

- El nombre de la base de datos con la cual se desea conectar
- El número de puerto de la instancia DB2 en el servidor donde se encuentra la base de datos. También puede usarse el nombre del servicio, siempre cuando exista la correspondiente entrada en el archivo de servicios de TCP/IP
- El identificador de usuario y la clave previamente definidos en el sistema operativo del servidor que serán usados para conectarse a la base de datos.

Toda esta información puede introducirse en el cliente DB2 usando el Asistente de configuración, para ello ejecute el Asistente para añadir bases de datos dando clic en *Seleccionado -> Añadir base de datos utilizando un asistente*, como se muestra en la *Figura 7.6*.

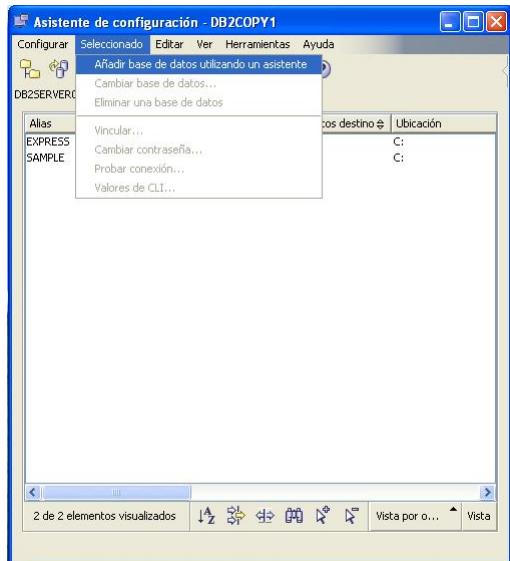


Figura 7.6 – Invocando al Asistente para añadir bases de datos

Otra forma de llamar a este asistente es dando clic con el botón derecho del ratón en el espacio en blanco al medio del Asistente de configuración y seleccionando *Añadir base de datos utilizando un asistente*. En la Figura 7.7 se ve al Asistente para añadir base de datos.

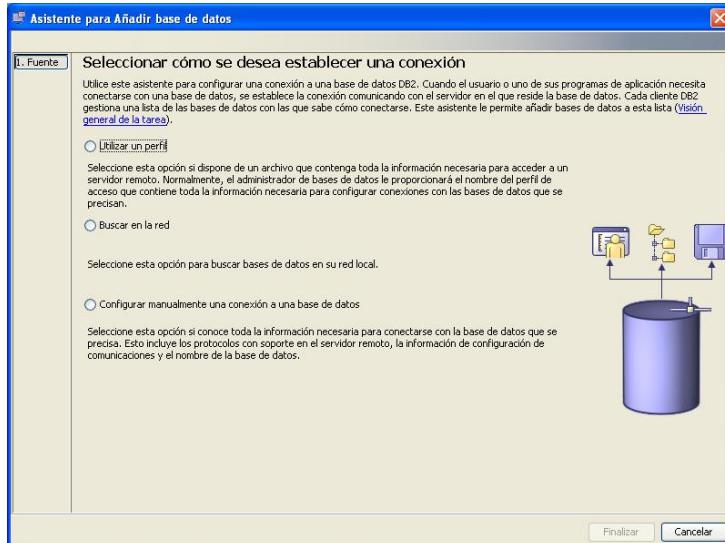


Figura 7.7 – Asistente para añadir base de datos

En el Asistente para añadir base de datos hay tres opciones:

1. Utilizar un perfil

Puede haber situaciones en las cuales sea necesario configurar varios clientes para conectarse al mismo servidor DB2. En esos casos, es conveniente realizar todas las tareas de configuración en uno de los clientes y guardar esa configuración en un archivo de “perfil”, luego utilizando este archivo se puede cargar esta información directamente a otros clientes. Por ejemplo en la *Figura 7.7*, si se elige *Utilizar un perfil*, el asistente pedirá luego el nombre del archivo de perfil desde donde se cargará la información. Más adelante en este capítulo veremos como crear perfiles de servidor y de cliente.

2. Buscar en la red

Este método, también conocido como Descubrimiento, le dice a DB2 que busque en la red un servidor, instancia y base de datos determinados. Para que este método funcione, el DAS debe estar activo en cada uno de los servidores donde residan bases de datos que deben ser descubiertas. Con este método hay dos formas posibles de realizar la búsqueda:

- Búsqueda en la red:

Aquí se busca en la red completa. No es muy recomendado si la red es muy grande y tiene muchos nodos, ya que tomará mucho tiempo recolectar la información de todos y cada uno de los sistemas

- Sistemas conocidos:

Buscar en la red por un servidor conocido en la dirección provista.

Ambos métodos se aprecian en la *Figura 7.8*

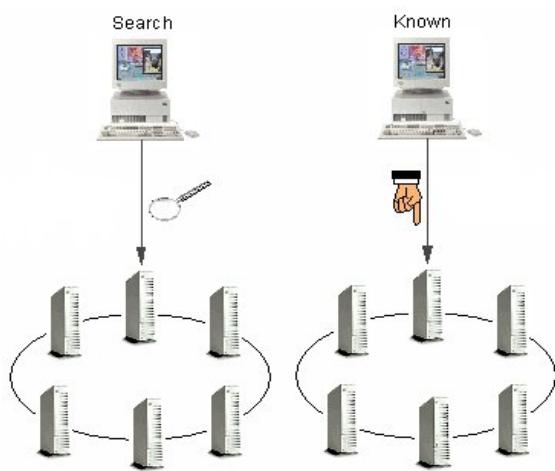


Figura 7.8 – Los métodos de búsqueda en red y sistemas conocidos (Descubrimiento)

Puede haber circunstancias en las cuales un administrador no desea que los clientes busquen en la red bases de datos que contienen información confidencial. Esto puede evitarse al nivel del DAS, de la instancia o de la base de datos. En la *Figura 7.9* se ve este tema más detalladamente.

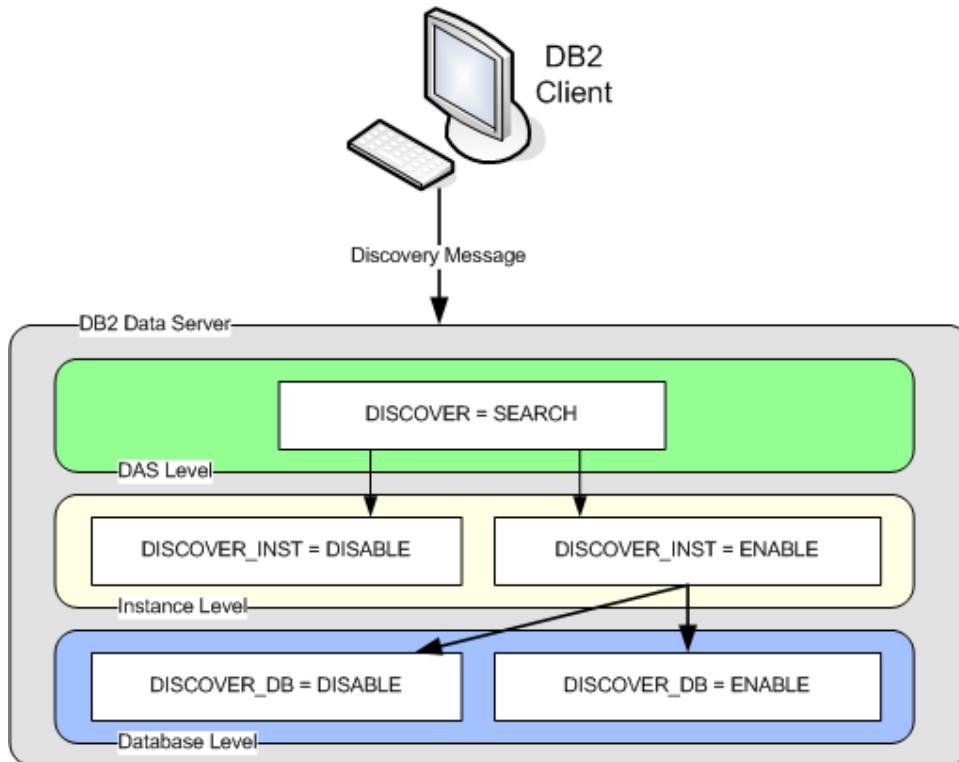


Figura 7.9 – Configuración de los parámetros para permitir el descubrimiento

En la *Figura 7.9* se ven los diferentes niveles en los cuales se puede permitir o bloquear el descubrimiento. Al nivel del DAS, se puede dar al parámetro DISCOVER el valor SEARCH si es que se desea permitir que el servidor responda a búsquedas en la red o KNOWN si solo debe responder a búsquedas dirigidas específicamente a su dirección (servidor conocido). Al nivel de la instancia, al parámetro DISCOVER_INST del manejador de base de datos se le puede dar el valor de DISABLE, si es que se quiere que la instancia no responda a las búsquedas o ENABLE, en caso contrario. Por ultimo, al nivel de la base de datos, al parámetro DISCOVER_DB puede dársele el valor de ENABLE, si es que la base de datos debe responder a las búsquedas o DISABLE si no debe hacerlo. Con estos tres parámetros se puede configurar el descubrimiento con la granularidad requerida.

3. Configurar manualmente una conexión a una base de datos

Si se usa este método, toda la información, como ser nombre del servidor, números de puerto y datos de la base, deben ser ingresados manualmente al Asistente de

configuración, quien generará comandos de catálogo para ejecutar la configuración de la conectividad. El Asistente de configuración no valida que esta información sea correcta. Ya se dará usted cuenta si no es válida cuando no pueda conectarse al servidor. Asimismo asegúrese que la identificación de usuario y la clave sean las correctas para conectarse a la base de datos remota. Por omisión, la autenticación tiene lugar en el servidor DB2 al que se está intentando conectar, por lo tanto se deben proveer una identificación de usuario y una clave definidas previamente en ese servidor.

7.2.3 Creación de perfiles de cliente y servidor

Si se está configurando un número elevado de servidores o clientes, en lugar de configurar uno a uno individualmente, se puede preparar uno y luego exportar un perfil del mismo, y posteriormente aplicar ese perfil a los otros clientes o servidores. Esto ahorra un montón de tiempo de administración cuando se está montando el ambiente.

Para crear un perfil personalizado desde el Asistente de configuración, se selecciona el menú de *Configuración* y después *Exportar Perfil -> Personalizar*, tal como se muestra en la *Figura 7.10*

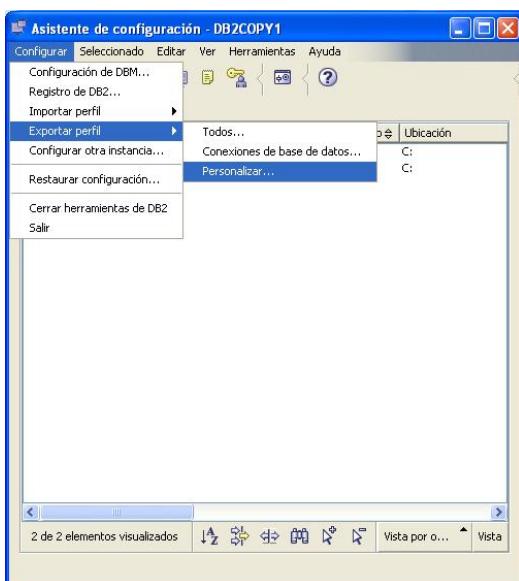


Figura 7.10 – Exportando un perfil

La *Figura 7.11* muestra los campos a completarse para exportar un perfil

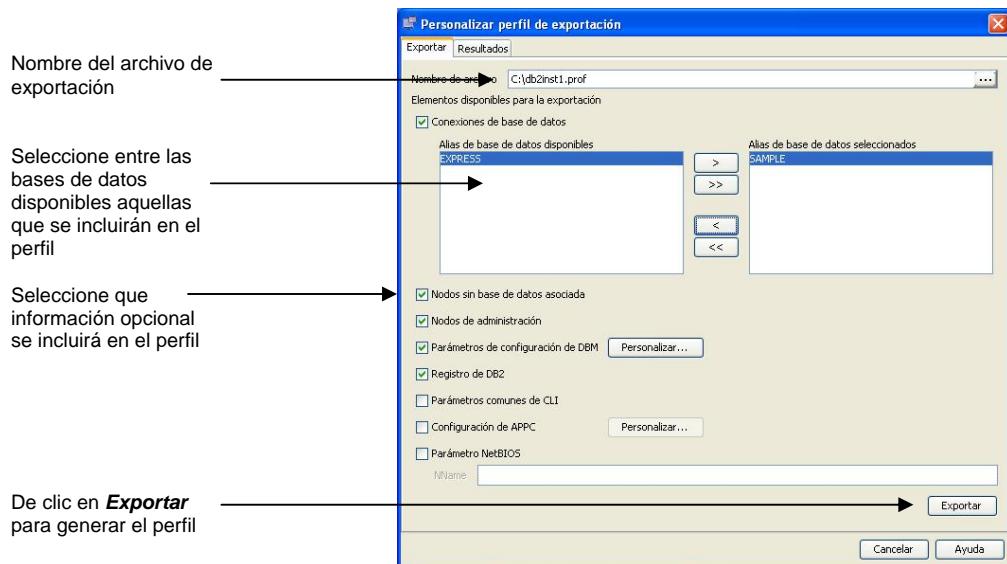


Figura 7.11 – Diálogo de personalización para exportar un perfil

La Figura 7.12 muestra el resultado luego de dar clic en “Exportar” en el diálogo de Personalización para exportar un perfil.

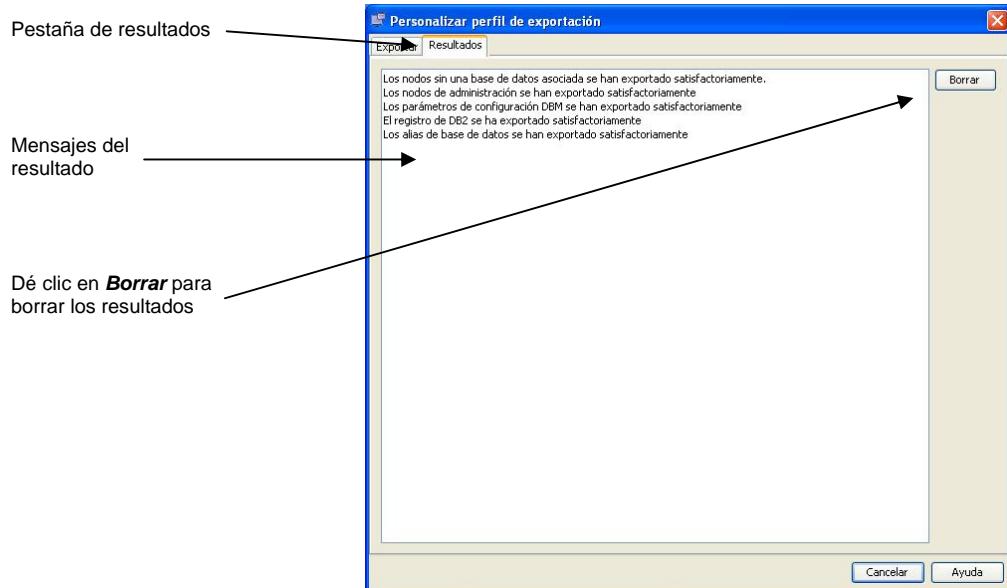


Figura 7.12 – Resultado luego de exportar un perfil

Para importar un perfil personalizado usando el Asistente de configuración, se debe dar clic en el menú Configurar, y luego seleccionar *Importar perfil -> Personalizar*, tal como se ve en la *Figura 7.13*

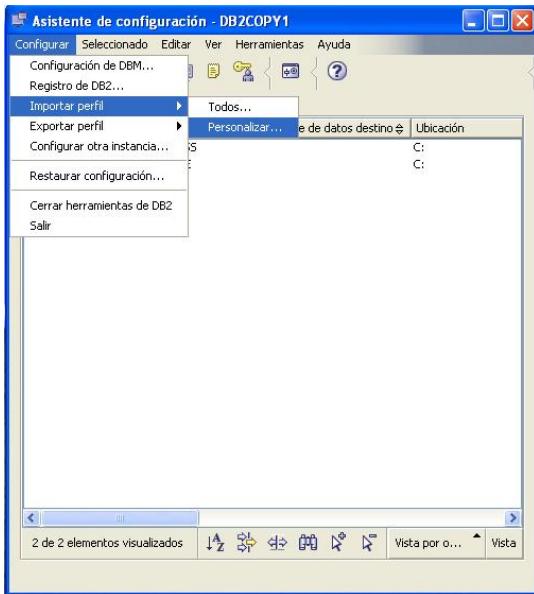


Figura 7.13 – Importando un perfil

La *Figura 7.14* muestra los campos que deben completarse para importar un perfil

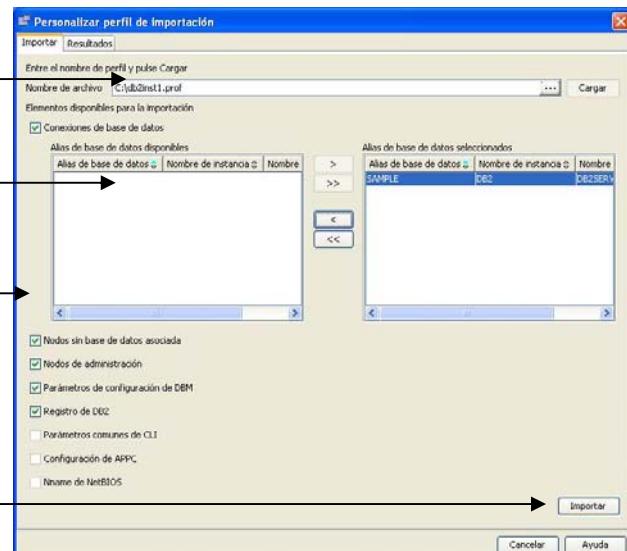


Figura 7.14 – Personalizando la importación de un perfil

7.3 Resumen

Un aspecto central del manejo de bases de datos relacionales es la conexión entre clientes y servidores. En este capítulo hemos visto como conectar un cliente, partiendo con la descripción del propósito y contenido de los directorios de bases de datos y nodos asociados con DB2.

A continuación se vio como usar la interfaz gráfica del Asistente de configuración para preparar la conexión entre cliente y servidor, incluyendo lo que es necesario configurar a ambos lados de la conexión.

También vimos como el Asistente para añadir base de datos se utiliza para definir una conexión al servidor mediante uno de tres métodos posibles: usando un perfil almacenado, buscando en la red (también conocido como Descubrimiento) o bien entrando la información del servidor en forma manual. Finalmente se desarrolló con más detalle el proceso de creación de perfiles de servidor y cliente.

7.4 Ejercicios

La configuración de bases de datos remotas puede hacerse fácilmente usando el Asistente de configuración. En este ejercicio catalogaremos una base de datos ubicada en un servidor DB2 remoto (a los efectos del ejercicio, este servidor será la estación de trabajo de su compañero de estudios). Usaremos tanto el modo de búsqueda como el de descubrimiento. Una vez que la base de datos está catalogada, se podrá tener acceso a ella tal como si estuviera en el sistema local, todos los procesos de comunicación son realizados por DB2 en forma transparente.

En este ejercicio se asume que se está trabajando conectado a una red, pero si no es el caso, siempre se puede utilizar el computador propio tanto como cliente como servidor, y seguir las instrucciones que se dan a continuación para conectarse al propio sistema.

Procedimiento

1. Obtenga la siguiente información de parte de su compañero de studio (o del instructor):
2. Información de la base de datos remota:

(PR)	Protocolo	<u>TCPIP</u>
(IP)	Dirección IP o nombre del servidor	<u></u>
(PN)	Número de puerto de la instancia	<u></u>
(DB)	Nombre de la Base de Datos	<u>SAMPLE</u>

Ayudas:

Para obtener el nombre del servidor en Windows, escriba `hostname` en una ventana de mandatos

Para obtener la dirección IP en Windows, escriba `ipconfig` en una ventana de mandatos

3. Abra el Asistente de configuración. (Ayuda: Se llega a él desde el menu de Arranque.
4. Abra el menu *Seleccionado* y elija *Añadir base de datos utilizando un asistente*.
5. En la página *Fuente* del asistente, seleccione la opción *Configurar manualmente una conexión a una base de datos*. Dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente.
6. En la página *Protocolo* del asistente, seleccione la opción TCP/IP. Dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente.
7. En la página *TCP/IP* del asistente, ingrese el nombre completo del sistema principal o la dirección IP que anotó en el paso (1). Entre el número de puerto que anotó en el paso (1). Dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente.
8. Nota: Se puede utilizar la opción de Nombre de servicio si es que hay una entrada en el archivo de servicios local con el número de puerto correspondiente al puerto dónde el servidor remoto está escuchando por conexiones. Cuando se usa esta opción, DB2 buscará la entrada en el archive de servicios de la máquina local, no en el servidor. Si se desea usar esta opción debe agregarse la entrada en el archive de servicios.
9. En la página *Base de datos* del asistente, ingrese en el campo *Nombre de la base de datos*, el nombre de la base de datos definida en el servidor remoto que anotó en el paso (1). Note que el campo *Alias de la base de datos* se llena automáticamente con el mismo valor. Este alias es nombre que las aplicaciones locales usarán para conectarse con esta base de datos. Dado que ya existe una base de datos local llamada SAMPLE, DB2 no le dejará catalogar otra base con el mismo nombre, por lo tanto se debe usar otro alias distinto. Para este ejemplo, cambie el alias de la base de datos por SAMPLE1. Si lo desea puede agregar un comentario opcional asociado a la base de datos. Dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente.
10. En la página *Fuente de datos* del asistente, opcionalmente se puede registrar esta nueva base de datos (fuente de datos) como una fuente de datos ODBC. Esto la registrará automáticamente en el Manejador ODBC de Windows. Para este ejemplo, desmarque *Registrar esta base de datos para CLI/ODBC* dado que no usaremos ODBC. Dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente.
11. En la página *Opciones de nodo* del asistente, especifique que sistema operativo se usa en el servidor que contiene a la base de datos remota. Dado que todas las estaciones de trabajo que se usan para este laboratorio usan Microsoft Windows, asegúrese que se ha seleccionado *Windows* en la lista de posibles sistemas

operativos. El nombre de la instancia debe ser *DB2*, si no lo es, cámbielo por *DB2*. Dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente.

12. La página *Opciones del sistema* brinda la oportunidad de verificar que tanto el nombre del sistema, el nombre del sistema principal y el sistema operativo han sido ingresados correctamente. Dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente.
13. La página *Opciones de seguridad* permite definir si donde tendrá lugar la autenticación del usuario y que método se usará para ello. Seleccione la opción Utilizar el valor de autentificación de la configuración DBM del servidor, de este modo se usará el método especificado en el parámetro **AUTHENTICATION** del archivo de configuración de la instancia remota. Dando clic en el botón Finalizar, se cataloga la base de datos remota y se cierra el asistente. En ese momento debería aparecer una ventana de confirmación, dé clic en el botón Probar conexión para asegurarse que se puede establecer satisfactoriamente la conexión con la base de datos remota. También asegúrese que el nombre de usuario y clave que ingrese para conectarse sea uno válido y definido en el servidor remoto (dado que lo más probable es que el parámetro **AUTHENTICATION** en el servidor, tenga asignado el valor **SERVER**). Si la conexión de prueba funciona, entonces la base de datos remota ha sido bien catalogada. Si no funciona, revise hacia atrás las pantallas del asistente y asegúrese que se han ingresado los valores correctos. (Dé clic en el botón Cambiar para volver a las pantallas del asistente).
14. Abra ahora el Centro de control y trate de ver las diferentes tablas en la recientemente catalogada base de datos remota.
15. Regrese al Asistente de configuración y pruebe de catalogar otra base de datos, esta vez usando la opción Buscar en la red y recorra las pantallas del asistente de la misma forma que lo hizo cuando configuró la conexión manualmente. Tenga en cuenta que si la red es grande, la búsqueda puede demorar largo tiempo antes de entregar algún resultado.

8

Capítulo 8 – Trabajando con los objetos de la Base de datos

En este capítulo veremos objetos de la base de datos, tales como esquemas, tablas, vistas, secuencias, etc. Otros objetos más avanzados tales como activadores, funciones definidas por el usuario (*User Defined Functions - UDFs*) y procedimientos almacenados se ven en el *Capítulo 14, Introducción al desarrollo de aplicaciones con DB2*.

Nota:

Para mayor información acerca de cómo trabajar con objetos de base de datos, vea este video (en inglés): <http://www.channeldb2.com/video/video/show?id=807741:Video:4242>

8.1 Esquemas

Los esquemas son una forma de agrupar bajo un nombre un conjunto de objetos de bases de datos. Fundamentalmente son usados para:

- Indicar quien es el dueño o propietario de un objeto, o relacionarlo con una aplicación
- Agrupar lógicamente un conjunto de objetos

Todos los objetos de base de datos DB2, excepto los sinónimos públicos tienen un nombre calificado de dos partes; el esquema es la primera mitad de ese nombre tal como se ve aquí:

```
<schema_name>. <object_name>
```

El nombre totalmente calificado de un objeto debe ser único. Si uno se conecta a una base de datos y crea o referencia a un objeto sin especificar el esquema, DB2 utiliza como nombre del esquema el identificador de usuario con el cual uno se ha conectado. Por ejemplo, si usted se conecta a la base de datos **SAMPLE** con el identificador de usuario **jmartel**, y crea la tabla **artistas** con la siguiente sentencia **CREATE TABLE**:

```
CREATE TABLE artistas ...
```

El nombre completo calificado de la tabla que ha creado es **jmartel.artistas**.

Se puede usar la sentencia **set schema** para fijar el esquema a usarse durante la sesión. En el *Listado 8.1* podemos ver un ejemplo.

```
connect to sample user jmartel using miclave
select * from staff ## Esto selecciona desde la tabla jmartel.staff
set schema db2admin
select * from staff ## Esto selecciona desde la tabla db2admin.staff
```

Listado 8.1 – Un ejemplo de la sentencia **set schema**

Para ilustrar el uso de los esquemas usaremos un "sistema de concursos". Digamos que una compañía lanza un concurso en el cual los participantes deben crear sus propias tablas y ejecutar algunas operaciones SQL. A todos los concursantes se les da el mismo identificador de usuario para conectarse a la base de datos, y el mismo proceso para crear las tablas en el cual ningún objeto está calificado, lo que es decir que no tienen un esquema asociado. Luego que los participantes se conectan al sistema del concurso, éste genera el nombre del esquema basado en el reloj del sistema. De esta forma el concursante A trabajará con una tabla con el mismo nombre que el concursante B, pero con esquemas diferentes y por ello no habrá conflictos entre ellos.

8.2 Sinónimos públicos (o aliases)

nuevo en
V9.7

Un concepto Nuevo que introduce DB2 9.7 es el de sinónimos públicos, también conocidos como aliases públicos. Los sinónimos públicos permiten referirse a objetos sin necesidad de especificar un esquema. Un ejemplo de esto puede verse en el *Listado 8.2*.

```
connect to sample user jmartel using miclave
create public synonym jorge for table jmartel.staff
select * from jorge
select * from jmartel.jorge ## Error
connect to sample user db2admin using psw
select * from jorge
```

Listado 8.2 – Ejemplo de un sinónimo público

Como se ve en el *Listado 8.2*, primero nos conectamos con el usuario **jmartel** y creamos el sinónimo público **jorge** el cual hace referencia a la tabla **jmartel.staff**. El sinónimo en sí mismo no utiliza un esquema, si tratamos de usar uno recibiremos un mensaje de error. Otros usuarios también pueden usar el sinónimo **jorge** ya que es público, esto se ve también en el ejemplo del *Listado 8.2*.

Si en el ejemplo no hubiéramos usado la palabra **public**, el sinónimo se hubiera creado como privado. El *Listado 8.3* nos muestra el mismo ejemplo pero con un sinónimo privado.

```
connect to sample user jmartel using miclave
create synonym jorge for table jmartel.staff
select * from jorge
select * from jmartel.raul ## Está bien, también funciona
connect to sample user db2admin using psw
```

```
select * from jorge ## Error, db2admin.jorge no existe
select * from jmartel.jorge ## Está bien, funciona
```

Listado 8.3 – Ejemplo de un sinónimo privado

En el *Listado 8.3* se ve claramente que dado que el sinónimo es privado, otro usuario no puede referirse a él sin especificar el esquema.

8.3 Tablas

Una tabla es una colección de datos relacionados agrupada en forma lógica en filas y columnas. En el *Listado 8.4* se ve un ejemplo de como crear una tabla usando la sentencia **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        BLOB(2M)     not logged compact
)
IN mytbls1
```

Listado 8.4 – Ejemplo de una sentencia CREATE TABLE

En las secciones siguientes iremos viendo las partes principales de esta sentencia **CREATE TABLE**

8.3.1 Tipos de dato

En la *Figura 8.1*, obtenida desde el Centro de información DB2, están representados todos los tipos de dato soportados por DB2

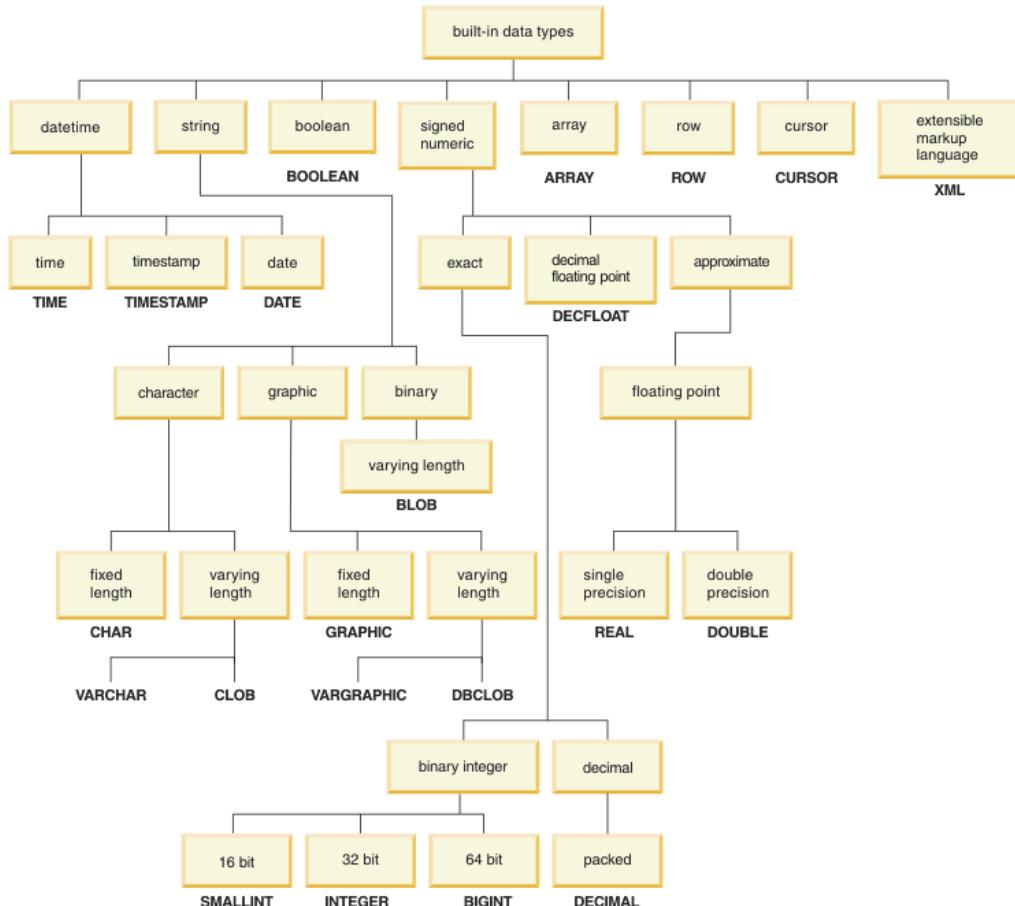


Figura 8.1 – Tipos de datos incluidos en DB2

Los tipos de datos que se ven en la *Figura 8.1* están descritos en detalle en la documentación de DB2; la mayoría de ellos comunes o muy similares entre los distintos sistemas de bases de datos relacionales, por lo tanto no los describiremos aquí. Por otro lado, algunos de estos tipos de datos tales como los objetos grandes (*Large Objects - LOBs*) pueden no ser tan intuitivos para los nuevos usuarios.

Los tipos de datos de objetos grandes se utilizan para almacenar cadenas de caracteres o binarias grandes, o bien archivos, tal como se ve en la *Figura 8.2*.

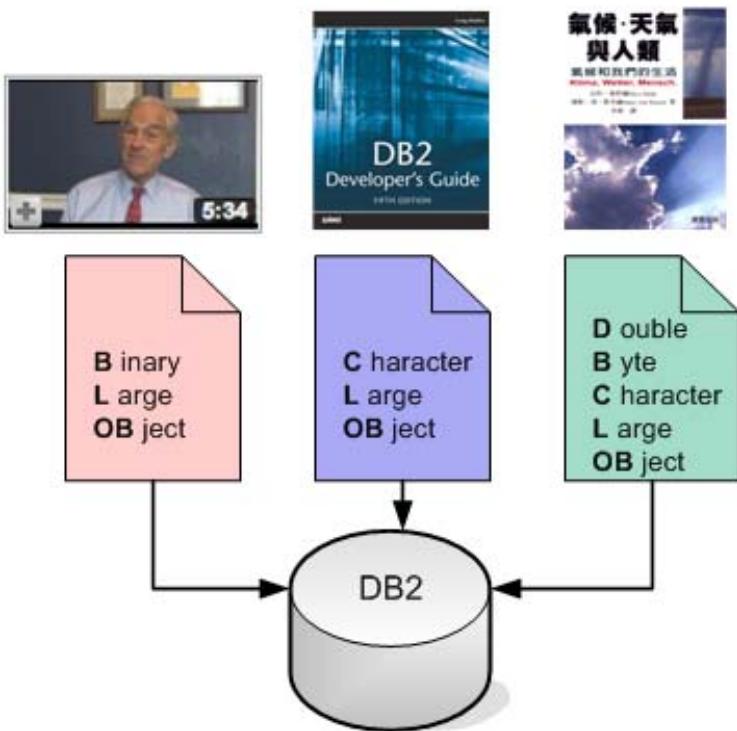


Figura 8.2 – Tipos de datos LOB

Para mayor claridad usualmente nos referimos a estos grandes objetos mediante abreviaturas: un objeto binario grande es un BLOB, uno de caracteres un CLOB, y a uno de caracteres de doble byte se lo denomina DBCLOB.

En la Figura 8.1 aparecen además unos nuevos tipos de datos que fueron introducidos con DB2 9.7:

- BOOLEAN
- ARRAY
- ROW
- CURSOR

Estos tipos de datos son parte de los usados en los servidores de datos Oracle, y son ahora también soportados por DB2. Más adelante en este capítulo trataremos los tipos de datos del servidor de datos Oracle con mayor detalle.

nuevo en
V9.7

8.3.1.1 Tipos de dato definidos por el usuario

DB2 nos permite definir nuestros propios tipos de datos mediante los tipos de datos definidos por el usuario (*User-defined types - UDTs*). Los UDTs pueden ser clasificados como:

- Tipo inequívoco (*Distinct type*)
- Tipo estructurado
- Tipo de referencia
- Tipo de arreglo
- Tipo de fila
- Tipo de cursor

nuevo en
V9.7

Los tipos de referencia, arreglo, fila y cursor son nuevos dentro de DB2 9.7 y son usados en las rutinas de SQL PL. Los tipos de datos inequívocos definidos por el usuario se basan en los tipos de datos que vienen definidos con DB2. Estos UDTs son útiles cuando:

- Hay necesidad de establecer un contexto para los valores
- Existe la necesidad que DB2 imponga una tipología de datos inflexible (*strong typing*)

Con las sentencias SQL del *Listado 8.5* tenemos un ejemplo de como y cuando usar UDTs inequívocos:

```
CREATE DISTINCT TYPE LIBRA AS INTEGER WITH COMPARISONS
CREATE DISTINCT TYPE KILOGRAMO AS INTEGER WITH COMPARISONS
CREATE TABLE persona
  (nombre      VARCHAR(30),
   peso_en_l  LIBRA NOT NULL,
   peso_en_k  KILOGRAMO NOT NULL )
```

Listado 8.5 – Un ejemplo de tipos de datos inequívocos

En este ejemplo, se han creado dos UDTs inequívocos: LIBRA y KILOGRAMO. Ambos están basados en el tipo de datos incluido INTEGER. La cláusula WITH COMPARISONS incluida en la sintaxis indica que se crearán además funciones de conversión con el mismo nombre que los tipos de datos.

La tabla persona usa los dos nuevos UDTs en las columnas peso_en_l y peso_en_k, respectivamente. Si ahora pretendemos ejecutar la siguiente sentencia:

```
SELECT F_NAME FROM PERSONA
  WHERE peso_en_l > peso_en_k
```

Recibiremos un mensaje de error, ya que estamos comparando dos columnas con distintos tipos de datos. Incluso aunque peso_en_l y peso_en_k usan los tipos de dato

LIBRA y KILOGRAMO respectivamente, ambos de los cuales han sido creados en base al tipo de datos INTEGER, al crear los UDTs, hemos hecho este tipo de comparación imposible, y esto es exactamente lo que queríamos ya que, en la vida real, comparar libras con kilogramos (o peras con manzanas) no tiene ningún sentido.

En el ejemplo siguiente pretendemos comparar la columna `peso_en_l` con un número entero; sin embargo, también aquí estos dos tipos de datos son diferentes y por lo tanto recibiremos un mensaje de error a menos que usemos una función de conversión.

Como se ve en la sentencia más abajo, usamos la función de conversión `LIBRA()` para hacer esta comparación posible. Como ya hemos dicho, la función de conversión `LIBRA()` se crea junto con el UDT cuando se usa la cláusula `WITH COMPARISONS` en la sentencia `CREATE DISTINCT TYPE`.

```
SELECT NOMBRE FROM PERSONA
WHERE peso_en_l > LIBRA(30)
```

8.3.1.2 Tipos de datos del servidor de base de datos Oracle

Los siguientes tipos de datos usados en el servidor de base de datos Oracle son ahora tambien soportados por el servidor de datos DB2: NUMBER, VARCHAR2, TIMESTAMP(n), "DATE", BOOLEAN, INDEX BY, VARRAY, Row Type, Ref Cursor. Para poder trabajar con ellos es necesario primero definir la variable de registro `DB2_COMPATIBILITY_VECTOR` de la siguiente forma:

```
db2set DB2_COMPATIBILITY_VECTOR=FF
db2stop
db2start
```

Una vez que se ha definido esta variable, las nuevas bases de datos serán capaces de soportar estos tipos de datos. Algunos de estos tipos solo pueden usarse dentro del contexto en el cual se use SQL PL.

Nota:

Si se está utilizando una edición de DB2 que soporta la característica de compatibilidad de SQL (*SQL Compatibility feature* – descrito en el *Capítulo 2*), tambien se pueden usar estos tipos de datos en todo sitio donde se use PL/SQL. Para este caso el valor a asignar a la variable de registro `DB2_COMPATIBILITY_VECTOR` debe ser 'FFF' en lugar de 'FF' como se ha dicho más arriba.

8.3.1.3 Conversión implícita o tipología flexible

Muchos lenguajes dinámicos tales como Ruby on Rails o PHP permiten la conversión implícita. Para DB2 esto era un problema dado su requerimiento de tipología inflexible. En DB2 9.7, estas reglas se han relajado un poco de modo que se permite la conversion

nuevo en
V9.7

nuevo en
V9.7

implícita, también conocida como tipología flexible. Todo esto significa que ahora, por ejemplo, se pueden asignar o comparar cadenas de caracteres con tipos numéricos, tal como se muestra en este ejemplo:

```
create table t1 (col1 int)
select * from t1 where col1 = '42'
```

Aquí se ve como se puede ahora comparar la cadena de caracteres '42' con la columna numérica `col1`.

Adicionalmente DB2 9.7 ahora permite especificar marcadores de parámetros sin tipificar y también valores no definidos o nulos (`NULL`) sin tipificar en más situaciones. Anteriormente había que explícitamente convertirlos a un tipo de datos dado, por ejemplo ahora la siguiente sentencia ejecuta sin errores:

```
select ?, NULL, myUDF(?, NULL) from t1
```

8.3.1.4 Valores no definidos o nulos (`Null`)⁶

Un valor `null` representa un estado desconocido. La sentencia CREATE TABLE puede definir una columna usando la cláusula NOT NULL para asegurar que la columna contendrá un valor conocido. También se puede especificar un valor por omisión para la columna cuando se la declara NOT NULL. La siguiente sentencia ejemplifica este caso:

```
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
)
```

En este ejemplo las columnas `ID` y `DEPT` han sido definidas como NOT NULL. Adicionalmente a la columna `DEPT` se le ha dado el valor por omisión 10, que será asignado a la columna en caso que no se provea otro valor.

8.3.2 Columnas de identidad

Una columna de identidad es una columna numérica la cual genera automáticamente un valor numérico único cada vez que se inserta una fila. Solo puede definirse una columna de identidad por tabla.

⁶ De ahora en adelante nos referiremos a los valores no definidos o nulos por su nombre en inglés: `null` (N. del T.)

Existen dos formas de generar valores para una columna de identidad, dependiendo de la manera en que ésta ha sido definida:

- **GENERATED ALWAYS:** en este caso, los valores son generados siempre por DB2. Las aplicaciones no pueden proveer un valor explícito.
- **GENERATED BY DEFAULT:** La aplicación puede dar un valor explícito a la columna, pero si no es así, en ese caso DB2 genera uno, pero DB2 no puede garantizar la unicidad. Esta opción está orientada para ser usada en casos de propagación de datos o para descargar y recargar una tabla

Veamos el siguiente ejemplo:

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

En el ejemplo, la columna ***subscriberID*** es un INTEGER definida como columna de identidad que será siempre generada. El valor generado comenzará desde 100 y será incrementado de a 100 cada vez.

8.3.3 Objetos de secuencia

A pesar que los objetos de secuencia son independientes de las tablas, hablaremos de ellos en esta sección dado que trabajan en forma similar a las columnas de identidad. La diferencia reside en que los objetos de secuencia generan números únicos para toda la base de datos mientras que las columnas de identidad lo hacen sólo dentro de una tabla. En las siguientes sentencias se ve un ejemplo de esto:

```
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
```

Listado 8.6 – Ejemplo de secuencias

PREVVAL entrega el valor actual de la secuencia, mientras que NEXTVAL provee el siguiente valor. En el ejemplo anterior también se usa SYSIBM.SYSDUMMY1. Esta es una tabla del catálogo que contiene una columna y una fila; y puede ser usada en situaciones en que se requiera que la consulta entregue un solo valor. Hablaremos de las tablas del catálogo del sistema en la sección siguiente.

8.3.4 Las tablas del catálogo del sistema

Cada base de datos tiene sus propias tablas y vistas de catálogo, donde están almacenados los *metadatos* sobre los objetos de la base de datos. Si estas tablas del sistema se corrompen, la base de datos podría terminar inusable. Estas tablas pueden consultarse tal como cualquier otra tabla de la base de datos. Para identificar las tablas del catálogo del sistema se usan tres esquemas distintos:

- SYSIBM: Tablas de base, optimizadas para el uso de DB2
- SYSCAT: Vistas basadas en las tablas SYSIBM, optimizadas para ser utilizadas fácilmente por los usuarios
- SYSSTAT: Estadísticas de la base de datos

Estas son algunas de las vistas del catálogo:

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

8.3.5 Tablas globales temporales declaradas (*Declared global temporary tables - DGTTs*)

Las tablas globales declaradas temporales son tablas creadas en memoria y que son dadas de baja automáticamente cuando la aplicación termina. Estas tablas solo pueden ser usadas por la aplicación que las ha creado y no están definidas en ninguna de las tablas del catálogo DB2. El acceso a estas tablas se realiza de forma muy eficiente ya que no hay contención en el catálogo, las filas no quedan trabadas por reserva, no hay registro de cambios en los archivos de anotación cronológica (la anotación es opcional) y no hay verificaciones de seguridad. También pueden definirse índices estándar en estas tablas y se puede ejecutar el utilitario RUNSTATS contra las mismas.

Las tablas declaradas temporales residen dentro de un espacio de tabla temporal del usuario, el que debe estar definido antes de crear cualquier tabla declarada temporal. En el *Listado 8.7* se ve un ejemplo de como crear tres tablas declaradas temporales:

```
CREATE USER TEMPORARY TABLESPACE apptemps  
    MANAGED BY SYSTEM USING ('apptemps');  
  
DECLARE GLOBAL TEMPORARY TABLE tempemployees  
    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;
```

Listado 8.7 – Trabajando con las DGTTs

El esquema que se usa al crear una tabla declarada temporal, es SESSION; este esquema debe ser especificado al referenciar la DGTT. El usuario que ha creado la tabla temporal, tendrá todos los privilegios sobre la misma. Cada aplicación que crea una tabla temporal tendrá su propia copia independiente, tal como se ve en la *Figura 8.3*.



Figura 8.3 – Alcance de las tablas globales declaradas temporales

En el *Listado 8.8* se ven las limitaciones del alcance de las tablas globales temporales declaradas. Se asume que ya se ha creado un espacio de tabla temporal de usuario

Listado de la Ventana de mandatos DB2 #1:

```
db2 connect to sample
db2 declare global temporary table mydgtt (col1 int, col2 varchar(10)) on
commit preserve rows
db2 insert into session.mydgtt values (1,'hello1'),(2,'hello2'),
(3,'hello3')
db2 select * from session.mydgtt

COL1          COL2
-----
1 hello1
2 hello2
3 hello3
3 record(s) selected.
```

Listado de la Ventana de mandatos DB2 #2:

```
db2 connect to sample
db2 select * from session.mydgtt
SQL0204N "SESSION.MYDGTT" is an undefined name.  SQLSTATE=42704
```

Listado 8.8 – Alcance de las DGTTs

Tal como puede verse en el *Listado 8.8*, cuando se trata de usar **SESSION.MYDGTT** en la segunda sesión (Ventana de mandatos DB2 #2), se recibe un mensaje de error debido a que la DGTT no ha sido definida en esa sesión. Nótese que en la definición de la DGTT se usa la cláusula ON COMMIT PRESERVE ROWS dado que al trabajar en la Ventana de mandatos DB2 luego de cada sentencia ingresada se aseguran los cambios en la base de datos (*commit*).

nuevo en
V9.7

8.3.6 Tablas creadas globales temporales (Create Global Temporary Tables - CGTTs)

A pesar que las DGTTs permiten crear una tabla temporal, ésta no puede ser compartida por otras conexiones o sesiones. Se debe ejecutar la sentencia DECLARE GLOBAL TEMPORARY TABLE cada vez que se establece una sesión. Por otro lado, con las tablas creadas globales temporales (*Create Global Temporary Tables - CGTTs*) esta definición solo precisa ser realizada una vez y es almacenada en el catálogo DB2 en forma permanente, lo cual significa que otras conexiones pueden simplemente usar la tabla sin necesidad de crearla nuevamente. A pesar que la estructura de la tabla puede ser usada inmediatamente, los datos de cada conexión son independientes y desaparecerán al cerrarse la conexión. Veamos un ejemplo de esto en el *Listado 8.9*, allí se asume que un espacio de tabla temporal del usuario ya ha sido creado.

Listado de la Ventana de mandatos DB2 #1:

```
db2 connect to sample
db2 create global temporary table mycgtt (col1 int, col2 varchar(10)) on
commit preserve rows
db2 insert into mycgtt values (1,'hello1'),(2,'hello2'), (3,'hello3')
db2 select * from mycgtt

COL1      COL2
-----
1 hello1
2 hello2
3 hello3
3 record(s) selected.
```

Listado de la Ventana de mandatos DB2 #2:

```
db2 connect to sample
db2 select * from mycgtt

COL1      COL2
-----
0 record(s) selected.
```

Listado 8.9 – Alcance de las CGTTs

En el *Listado 8.9*, vemos que dentro de la Ventana de comandos DB2 #2 (otra sesión o conexión), no hay necesidad de crear nuevamente la CGTT, simplemente se utiliza, pero al consultarla, no se obtiene ninguna fila, ya que los datos insertados en la primera sesión son particulares de la misma.

8.4 Vistas

Una vista es una representación de los datos de una tabla. Los datos de la vista no se almacenan en forma separada, sino que se obtienen desde las tablas cada vez que se invoca la vista. También pueden definirse vistas anidadas, esto es vistas definidas sobre otras vistas. Toda la información referida a las vistas se almacena en las siguientes tablas del catálogo DB2: SYSCAT.VIEWS, SYSCAT.VIEWDEP, y SYSCAT.TABLES. En el *Listado 8.10* se ve un ejemplo de como crear y usar una vista.

```
CONNECT TO MYDB1;

CREATE VIEW MYVIEW1
AS SELECT ARTNO, NAME, CLASSIFICATION
FROM ARTISTS;

SELECT * FROM MYVIEW1;
```

Output:

ARTNO	NAME	CLASSIFICATION
10	HUMAN	A
20	MY PLANT	C
30	THE STORE	E
...		

Listado 8.10 – Trabajando con vistas

8.5 Índices

Un índice es un conjunto ordenado de claves cada una de las cuales apunta a una fila en una tabla. Mediante un índice se puede garantizar la unicidad de las claves de una tabla además de mejorar el rendimiento al utilizarla. Algunas de las características que pueden definirse en un índice son:

- El orden del índice puede ser ascendente o descendente
- Las claves del índice pueden ser únicas o no
- Se puede usar más de una columna de la tabla en el índice (lo que se conoce como índice compuesto)
- Si los datos de la tabla están almacenados físicamente en el mismo orden que el índice, éste se conoce como índice de agrupación (*cluster index*)

Por ejemplo:

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

8.5.1 Asistente de diseño

El Asistente de diseño es una herramienta excelente que ayuda a obtener el diseño óptimo para una base de datos, dada una determinada carga de trabajo SQL. Ayuda además en el diseño de índices, tablas de consulta materializadas (*Materialized Query Tables - MQTs*), agrupación multidimensional (*Multi-dimension clustering - MDC*), y en la partición de la base de datos. Para ejecutar el Asistente de diseño desde el Centro de control, se da clic con el botón derecho del ratón sobre una base de datos y luego se selecciona Asistente de diseño, tal como se ve en la *Figura 8.4*.

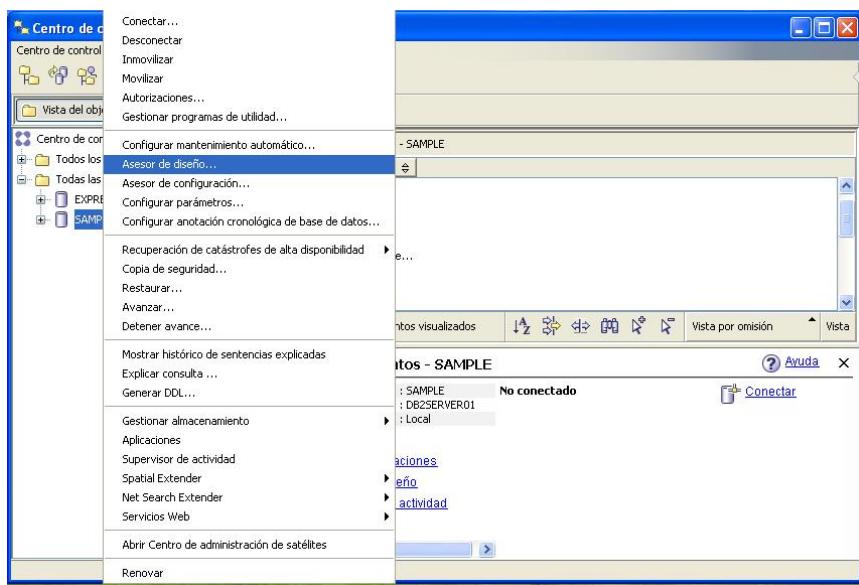


Figura 8.4 – Ejecutando el Asistente de diseño desde el Centro de control

En la Figura 8.5 se ve al Asistente de diseño. Siguiendo los pasos de este asistente se obtienen las recomendaciones de diseño que entrega DB2.

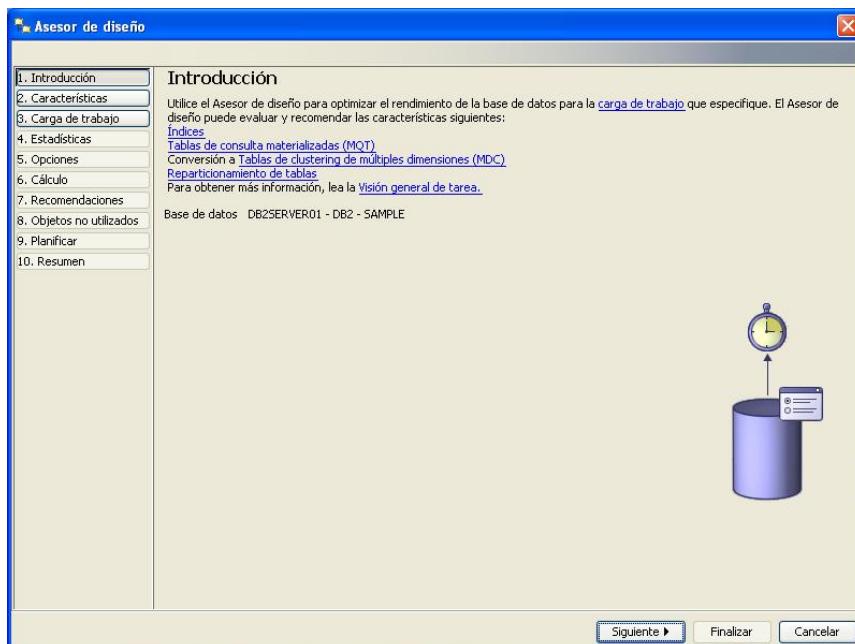


Figura 8.5 – El Asistente de diseño

8.6 Integridad referencial

Mediante la integridad referencial, la base de datos puede manejar relaciones entre las tablas de tipo madre-hija, tal como se ve en la *Figura 8.6*. Allí hay dos tablas, DEPARTMENT y EMPLOYEE, relacionadas por el número de departamento. La columna WORKDEPT en la tabla EMPLOYEE solo puede almacenar números de departamento que ya existan en la tabla DEPARTMENT (columna DEPTNO). Todo esto porque en el ejemplo la tabla DEPARTMENT es la tabla madre y la tabla EMPLOYEE es la tabla hija o tabla dependiente. En la figura se ve además las sentencia CREATE TABLE para la tabla EMPLOYEE con las cláusulas necesarias para establecer la relación.

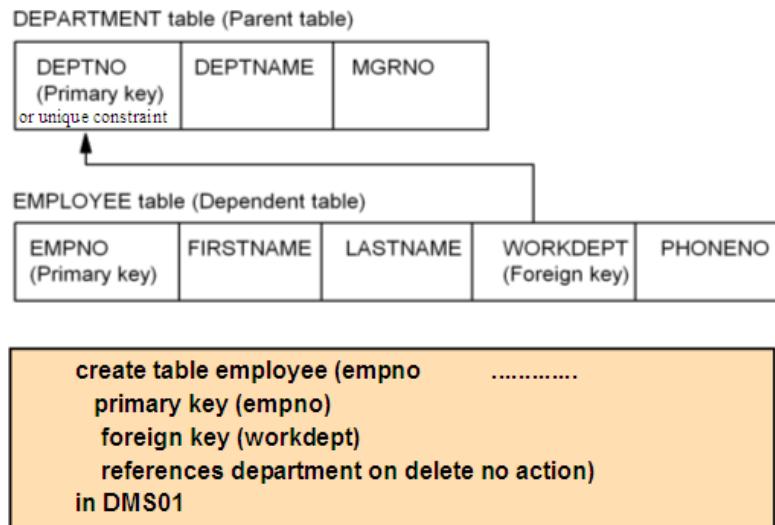


Figura 8.6 – Ejemplo de integridad referencial entre tablas

Los conceptos más usados en integridad referencial pueden verse en la *Tabla 8.1*.

Concepto	Descripción
Tabla madre	Tabla de datos controladora donde reside la clave madre (<i>parent key</i>)
Tabla dependiente	Tabla que depende de los datos en la tabla madre y que contiene una clave foránea. Para cada fila que existe en la tabla dependiente, debe existir una fila correspondiente en la tabla madre.
Clave primaria	Es la clave madre de la tabla madre. Sus valores deben ser únicos y no deben ser nulos (NULL). Puede consistir de una o más columnas de la tabla.
Clave foránea	Referencia a la clave primaria de la tabla madre

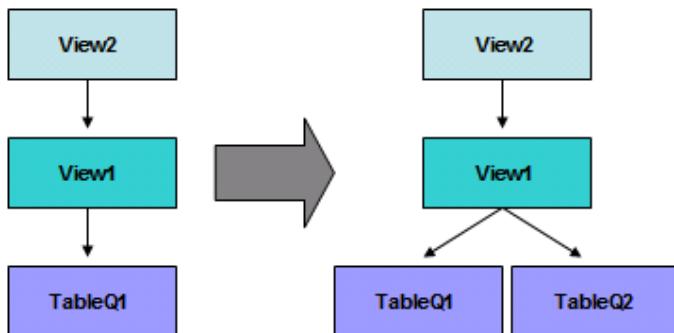
Tabla 8.1 – Conceptos principales de integridad referencial

Con la integridad referencial se pueden relacionar los datos entre una o más tablas. También se pueden imponer restricciones (constraints) a los valores de los datos de modo que guarden conformidad con ciertas propiedades o reglas del negocio. Por ejemplo, si una columna de una tabla almacena el sexo de una persona, se puede definir una restricción de modo que solamente se puedan ingresar los valores "H" por hombre o "M" por mujer.

nuevo en
V9.7

8.7 Evolución de los esquemas

Cuando se requieren cambios de negocio, la infraestructura de la tecnología de la información (TI) y los sistemas también deben cambiar. En el mundo de las bases de datos esto significa que se deberán crear nuevas tablas, tablas existentes tendrán que ser dadas de baja o modificadas, se tendrá que modificar la lógica de los activadores, etc. A pesar que estos cambios pueden parecer simples de realizar, en realidad son bastante difíciles y complejos. Se requerirán ventanas de mantenimiento prolongadas y habrá que realizar procedimientos intrincados y riesgosos. Una de las razones por las cuales estos cambios eran complejos de realizar en el pasado era que DB2 requería que todos los objetos se mantengan consistentes en todo momento. Por ello, o bien no se permitían actividades que afectaran a objetos que dependieran del objeto que estuviese siendo alterado, o bien dichos objetos dependientes debían ser dados de baja. En la *Figura 8.7* puede verse un escenario de ejemplo.

**Figura 8.7 – Ejemplo de evolución de un esquema**

Supongamos por ejemplo que en la *Figura 8.7*, la definición de la vista *view1* precisa ser alterada de forma que no esté solamente basada en la tabla *TableQ1* (donde se almacena la información financiera del primer trimestre de la compañía) sino en el conjunto de las tablas *TableQ1* y *TableQ2* con los dos primeros trimestres respectivamente. Normalmente, uno debería dar de baja la vista *View1* y recrearla con la nueva definición, sin embargo, la vista *View2* depende de la vista *View1*. Antes de DB2 9.7, DB2 no nos permitiría dar de baja la vista *View1* debido a la vista dependiente *View2*. Hubiera sido necesario dar de baja a la vista *View2* en primer lugar, de modo de poder dar de baja a la vista *View1* y luego reconstruir ambas vistas. Con DB2 9.7, las reglas se han relajado de

modo que se permiten cambios que afectan a objetos dependientes. El objeto dependiente (*view2* en el ejemplo), debe ser revalidado antes de ser usado nuevamente, pero esto es algo que DB2 hace automáticamente. Esto es lo que se conoce como **revalidación automática**. Mediante el parámetro AUTO_REVAL del db cfg se puede activar o desactivar la revalidación automática y también determinar cuando ocurrirá. Por ejemplo, si se asigna a este parámetro el valor DEFERRED_FORCE, la revalidación será postergada hasta que el objeto inválido o los objetos dependientes sean usados, pero cuando se desee crear un objeto dependiente nuevo, se permitirán las sentencias CREATE, pero arrojarán un mensaje de advertencia.

Otros cambios que afectan el modelo de dependencia incluyen la implementación de características tales como la sintaxis CREATE OR REPLACE para vistas, funciones procedimientos, activadores, alias, etc. Por ejemplo:

```
create or replace procedure p1 begin ... end
```

Con esta sintaxis, si un objeto (por ejemplo el procedimiento P1) no existe, será creado. Si ya existiera, será reemplazado. Es esto último lo que es importante para la dependencia de los objetos. Cuando P1 es reemplazado, los objetos dependientes de P1 son revalidados automáticamente. Una situación similar ocurre con nuevas características tales como RENAME COLUMN, y cuando se realizan cambios en el tipo de los datos con ALTER COLUMN, sentencia que ha sido mejorada para soportar mayor número de cambios en los tipos de datos.

Otro concepto relacionado es conocido como **invalidación blanda** permite dar de baja un objeto aun cuando haya otras transacciones simultaneas que lo estén usando, pero cualquier nueva transacción no podrá tener acceso al objeto dado de baja.

8.8 Resumen

En este capítulo hemos visto los objetos de las bases de datos DB2: que son, como se crean y como se usan. Presentamos los esquemas y los comparamos con los nuevos sinónimos públicos, los que a su vez comparamos con los sinónimos privados.

Luego hablamos en detalle de las tablas y sus elementos: tipos de datos (incluidos y definidos por el usuario), columnas de identidad, objetos de secuencia y tablas globales temporales. Más tarde nos referimos a las vistas, los índices y al uso de la herramienta gráfica del Asistente de diseño como una forma de mejorar la forma de tener acceso y recuperar los datos desde una tabla.

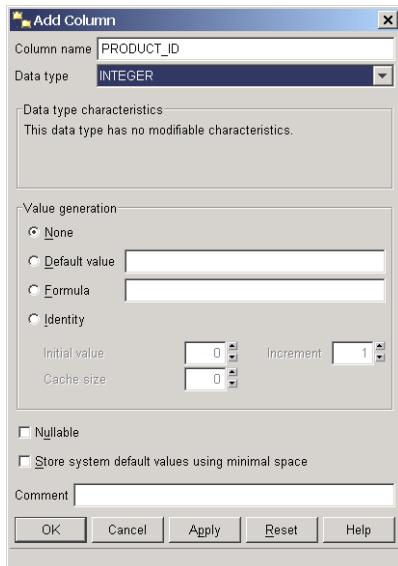
Finalmente, examinamos la integridad referencial como un método de definir relaciones entre las tablas y el novedoso concepto de evolución de esquema, el cual permite cambiar los objetos de datos sin mayor complicación.

8.9 Ejercicios

Hasta ahora, hemos utilizado las tablas ya existentes en la base de datos **SAMPLE** para ilustrar diversos conceptos. Eventualmente en algún momento será necesario crear nuestras propias tablas en la base de datos. En este ejercicio usaremos el *Asistente de creación de tablas* para crear dos tablas nuevas en la base de datos **SAMPLE**.

Procedimiento

1. Ejecute el Asistente de creación e tablas tal como se ha visto anteriormente en el capítulo (*Centro de control -> Todas las bases de datos -> SAMPLE -> (clic derecho) Objeto tabla -> Crear ...*)
2. Defina el nombre de la tabla, las columnas y cualquier restricción. Esta tabla sera usada para almacenar información sobre los suministros de oficina utilizados por los proyectos de la base de datos **SAMPLE**. Cada vez que se adquiera un suministro, se agregará una fila a esta tabla. La tabla tendrá seis columnas:
 - product_id: identificador único del ítem que se ha comprado
 - description: descripción del ítem
 - quantity: la cantidad adquirida
 - cost: costo del ítem
 - image: una foto del ítem (si está disponible)
 - project_num: el proyecto para el cual se ha comprado este producto
3. En la primera página del asistente ingrese como nombre del esquema el identificador de usuario con el que está conectado y use **SUPPLIES** como nombre de la tabla. Si lo desea, ingrese un comentario. Dé clic en el botón *Siguiente* para continuar con la siguiente página del asistente.
4. En esta página se pueden añadir columnas a la tabla. Dé clic en el botón *Añadir* para añadir columnas



5. Ingrese el nombre de la columna ***product_id*** y seleccione el tipo de datos **INTEGER**. Desmarque *Anulable*, y dé clic en el botón *Aplicar* para definir la columna.
6. Repita el paso anterior para las restantes columnas de la tabla usando las opciones de la tabla más abajo. Una vez que todas las columnas han sido agregadas (Aplicadas), dé clic en el botón *Bien* para desplegar un sumario de las columnas recién creadas y luego de clic en el botón *Siguiente* para continuar con la siguiente página del asistente.

Nombre de la columna	Atributos
product_id (listo)	INTEGER, NOT NULL
description	VARCHAR, longitud 40, NOT NULL
quantity	INTEGER, NOT NULL
cost	DECIMAL, precisión 7, escala 2, NOT NULL
image	BLOB, 1MB, NULLABLE, NOT LOGGED
project_num	CHAR, longitud 6, NOT NULL

Nota: Se puede especificar la opción NOT LOGGED cuando se declaran columnas LOB y es obligatorio para columnas mayores de 1GB en tamaño. Generalmente se recomienda además para LOBs mayores de 10MB, dado que los cambios a columnas muy grandes llenarían los archivos de anotación cronológica rápidamente en caso contrario. Aun cuando se use NOT LOGGED, los cambios a

los archivos LOB durante una transacción pueden ser desechos satisfactoriamente. Tenga en cuenta además que la columna para la foto es la única que ha sido definida como **NULLABLE**. ¿Por qué piensa que ha sido así?

7. Hasta este punto, ya hemos ingresado toda la información obligatoria para crear una tabla. Saltando los otros paneles del asistente se eligen las opciones por omisión para esas opciones. Siempre se puede agregar claves y restricciones después de haber creado la tabla.
 8. Agregue ahora una restricción para limitar los valores en la columna *quantity*. En la página de *Restricciones* del asistente, dé clic en el botón *Agregar*. En el campo *Nombre* ingrese **valid_quantities**. En el campo *Condición* ingrese **quantity > 0**
- Dé clic en el botón *Bien*, debería ver un resumen de la restricción que acaba de agregar en la página *Restricciones* del asistente. Dé clic en el botón *Siguiente* para continuar con la próxima página del asistente.
9. Puede ahora continuar a través del asistente, cambiando otros parámetros de la tabla, o bien saltar a la página *Resumen*, o simplemente dar clic en el botón *Finalizar* para crear la tabla.
 10. En el Centro de control, de clic en la carpeta *Tablas* bajo la base de datos **SAMPLE** en el panel de Árbol de objetos. La tabla que acaba de crear debería estar en la lista, si no es así, tal vez sea necesario refrescar el Centro de control para ver los cambios.
 11. Probemos ahora la conversión implícita usando la tabla **STAFF** en la base de datos **SAMPLE**. Ingrese lo siguiente:

```
C:\>db2 describe table staff
```

Note que la columna ID está definida como **SMALLINT**

```
C:\>db2 select * from staff where id = '350' --> Note que '350' es una
cadena de caracteres
```

```
C:\>db2 select * from staff where id = 350 --> Y aquí 350 es un
número
```

En ambos casos la respuesta debería ser:

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
350	Gafney	84	Clerk	5	43030.50	188.00

En la primera sentencia SELECT donde usamos '350' como cadena de caracteres, DB2 realizó una conversión implícita a un número (**SMALLINT**).

9

Capítulo 9 – Utilitarios para el movimiento de datos

En este capítulo trataremos acerca de las herramientas o comandos usados para mover datos dentro de la misma base de datos o entre bases de datos que pueden estar en la misma o diferentes plataformas. En la *Figura 9.1* tenemos una vista general de los utilitarios para el movimiento de datos.

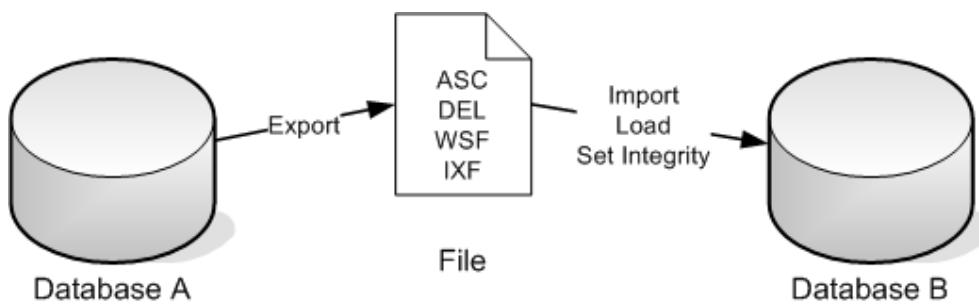


Figura 9.1 – Utilitarios para el movimiento de datos

En la *Figura 9.1* tenemos dos bases de datos, la base de datos A, y la B. Con el utilitario EXPORT se puede exportar los datos desde una tabla a un archivo, el cual puede tener cualquiera de estos formatos:

- ASC = ASCII
- DEL = ASCII delimitado
- WSF = Formato de hoja de cálculo
- IXF = Formato integrado de intercambio (*Integrated Exchange Format*)

Los formatos de archivo ASC y DEL son archivos de texto que pueden ser abiertos y revisados con cualquier editor de texto. El formato WSF puede ser usado para mover datos a una planilla de cálculo, como ser Excel o Lotus® 1-2-3. El formato IXF no solo incluye los datos sino además las sentencias del lenguaje de definición de datos (*Data Definition Language - DDL*) de la tabla en cuestión. Este formato es conveniente, ya que en caso que la tabla tenga que ser reconstruida, esto puede hacerse directamente desde el archivo exportado en formato IXF, lo que no es posible si se usan los otros formatos.

Una vez que los datos han sido exportados a un archive, usando el utilitario IMPORT, se puede importar los datos desde el archivo hacia otra tabla. Si se han usado los formatos ASC, DEL o WSF, la tabla debe existir previamente antes de ejecutar IMPORT, cosa que no es necesaria si se ha usado el formato IXF. Otra forma de cargar datos dentro de una tabla es usando el utilitario LOAD, el que es más rápido ya que graba directamente dentro de las páginas de la base de datos sin pasar por el motor DB2. Sin embargo, este método no valida restricciones y los activadores no se disparan. Una forma usual de garantizar la consistencia de los datos que se han cargado es ejecutar posteriormente a la carga el comando SET INTEGRITY.

Las secciones siguientes describen con mayor detalle los utilitarios EXPORT, IMPORT, y LOAD.

Nota:

Para mayor información sobre como trabajar con los utilitarios de movimiento de datos, vea este video (en inglés):

<http://www.channeldb2.com/video/video/show?id=807741:Video:4262>

9.1 Utilitario EXPORT

Como ya hemos dicho, el utilitario EXPORT se usa para extraer datos desde una tabla hacia un archivo. Pero lo que realmente se está ejecutando por detrás, es una sentencia SQL SELECT. En el ejemplo siguiente, se exportan 10 filas de la tabla `employee` hacia el archivo `employee.ixf` con formato IXF.

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

Les alentamos a tratar de ejecutar este ejemplo. La tabla `employee` es parte de la base de datos `SAMPLE`, por lo que primero es necesario conectarse a esta base de datos que hemos creado en un capítulo anterior.

En caso que prefiera trabajar con una herramienta gráfica, el utilitario EXPORT puede ser invocado desde el Centro de control, tal como se ve en la *Figura 9.2*.

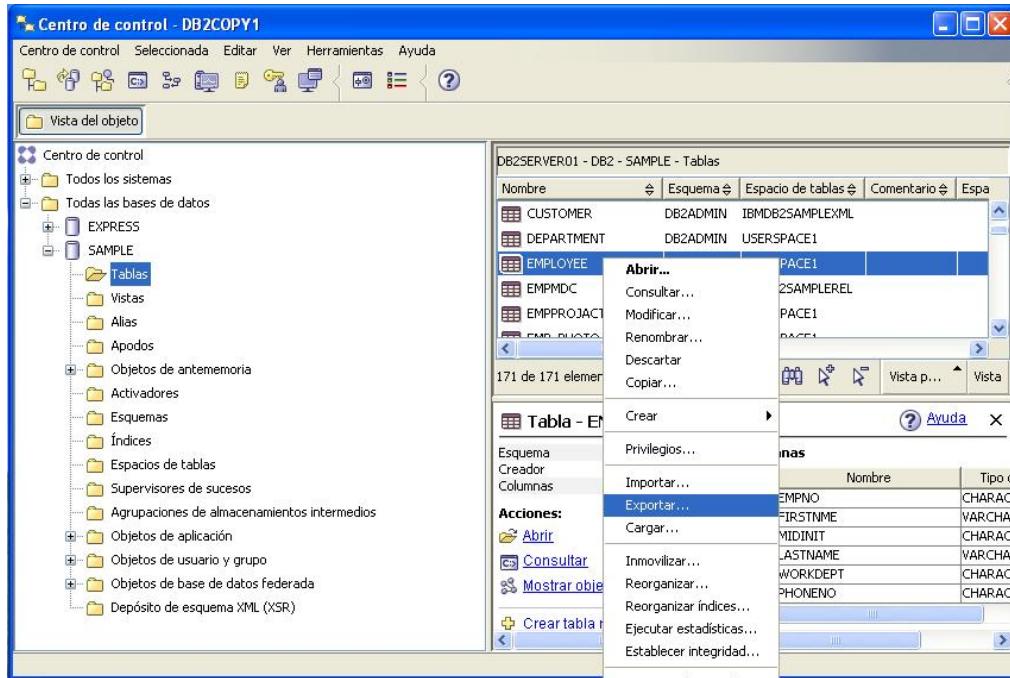


Figura 9.2 – Lanzando el diálogo para ejecutar EXPORT

Tal como se ve en la figura, primero se selecciona la tabla *employee* con un solo clic, y luego otro clic con el botón derecho del ratón hará que se despliegue un menú emergente, desde donde se podrá seleccionar la opción *Exportar*. Una vez elegida esa opción se abrirá el asistente. De aquí en adelante simplemente siga las instrucciones provistas por el asistente para completar la operación.

9.2 Utilitario IMPORT

Tal como hemos visto anteriormente, el utilitario IMPORT se usa para cargar datos desde un archivo hacia una tabla, ejecutando por detrás una operación de SQL INSERT. Dado que lo que se está ejecutando es una operación de INSERT, todos los activadores se disparan, las restricciones se hacen valer y se utiliza la agrupación de almacenamientos intermedios. En el siguiente ejemplo se cargan todos los datos desde el archivo *employee.ixf* con formato IXF hacia la tabla *employee_copy*. Les alentamos a probar este ejemplo, pero recuerden que deben haber ejecutado previamente el utilitario EXPORT en la sección anterior.

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

La opción REPLACE_CREATE es una de las muchas opciones disponibles para el utilitario IMPORT. Con esta opción se reemplaza el contenido de la tabla `employee_copy` si es que ésta ya existía al momento de ejecutar el utilitario IMPORT o bien se creará la tabla y luego se cargarán los datos si es que la tabla no existiera previamente.

Si se prefiere trabajar con el Centro de control, se puede ejecutar el utilitario IMPORT seleccionando cualquier tabla, dando clic con el botón derecho del ratón sobre ella y eligiendo la opción *Importar*, tal como se ve en la *Figura 9.3*.

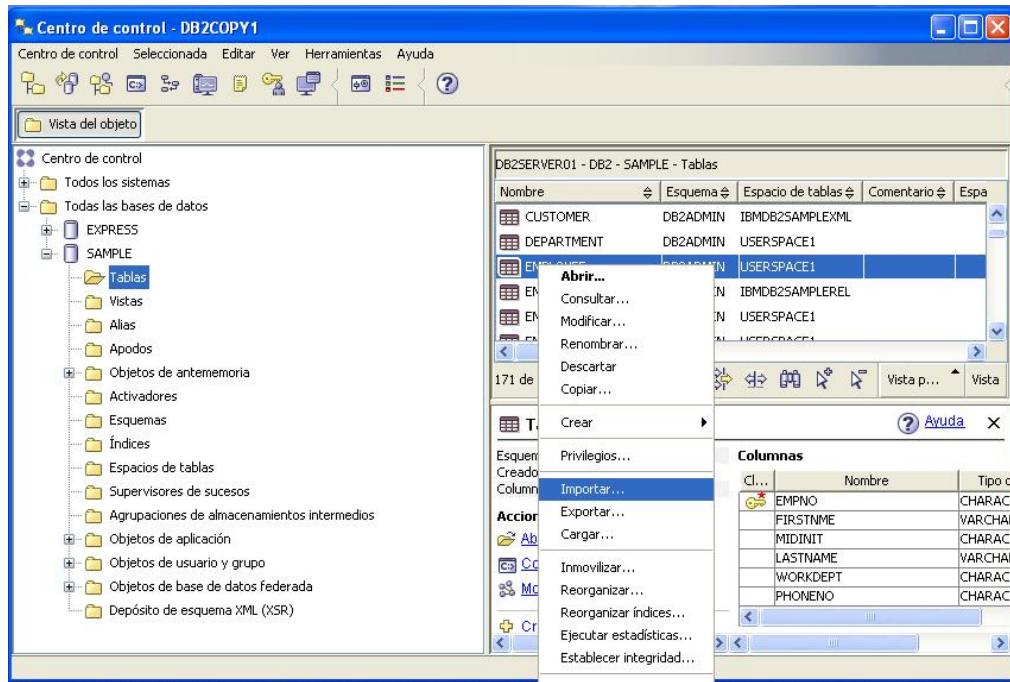


Figura 9.3 – Lanzando el diálogo para ejecutar IMPORT

9.3 Utilitario LOAD

El utilitario LOAD es una forma más rápida de cargar datos desde un archivo hacia una tabla. Como ya se ha dicho, el utilitario LOAD no pasa por el motor DB2, por lo tanto los activadores no se disparan, la agrupación de almacenamiento intermedio no es usada y las restricciones no se hacen valer, aunque estas últimas pueden hacerse valer en un paso separado. Por otro lado una operación de LOAD es más rápida que un IMPORT dado que es una carga de datos a bajo nivel que graba directamente las páginas de datos en el disco. Su trabajo se compone de tres fases: LOAD, BUILD, y DELETE.

En el siguiente ejemplo, todos los datos en el archivo `employee.ixf` de formato IXF, se cargan en la tabla `employee_copy`. La opción REPLACE es una de las muchas opciones

disponibles para el utilitario LOAD. En este caso es usada para reemplazar todo el contenido de la tabla `employee_copy`.

```
LOAD FROM employee.ixf OF IXF
    REPLACE INTO employee_copy
```

Una vez ejecutado el comando anterior, el espacio de tabla donde reside la tabla puede haber sido puesto en estado de CHECK PENDING. Esto significa que será necesario ejecutar el comando SET INTEGRITY para verificar la consistencia de los datos. Esto se hace de la siguiente forma:

```
SET INTEGRITY FOR employee_copy
    ALL IMMEDIATE UNCHECKED
```

Si se quisiera hacerlo desde el Centro de control, se pueden ejecutar los utilitarios LOAD y SET INTEGRITY desde allí, tal como se ve en las *Figura 9.4* y *9.5* respectivamente.

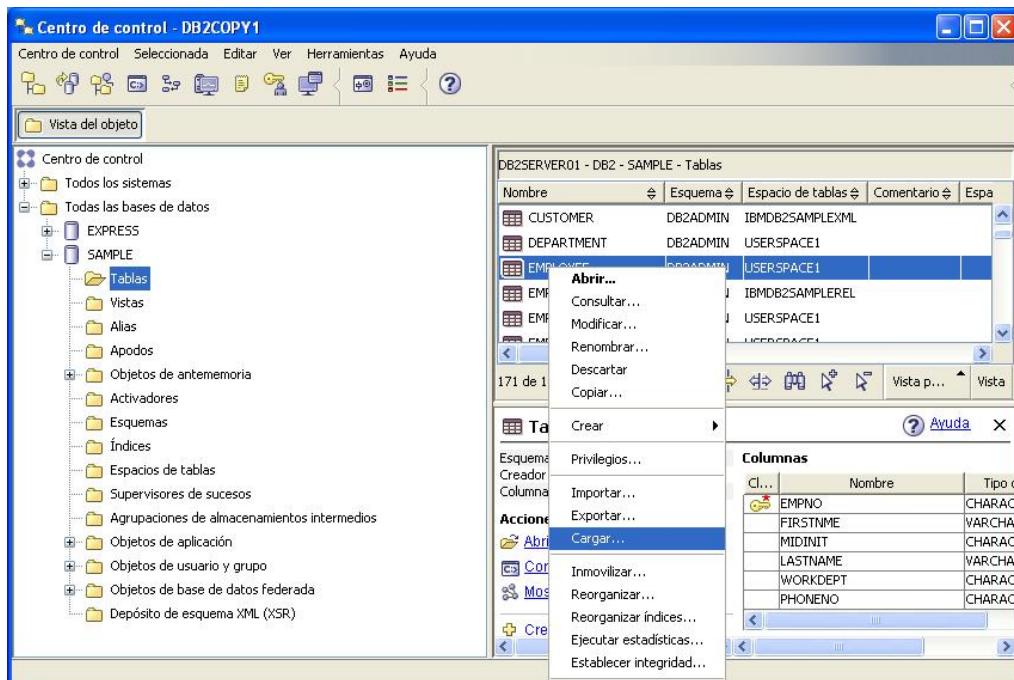


Figura 9.4 – Lanzando el diálogo para ejecutar LOAD

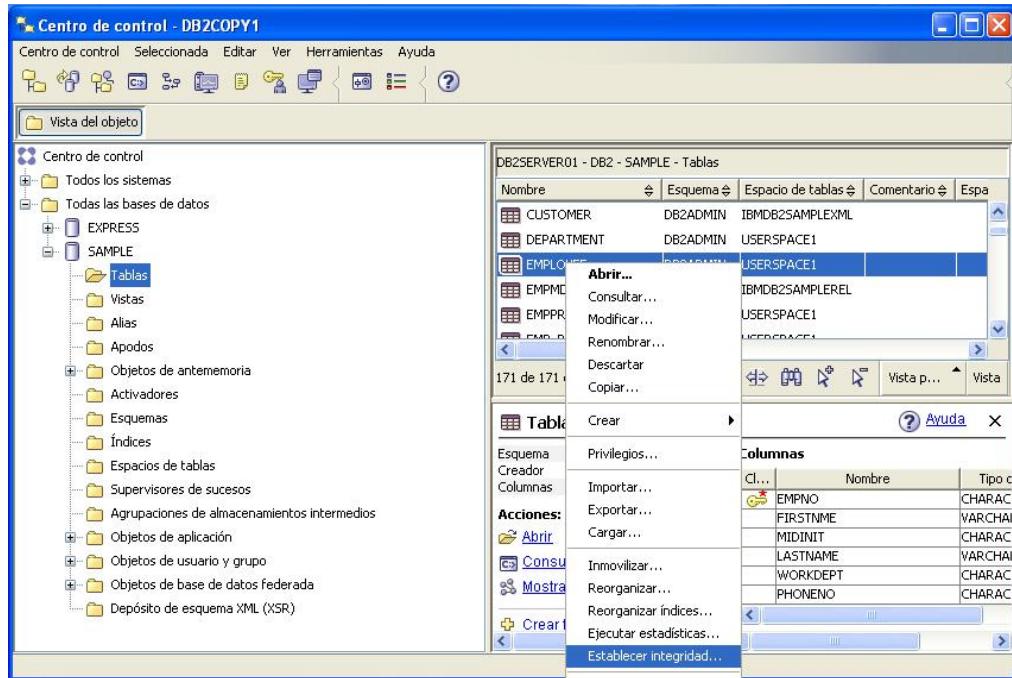


Figura 9.5 – Lanzando el asistente para ejecutar SET INTEGRITY

9.4 El utilitario db2move

Los utilitarios EXPORT, IMPORT, y LOAD trabajan de a una tabla por vez. Por supuesto que puede escribirse una secuencia de comandos que a su vez genere los comandos para ejecutar los utilitarios contra cada tabla de una base de datos; pero existe otro utilitario, denominado **db2move** que ya realiza este trabajo. El utilitario **db2move** solo puede trabajar con archivos IXF, y los nombres de archive serán generados automáticamente por **db2move**. Más abajo se ve un ejemplo de como ejecutar **db2move** con las opciones de exportar e importar respectivamente, usando la base de datos **SAMPLE**.

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

No existe una opción para **db2move** en el Centro de control.

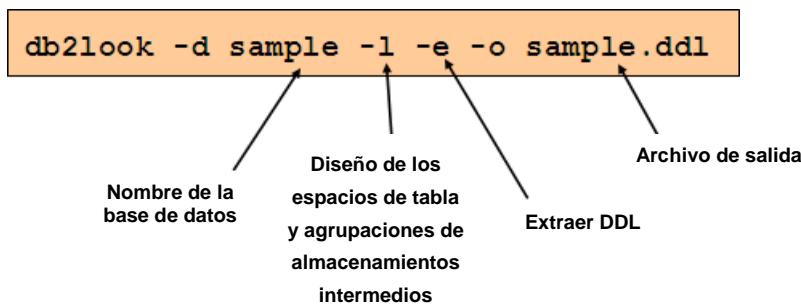
9.5 El utilitario db2look

Mientras que los utilitarios EXPORT, IMPORT, LOAD y **db2move** permiten mover datos desde una tabla a otra, ya sea dentro de la misma base de datos como a través de varias de ellas, el utilitario **db2look** es usado para extraer las sentencias DDL, las estadísticas de una base de datos y las características de los espacios de tabla de una base de datos y

almacenarlos como una cadena de comandos que más tarde pueden ejecutarse en otro sistema. Por ejemplo, si se desea clonar una base de datos desde un servidor DB2 corriendo en Linux hacia otro servidor DB2 en Windows, lo primero es ejecutar el utilitario **db2look** en el servidor DB2 en Linux para obtener la estructura de la base de datos y almacenar dicha estructura en un archivo de cadena de comandos. Luego se copia este archivo de comandos al servidor DB2 Windows y se lo ejecuta para comenzar a construir la base de datos clonada. Una vez hecho esto ya se ha clonado la estructura de la base de datos y el paso siguiente sería ejecutar el utilitario **db2move** con la opción de exportar en el servidor DB2 Linux y luego copiar los archivos generados al servidor DB2 Windows. A continuación se ejecuta el utilitario **db2move** en el servidor DB2 Windows, ya sea con la opción de importar o cargar los datos. Al completarse esta operación, la base de datos ha sido completamente clonada desde un servidor a otro en una plataforma completamente distinta.

El escenario que hemos descrito sería necesario en caso de trabajar con bases de datos en diferentes plataformas, como ser el caso de Linux y Windows. Si ambos servidores corren en la misma plataforma, sería más adecuado utilizar los comandos de respaldo y recuperación (**backup** y **restore**). Veremos los comandos **backup** y **restore** con más detalle en un capítulo posterior.

En el siguiente ejemplo se ve como extraer los diseños de los espacios de tabla y agrupaciones de almacenamientos intermedios, además de las sentencias DDL desde la base de datos **SAMPLE** y almacenarlos dentro del archivo **sample.ddl**. Les alentamos a ejecutar este comando y revisar el archivo de texto de salida **sample.ddl**.



El comando **db2look** tiene numerosas opciones como para describir todas en este libro; sin embargo se puede usar la opción **-h** para obtener una descripción breve de las opciones disponibles:

```
db2look -h
```

También se puede ejecutar el utilitario **db2look** desde el Centro de control, tal como se aprecia en la *Figura 9.6*

176 Conociendo al DB2 Express-C

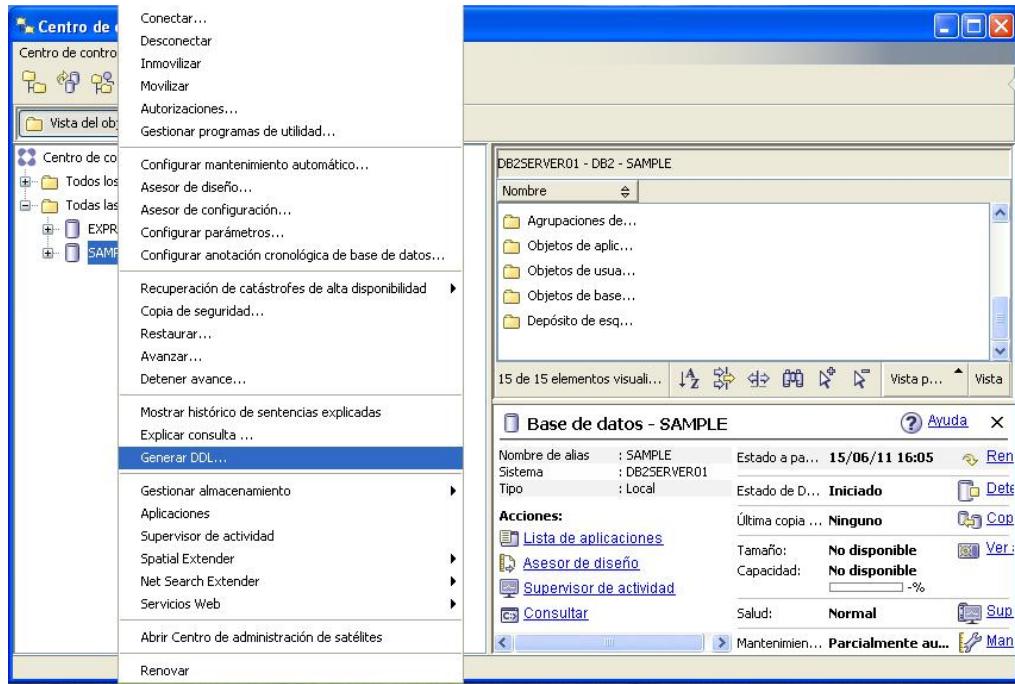


Figura 9.6 – Extracción de DDL desde el Centro de control

En la *Figura 9.6*, se selecciona la base de datos desde la que se desea obtener el DDL, se da clic con el botón derecho del ratón sobre ella y se elige *Generar DDL*; con esto aparece la ventana *Generar DDL*, mostrando varias opciones de extracción, tal como se ve en la *Figura 9.7*.

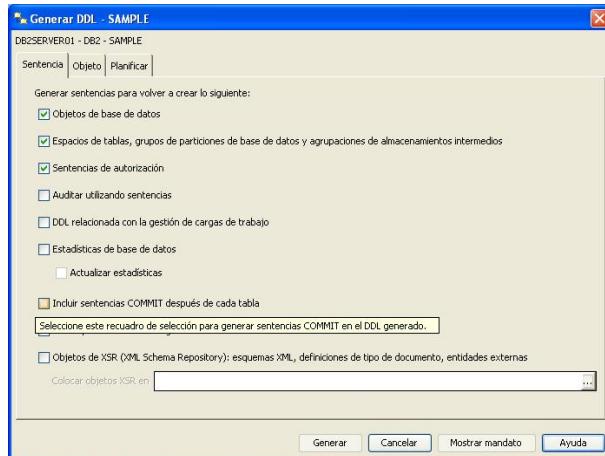


Figura 9.7 – Extrayendo el DDL desde el Centro de control

9.6 Resumen

En este capítulo hemos visto varias funciones de DB2 para exportar e importar. Comenzamos viendo los distintos formatos de exportación (ASC, DEL, WSF, e IXF), y más tarde examinamos con mayor profundidad el utilitario EXPORT. Luego hablamos de los utilitarios para importar, IMPORT y LOAD, conjuntamente con la necesidad de la sentencia SET INTEGRITY cuando se usa LOAD.

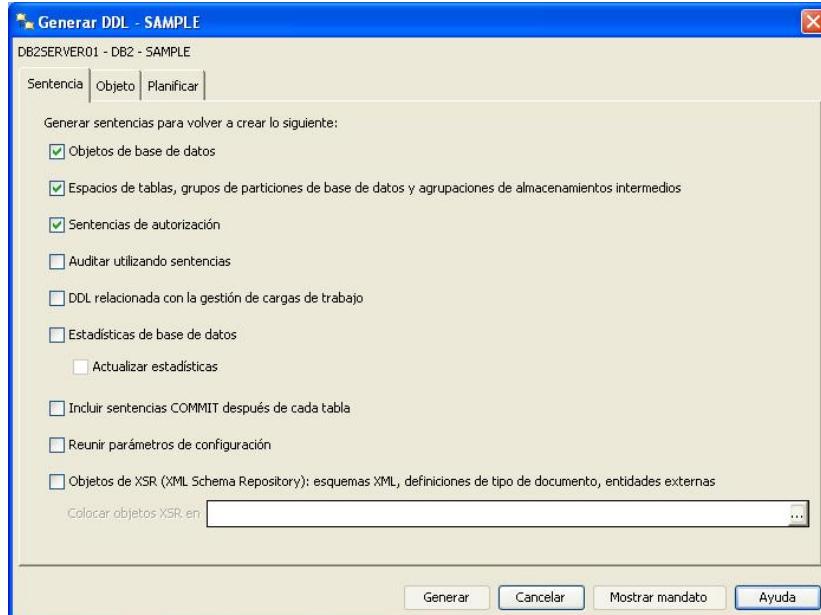
Vimos además como el comando db2move nos facilita la tarea de exportar, transferir e importar. Finalmente describimos como un comando algo más complejo, db2look, nos permite extraer y almacenar todos los elementos necesarios para recrear una base de datos completa en forma independiente, si esto fuera requerido.

9.7 Ejercicios

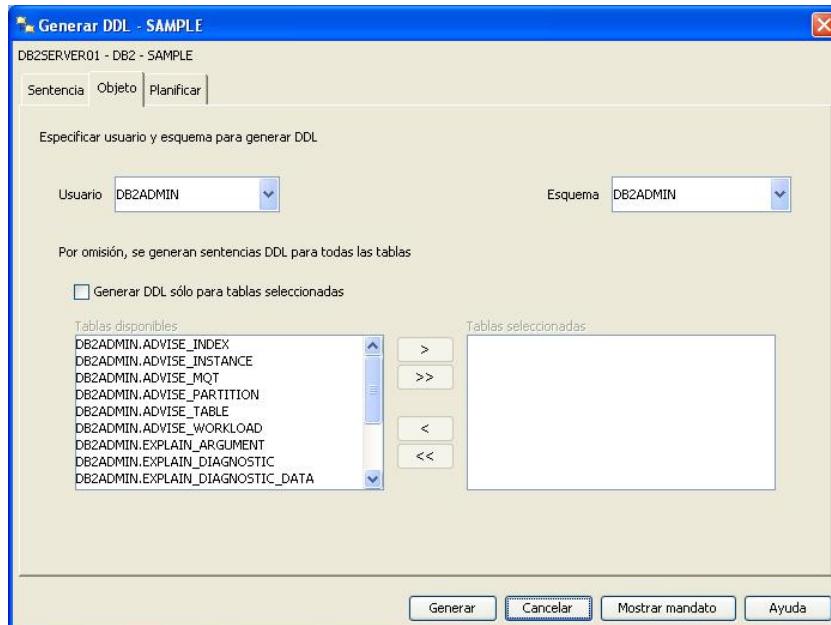
Cuando se clona una base de datos, el objetivo debería ser recrearla en una forma que fuera simple y reproducible. Esto se consigue utilizando cadenas de comandos SQL, las que pueden ejecutarse inmediatamente después de haber instalado DB2. En este ejercicio, usando el Centro de control, extraeremos la definición de los objetos en la base de datos **EXPRESS**, la cual fue creada en un ejercicio anterior.

Procedimiento

1. Abra el Centro de control.
2. Dé clic con el botón derecho del ratón sobre la base de datos **EXPRESS** en el árbol de objetos y seleccione *Generar DDL* en el menú. Esto lanzará la ventana de diálogo *Generar DDL*.
3. En la ventana de *Generar DDL*, especifique las opciones para la generación del DDL, tal como se ve más abajo. Si usted ha creado objetos adicionales en su ambiente, tales como espacios de tablas, agrupaciones de almacenamiento intermedio, éste es el momento de seleccionarlos aquí. Dado que no lo ha hecho, desmarque la opción. No se han incluido las estadísticas dado que si estamos moviendo desde un ambiente de desarrollo a uno de producción, lo más probable es que este último contenga un conjunto de estadísticas diferente al ambiente de desarrollo. Si fuera el caso que ambos ambientes tuvieran exactamente la misma configuración, se podrían incluir estas opciones adicionales.



- Cámbiese a la pestaña de *Objetos*. Allí podrá seleccionar aquellos objetos específicos para los cuales desea generar el DDL. En este caso seleccione el usuario y esquema que ha usado para crear todos sus objetos y genere el DDL para todos los objetos en dicho esquema. Dé clic en el botón *Generar* para comenzar la generación del DDL.



5. Revise el DDL obtenido. El resultado del paso anterior es un archivo de cadena de comandos con todas las sentencias SQL para los objetos seleccionados. Ahora organizará este archivo de comandos en agrupaciones lógicas.
6. Cree un directorio de nombre C :\express en el sistema de archivos y guarde el archivo del DLL generado en este nuevo directorio con el nombre schema .ddl. (Dé clic en el botón Guardar)



7. Abra este Nuevo archivo en el Editor de mandatos. (Ayuda: en el Editor de mandatos seleccione Archivo -> Abrir)
8. Aunque lo único que queríamos era el DDL de las tablas, podrá notar que se ha incluido además el DDL de otros objetos de la base de datos. Mueva todas las sentencias **CREATE TRIGGER** a un nuevo archivo de nombre triggers.ddl. Aun cuando hemos creado un solo activador, es una buena medida separar los objetos por su tipo.
9. En este caso también le recomendamos remover todas las sentencias:
 - CONNECT TO
 - DISCONNECTEn este punto, usted debería tener dos archivos de comando:
C :\express\schema .ddl con todo el DDL de las tablas, vistas, índices y restricciones.
C :\express\triggers.ddl con el DDL de los activadores
10. Limpie el archivo de comandos para el paso a producción:
 - Elimine todos los comentarios innecesarios (por ej. -- CONNECT TO...)
 - Separe todas las funciones y procedimientos dentro de sus propios archivos (esto es útil cuando hay gran cantidad de funciones y procedimientos). Podría también ser apropiado agruparlos por función o aplicación (por ej. billing.ddl, math.ddl, stringfunc.ddl, etc.)

11. Algo que puede haber notado al revisar los comandos es que se utiliza un símbolo especial (@) para delimitar el final de los activadores, las funciones y los procedimientos. Esto es necesario para poder diferenciar el final de la sentencia **CREATE <object>**, del final de una sentencia de procedimiento dentro del objeto.

10

Capítulo 10 – Seguridad de la base de datos

Este capítulo trata de como DB2 maneja la seguridad. Una visión general puede apreciarse en la *Figura 10.1*.

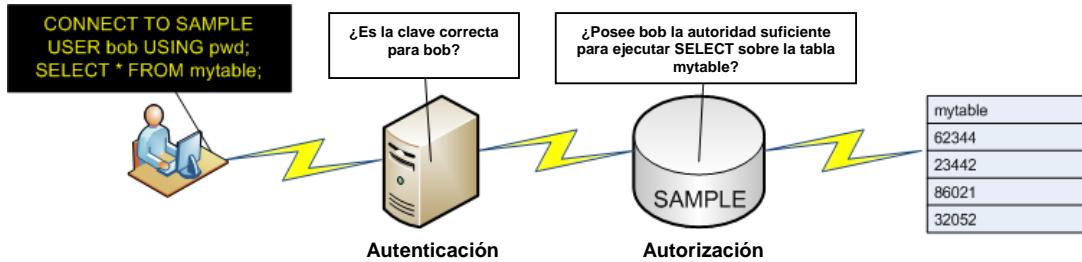


Figura 10.1 – Visión general de la seguridad en DB2

Tal como se ve en la *Figura 10.1*, la seguridad en DB2 consiste de dos partes:

- **Autenticación**

Este es el proceso mediante el cual se valida la identidad del usuario. La autenticación es realizada por una componente externo a DB2 mediante un conector de seguridad entre ambos. El conector de seguridad por omisión se basa en la seguridad del sistema operativo, pero se pueden usar conectores de seguridad para Kerberos, LDAP, o bien construir un conector propio para la autenticación. Cuando se utiliza el conector por omisión basado en el sistema operativo, el identificador de usuario y la clave son enviados al servidor de base de datos (como parte de la sentencia CONNECT). Entonces el servidor de base de datos invoca a los servicios de autenticación del sistema operativo para validar ambos.

- **Autorización**

En esta etapa, DB2 verifica si el usuario ya autenticado puede realizar la operación solicitada. Esta información de autorización está almacenada en el catálogo DB2 y en el archivo de configuración DBM.

Por ejemplo, en la *Figura 10.1*, el usuario *bob* se conecta a la base de datos **SAMPLE** con esta sentencia:

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

Tanto *bob* como *pwd* son transferidos al sistema operativo o al agente de autenticación externo para realizar la aprobación de autenticación, verificando que un usuario llamado *bob* ya está definido y que la clave provista es la que corresponde a ese usuario. Si no hay errores en esta parte, el sistema operativo transferirá el control de vuelta a DB2. Ahora, cuando el usuario *bob* ejecuta una sentencia tal como:

```
SELECT * FROM mytable
```

DB2 toma el control de la verificación de seguridad para realizar una chequeo de autorización y confirmar que el usuario *bob* tiene el privilegio de SELECT sobre la tabla *mytable*. Si falla el chequeo de autorización, DB2 retornará un mensaje de error, en caso contrario la sentencia será ejecutada sobre *mytable*.

Nota:

Para mayor información sobre como trabajar con la seguridad en DB2, vea este video (en inglés): <http://www.channeldb2.com/video/video/show?id=807741:Video:4267>

10.1 Autenticación

Aun cuando la autenticación propiamente dicha es realizada por el sistema operativo a través del conector de seguridad por omisión (o por otro agente de seguridad externo), DB2 decide a que nivel ocurrirá esta autenticación.

El parámetro de configuración de la base de datos AUTHENTICATION, que se fija en el servidor DB2, tiene un rango de valores posibles. Por ejemplo si a este parámetro se le asigna el valor SERVER (valor por omisión), la autenticación se realiza por el sistema operativo o por en agente de seguridad externo, en el servidor. Por otro lado, si el valor de AUTHENTICATION es CLIENT, la autenticación se realiza por el sistema operativo o por en agente de seguridad externo, en el cliente. Esto se muestra en la Figura 10.2.

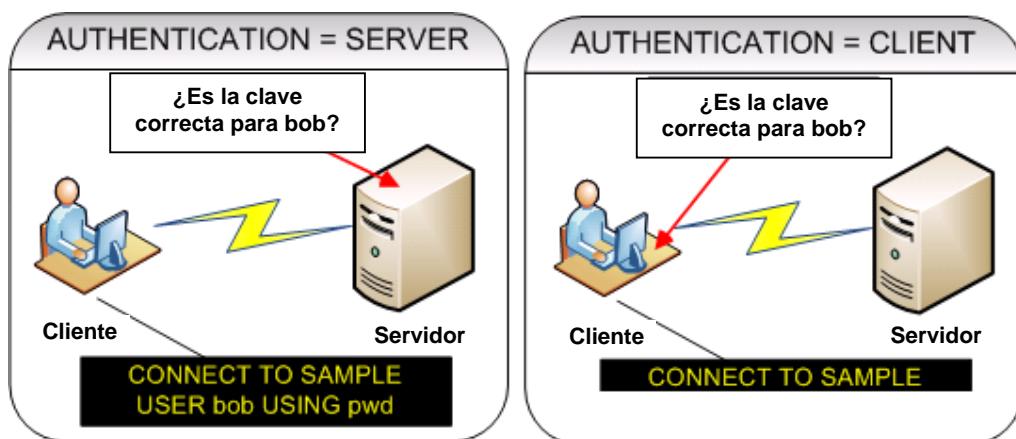


Figura 10.2 – Donde ocurre la autenticación

Al parámetro AUTHENTICATION se le pueden asignar cualquiera de los valores listados en la *Tabla 10.1*

Valor	Descripción
SERVER (por omisión)	La autenticación tiene lugar en el servidor
CLIENT	La autenticación tiene lugar en el cliente
SERVER_ENCRYPT	Tal como en SERVER excepto que los identificadores de usuario y las claves están cifrados
KERBEROS	La autenticación se realiza utilizando un mecanismo de seguridad Kerberos
SQL_AUTHENTICATION_DATAENC	Autenticación en el servidor con el agregado que las conexiones deben además cifrar los datos
SQL_AUTHENTICATION_DATAENC_CMP	Como en el caso anterior, salvo que el cifrado de los datos se hará solo si la facilidad está disponible
GSSPLUGIN	La autenticación usa un conector de seguridad externo basado en un mecanismo GSS API

Tabla 10.1 – Valores válidos para el parámetro AUTHENTICATION

10.2 Autorización

El proceso de autorización consiste de privilegios, autoridades, roles y credenciales de control de acceso basado en etiquetas (*Label-Based Access Control -LBAC*) las cuales están almacenadas en tablas del sistema DB2 y son administradas por él mismo.

Un privilegio le permite a un usuario ejecutar un único tipo de operación contra una base de datos, como ser CREATE, UPDATE, DELETE, INSERT, etc.

Un rol permite agrupar diferentes privilegios los que pueden ser otorgados a un usuario, un grupo o a otros roles.

Una autoridad es un rol predefinido que consiste de varios privilegios.

Las credenciales LBAC incluyen políticas y etiquetas que permiten tener acceso granular a filas y columnas específicas a usuarios determinados. DB2 Express-C no incluye LBAC, pero usted puede leer más sobre el tema en el *Capítulo 2*.

10.2.1 Privilegios

En la *Figura 10.3* se ven algunos de los diferentes privilegios en DB2.

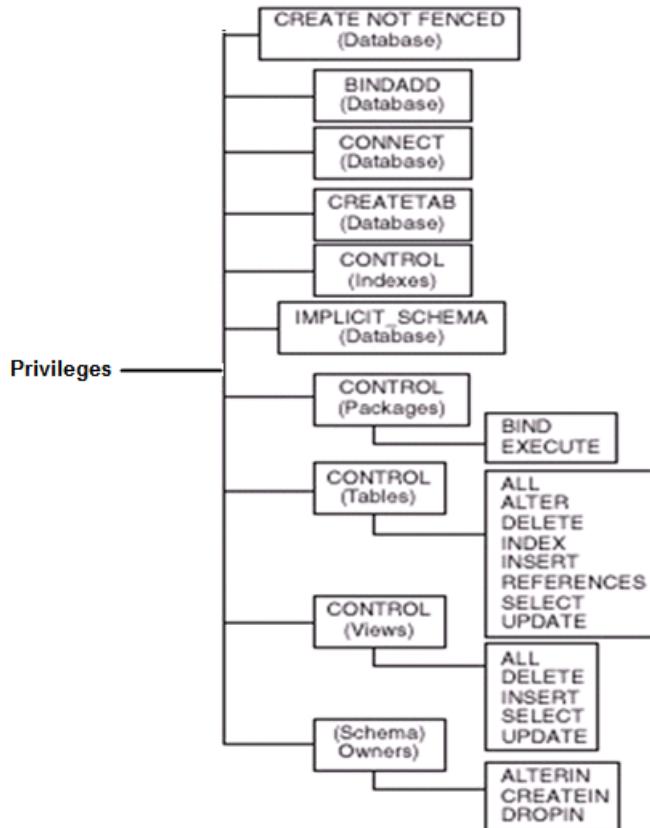


Figura 10.3 – Algunos privilegios en DB2

Si un usuario o grupo recibe el privilegio CONTROL, ello implica que puede además otorgar este privilegio a otro usuario o grupo. Detalle de los otros privilegios se encuentra en el Centro de información DB2.

10.2.2 Autoridades

Las autoridades están clasificadas en dos grupos:

- Autoridades a nivel de instancia: Son aquellas que pueden operar al nivel de la instancia, por ejemplo, SYSADM.
- Autoridades a nivel de la base de datos: Las que solo operan a nivel de la base de datos, por ejemplo, DBADM.

10.2.2.1 Autoridades a nivel de la instancia

La *Tabla 10.2* lista las autoridades a nivel de la instancia.

Autoridad	Descripción
SYSADM	Maneja la instancia como un todo
SYSCTRL	Administra una instancia del manejador de la base de datos
SYSMAINT	Mantiene las bases de datos dentro de la instancia
SYSMON	Monitorea la instancia y sus bases de datos

Tabla 10.2 – Autoridades a nivel de instancia

Para otorgar autoridad de SYSADM, SYSCTRL, SYSMAINT, o SYSMON a un grupo, a los parámetros del DBM CFG: SYSADM_GROUP, SYSCTRL_GROUP, SYSMAINT_GROUP, y SYSMON_GROUP, respectivamente, se les puede asignar un grupo del sistema operativo.

Por ejemplo, para otorgar la autoridad SYSADM al grupo del sistema operativo *myadmns*, se utiliza este comando:

```
update dbm cfg using SYSADM_GROUP myadmns
```

Cada instancia DB2 tiene su propia definición de los grupos de autoridad. En Windows, estos parámetros están vacíos por omisión, lo cual significa que el grupo de Administradores Locales tendrá la autoridad SYSADM. En DB2 9.7, también tendrán autoridad SYSADM, el grupo DB2ADMNS (si está activada la seguridad extendida) y la cuenta *LocalSystem*. El identificador de autorización para la cuenta *LocalSystem* es SYSTEM. En Linux, el grupo propietario de la instancia es el grupo SYSADM por omisión.

En la *Figura 10.4*, extraída desde el Centro de información DB2 se ven las diferentes autoridades a nivel de la instancia y las distintas funciones que pueden realizar. Tal como lo muestra la figura, la autoridad SYSADM incluye todas las funciones de la autoridad SYSCTRL y algo más. La autoridad SYSCTRL incluye todo lo de la SYSMAINT y algo más, y así siguiendo.

nuevo en
V9.7

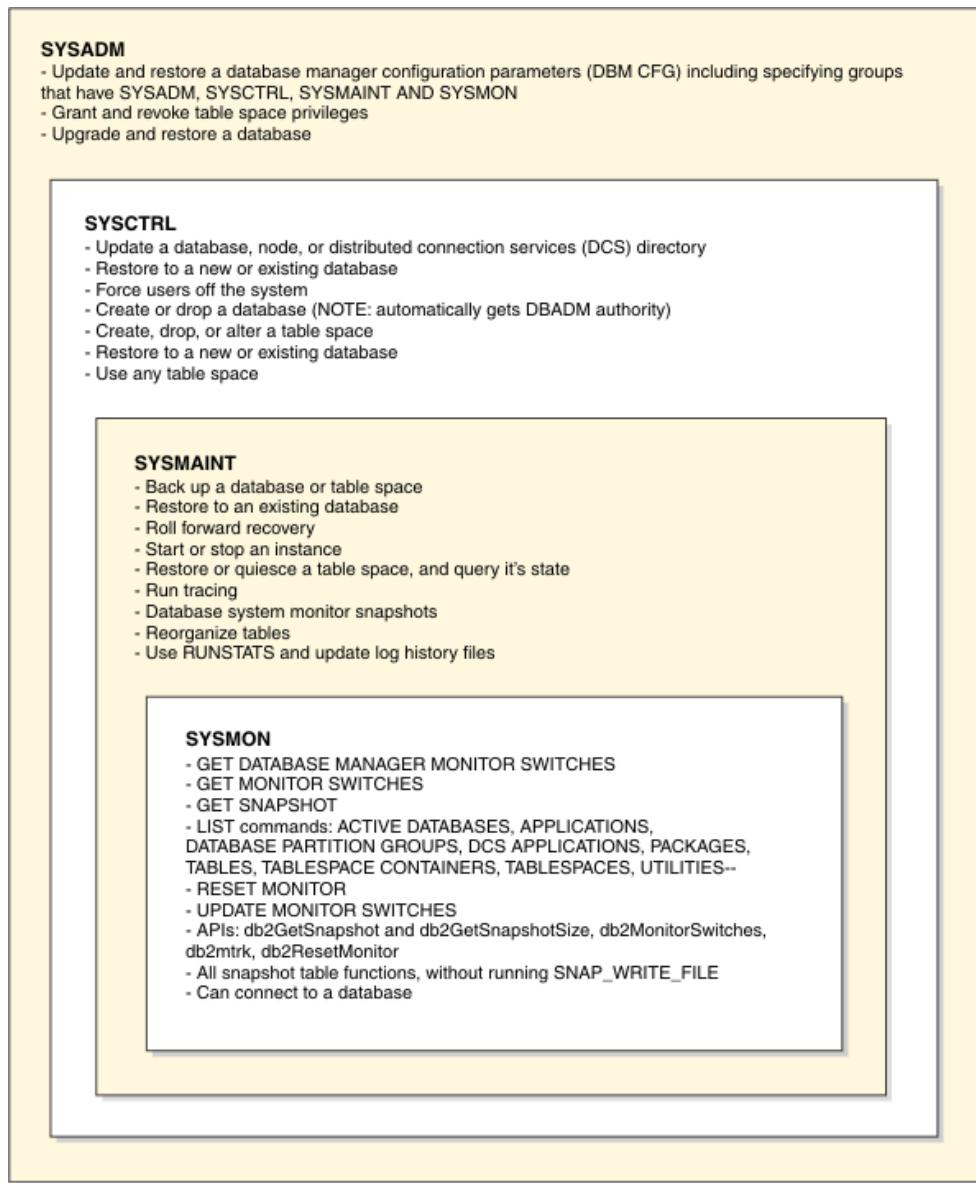


Figura 10.4 – Autoridades a nivel de instancia y sus funciones

10.2.2.2 Autoridades a nivel de la base de datos

La Tabla 10.3 lista las autoridades a nivel de la base de datos.

Autoridad	Descripción
SECADM	Maneja la seguridad dentro de la base de datos
DBADM	Administra la base de datos
ACCESSCTRL <small>nuevo en V9.7</small>	Otorga y revoca autoridades y privilegios (que no sean las autoridades SECADM, DBADM, ACCESSCTRL, y DATAACCESS. Nótese que se requiere autoridad SECADM para otorgar y revocar esas autoridades)
DATAACCESS <small>nuevo en V9.7</small>	Provee la habilidad de tener acceso a los datos en la base de datos
SQLADM	Monitorea y sintoniza las consultas SQL
WLMADM	Maneja las cargas de trabajo
EXPLAIN	Para usuarios que precisan ejecutar la sentencia EXPLAIN sobre los planes de consulta (la autoridad EXPLAIN no otorga acceso a los datos mismos)

Tabla 10.3 – Autoridades a nivel de base de datos

Para poder otorgar una autoridad a nivel de base de datos, se usa la sentencia GRANT. Por ejemplo, para otorgar DBADM en la base de datos **SAMPLE** al usuario **bob**, se usa:

```
connect to sample
grant DBADM on database to user bob
```

En el ejemplo anterior, primero se necesita conectarse a la base de datos en cuestión, en este caso la base de datos **SAMPLE**, tras lo cual se puede otorgar DBADM a un usuario. Para otorgar la autoridad DBADM y cualquier otra autoridad a nivel de la base de datos se requiere ser SECADM.

Tenga en cuenta que un DBADM no puede crear espacios de tabla, aun cuando estos son objetos dentro de la base de datos, porque los espacios de tablas se relacionan con contenedores (discos) y agrupaciones de memoria intermedia (memoria) los cuales son recursos físicos del sistema. Estas operaciones solo pueden ser hechas por un SYSADM.

La Figura 10.5, tomada desde el Centro de Información DB2, nos muestra las diferentes autoridades a nivel de la base de datos y las funciones que pueden realizar.

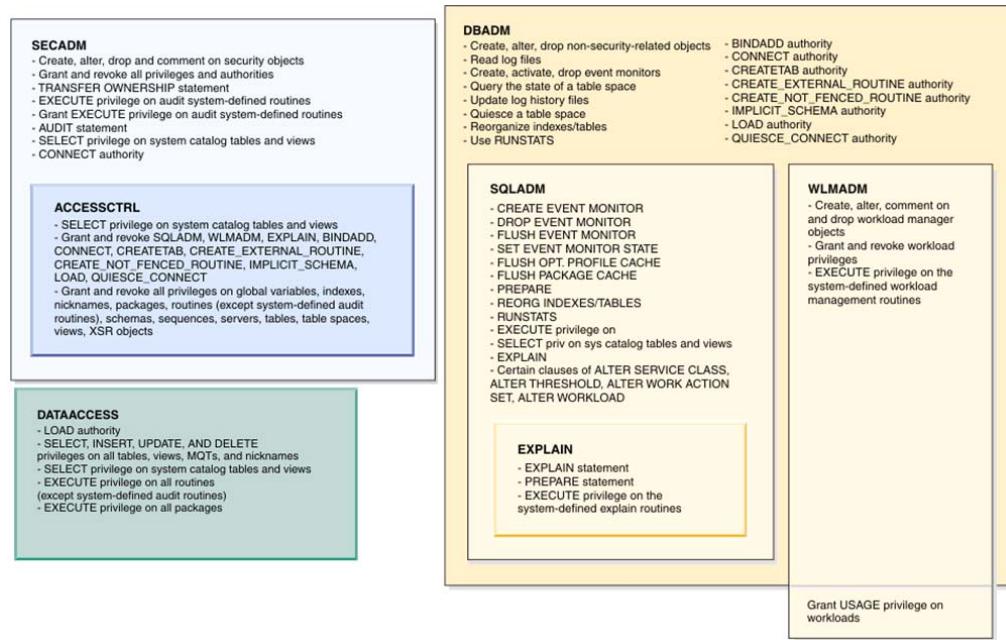


Figura 10.5 – Autoridades a nivel de base de datos y sus funciones

nuevo en
V9.7

Nota:

En DB2 9.7, para garantizar una mejor privacidad de los datos y el cumplimiento de normativas, el modelo de autorización ha sido actualizado de modo de separar claramente las tareas del administrador del sistema, del administrador de la base de datos y del administrador de la seguridad.

En general, el ámbito funcional de varias autoridades ha sido reducido en comparación con las anteriores versiones de DB2. Por ejemplo, un SYSADM ya no tiene acceso a los datos de ninguna base de datos. Un DBADM tampoco tiene acceso a los datos de las bases que administra. Por otro lado, SECADM puede realizar más funciones tales como otorgar y revocar autoridades a usuarios, roles y grupos.

Se han creado además nuevas autoridades para permitir un control más granular de la seguridad del sistema, lo que minimiza el riesgo de exponer los datos, ya que se otorga a los usuarios solo el acceso que requieren para realizar su trabajo.

10.2.2.3 Como hacer que SYSADM y DBADM trabajen como en las versiones previas de DB2

Si se desea que un SYSADM se comporte como lo hacía en las versiones anteriores a DB2 9.7, hay que considerar dos casos:

- Si el SYSADM es el creador de la base de datos, entonces automáticamente recibe las autoridades de DATAACCESS, ACCESSCTRL, SECADM y DBADM para esa base de datos, por lo tanto el SYSADM tendrá las mismas habilidades que en las versiones anteriores a DB2 9.7.
- Si el SYSADM no es el creador de la base de datos, entonces, para obtener las mismas capacidades que en las versiones anteriores de DB2 (excepto SECADM); un SECADM debe ejecutar una sentencia de GRANT DBADM con DATAACCESS y ACCESSCTRL (lo cual es por omisión) al SYSADM en la base de datos en cuestión.

Algunos casos a considerar para un SECADM:

- El SECADM por omisión es el creador de la base de datos.
- Si un usuario con autoridad de SECADM le otorga SECADM a un usuario que tenga autoridad SYSADM, entonces el SYSADM puede otorgar SECADM a otros usuarios.
- Si un usuario con autoridad SECADM le otorga DBADM a otro usuario, el DBADM también recibe DATAACCESS y ACCESSCTRL por omisión.

Si se está migrando una base de datos DB2 9.5, las capacidades del SYSADM y del DBADM no cambiarán, dado que DB2 automáticamente le otorga DBADM, DATAACCESS y ACCESSCTRL al grupo SYSADM durante la migración. DB2 también otorgará DATAACCESS y ACCESSCTRL a cada identificador de autorización que sea DBADM al momento de la migración. Adicionalmente, DB2 otorgará automáticamente la autoridad de SECADM al identificador de usuario que esté realizando la migración si es que no existiera una identificador de autorización de tipo USER que sea SECADM en la base de datos. El SYSADM perderá su habilidad implícita de otorgar o revocar DBADM y SECADM lo cual ahora solo puede ser realizado por un SECADM.

10.2.3 Roles

Los roles permiten al administrador de seguridad asignar privilegios o autoridades a varios usuarios o grupos conjuntamente. En ese aspecto, los roles son parecidos a los grupos, pero están definidos dentro de DB2, y por lo tanto este hecho da algunas ventajas. Por ejemplo, los privilegios y autoridades otorgados a los roles siempre se usa al crear objetos tales como vistas o activadores, lo que no ocurre en el caso de los grupos. Por otro lado, no se pueden asignar a un rol autoridades a nivel de instancia, tal como SYSADM, solo privilegios y autoridades a nivel de la base de datos; mientras que a un grupo se le pueden asignar todos los privilegios y autoridades que se deseen.

Para trabajar con roles se requiere completar los siguientes pasos:

12. Un administrador de seguridad (SECADM) debe crear primero un rol usando un comando tal como

```
CREATE ROLE TESTER
```

13. A continuación, un DBADM debe otorgar privilegios o autoridades al rol. Por ejemplo para otorgar el privilegio de SELECT sobre las tablas STAFF y DEPT en la base de datos SAMPLE al rol **TESTER**, se debe ejecutar:

```
GRANT SELECT ON TABLE STAFF TO ROLE TESTER  
GRANT SELECT ON TABLE DEPT TO ROLE TESTER
```

14. Ahora el administrador de seguridad les otorga el rol **TESTER** a los usuarios JORGE y ADRIAN:

```
GRANT ROLE TESTER TO USER JORGE, USER ADRIAN
```

15. Si luego ADRIAN dejara el departamento de pruebas, el administrador de seguridad revocaría el rol **TESTER** del usuario ADRIAN:

```
REVOKE ROLE TESTER FROM USER ADRIAN
```

10.3 Consideraciones cuando se usan privilegios a grupos

Si se decide utilizar grupos en lugar de roles, se debe tener en cuenta lo siguiente:

- Cuando a un grupo se le otorgan privilegios, los miembros de ese grupo obtienen privilegios implícitos heredados de la pertenencia al grupo.
- Cuando se remueve a un usuario de un grupo, éste pierde los privilegios implícitos del grupo, pero mantiene todos aquellos que le fueron otorgados explícitamente. Estos privilegios explícitos deben ser revocados explícitamente del usuario.

10.4 El grupo PUBLIC

DB2 define internamente un grupo denominado PUBLIC. Cualquier usuario identificado ya sea por el sistema operativo o por el servicio de autenticación de la red, es implícitamente miembro del grupo PUBLIC. Cuando se crea una base de datos, ciertos privilegios son otorgados al grupo PUBLIC automáticamente:

- CONNECT,
- CREATETAB,
- IMPLICIT SCHEMA,
- BINDADD

Para una mayor seguridad, recomendamos revocar todos los privilegios del grupo PUBLIC tal como se ve a continuación:

```
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 Las sentencias GRANT y REVOKE

Las sentencias GRANT y REVOKE son parte del estándar SQL, y son usadas para otorgar y revocar privilegios a un usuario, a un grupo o a un rol. El usuario que ejecuta esta sentencia debe al menos tener la autoridad de ACCESSCTRL. A continuación algunos ejemplos de estas sentencias:

Para otorgar el privilegio SELECT sobre la tabla T1 al usuario USER1:

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

Para otorgar todos los privilegios sobre la tabla T1 al grupo GROUP1:

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

Para revocar todos los privilegios sobre la tabla T1 del grupo GROUP1:

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

Para otorgar el privilegio EXECUTE sobre el procedimiento p1 al usuario USER1:

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

Para revocar el privilegio EXECUTE sobre el procedimiento p1 del usuario USER1:

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

10.6 Chequeo de las autorizaciones y privilegios

La forma más sencilla de chequear las autorizaciones y privilegios es mediante el Centro de control. La *Figura 10.6* nos muestra como lanzar el diálogo de Privilegios de tabla para la tabla **EMPLOYEE** desde el Centro de control.

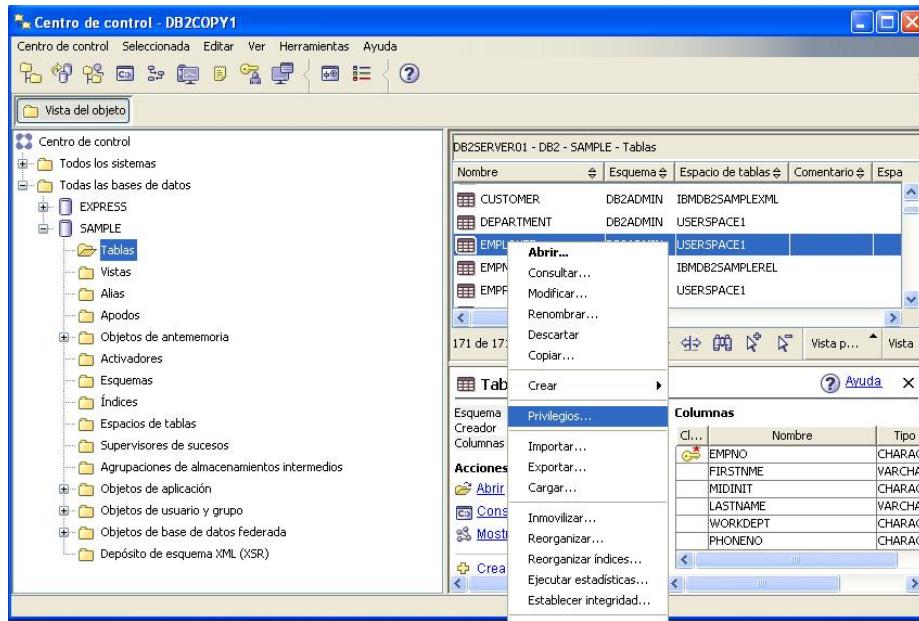


Figura 10.6 – Lanzando el diálogo de privilegios de tabla

Tal como se ve en la *Figura 10.6*, primero se selecciona la tabla deseada, se le da clic con el botón derecho del ratón y se selecciona Privilegios. Una vez seleccionado, se abre la ventana de Privilegios de tabla, como puede verse en la *Figura 10.7*. En esta figura también se explican los diferentes campos y elementos del cuadro de diálogo.

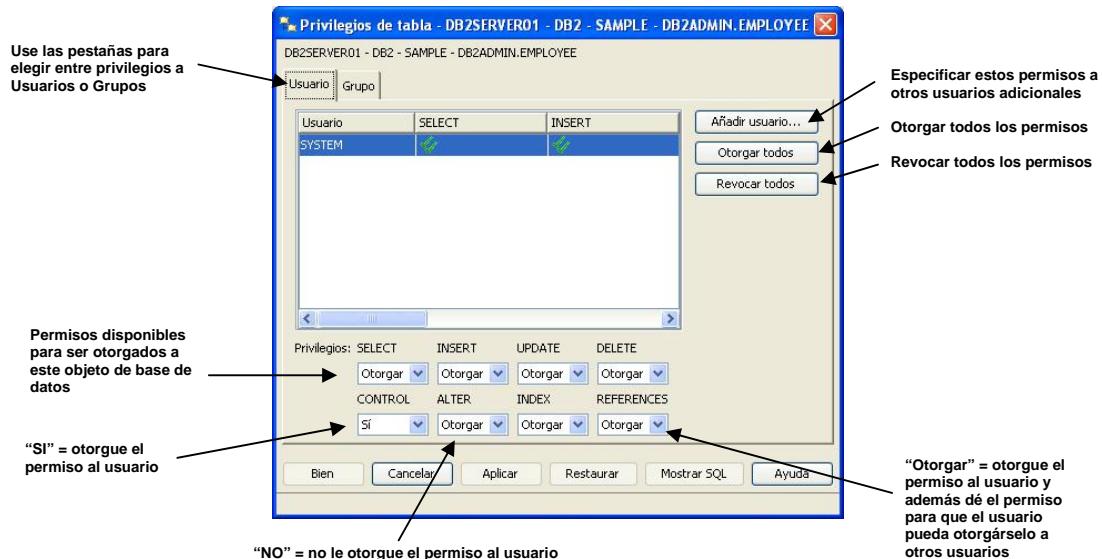


Figura 10.7 – El cuadro de diálogo de los privilegios de tabla

Alternativamente, se puede consultar las vistas SYSCAT del catálogo DB2 que contienen información de autorización. Por ejemplo, si se desea saber si el usuario **DB2ADMIN** tiene el privilegio SELECT sobre la tabla T2, y además se desea saber quien le otorgó ese privilegio, se puede ejecutar una consulta como esta:

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

GRANTOR	GRANTEE	SELECTAUTH
JMARTEL	DB2ADMIN	Y

En el ejemplo anterior el usuario **JMARTEL** le otorgó el privilegio SELECT al usuario **DB2ADMIN**.

10.7 Seguridad extendida en Windows

Para prevenir el acceso a través del sistema operativo Windows a los archivos y directorios de DB2 (como ser aquellos en que DB2 almacena información de la instancia), DB2 activa la seguridad extendida por omisión al momento de la instalación. La seguridad extendida crea dos grupos:

- DB2ADMNS: Este grupo conjuntamente con el de los administradores locales tiene completo acceso a todos los objetos DB2 a través del sistema operativo.
- DB2USERS: Este grupo tiene acceso de lectura y ejecución a todos los objetos DB2 a través del sistema operativo.

nuevo en
V9.7

Con DB2 9.7 los miembros del grupo DB2ADMNS tendrán automáticamente autoridad SYSADM en DB2 si la seguridad extendida está activada y si al parámetro de configuración de la base de datos SYSADM_GROUP no se le ha asignado un valor.

10.8 Resumen

Este capítulo ha cubierto los aspectos de seguridad en DB2, comenzando con una detallada discusión entre la importancia de autenticar y autorizar. A partir de allí vimos los diferentes niveles de autoridad que proveen seguridad a la instancia y a la base de datos.

A continuación desarrollamos el Nuevo concepto de roles y como pueden ser usados en nuestro beneficio en relación a la seguridad, además de ver las limitaciones de establecer la seguridad utilizando grupos. Hablamos en particular del grupo PUBLIC y dimos algunas sugerencias acerca de como asegurarlo de modo que los usuarios generales notuvieran acceso al servidor de datos.

Adicionalmente examinamos las sentencias GRANT y REVOKE y finalmente vimos como utilizar el Centro de control y las tablas del catálogo del sistema para chequear los niveles de autorización y privilegios.

10.9 Ejercicios

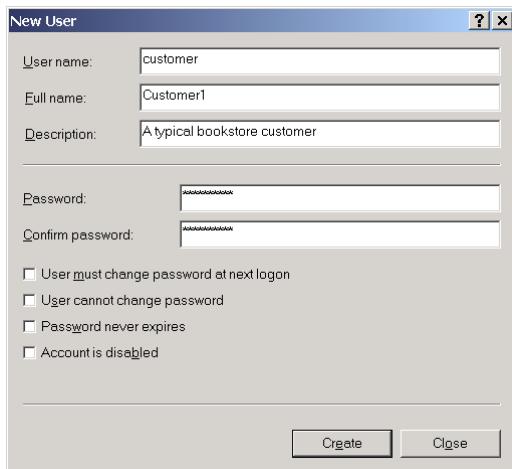
Hasta ahora hemos estado usando la cuenta del administrador de la instancia (SYSADM) para emitir todos los comandos de la base de datos. Esta cuenta tiene amplio acceso a todos los utilitarios, datos y objetos de la base de datos. Por lo tanto es muy importante resguardar esta cuenta para evitar pérdidas accidentales o deliberadas de datos. En la mayoría de los casos crearemos varias cuentas de usuarios diferentes o grupos con un conjunto limitado de permisos. En este ejercicio crearemos una nueva cuenta para luego asignarle privilegios específicos.

Parte 1 – Trabajando con privilegios

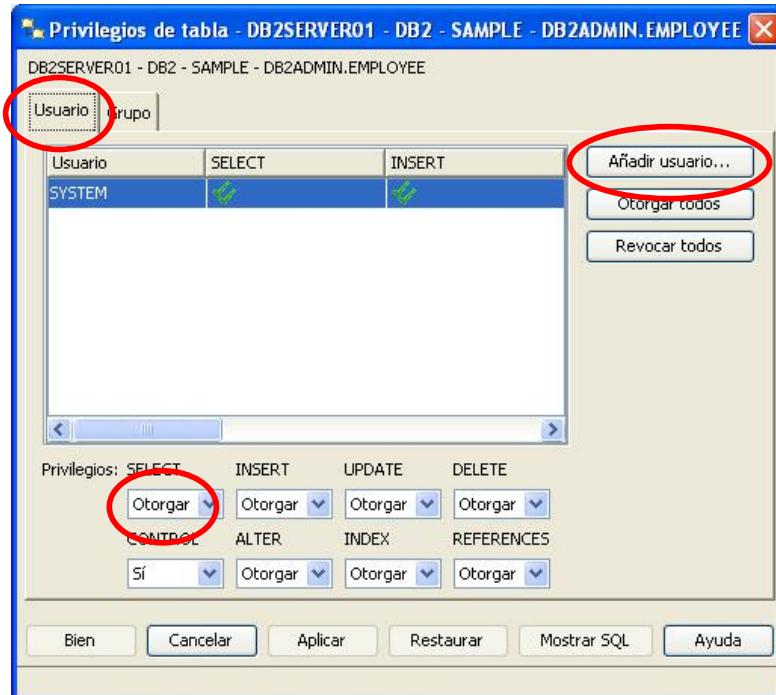
En esta parte del ejercicio practicaremos como otorgar y revocar privilegios a los usuarios mediante el Centro de control.

Procedimiento

16. Abra la consola de *Manejo del Computador de Windows* dando clic con el botón derecho del ratón sobre el ícono de *Mi computador* en el escritorio y seleccione el ítem *Manejar* del menú.
17. Expanda la selección de *Herramientas del sistema* en el árbol del panel de la izquierda de la ventana y luego expanda la carpeta de *Usuarios y grupos locales*. Dé clic con el botón derecho del ratón sobre la carpeta de *Usuarios* y seleccione el ítem *Usuario nuevo*.
18. En la ventana de diálogo de *Usuario Nuevo*, entre la siguiente información: en el campo *Nombre del usuario* entre *customer* y en el campo *Nombre completo* entre *Customer1*. En el campo *Descripción* entre *Un típico cliente de librería*. En los campos *Clave* y *Confirmación* ingrese *ibmdb2ibm*. Remueva la marca de la opción *El usuario debe cambiar la clave en la próxima conexión*, y de clic en el botón *Crear* para crear el nuevo usuario, y luego en el botón *Cerrar* para cerrar la ventana de diálogo.



19. Abra el Centro de control, y elija la vista avanzada. Para cambiarse a la vista avanzada, seleccione *Herramientas ->Personalizar* en el menú del Centro de control y seleccione la opción *Avanzada*. Luego dé clic en el botón *Bien*.
20. Expanda el árbol de objetos del Centro de control en el panel izquierdo de la siguiente forma: *Todas las bases de datos -> EXPRESS -> Tablas*.
21. Otorgue los privilegios requeridos al usuario recientemente creados. De la lista de tablas en la base de datos **EXPRESS**, dé clic con el botón derecho del ratón en la tabla **CUSTOMERS** y luego seleccione *Privilegios* para desplegar la ventana de diálogo de *Privilegios de tabla*.
22. Dé clic en el botón *Agregar usuario* y seleccione el usuario **customer** que acaba de crear. Dé clic en el botón *Bien* para cerrar el cuadro de diálogo de *Añadir usuario*.
23. Notará que el usuario **customer** ha sido agregado a la lista de usuarios, pero no tienen ningún privilegio asignado. Para otorgar los privilegios SELECT, INSERT, UPDATE, y DELETE al usuario, cambie cada lista desplegable a Si. Un usuario Internet debe ser capaz de ver/agregar/actualizar/eliminar los datos de su cuenta. No le otorgamos otros permisos ya que no los necesita. Dé clic en el botón Bien para cerrar la ventana de diálogo de *Privilegios de tabla* y aceptar los cambios hechos.



24. Repita los pasos 6 a 8 para las tablas ***BOOKS*** y ***SALES***. Para la tabla ***BOOKS***, solo otorgue el privilegio SELECT ya que el cliente no debería ser capaz de modificar nada en el inventario de la tienda. Para la tabla ***SALES***, solo otorgue los privilegios SELECT e INSERT, el cliente NO debe tener los privilegios DELETE o UPDATE dado que solo los empleados de la tienda deberían modificar las transacciones de ventas.
25. Conéctese a la base de datos usando el identificador de usuario ***customer*** creado más arriba usando la Ventana de Mandatos DB2 tal como sigue:

```
db2 connect to express user customer using ibmdb2ibm
```

Trate de leer datos desde la tabla ***customers*** con una sentencia SELECT. ¿Qué sucede? Trate de ejecutar sentencias DELETE o UPDATE contra la tabla ***SALES***. ¿Qué sucede?

Parte 2 – Trabajando con las autoridades SYSADM, DBADM y SECADM

En esta parte del ejercicio practicaremos como asignar las autoridades SYSADM y DBADM y entenderemos como trabajan esas autoridades.

Procedimiento

1. Siga los mismos pasos de la parte 1 para crear un nuevo usuario: ***mysysadm***
2. Cree el grupo de Windows ***mysysadmgrp***. Siga los mismos pasos para crear un usuario, pero en lugar de dar clic con el botón derecho del ratón en la carpeta *Usuarios*, hágalo en la carpeta *Grupos* y elija *Nuevo grupo*. Como nombre del grupo, ingrese ***mysysadmgrp***. En la sección de *Miembros*, de clic en *Añadir* para agregar un nuevo miembro e ingrese ***mysysadm***. Dé clic en el botón *Verificar nombres* para confirmar que ha ingresado correctamente el nombre del miembro. Si es así, dé clic en *Crear* y luego en *Cerrar*.
3. Hasta ahora ha creado el usuario ***mysysadm*** y el grupo ***mysysadmgrp*** al cual pertenece el usuario ***mysysadm***. Todo esto ha sido hecho en el sistema operativo Windows. Ahora tenemos que informar a DB2 que queremos que el grupo ***mysysadmgrp*** sea el grupo SYSADM por medio de este comando en la Ventana de mandatos DB2:

```
db2 update dbm cfg using SYSADM_GROUP mysysadmgrp
```

Dado que el parámetro SYSADM_GROUP no es dinámico, se debe detener y luego arrancar la instancia. La opción *force* en **db2stop** garantiza que todas las conexiones serán removidas antes del **db2stop**.

```
db2stop force  
db2start
```

4. Conéctese a la base de datos SAMPLE con el usuario ***mysysadm*** desde la Ventana de mandatos DB2 y ejecute una sentencia SELECT * sobre la tabla STAFF. Tenga en cuenta que debe usar el esquema correcto que fue usado al crear la tabla. En nuestro ejemplo usaremos ***jmartel*** como esquema.

```
db2 connect to sample user mysysadm using ibmdb2ibm  
db2 select * from jmartel.staff
```

Usted debería recibir un mensaje como este. ¿Por qué? ¿Acaso no es SYSADM?

```
SQL0551N "MYSYSADM" does not have the required authorization or  
privilege to perform operation "SELECT" on object "JMARTEL.STAFF".  
SQLSTATE=42501
```

A partir de DB2 9.7, SYSADM no obtiene la autoridad DBADM por omisión. Esa es la razón por la cual ha recibido el mensaje de error.

5. Conéctese a la base de datos **SAMPLE** con el usuario Windows con el que ha creado la misma. En este ejemplo, **JMARTEL** es ese usuario. A continuación otorgue DBADM sin DATAACCESS al usuario ***mysysadm***, y trate de Nuevo la

sentencia SELECT sobre STAFF con el usuario *mysysadm*. ¿Funcionó ahora?
¿Por qué?

```
db2 connect to sample user jmartel using ibmdb2ibm
db2 grant dbadm without dataaccess on database to user mysysadm
db2 connect to sample user mysysadm using ibmdb2ibm
db2 select * from jmartel.staff
```

Como habrá visto, todavía recibe el mismo error aun cuando se le ha otorgado DBADM a *mysysadm*. Esto es lo esperado, dado que hemos incluido la cláusula WITHOUT DATAACCESS lo que significa que la autoridad DATAACCESS no estaba incluida, por lo tanto *mysysadm* todavía no tiene la autoridad necesaria para tener acceso a los datos. Esto le demuestra como puede restringir el acceso a los datos al DBADM.

6. Otorguemos ahora DATAACCESS a *mysysadm* y tratemos el SELECT nuevamente.

```
db2 connect to sample user jmartel using ibmdb2ibm
db2 grant DATAACCESS on database to user mysysadm
db2 connect to sample user mysysadm using ibmdb2ibm
db2 select * from jmartel.staff
```

¡Ahora sí que el SELECT debería funcionar! Este ejercicio le ha demostrado el nuevo comportamiento de SYSADM y DBADM a partir de DB2 9.7. La idea principal que debe extraer de este ejercicio es que ahora hay una separación entre el acceso a los datos y lo que SYSADM y DBADM son capaces de hacer.

7. Cambie de vuelta el valor de SYSADM_GROUP a NULL de modo que el grupo de Administradores locales y la cuenta LocalSystem vuelvan a ser SYSADM:

```
db2 update dbm cfg using sysadm_group NULL
db2stop force
db2start
```

11

Capítulo 11 – Respaldo y recuperación

En este capítulo veremos como DB2 va registrando los cambios en las bases de datos (*logging*), como hacer una copia total o parcial de una base de datos usando el utilitario BACKUP y como recuperar los datos mediante el utilitario RESTORE.

Nota:

Para mayor información sobre la anotación cronológica, respaldo y recuperación, vea este video (en inglés): <http://www.channeldb2.com/video/video/show?id=807741:Video:4282>

11.1 Anotación cronológica (*logging*)

Cuando usted trabaja con un editor de texto, lo más probable es que de tanto en tanto quiera asegurarse que su documento está resguardado, por lo que seguramente le dará clic al botón de *Salvar*. En el mundo de las bases de datos la sentencia COMMIT hace justamente eso; cada vez que se ejecuta una sentencia COMMIT, usted se asegura que cualesquiera cambios se hayan introducido en los datos, estos han sido salvados en algún sitio.

Algo similar ocurre al trabajar con un documento de texto, algunas veces se ve en la esquina inferior izquierda un breve mensaje que dice “salvado automático”. En el mundo de las bases de datos, esto ocurre también, dado que toda operación contra los datos ya sea un UPDATE, INSERT o DELETE, se salvará en alguna parte a medida que se ejecuta.

Aquel “alguna parte” de los párrafos precedentes se refiere a los archivos de anotación cronológica (*logs*) de la base de datos. Los archivos de anotación cronológica se almacenan en disco y son usados para registrar las acciones de las transacciones. Si hubiera una caída del sistema o de la base de datos, estos archivos de anotación cronológica se usan para reproducir y rehacer las transacciones completadas durante el proceso de recuperación.

La *Figura 11.1* nos da en forma gráfica una visión general de lo que sucede desde el punto de vista de la anotación cronológica cuando se está trabajando con una base de datos.

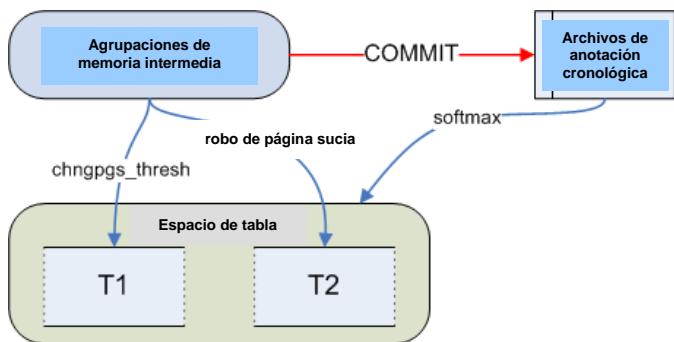


Figura 11.1 – Anotación cronológica

En la *Figura 11.1*, podemos ver un espacio de tabla y los archivos de anotación cronológica. Ambos residen en disco, aunque le recomendamos que no sea en el mismo disco. Cuando una operación de UPDATE toma lugar, las páginas de la(s) fila(s) en cuestión son traídas a la agrupación de almacenamientos intermedios (memoria). Los cambios se efectúan en la agrupación de almacenamientos intermedios y los valores anterior y nuevo se almacenan en los archivos de anotación cronológica, algunas veces en forma inmediata y otras veces cuando el almacenamiento intermedio del archivo de anotación cronológica se llena. Si después del UPDATE se ejecuta un COMMIT, el valor anterior y el nuevo son grabados en el archivo de anotación cronológica inmediatamente. Este proceso se repite para muchos otras operaciones SQL que se ejecutan en la base de datos. Las páginas en la agrupación de almacenamientos intermedios son “exteriorizadas”, o sea grabadas en disco en el espacio de tablas, solo cuando se cumplen ciertas condiciones, como por ejemplo que se alcance el límite de páginas modificadas especificado en el parámetro CHNGPGS_THRES. Este parámetro indica el porcentaje de páginas “sucias”, o sea páginas que contienen cambios, que la agrupación de almacenamientos intermedios puede contener sin que éstas sean exteriorizadas.

Desde el punto de vista del rendimiento, no tiene ningún sentido realizar dos grabaciones en el disco para cada operación COMMIT: Una para grabar los archivos de anotación cronológica y otra para grabar en el espacio de tabla; por ello la exteriorización de los datos hacia el espacio de tabla en el disco solo ocurre cuando se alcanzan límites tales como los del parámetro CHNGPGS_THRES.

11.2 Tipos de archivos de anotación cronológica

Existen dos tipos de archivos de anotación cronológica:

- **Archivos de anotación cronológica primarios**

Estos están predefinidos en el disco y el número de ellos disponible se determina mediante el parámetro de configuración de la base de datos, LOGPRIMARY.

- **Archivos de anotación cronológica secundarios**

Estos son definidos dinámicamente a medida que DB2 los va necesitando. El parámetro de configuración de la base de datos LOGSECOND determina el número máximo de estos archivos. La definición dinámica de estos archivos es costosa, por lo tanto, para las operaciones del día a día, es preferible mantenerse con los archivos primarios. Los archivos secundarios son eliminados una vez que se terminan todas las conexiones a la base de datos.

Se podría generar estos archivos de anotación cronológica *ad infinitum* si se le asigna -1 al parámetro LOGSECOND; sin embargo es algo que no se recomienda, ya que se podría llenar todo el espacio en disco del sistema.

11.3 Tipos de anotación cronológica

Hay dos tipos de anotación cronológica: anotación cronológica circular (por omisión) y anotación cronológica con archivo.

11.3.1 Anotación cronológica circular

Por omisión se elige la anotación cronológica circular, y queda activada cuando el valor de los parámetros de configuración de la base de datos LOGARCHMETH1 y LOGARCHMETH2 tienen el valor OFF. Estos parámetros indican el método a usar para archivar las anotaciones cronológicas, pero si ambos tienen el valor OFF, eso significa que no se desea archivar las anotaciones cronológicas, que es la forma en que la anotación circular trabaja. La *Figura 11.2* nos muestra un ejemplo de anotación cronológica circular.

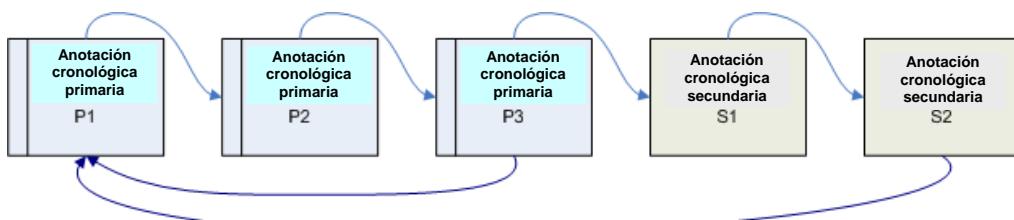


Figura 11.2 – Trabajando con anotaciones cronológicas primarias y secundarias

En la *Figura 11.2* hay tres anotaciones cronológicas primarias, por lo tanto podemos asumir que el valor del parámetro LOGPRIMARY es 3. Para mantener el ejemplo simple, solo se está ejecutando una transacción. A medida que ésta procesa, la anotación cronológica P1 comienza a llenarse, y luego sigue P2. Si ocurre un COMMIT y los datos son más tarde exteriorizados de la agrupación de almacenamiento intermedio al espacio de tabla en disco, entonces P1 y P2 pueden ser reutilizados, ya que la información que contienen ya no es necesaria para una recuperación en caso de caída (*crash recovery* - hablaremos en detalle de este tema más adelante en este capítulo). Si por otro lado, la transacción es tan larga que usa P1, P2, P3, y aun requiere más espacio de anotación

cronológica debido a que no ha ejecutado un COMMIT o bien sus datos no se han exteriorizado aun, entonces se define dinámicamente una anotación cronológica secundaria (S1 en la figura). Si la transacción continua, se van definiendo más anotaciones cronológicas secundarias hasta que se alcanza el valor máximo especificado en el parámetro LOGSECOND. Si aun se necesitan más anotaciones cronológicas, se desplegaría un mensaje indicando que se ha llenado la anotación cronológica y la transacción es vuelta atrás (*rolled back*). If still more logs are needed, an error message indicating a log full condition is reached will be returned to the user, and the transaction will be rolled back. Otra posibilidad es usar el parámetro de configuración DB2 BLK_LOG_DSK_FUL para seguir grabando las anotaciones cronológicas cada 5 minutos pero dejando que algunas transacciones se cuelguen, lo que le da al administrador de bases de datos algo de tiempo para encontrar espacio de forma que la transacción pueda continuar.

11.3.2 Anotación cronológica con archivo

En la anotación cronológica con archivo, también conocida como anotación cronológica retenida, las anotaciones cronológicas no se reutilizan, sino que se archivan, ya sea en línea o fuera de línea. Las anotaciones cronológicas archivadas en línea se mantienen junto con las anotaciones cronológicas activas ya que se necesitan para recuperaciones en caso de caídas. Las anotaciones cronológicas archivadas fuera de línea son trasladadas a otro medio como ser cintas, y esto puede ser hecho con rutinas de USEREXIT, con el *Tivoli Storage Manager*, u otro producto para archivar de algún otro proveedor.

Para activar las anotaciones cronológicas con archive, se debe asignar un valor al parámetro de configuración de la base de datos LOGARCHMETH1 o al parámetro LOGARCHMETH2 (o a ambos), que no sea OFF. Otra forma de activarlas es poner el valor RECOVERY en el parámetro de configuración LOGRETAIN. Esto hace que automáticamente el parámetro LOGARCHMETH1 tenga el valor LOGRETAIN. Sin embargo el parámetro LOGRETAIN está obsoleto y principalmente está aun allí para mantener la compatibilidad con versiones anteriores de DB2.

En los sistemas de producción normalmente se usa la anotación cronológica con archivo, ya que gracias a que las anotaciones cronológicas se archivan, es posible recuperar una base de datos a cualquier punto en el tiempo tan antiguo como la anotación cronológica archivada más antigua. Un administrador de bases de datos, usando la anotación cronológica con archivo, es capaz de recuperar la base de datos de errores causados por los usuarios. Por ejemplo, supongamos que por varios días un usuario, sin advertirlo, ha estado ejecutando una transacción incorrecta. Al detectarse el problema, el administrador de bases de datos puede recuperar el sistema a un punto en el tiempo anterior al comienzo del problema. Sin embargo, podrían necesitarse algunas acciones manuales antes que la transacción pueda ejecutarse nuevamente.

Si se desea hacer recuperación en avance o respaldo en línea, es necesario contar con las anotaciones cronológicas con archivo. En la *Figura 11.3* se detalla el proceso de las anotaciones cronológicas con archivo.

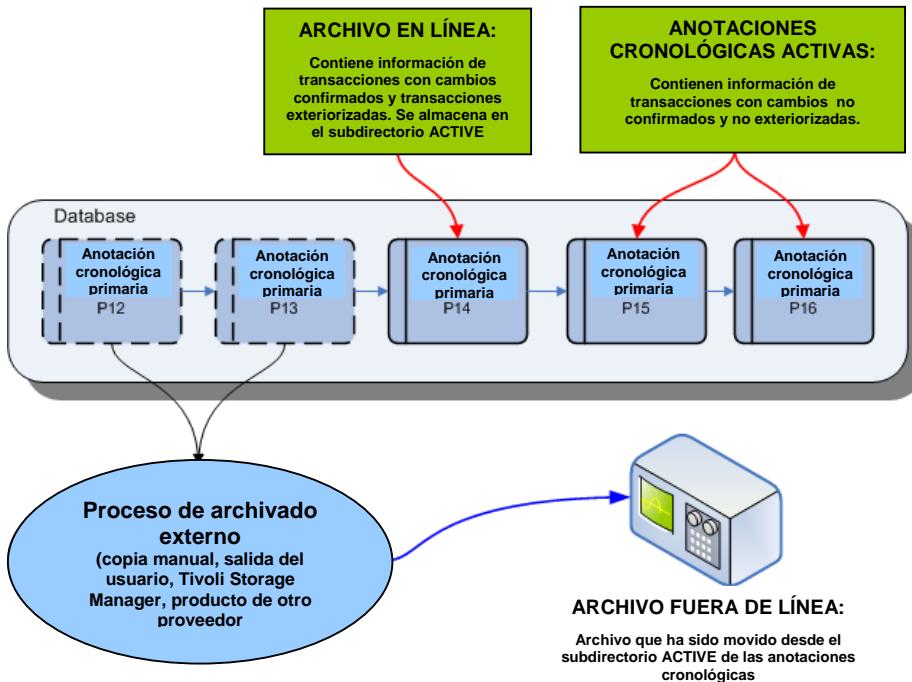


Figura 11.3 – Anotaciones cronológicas con archivo

11.4 Manejo de la anotación cronológica desde el Centro de control

Es posible configurar las anotaciones cronológicas de la base de datos desde el centro de control; simplemente de clic con el botón derecho del ratón sobre la base de datos en cuestión y seleccione *Configurar anotación cronológica de base de datos* tal como se ve en la *Figura 11.4*.

204 Conociendo al DB2 Express-C

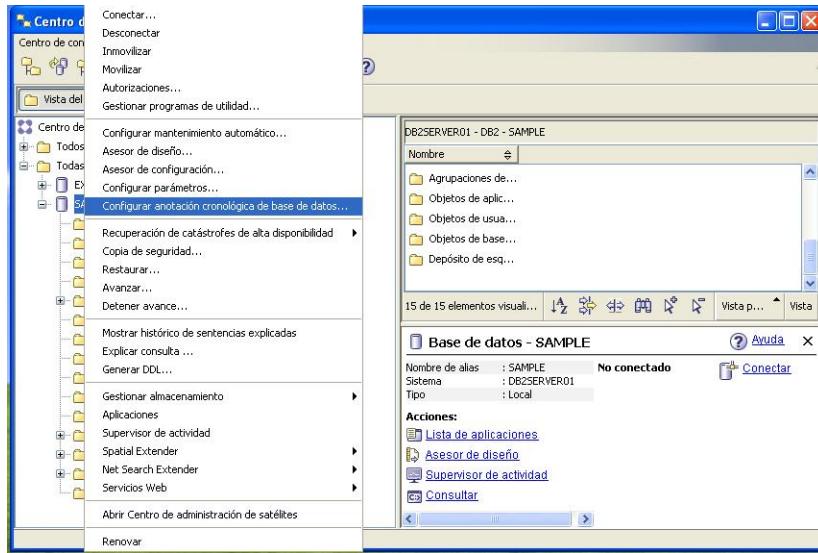


Figura 11.4 – Configurar anotación cronológica de base de datos desde el Centro de control

La Figura 11.5 muestra el Asistente para configurar anotación cronológica de base de datos, donde se puede elegir anotación cronológica circular o anotación cronológica con archivo (Archivar anotación cronológica).

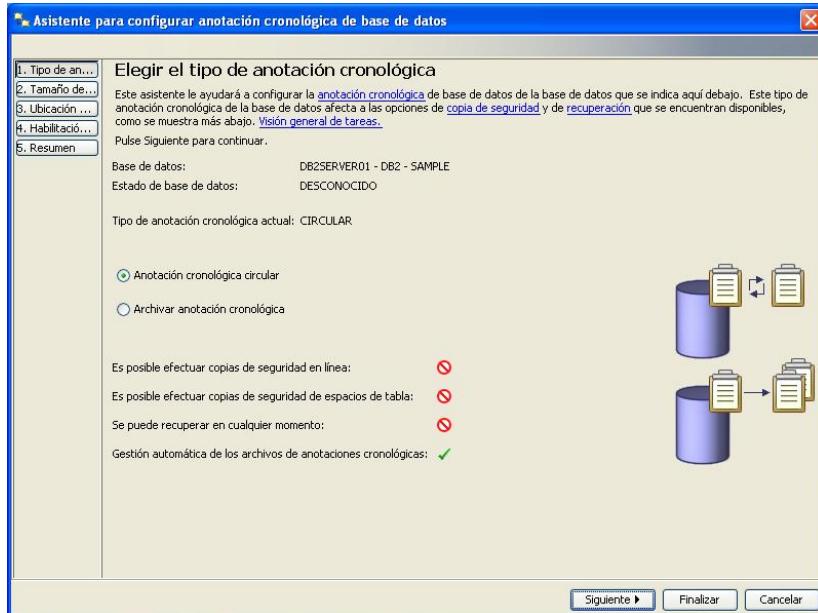


Figura 11.5 – Asistente para configurar anotación cronológica de base de datos

11.5 Parámetros para la anotación cronológica

Hay un número de parámetros de DB CFG relacionados con la anotación cronológica. La Tabla 11.1 lista los principales.

Parámetro	Descripción
logbufsz	La cantidad de memoria a usar como almacenamiento intermedio para los registros de anotación cronológica antes que sean grabados en disco.
logfilsz	El tamaño de cada anotación cronológica en páginas de 4KB.
logprimary	El número de anotaciones cronológicas primarias de tamaño logfilsz que se crearán.
logsecond	El número de anotaciones cronológicas secundarias que se crearán y serán usadas para recuperación en caso de ser necesarias.
newlogpath	Las anotaciones cronológicas activas y las archivadas en línea se almacenan por omisión en el directorio de la base de datos, bajo el subdirectorio SQLOGDIR. Es posible cambiar esta ubicación dándole a este parámetro el valor del directorio o disco donde se desea ubicar estas anotaciones cronológicas.
mirrorlogpath	Para estar a salvo de la pérdida de las anotaciones cronológicas ubicadas en el directorio anterior (primario), ya sea por una falla del disco o un borrado accidental, es posible definir aquí un directorio secundario (<i>mirror</i>) donde DB2 mantendrá un conjunto idéntico de anotaciones cronológicas
logarchmeth1 / logarchmeth2	Especifica una ubicación diferente de aquella para las anotaciones cronológicas activas, que se usarán para almacenar las anotaciones cronológicas archivadas. Si se especifican ambos parámetros, cada anotación cronológica es archivada dos veces, lo que significa que se pueden tener dos copias de las anotaciones cronológicas archivadas, cada una de ellas en una ubicación distinta. Los posibles valores para estos parámetros son: OFF (lo que significa que se usará anotación cronológica circular), LOGRETAIN, USEREXIT, DISK, TSM, VENDOR
loghead	El nombre de la anotación cronológica que está active ahora
softmax	Pone un límite al número de anotaciones cronológicas a usarse en caso de una recuperación por caída.

overflowlogpath	Especifica la ubicación donde DB2 podrá encontrar las anotaciones cronológicas requeridas para una operación de recuperación en avance. Es similar a la opción OVERFLOW LOG PATH del comando ROLLFORWARD.
blk_log_dsk_ful	Se utiliza para prevenir errores de disco lleno cuando DB2 es incapaz de crear una nueva anotación cronológica en el directorio de anotaciones cronológicas activas. En lugar de dar error, DB2 tratará de crear la anotación cronológica cada cinco minutos hasta que lo logre. Aquellas sentencias SQL sin bloqueo o solo de lectura continuarán su proceso sin ser afectadas.
max_log	Porcentaje máximo de una anotación cronológica que puede ser utilizado por una sola transacción.
num_log_span	Número de anotaciones cronológicas activas por cada unidad de trabajo activa
mincommit	Número de confirmaciones de escritura a agrupar antes de grabar en disco

Tabla 11.1 – Parámetros para anotación cronológica

11.6 Respaldo de base de datos

El comando BACKUP permite tomar una copia fotográfica de la base de datos en el momento en que es ejecutado. En su forma más simple el comando es:

```
BACKUP DATABASE <nombre_de_la_base_de_datos> [ TO <directorio> ]
```

La mayoría de los comandos y utilitarios pueden ejecutarse en línea o fuera de línea. En línea significa que otros usuarios pueden estar conectados y estar ejecutando operaciones sobre la base de datos al mismo tiempo que nuestro comando se está ejecutando. Fuera de línea implica que ningún otro usuario puede estar conectado a la base de datos mientras nuestro comando ejecuta. Para hacer que la operación ejecute en línea, se debe añadir la palabra clave ONLINE a la sintaxis del comando, de lo contrario el comando asumirá que deseamos ejecutarlo fuera de línea.

Por ejemplo, supongamos que deseamos tomar un respaldo a la base de datos **SAMPLE** en el directorio C:\BACKUPS , para ello se emite este comando desde la Ventana de mandatos DB2 en Windows o desde el *shell* de Linux:

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

Nótese que el directorio C:\BACKUPS debe existir antes de ejecutar el comando. Además hay que asegurarse que no hay conexiones a la base de datos al momento de ejecutar el comando, pues de lo contrario recibiremos un mensaje de error, ya que un respaldo fuera de línea no puede ejecutarse mientras haya conexiones.

Para ver si hay conexiones a las bases de datos de una instancia, se utiliza este comando desde la Ventana de mandatos DB2 en Windows o desde el *shell* de Linux:

```
db2 list applications
```

Para terminar por la fuerza todas las conexiones a todas las bases de datos de una instancia, se utiliza este comando desde la Ventana de mandatos DB2 en Windows o desde el *shell* de Linux:

```
db2 force applications all
```

Claro que no sería muy apropiado ejecutar este último comando en un ambiente de producción con muchos usuarios, a menos que usted quiera hacerse famoso y recibir unas cuantas llamadas telefónicas no muy amables de sus colegas. Tenga en cuenta que este comando ejecuta en forma asíncrona, lo que significa que si trata de ejecutar el comando de respaldo inmediatamente después, podría ser que aún no funcione. En ese caso, espere unos segundos y trate nuevamente.

Después que el comando BACKUP finaliza satisfactoriamente, se crea un Nuevo archive con la imagen de respaldo de la base de datos. El nombre de este archive se forma siguiendo la convención de la *Figura 11.6*.

Linux/UNIX/Windows

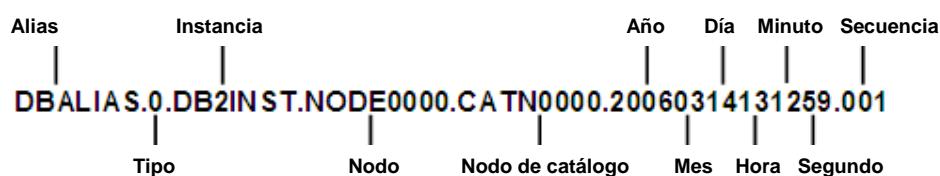


Figure 11.6 – Backup image naming convention

Si el tipo es "0" significa que es un respaldo completo, si es "3" es un respaldo de un espacio de tabla. El nodo es siempre NODE0000 para bases de datos no particionadas, que es el caso en todas las ediciones de DB2 excepto para DB2 Enterprise Edition con el agregado de DPF. El nodo del catálogo es siempre CATN0000. Para mayor información diríjase a los manuales de DB2.

Cuando se toman varios respaldos y se almacenan en el mismo directorio, la indicación de fecha y hora al final del nombre del archive se usa para distinguir entre las imágenes de respaldo. Como veremos en la próxima sección, el comando RESTORE usa esta indicación de fecha y hora para recuperar desde un respaldo específico.

11.7 Recuperación de base de datos

Recuperar una base de datos implica restaurar la base desde un respaldo y/o desde las anotaciones cronológicas. Si solo se recupera desde un respaldo, estaremos recreando la base de datos tal como existía al momento de tomar ese respaldo.

Si se ha activado la anotación cronológica con archive antes de tomar el respaldo, no solamente se puede restaurar desde la imagen de respaldo sino que además desde las anotaciones cronológicas. Tal como se verá en la próxima sección, una recuperación en avance permite recuperar una imagen de respaldo y a partir de allí aplicar (en avance) las anotaciones cronológicas hasta el final de ellas o hasta un punto específico en el tiempo.

Siempre que hablamos de recuperación en esta sección nos estaremos refiriendo al comando RESTORE.

11.7.1 Tipos de recuperación

Existen tres tipos de recuperación:

- **Recuperación de caída o reinicio**

Suponga que usted está feliz trabajando en su computador ejecutando transacciones muy importantes contra su base de datos DB2 y repentinamente se corta la energía eléctrica o alguien accidentalmente desenchufa la máquina. ¿Qué pasa entonces con la base de datos?

Cuando su computador y DB2 arrancan la próxima vez, la recuperación de caída ejecutará automáticamente el comando RESTART DATABASE y leerá las anotaciones cronológicas activas y con ellas rehará o deshará las transacciones sobre la base. Al terminar este comando está garantizado que la base de datos estará en un estado consistente, o sea que lo que se ha confirmado será guardado mientras que lo no confirmado se volverá hacia atrás.

- **Recuperación de versión o imagen**

Este tipo de recuperación implica que se está restaurando la base solamente desde una imagen de respaldo; por lo tanto la base de datos volverá al estado en que estaba al momento que se tomó el respaldo. Todas aquellas transacciones ejecutadas contra la base de datos después de ese momento se perderán.

- **Recuperación en avance**

Con este tipo de recuperación, no solamente se restaura una imagen de respaldo sino que adicionalmente se ejecuta el comando ROLLFORWARD para aplicar las anotaciones cronológicas encima del respaldo, de modo que se puede recuperar hasta un punto específico en el tiempo. Con este tipo de recuperación se minimiza la pérdida de datos.

11.7.2 Recuperación de la base de datos

Para recuperar una base de datos desde una imagen de respaldo se usa el comando RESTORE. La forma más simple de usar este comando es la siguiente:

```
RESTORE DATABASE <nombre_de_la_base_de_datos> [from <directorio>] [taken at <indicación_de_fecha_y_hora>]
```

Por ejemplo, si se tiene una imagen de respaldo de la base de datos *sample* con este nombre:

Alias	Instancia		Año	Día	Minuto	Secuencia
SAMPLE.0	DB2INST.NODE0000.CATN0000		2006	0314	131259	.001
Tipo	Nodo	Nodo de catálogo	Mes	Hora	Segundo	

Puede ejecutarse el comando:

```
RESTORE DB sample FROM <directorio> TAKEN AT 20060314131259
```

11.8 Otras operaciones con BACKUP y RESTORE

A continuación una lista de algunas cosas que se pueden hacer con los comandos BACKUP y RESTORE. Les alentamos a revisar los manuales de DB2 para detalles adicionales.

- Respaldar una base de datos en una instancia de 32-bit y recuperarla en una de 64-bit
- Recuperar sobre una base de datos existente
- Usar recuperación redirigida para recuperar en un sistema con un número diferente de discos a los especificados en la imagen de respaldo
- Respaldar o recuperar a nivel de espacio de tabla en vez de la base de datos completa
- Realizar respaldos delta e incrementales; los respaldos delta registran solo los cambios entre respaldos, mientras que los incrementales registran todos los cambios y los acumulan en cada imagen de respaldo
- Respaldo desde una copia *flash* (se requiere el equipamiento correcto)
- Recuperar tablas dadas de baja (si la opción fue activada para la tabla dada)
- Respaldar en una plataforma (por ejemplo, Windows) y restaurar en otra (por ejemplo, Linux) no es posible. En este caso se debe utilizar db2look y db2move. Para aquellas ediciones de DB2 que soportan el sistema operativo UNIX, en

algunas plataformas se puede respaldar desde una plataforma UNIX y recuperar en otra.

11.9 Resumen

En este capítulo hemos examinado la función de anotación cronológica de DB2, incluyendo los dos tipos de anotaciones cronológicas (primaria y secundaria) y los dos tipos de anotación cronológica (circular y con archivo), además de los distintos parámetros de base de datos relacionados con la anotación cronológica. Para cada tipo de anotación cronológica hemos mencionado cuándo y por qué se usan y como establecerlas desde el Centro de control.

También vimos como ejecutar actividades de respaldo y recuperación usando la línea de mandatos DB2, dando una mirada en profundidad a los tres tipos de recuperación de la base de datos: caída, versión y por avance.

11.10 Ejercicios

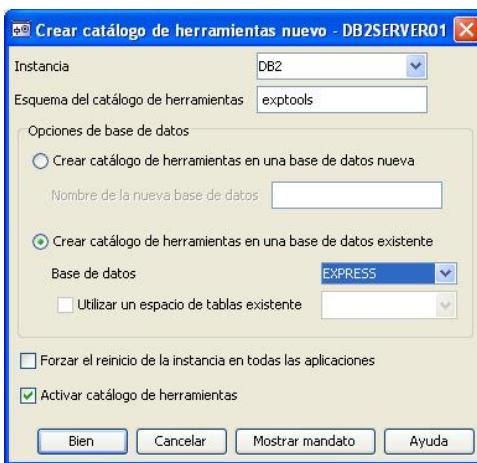
A pesar que DB2 es capaz de automatizar numerosas actividades de mantenimiento de la base de datos, hay veces en que se debe adaptar cuando ciertas actividades deben ocurrir. En este ejercicio crearemos una programación adaptada para un respaldo nocturno de la base de datos **EXPRESS**.

Procedimiento

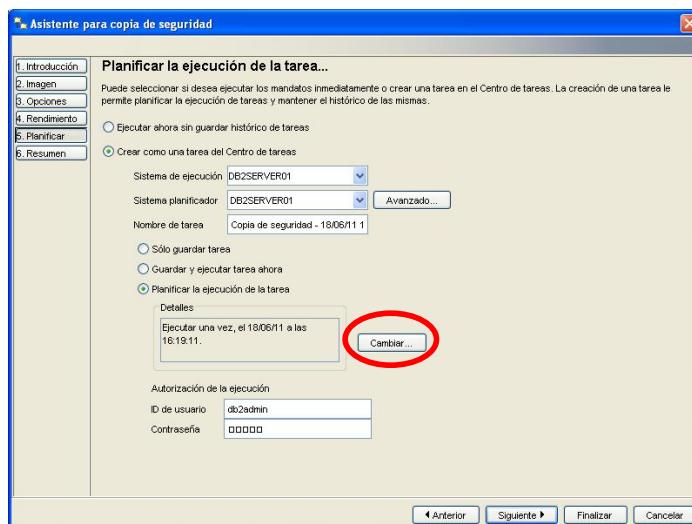
1. Desde el árbol de objetos en el Centro de control navegue hacia *Centro de control -> Todas las bases de datos*. Dé clic con el botón derecho del ratón sobre la base de datos **EXPRESS** y seleccione *Copia de seguridad*. Esto lanza el *Asistente para copia de seguridad*.
2. La página de *Introducción* del asistente resume el estado actual de la base de datos, incluyendo el momento del último respaldo y el método de anotación cronológica. Dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente.
3. En la página de *Imagen* del asistente, seleccione la destinación de la imagen de respaldo. Habitualmente se selecciona un disco físico diferente al cual la base de datos está almacenada. Para este ejercicio cree una nueva carpeta en el sistema de archivos llamada **C:\db2backup**, y especifique esa carpeta como destino del respaldo. Luego en el asistente despliegue la lista de *Tipo de medio* y allí seleccione *Sistema de archivos*. Dé clic en el botón *Añadir*, seleccione la carpeta que acaba de crear y luego dé clic en el botón *Bien*. Luego dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente.
4. Dé un vistazo a las páginas de Opciones y Rendimiento, pero las opciones por omisión son generalmente suficientes dado que DB2 automáticamente realiza el

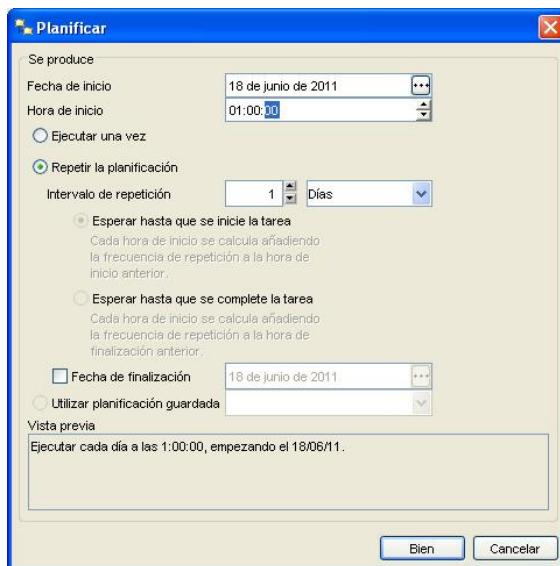
respaldo de la base de datos en la forma más eficiente. Continúe con la página *Planificar* cuando se encuentre satisfecho.

- En la página *Planificar*, si el planificador aún no está activado, seleccione activarlo ahora. Seleccione en que sistema crear el catálogo de herramientas, bajo que esquema y elija crearlo en la base de datos **EXPRESS** (ya existente). El catálogo de herramientas almacena los metadatos acerca de todas las tareas planificadas. Dé clic en el botón *Bien* para continuar. Dé clic en el botón *Siguiente* para moverse a la siguiente página del asistente una vez que el catálogo de herramientas ha sido creado.



- En la página *Planificar*, elija crear un plan para ejecutar una tarea. Planifique que el respaldo corra una vez al día, comenzando a la una de la mañana. Dé clic en el botón *Siguiente* para moverse a la página siguiente.





7. En la página Resumen se pueden revisar las tareas planificadas que se van a crear. Una vez hecha la revisión, dé clic en el botón Finalizar para crear la tarea.
8. Lance el Centro de tareas para ver o modificar la tarea de respaldo recién creada.

12

Capítulo 12 – Tareas de Mantenimiento

Este capítulo discute alguna de las tareas requeridas para mantener en buena forma a su base de datos. En general, en DB2 Express-C al igual que en las otras ediciones de DB2, se automatiza la mayoría de estas tareas. Esta capacidad de auto administración es un gran beneficio para compañías de pequeño y mediano tamaño, que no cuentan con suficientes recursos económicos para contratar a un Administrador de Bases de Datos (DBA). Por otro lado, si un DBA es contratado, este tendrá mas tiempo libre para realizar actividades avanzadas que agregarán valor a la compañía.

Nota:

Para mas información sobre tareas de mantenimiento, observe este video:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4302>

12.1 REORG, RUNSTATS, REBIND

Estas son las tres tareas principales en DB2, como se describe en la Figura 12.1: REORG, RUNSTATS y REBIND.

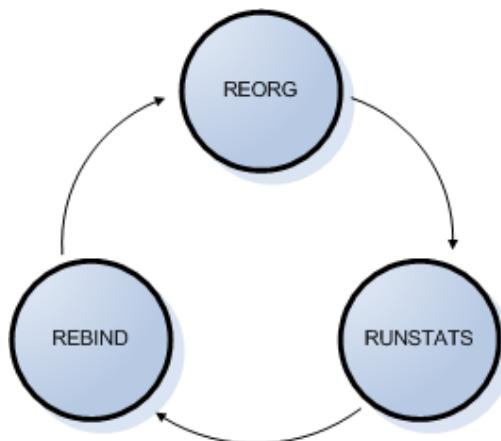


Figura 12.1 – Tareas de Mantenimiento: REORG, RUNSTATS, REBIND

La Figura 12.1 nos muestra que las tareas de mantenimiento son realizadas de manera circular. Si se ejecuta un REORG, es recomendable correr también un RUNSTATS, seguido por un REBIND. En general, las tablas en una base de datos se ven modificadas por operaciones de UPDATE, DELETE e INSERT en manera constante. Luego de algún tiempo es bueno correr estas operaciones de mantenimiento. El ciclo de mantenimiento puede ser iniciado con un REORG.

12.1.1 El comando REORG

Con el tiempo, las operaciones INSERT, UPDATE y DELETE pueden causar fragmentación de los datos guardados en un base de datos. Esto significa que los datos aparecerán en diversas páginas internas en el disco y no de manera contigua. El comando REORG toma el espacio desperdiciado y reorganiza los datos para hacer la extracción más eficiente. La mayoría de las tablas que son frecuentemente modificadas se beneficiarán al ejecutar REORG. Usted puede hacer REORG tanto en índices como en tablas y este comando REORG puede ser ejecutado en línea (online) o fuera de línea (offline).

El comando REORG es mas rápido y eficiente si es offline, pero no permite el acceso a la tabla, mientras que la versión online de REORG permite el acceso a la tabla, pero puede consumir una gran cantidad de recursos del sistema; trabaja mucho mejor para tablas pequeñas.

Sintáxis:

```
REORG TABLE <nombre de tabla>
```

Ejemplo:

```
REORG TABLE empleado
```

El comando REORGCHK puede usarse antes del comando REORG par determinar si una tabla o índice necesitan ser reorganizados.

12.1.2 El comando RUNSTATS

El optimizador de DB2 es el “cerebro” de DB2. Este encuentra el camino de acceso más eficiente para localizar y extraer los datos. El optimizador se basa en el costo de la localización y extracción de datos en el sistema y usa estadísticas que son almacenadas en tablas de catálogo para calcular el camino de menor costo. Las tablas de catálogo tienen estadísticas sobre cuantas columnas tiene una tabla, cuantas filas hay, cuantos y que tipos de índices hay por cada tabla y otra información al respecto.

La información de las estadísticas no es actualizada dinámicamente. Esto es por diseño, ya que no es deseable que DB2 actualice las estadísticas para cada operación realizada en la base por que afectaría negativamente la performance general. En cambio, DB2 provee el comando RUNSTATS para actualizar estas estadísticas. Correr este comando es esencial para mantener las estadísticas actualizadas. El optimizador DB2 puede hacer

cambios radicales en el camino de acceso si sabe que la tabla tiene una fila o si tiene un millón de filas. Cuando las estadísticas de la base de datos están actualizadas, DB2 puede elegir el mejor plan de acceso. La frecuencia en que se recomienda correr el RUNSTATS depende de que tan seguido se realizan cambios en los datos.

Sintaxis:

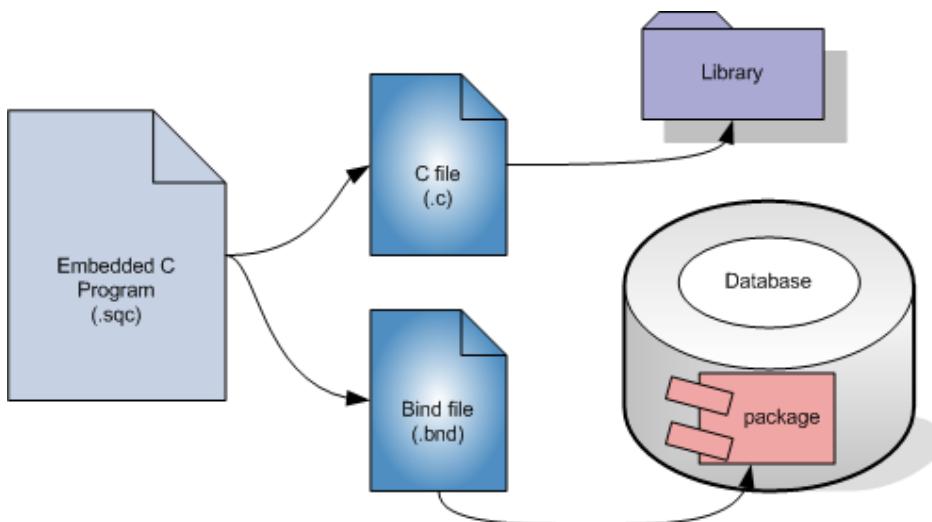
```
RUNSTATS ON TABLE <esquema.nombredetabla>
```

Ejemplo:

```
RUNSTATS ON TABLE miesquema.empleado
```

12.1.3 BIND / REBIND

Luego de correr exitosamente un comando RUNSTATS, no todas las consultas usarán las últimas estadísticas. Los planes de acceso del SQL estático son definidos cuando se ejecuta un comando BIND. Las estadísticas usadas cuando se ejecutó el BIND no serán siempre las mismas que las actuales. La figura 12.2 ayuda a ilustrar esta idea.



La Figura 12.2 – Proceso bind de SQL Estático

En la figura 12.2 un programa C embebido (almacenado como un archivo con una extensión “.sqc”) es precompilado. Luego de la pre-compilación, dos files son generados, un archivo “.c” que contiene el código C con todas las sentencias SQL comentadas; y un archivo “.bnd” conteniendo todas las sentencias SQL. El archivo C con la extensión “.c” es compilado con cualquier compilador C, creando una “biblioteca” o “library” como se muestra en el vértice superior derecho de la figura. El archivo “.bnd” es ‘compilado’ en forma similar con un comando BIND, generando un paquete que se almacena en la base de datos. “Binding” es equivalente a compilar las sentencias SQL donde el mejor plan de

acceso es determinado, basándose en las estadísticas en ese momento y luego guardadas en el paquete.

Ahora, ¿qué pasa si son insertadas un millón de filas en la tabla que se usa en el SQL del programa C embebido? Luego de la inserción, si un RUNSTATS es ejecutado, las estadísticas serán actualizadas, sin embargo el package no será actualizado para recalcular el camino de acceso basado en las últimas estadísticas. El comando db2rbind puede utilizarse para ‘rebind’ todos los paquetes existentes para que tomen en cuenta las últimas estadísticas.

Sintaxis:

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

Ejemplo:

Para que todos los paquetes de la base de datos *sample* sean alcanzados por un rebind general y almacene el resultado en el archivo mylog.txt, ejecute este comando:

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

12.1.4 Tareas de mantenimiento desde el Centro de control

Desde el Centro de Control usted puede correr REORG y RUNSTATS. La figura 12.3 muestra cómo.

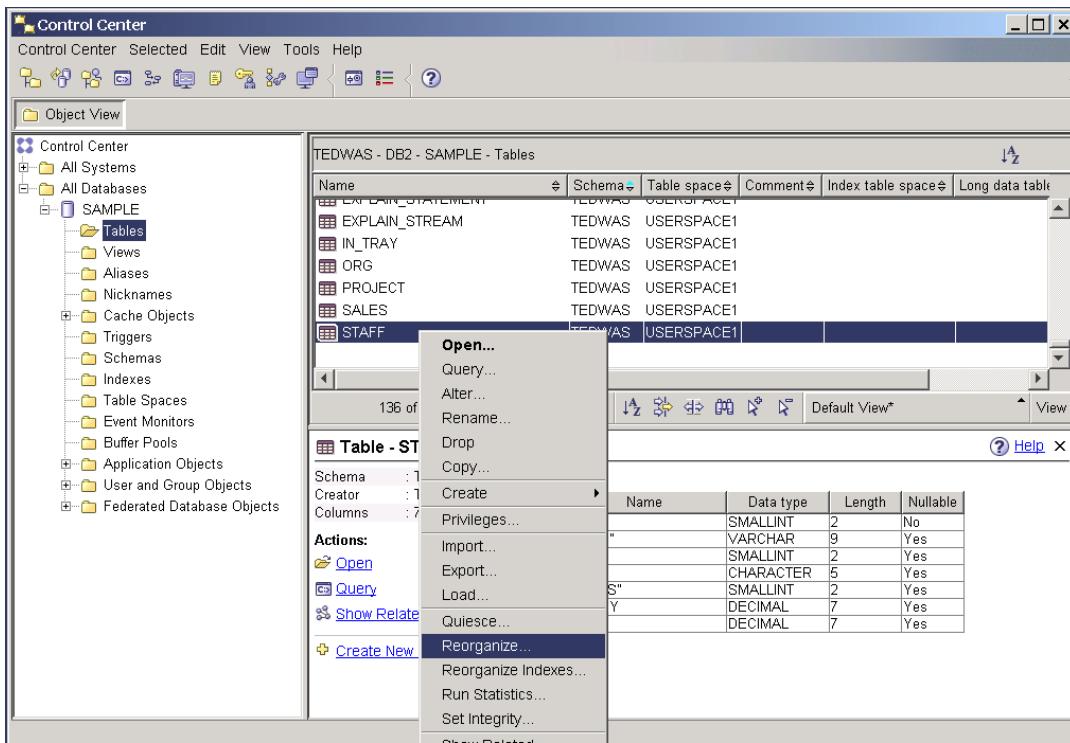


Figura 12.3 – REORG y RUNSTATS desde el Centro de control

Al elegir la tabla sobre la que quiere operar, haga clic botón derecho y seleccione Reorganizar (para ejecutar REORG) o Correr Estadísticas (para ejecutar RUNSTATS).

La vista operacional de la base de datos

Cuando selecciona una base de datos, la vista operacional de la base de datos en el lado derecho de el Centro de control proveerá de información sobre la Base de Datos, tales como su tamaño, cuando fue por ultima vez respaldada y si el mantenimiento automático esta configurado, etc. Esta vista permite identificar rápidamente las necesidades de mantenimiento para su Base de Datos. La figura 12.4 muestra esta información.

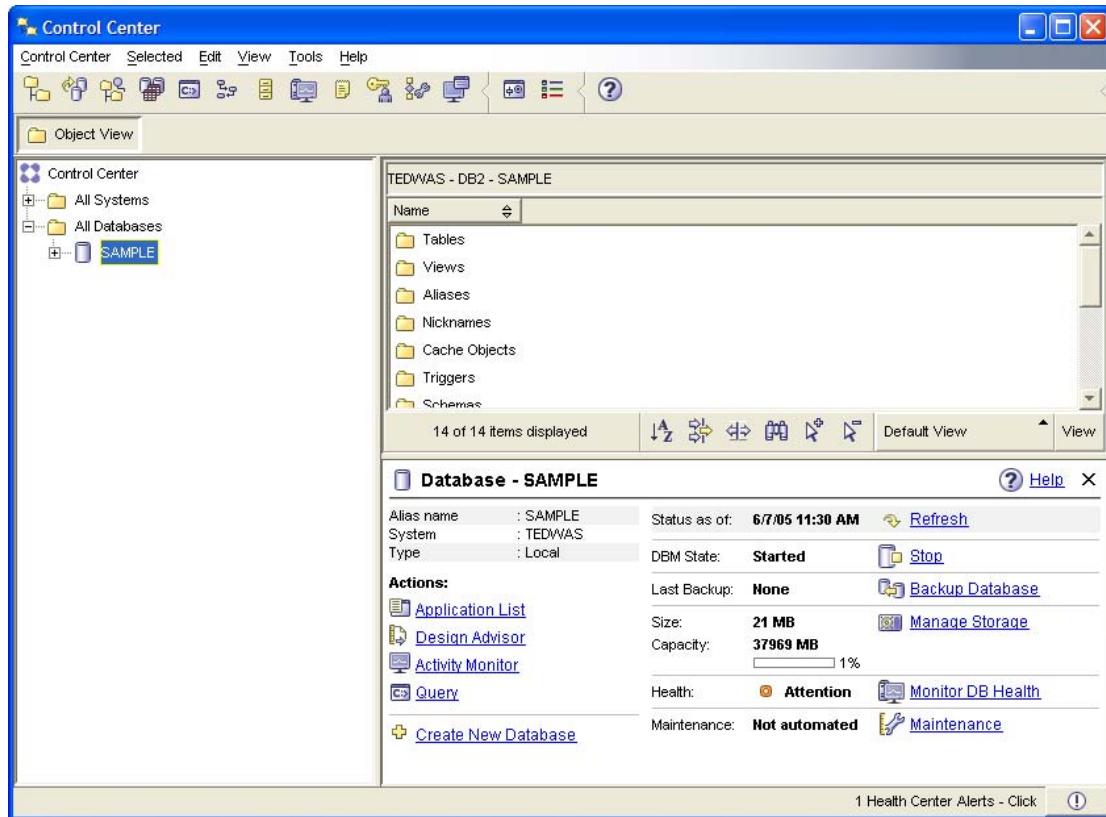


Figura 12.4 – La vista operacional de la Base de datos desde el Centro de Control

12.2 Las opciones de Mantenimiento

Hay tres modos para realizar tareas de mantenimiento:

1. Mantenimiento Manual

Cuando el mantenimiento es requerido, se hace manualmente en el momento que se necesita.

2. Crear scripts para realizar el mantenimiento

Se pueden crear scripts con los comandos de mantenimiento, y programar su ejecución periódica.

3. Mantenimiento Automatizado

Haga que DB2 automáticamente realice el mantenimiento por si mismo (REORG, RUNSTATS, BACKUP)

En esta sección nos concentraremos en el mantenimiento automatizado.

El mantenimiento automático consiste en lo siguiente:

- El usuario define una ventana de mantenimiento donde las tareas pueden ser ejecutadas con una mínima disrupción del funcionamiento operativo. Por ejemplo, si el sistema tiene menor actividad los domingos de 2:00am a 4:00am, este marco de tiempo funcionará como ventana de mantenimiento.
- Hay dos ventanas de mantenimiento: una para las operaciones en línea (online) y otra para las operaciones fuera de línea (offline).
- DB2 ejecutará las operaciones de mantenimiento automáticamente solamente cuando sea necesario durante la ventana de mantenimiento.

Desde el Centro de control, se puede lanzar el asistente de configuración del Mantenimiento automático como se muestra en la figura 12.5.

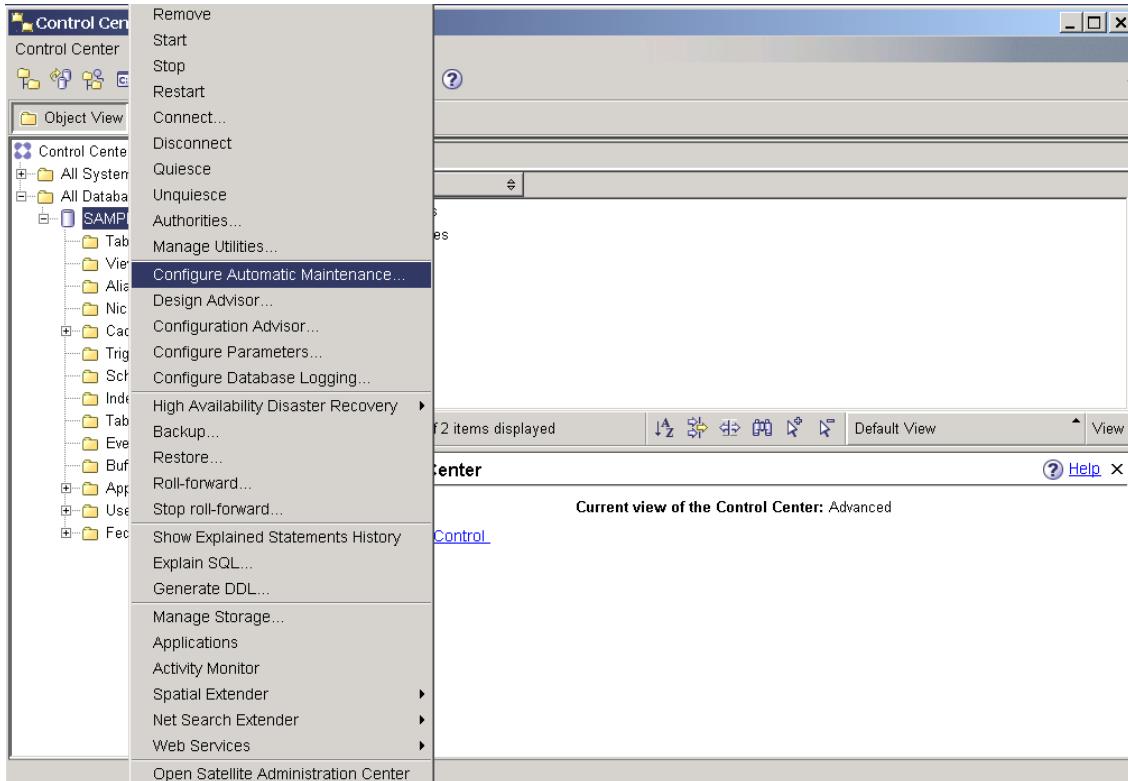


Figura 12.5 – Lanzando el Asistente de Configuración de Mantenimiento automatizado

La figura 12.6 muestra el Asistente de Configuración de Mantenimiento automatizado.

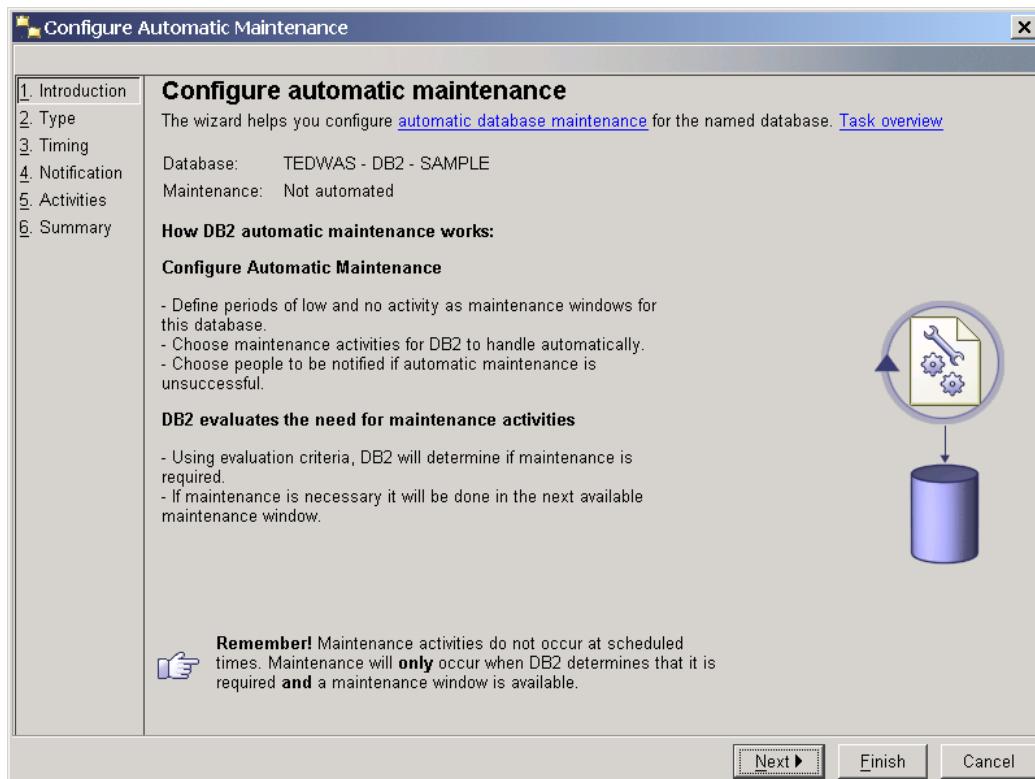


Figura 12.6 – El Asistente de Configuración de Mantenimiento Automatizado

12.3 Resumen

En este capítulo se examinó la importancia del mantenimiento en las bases de datos, incluyendo las tareas cíclicas REORG (reorganizar), RUNSTATS (calcular estadísticas), y REBIND (reenlazar). El comando REORG, reorganiza los datos para eliminar la fragmentación y mejorar la velocidad de respuesta de las consultas. RUNSTATS actualiza información estadística usada por las herramientas de optimización de DB2 para mejorar el rendimiento. Los procesos BIND o REBIND actualizan los paquetes de la base de datos con el último plan de ejecución resultante del cálculo del paso anterior.

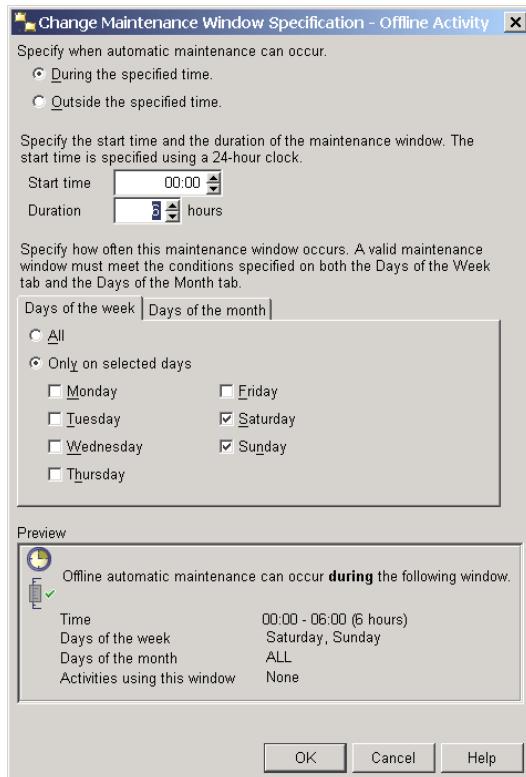
También se mostraron las herramientas gráficas que provee el Centro de Control de DB2 (ahora IBM Data Studio) para ejecutar actividades de mantenimiento, ya sea por medio de scripts, de forma manual o automáticamente.

12.4 Ejercicios

En este ejercicio, se configurará el mantenimiento automático sobre la base de datos **SAMPLE**.

Procedimiento

1. Desde el árbol de objetos del Centro de Control, hacer clic-derecho en la base de datos *SAMPLE* y seleccione el ítem del menú Configuración de Mantenimiento Automático. Esto dispara el Asistente de Configuración de Mantenimiento Automático.
2. La página de *Introducción* del Asistente muestra la configuración de Mantenimiento automático actual. Si ha creado la base de datos SAMPLE con la opción de Mantenimiento Automático, entonces este modo de mantenimiento ya está configurado. Puede usar este asistente para re-configurar las opciones de mantenimiento automatizado. Haga clic en el botón *Siguiente* para moverse a la siguiente página del Asistente.
3. La página *Tipo* del Asistente requiere que elija entre desactivar todo el mantenimiento automático, o cambiar las configuraciones de mantenimiento automático. Seleccione la opción para cambiar las configuraciones de mantenimiento automático, haga clic en *Siguiente*.
4. La página de frecuencia (*Timing*) del asistente requiere que se especifique las ventanas de mantenimiento. Configure la ventana de apagado (*Offline window*) para que cada Sábado y Domingo a la noche desde las 0:00 hasta las 6 AM como se muestra abajo. Haga clic en el botón de Cambio, junto al panel de visualización previa de ventana de mantenimiento *Fuera de Línea* y elija las horas desde-hasta deseadas. Al finalizar de ingresar la información haga clic en el botón *OK* para volver al Asistente. Deje la ventana online como está (así el mantenimiento en línea puede realizarse en cualquier momento). Haga clic en el botón *Next*.



5. En la página de *Notificación* del Asistente, se puede fijar un contacto en caso de que una actividad de mantenimiento automatizado falle. Salteemos este paso por ahora. Haga clic en el botón *Siguiente*.
6. En la pagina de *Actividades* del Asistente, puede elegir individualmente el automatizar o no, algunas de las tareas específica tanto como ser notificado de actividades particulares. En este ejemplo, asegúrese que todos los checkboxes de *Automatice* estén chequeados y que los checkboxes *Notificar* estén deschequeados. Haga clic en el botón *Siguiente*.
7. Antes de pasar a la próxima pagina del Asistente, debería configurar la ubicación del respaldo de la base de datos. Idealmente se debería almacenar los respaldos en un dispositivo diferente en caso de falla de disco. En la pagina *Actividades*, seleccione la opción *Backup database option*, y luego haga clic en el botón *Configuración de Seteos*.
8. En la oreja de *Criterio de Backup* de la ventana de dialogo de Configuración de Seteos, elija la opción *Balance de Recuperabilidad de Base de datos con Performance*. En la oreja de *Ubicación de Backup*, elija una ubicación preexistente y haga clic en el botón *Cambiar*. Especifique una diferente ubicación para realizar un respaldo backup (asegure que haya suficiente espacio en el disco). En la oreja

Modo del Backup, asegúrese que *Backup fuera de línea* esté seleccionado. Haga clic en el botón *OK* para cerrar este tab. Haga clic en el botón *Siguiente*.

9. La página de *Sumario del Asistente de Configuración de Mantenimiento Automatizado* contiene un resumen de las opciones que ha elegido en el proceso. Haga clic en el botón de *Finalizar* para aceptar e implementar el cambio.

13

Capítulo 13 – Concurrencia y Bloqueo

Este capítulo discute cómo permitir que múltiples usuarios tengan acceso a la misma base de datos al mismo tiempo sin interferir uno a otro, y mantener consistencia en sus operaciones. Discutiremos los conceptos de transacciones, de la concurrencia y del bloqueo.

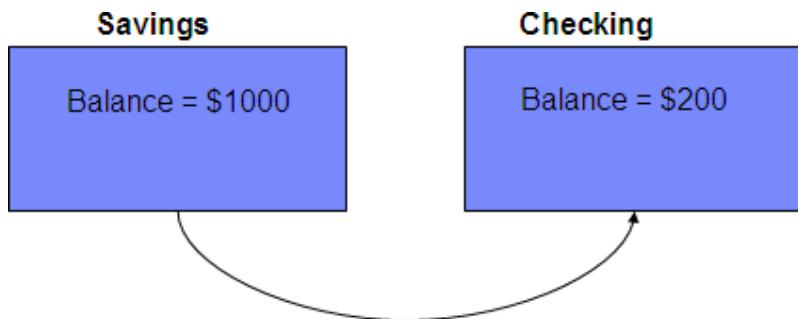
Nota:

Para más información acerca de concurrencia y bloqueo, vea este vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4322>

13.1 Transacciones

Una transacción o una unidad del trabajo consiste en una o más sentencias SQL que, cuando se ejecutan, se deben considerar como una sola unidad; es decir, si una de las sentencias en la transacción falla, la transacción entera falla, y cualquier sentencia ejecutada hasta el punto de la falla se restaura a un estado previo íntegro o consistente (se hace un *rollback*). Una transacción termina con una sentencia COMMIT, que también significa el comienzo de una nueva transacción. La figura 13.1 proporciona un ejemplo de una transacción.



Transfer \$100 from Savings to Checking:

- Debit \$100 from Savings account
- Credit \$100 to Checking account

Figura 13.1 – Un ejemplo de transacción

En la figura 13.1, por ejemplo, usted quiere transferir 100 dólares de su cuenta de ahorros a su cuenta de cheques. La siguiente secuencia de eventos puede ser requerida para completar esa tarea:

Realizar un débito de \$100 de la cuenta de ahorros y luego acreditar este monto en la cuenta de cheques.

Si la secuencia de eventos no se trata como una sola unidad de trabajo, una transacción, se imagina qué sucedería si ocurriera un apagón después de la operación de débito de la cuenta de ahorros, pero antes de que se acredite la operación en la cuenta de cheques. ¡Usted perdería \$100!

13.2 Concurrencia

Concurrencia implica que varios usuarios puedan trabajar al mismo tiempo sobre los mismos objetos de la base de datos. DB2 fue diseñada como una base de datos multiusuarios. El acceso a los datos debe ser coordinado correcta y transparentemente usando un mecanismo para asegurar integridad y consistencia de los datos. Considere la figura 13.2 como un ejemplo.

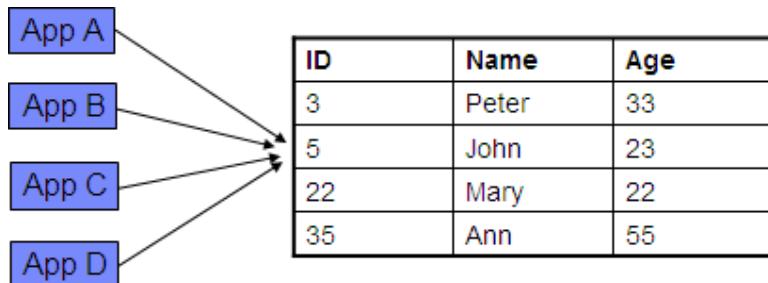


Figura 13.2 –Un ejemplo de concurrencia, y la necesidad por control de concurrencia

En la figura 13.2, hay cuatro aplicaciones, App A, App B, App C, y App D que están intentando tener acceso a la misma fila (fila 2) en una tabla. Sin ningún control de concurrencia, todas las aplicaciones podrían realizar operaciones en la misma fila. Asumiendo que todas las aplicaciones están actualizando la columna Age para la fila 2 con diversos valores, la aplicación que realiza la última actualización será probablemente el "ganador" en esta situación. Debe ser obvio en este ejemplo que una cierta clase de control de concurrencia es requerida para garantizar resultados consistentes. Este control de concurrencia se basa en el uso de bloqueos.

Los conceptos de bloqueo y concurrencia van de la mano. El bloqueo temporal impide a otras aplicaciones realizar sus operaciones hasta que otra operación termine. A mayor bloqueo que haya en un sistema, menor concurrencia será posible. Del otro lado, a menor bloqueo que haya en un sistema, mayor concurrencia será posible.

Los bloqueos ocurren automáticamente pues son necesarios para apoyar una transacción y se eliminan cuando la transacción termina (con un comando del COMMIT o del

ROLLBACK). Los bloqueos se pueden hacer en las tablas o las filas. Hay dos tipos básicos de bloqueos:

- Bloqueos Compartidos (S locks) – Se usa cuando una aplicación quiere leer y prevenir que otros actualicen la misma fila
- Bloqueos (X locks) – Se usan cuando una aplicación actualiza, inserta, o borra una fila

Ahora considere la figura 13.3, la cual es similar a la figura 13.2, pero esta muestra un bloqueo.

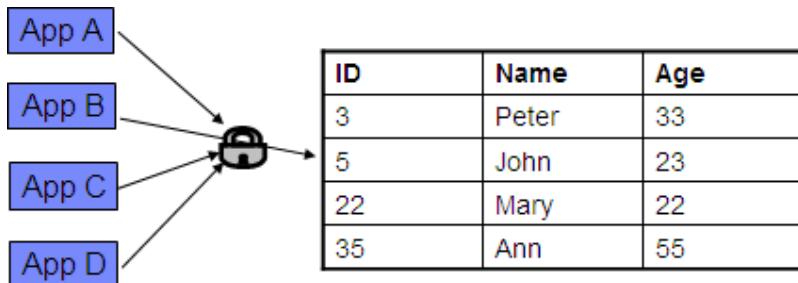


Figura 13.3 –Un ejemplo de concurrencia, y la necesidad de bloqueos

Por ejemplo, en la figura 13.3, si la App B es la primera aplicación en acceder a la fila 2, y está realizando un UPDATE, App B mantiene un bloqueo de X en la fila. Cuando la App A, la App C y la App D intentan tener acceso a la misma fila, no podrán hacer un UPDATE debido al bloqueo de X. Este control permite consistencia y la integridad de los datos.

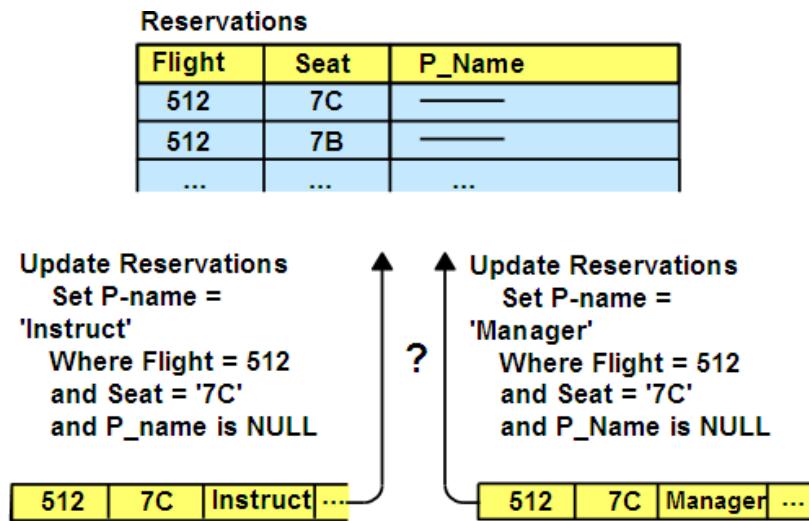
13.3 Problemas sin control de concurrencia

Sin alguna forma de control de concurrencia, se pueden presentar los siguientes problemas

- Actualizaciones perdidas
- Lecturas no confirmadas o Uncommitted read
- Lecturas no repetidas
- Lecturas fantasma

13.3.1 Actualizaciones perdidas

Las actualizaciones perdidas son un problema similar al que se explicó previamente en esta sección donde las aplicaciones realizaron actualizaciones y la última en hacerlo sería la “ganadora”.

**Figura 13.4 – Actualización perdida**

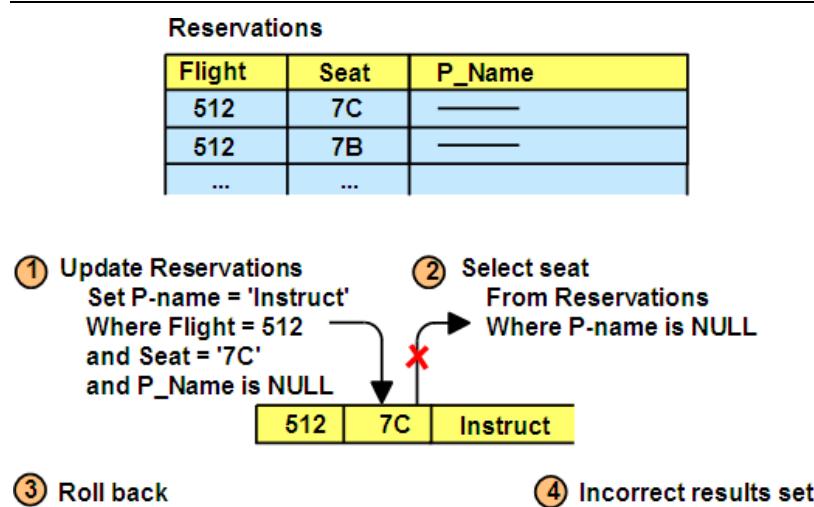
En la figura 13.4 hay dos aplicaciones intentando actualizar la misma fila. La aplicación App1 está del lado izquierdo, aplicación App2 del lado derecho. Entonces, la secuencia de eventos es:

1. App1 actualiza una fila
2. App2 actualiza la misma fila
3. App1 hace commit
4. App2 hace commit

La actualización de la aplicación App1 se pierde cuando la aplicación App2 hace su actualización, y de ahí el término “Actualización Perdida”.

13.3.2 Lecturas no confirmadas o Uncommitted read

Una lectura no confirmada, o también llamada una “lectura sucia”, permite a una aplicación leer información que no ha sido confirmada, y por lo tanto no es necesariamente correcta.

**Figura 13.5 – Lectura no Confirmada o Uncommitted Read**

En la figura 13.5 se sigue esta secuencia de eventos:

- La aplicación App1 actualiza una fila
- La aplicación App2 lee los nuevos valores desde esa fila
- La aplicación App1 hace roll back sobre el cambio de esa fila

La aplicación App2 está leyendo datos no confirmados, y por ello son datos inválidos, por eso este problema es llamado una “Lecturas no confirmadas”

13.3.3 Lecturas no repetidas

Una lectura no repetida implica que usted no puede obtener el mismo resultado después de realizar la misma lectura en la misma operación.

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

Figura 13.6 – Lectura no repetida

En la figura 13.6, considere que usted esté intentando reservar un vuelo de Dallas a Honolulu. La secuencia de eventos es:

- La aplicación App1 abre un cursor (también conocido como un conjunto de resultados o result set) obteniendo lo que usted ve en la figura 13.6
- La aplicación App2 borra una fila que calificó para el cursor (por ejemplo, la fila con destino "San Jose")
- La aplicación App2 confirma los cambios
- La aplicación App1 cierra y vuelve a abrir el cursor

En este caso, ya que la aplicación App1 no consiguió los mismos datos en la lectura, no puede reproducir el conjunto de resultados; a este problema es al que se le llama "lectura no repetida".

13.3.4 Lectura fantasma

La lectura fantasma es un problema similar al problema de lectura no repetida, pero la diferencia está en que las subsecuentes tomas o "fetchs", usted puede obtener filas adicionales. La figura 13.7 proporciona un ejemplo de este problema.

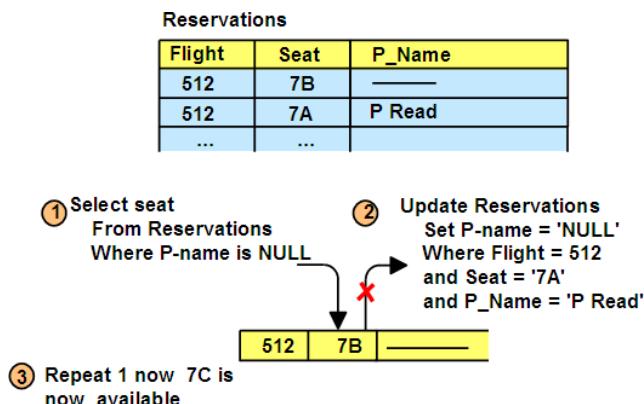


Figura 13.7 – Lectura fantasma

La figura 13.7 muestra la siguiente secuencia de eventos:

1. La aplicación App1 abre un cursor
2. La aplicación App2 agrega una fila a la base de datos que podría calificar para el cursor
3. La aplicación App2 commits los cambios
4. La aplicación App1 cierra y vuelve a abrir el cursor

En este caso, la aplicación App1 podría no conseguir los mismos datos en una lectura repetida, esta podría traer más filas, por esto este problema es llamado "lectura fantasma".

13.4 Niveles de aislamiento

Usted puede considerar los niveles de aislamiento como políticas de bloqueo donde, dependiendo del nivel de aislamiento elegido, usted puede conseguir diferentes comportamientos para una base de datos bloqueada con una aplicación.

DB2 provee diferentes niveles de protección para aislamientos de datos:

- Lectura no confirmada o Uncommitted Read (UR)
- Estabilidad de cursor o Cursor Stability (CS)
- Estabilidad de Lectura o Read Stability (RS)
- Lecturas repetidas o Repeatable Read (RR)

13.4.1 Lectura no confirmada o Uncommitted read

La lectura no confirmada también es conocida como una lectura “sucia”. Este es el menor nivel de aislamiento, y provee el mayor nivel de concurrencia. En operaciones de lectura, las filas no son bloqueadas, a menos que otra aplicación intente eliminar o modificar la tabla; y en operaciones de actualización, el comportamiento es como si se usara el nivel de aislamiento de Estabilidad de Cursor o cursor stability.

Los problemas que aún persisten con este nivel de aislamiento son:

- Lectura no confirmada
- Lecturas no repetidas
- Lecturas fantasmas

Problemas prevenidos con este nivel de aislamiento son:

- Pérdidas de actualizaciones

13.4.2 Estabilidad de cursor o Cursor stability

Estabilidad de cursor o Cursor stability es el nivel de aislamiento predeterminado. Esto provee el mínimo nivel de bloqueo. Básicamente, con este nivel de aislamiento la fila "actual" de un cursor está bloqueada. Si la fila es sólo leída, el bloqueo es mantenido hasta que una nueva fila es tomada (fetched) o la unidad de trabajo es terminada. Si la fila es actualizada, el bloqueo es mantenido hasta que la unidad de trabajo es terminada.

Los problemas que persisten con este nivel de aislamiento son:

- Lecturas no repetidas
- Lecturas fantasmas

Los problemas prevenidos con este nivel de aislamiento son:

- Pérdida de actualizaciones
- Lectura no confirmada

13.4.2.1 Actualmente confirmada (currently committed)

Anterior a DB2 9.7, cuando se utilizaba el nivel de aislamiento de estabilidad de cursor (cursor stability), una escritura (operación UPDATE) impedía a una lectura (operación SELECT) acceder a la misma fila. La lógica indicaba que como el que escribe está cambiando datos de una fila, el que lee debería esperar hasta que estos cambios se terminen para poder ver el último valor confirmado. En DB2 9.7, hay un nuevo comportamiento por omisión en el nivel de aislamiento de estabilidad de cursor. Esta nueva funcionalidad se implementa a través de las semánticas de una transacción **actualmente confirmada (currently committed) (CC)**. Con CC, el que escribe no impedirá al lector acceder a la misma fila. Este comportamiento era sólamente posible en el pasado utilizando el nivel de aislamiento de lectura no confirmada (uncommitted read) (UR); sin embargo, la diferencia ahora es que con UR el que lee obtiene el valor **no confirmado**, mientras que con CC, el lector obtiene el valor **actualmente confirmado (currently committed)**. Este valor es el confirmado antes de comenzar la operación de escritura.

Por ejemplo, se tiene una tabla T1 con los siguientes contenidos

:

FIRSTNAME	LASTNAME
Raul	Chong
Jin	Xie

Ahora la aplicación AppA invoca esta sentencia, pero no la confirma:

```
update T1 set lastname = 'Smith' where firstname = 'Raul'
```

Luego, la aplicación AppB ejecuta:

```
select lastname from T1 where firstname = 'Raul' with CS
```

En las versiones anteriores a DB2 9.7, esta situación provocaba que la aplicación AppB se quede esperando a que AppA libere el registro debido al bloqueo exclusivo que tiene la aplicación AppA sobre el registro en cuestión.

A partir de DB2 9.7 y la activación de actualmente confirmada (el valor por omisión de las nuevas bases de datos), la sentencia devuelve el último valor confirmado del registro, el cual es *Chong*.

Notar que aunque estabilidad de cursor (CS) es el comportamiento por omisión, se incluye la instrucción 'with CS' para mayor claridad. Esta cláusula se discute más adelante en este capítulo.

Si AppB ejecuta esta sentencia:

```
select lastname from T1 where firstname = 'Raul' with UR
```

Debido a que se utiliza lectura sin confirmar (uncommitted read) (UR), el resultado sería *Smith*, el cual es el valor sin confirmar.

Este ejemplo muestra que con actualmente confirmada (currently committed) (CC), se logra una mayor concurrencia para las aplicaciones debido a que las lecturas no tienen que esperar a que las escrituras terminen su tarea.

Otro escenario que hubiera provocado contención previo a DB2 9.7, es una lectura que impide a una escritura acceder a una fila. Esta situación era una de las razones de la recomendación de utilizar un COMMIT aún para las operaciones de lectura, ya que se aseguraría que el bloqueo compartido (share lock) (S) fuese liberado. Con CC, esta situación ya no existe debido a que la lectura no bloquea a la escritura.

En referencia a las operaciones INSERT que no son confirmadas, la operación de lectura las saltea por omisión; es decir, el resultado no muestra estas filas. Para los comandos DELETE, la operación de lectura debería también ignorar las filas afectadas, pero el comportamiento depende del valor de la variable de registro de DB2 DB2_SKIPDELETED. Otras variables de registro y propiedades en los comandos BIND y PREPARE pueden cambiar el comportamiento por omisión de actualmente confirmada (CC).

Para recordar: Actualmente Confirmada (Currently committed) significa que solamente se mostrará la información que en el momento de la operación se encuentra confirmada, por lo tanto, las operaciones INSERT o DELETE no confirmadas serán ignoradas.

Como se mencionó anteriormente, CC está habilitado por omisión en las nuevas bases de datos. Si desea desactivarlo, o habilitarlo para una base de datos creada anterior a DB2 9.7 actualizada a DB2 9.7, puede realizarlo actualizando el valor de configuración CUR_COMMIT de la base de datos a modificar. Por ejemplo, para deshabilitar CC en la base de datos **SAMPLE**, se debería ejecutar:

```
db2 update db cfg for sample using CUR_COMMIT off  
db2stop  
db2start
```

13.4.3 Estabilidad de lectura o Read stability

Con estabilidad de lectura, todas las filas que una aplicación devuelve dentro de una unidad de trabajo son bloqueadas. Para un cursor dado, el bloqueo se da sobre todas las filas que califican para el cursor. Por ejemplo, si tu tienes una tabla con 10,000 filas y la consulta regresa 10 filas, entonces sólo esas 10 filas son bloqueadas. Estabilidad de Lectura usa un moderado grado de bloqueo.

Los problemas persistentes con este nivel de aislamiento son:

- Lectura fantasma

Los problemas prevenidos con este nivel de aislamiento son:

- Pérdida de actualizaciones

- Lectura no confirmada
- Lecturas no repetidas

13.4.4 Lecturas repetidas

Lecturas repetidas es el más alto nivel de aislamiento. Proporciona el grado más alto de bloqueo, y la menor concurrencia. Los bloqueos se mantienen en todas las filas procesadas para construir el result set; es decir, las filas que no necesariamente están en el conjunto de resultados final pueden ser bloqueadas. Ninguna otra aplicación puede actualizar, borrar, o insertar una fila que afectaría el resultado bloqueado hasta que la unidad de trabajo termina. Las lecturas repetidas garantizan que la misma consulta hecha por una aplicación, en una unidad del trabajo, dará el mismo resultado cada vez.

Problemas que persisten con este nivel de aislamiento son:

- ninguno

Problemas prevenidos con este nivel de aislamiento son:

- Pérdida de actualizaciones
- Lecturas no confirmadas
- Lecturas no repetidas
- Lecturas fantasma

13.4.5 Comparando niveles de aislamiento

La figura 13.8 compara los diferentes niveles de aislamiento para un fetch. En la figura, nosotros vemos que el nivel de aislamiento lectura no confirmada o uncommitted read (UR) no tiene bloqueos. El nivel de aislamiento estabilidad de cursor o cursor stability (CS) tiene un bloqueo por la fila 1 cuando ésta está siendo tomada por el Fetch, pero la libera tan pronto como este proceso toma o hace fetch de la fila 2, y así continúa. Para el nivel de aislamiento Estabilidad de lectura o read stability (RS) o lecturas repetidas (RR), cualquier fila que es tomada con fetch será bloqueada, y el bloqueo no es liberado hasta el final de la transacción (cuando se ejecuta un commit).

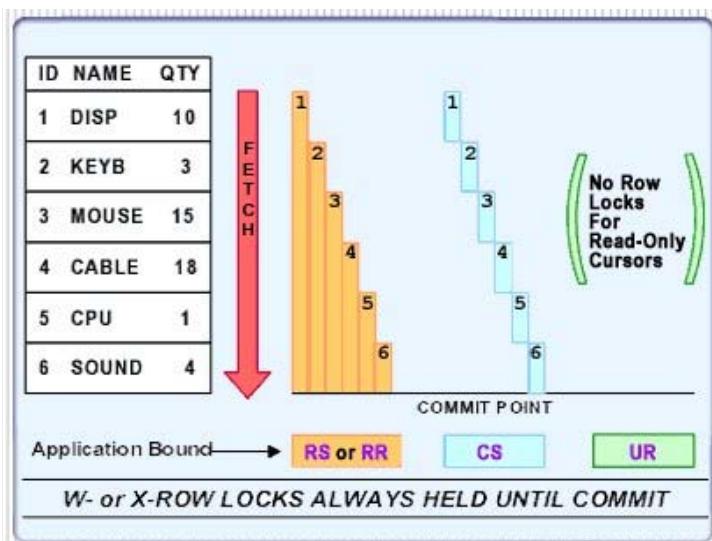


Figura 13.8 – Comparando niveles de aislamiento por un fetch

13.4.6 Fijando los niveles de aislamiento

Los niveles de aislamiento pueden ser especificados en muchos niveles:

- Sesión (aplicación)
- Conexión
- Sentencia

El nivel de aislamiento es definido normalmente en la sesión o en el nivel de aplicación. Si no es especificado un nivel de aislamiento en su aplicación, es puesto por default en Estabilidad de cursor. Por ejemplo, la tabla 13.1 muestra los posibles niveles de aislamiento para un programa .NET o JDBC y cómo esas propiedades corresponden con un nivel de aislamiento de 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

Tabla 13.1 - Comparación de la terminología para los de niveles de aislamiento

A nivel sentencia, el aislamiento puede ser fijado usando la cláusula WITH {nivel de aislamiento}. Por ejemplo:

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

Escenario de ejemplo:

Asuma que una aplicación necesita conseguir una cuenta aproximada del número de filas que hay en una tabla. La prioridad es tener el mejor performance. En este ejemplo, se puede utilizar el nivel de aislamiento de Estabilidad de cursor para toda la aplicación, pero para la sentencia usada para obtener el número de filas de la tabla se puede usar aislamiento UR como se muestra:

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

Para SQL embebido, el nivel es fijado cuando se relacionan las variables en la compilación (bind time). Para SQL dinámico, el nivel se fija en tiempo de ejecución.

La elección del nivel de aislamiento depende de la aplicación. Si su aplicación no necesita cuentas exactas como en el ejemplo previo, elija el aislamiento UR. Si su aplicación necesita un control preciso de los datos con los que trabaja, elija aislamiento RR.

nuevo en
V9.7

Para utilizar las semánticas de actualmente confirmada (CC) en el momento de enlace o de preparado, usar esta sintaxis:

BIND:

```
>-----+----->
'--CONCURRENTACCESSRESOLUTION--+-USE CURRENTLY COMMITTED--+-'
                               '--WAIT FOR OUTCOME-----'
```

PREPARE:

```
concurrent-access-resolution:
| +-USE CURRENTLY COMMITTED-----+
  '-WAIT FOR OUTCOME-----'
```

En una aplicación JDBC que utiliza el IBM Data Server Driver para JDBC y SQLJ, se puede usar la propiedad concurrentAccessResolution para habilitar actualmente confirmada (CC).

13.5 Escalamiento de Bloqueos

Cada bloqueo hecho por DB2 consume algo de memoria. Cuando el optimizador cree que es mejor tener un bloqueo en toda la tabla, en vez de múltiples filas bloqueadas, ocurre el fenómeno llamado Escalamiento de bloqueos. La figura 13.9 ilustra esto.

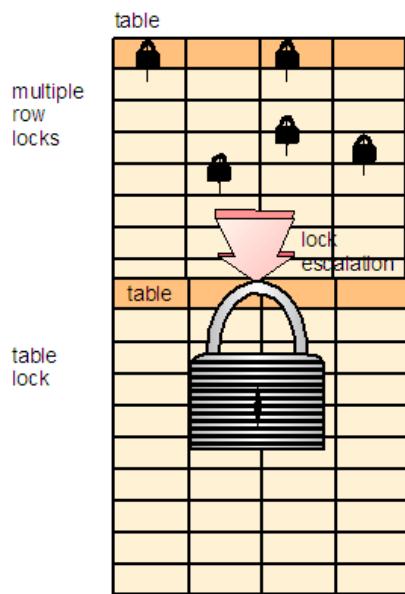


Figura 13.9 – Escalamiento de Bloqueos

Hay dos parámetros de configuración principales de una base de datos relacionadas al Escalamiento de los bloqueos:

- **LOCKLIST** – La cantidad de memoria (en páginas de 4k) reservada para manejar bloqueos para todas las aplicaciones conectadas. El valor predeterminado es cincuenta páginas de 4K (200 K) en Windows.
- **MAXLOCKS** – El máximo porcentaje de la lista entera de bloqueos que una sola aplicación puede usar. El valor predeterminado es 22%.

Por lo tanto, si el valor predeterminado es usado, ocurre escalamiento de bloqueo cuando una aplicación requiere más de 44K de memoria bloqueada ($200\text{ K} * 22\% = 44\text{K}$). Si un escalamiento de bloqueo ocurre frecuentemente con esas condiciones, incremente el valor de LOCKLIST y MAXLOCKS. El fenómeno escalamiento de bloqueo no es bueno para el rendimiento pues reduce concurrencia. El archivo de diagnóstico de DB2 (db2diag.log, que normalmente está localizado en el directorio C:\Program Files\IBM\SQLLIB\DB2) puede ser usado para determinar si está ocurriendo escalamiento de bloqueo.

13.6 Monitoreo de bloqueos

Usted puede monitorear el uso de bloqueos en DB2 usando la utilidad “snapshot” de DB2 que puede capturar instantáneamente lo que ocurre en su aplicación. Para habilitar la instantánea o snapshot para bloqueos, use este comando:

```
UPDATE MONITOR SWITCHES USING LOCK ON
```

Después de habilitar la opción, la información de monitoreo será recolectada. Para obtener un reporte de los bloqueos en un momento dado, ejecute este comando:

GET SNAPSHOT FOR LOCKS FOR APPLICATION AGENT ID <handle>

La figura 13.9 muestra la salida para un ejemplo de la aplicación instantánea de bloqueo.

```
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 = 0x05000700048001000000000052
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
```

Figura 13.9 – Aplicación instantánea de bloqueo

nuevo en
V9.7

Nota:

En DB2 9.7, se hace un esfuerzo para mover el control fuera del monitor de sistema y de la tecnología de situación estática (snapshot), hacia la posibilidad de tener acceso a la memoria interna a través de SQL, como por ejemplo las funciones de tabla de administración de carga de trabajo y las herramientas de IBM Data Studio. Ver la documentación oficial de DB2 para más información.

13.7 Espera de Bloqueo

Cuando dos o más aplicaciones necesitan realizar una operación sobre el mismo objeto, uno de ellos puede tener que esperar para obtener el bloqueo necesario. De forma predeterminada, una aplicación esperará indefinidamente. El tiempo que una aplicación espera por un bloqueo está controlado por el parámetro de configuración de la base de datos **LOCKTIMEOUT**. El valor predeterminado de este parámetro es -1 (espera infinita).

El registro CURRENT LOCK TIMEOUT puede ser usado para fijar el tiempo de Espera de bloqueo para una conexión dada. De forma predeterminada, este registro tiene el valor de **LOCKTIMEOUT**. Use la sentencia **SET LOCK TIMEOUT** para cambiar su valor. Una vez que el valor de este registro es fijado para una conexión, persistirá a través de las transacciones. Por ejemplo:

```
SET LOCK TIMEOUT=WAIT n
```

13.8 Bloqueo Mortal o Deadlock, causas y detección

Un bloqueo mortal o deadlock ocurre cuando dos o más aplicaciones conectadas a la misma base de datos esperan indefinidamente por un recurso. La espera nunca es resuelta porque cada aplicación está esperando un recurso que la otra necesita. En la mayoría de los casos los Deadlocks son un problema de diseño de la aplicación. La figura 13.10 ilustra un deadlock.

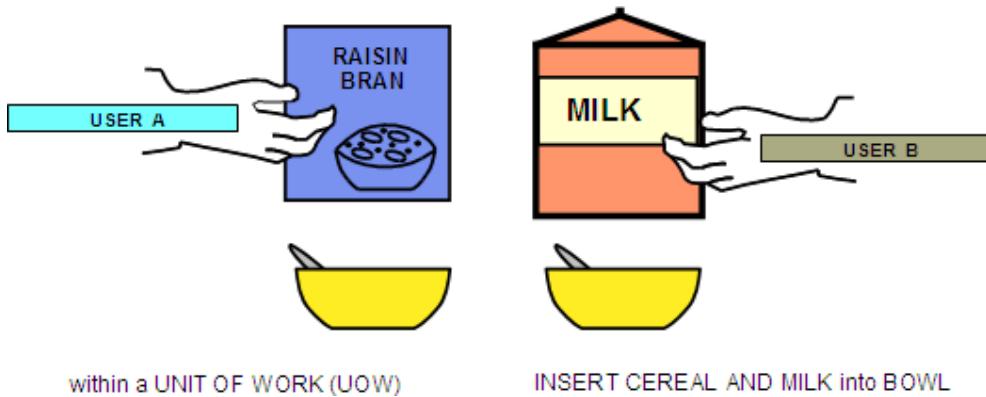


Figura 13.10 – Escenario de un Deadlock

En la figura 13.10, el usuario A está sosteniendo el cereal *raisin bran* y no lo soltará hasta que él tenga la leche. En la otra mano, el usuario B está sosteniendo la leche, y no la dejará hasta que tenga el cereal *raisin bran*. Por lo tanto, tenemos una situación de bloqueo mortal o deadlock.

nuevo en
V9.7

Con DB2 9.7, el uso de actualmente confirmada (CC) ha reducido considerablemente las situaciones de abrazos mortales (deadlocks), ya que la aplicación no necesita esperar por otra aplicación para liberar un bloqueo, sino que accede directamente al valor de la información actualmente confirmada.

Para simular una situación de deadlock en DB2, siga estos pasos:

1. Desactivar la opción de actualmente confirmada:

```
db2 update db cfg for sample using cur_commit off  
db2stop force  
db2start
```

2. Abra dos ventanas de comandos de DB2 (las cuales serán llamadas "CLP1" y "CLP2", respectivamente) representando dos diferentes aplicaciones que se conectan a la base de datos.
3. Desde CLP1 ejecute los comandos:

```
db2 connect to sample  
db2 +c update employee set firstnme = 'Mary' where empno = '000050'
```

Primero nos conectamos a la base de datos SAMPLE, y luego ejecutamos una sentencia UPDATE en la fila con "empno = 50000" en la tabla employee. La opción "+c" en la sentencia indica que nosotros no queremos que la ventana de comandos de DB2 automáticamente ejecute un COMMIT luego de esta sentencia. Esto lo hacemos deliberadamente porque queremos mantener el bloqueo.

4. Desde CLP2 ejecute los comandos:

```
db2 connect to sample  
db2 +c update employee set firstnme = 'Tom' where empno = '000030'
```

En la ventana CLP2, que representa la segunda aplicación, estamos conectando a la base de datos SAMPLE, pero estamos actualizando otra fila en la tabla employee.

5. Desde CLP1 ejecute:

```
db2 +c select firstnme from employee where empno = '000030'
```

Después de presionar *Enter* para ejecutar la sentencia SELECT, puede parecer que el SELECT se queda en un estado de no respuesta. Realmente no lo está, si no que está esperando por el liberamiento de un bloqueo exclusivo (o exclusive lock) que fue tomado por el CLP2 en esa fila en el paso 3. En este punto, si LOCKTIMEOUT ha sido dejado con su valor predeterminado de -1, la aplicación CLP1 podría esperar eternamente

6. Desde CLP2 ejecute:

```
db2 +c select firstnme from employee where empno = '000050'
```

Al ejecutar la sentencia SELECT, estamos creando un deadlock. Esta sentencia SELECT también parecerá quedarse en un estado de no respuesta, porque está esperando por el liberamiento del bloque exclusivo (o exclusive lock) que fue tomado por CLP1 en esta fila en el paso 2.

En este ejemplo de deadlock, DB2 verifica el valor del parámetro de configuración DLCHKTIME en la base de datos. Este parámetro fija el intervalo de tiempo para verificar si un deadlock ocurrió. Por ejemplo, si este parámetro es fijado a 10 segundos, DB2 verificará cada 10 segundos si un deadlock ha ocurrido. Si un deadlock ha ocurrido, DB2

usará un algoritmo interno para determinar cuál de las dos transacciones podría ser rolled back (deshecha), y cuál debería continuar.

Si usted está experimentando numerosos deadlocks, usted debería examinar sus transacciones existentes y ver si una reestructuración es posible.

13.9 Mejores prácticas para concurrencia y bloqueo

Siguiendo los consejos siguientes, usted puede mejorar la concurrencia de sus aplicaciones:

1. Asegurarse de habilitar la opción de actualmente confirmada (CC), en caso de que la lógica de la aplicación así lo permita.
2. Crear las transacciones lo mas cortas como sea posible. Esto puede ser posible utilizando constantemente la sentencia COMMIT (aun para transacciones de solo lectura) cuando la lógica de su aplicación lo permita.
3. Registrar (log) la información de las transacciones sólo cuando sea requerida.
4. Purgar los datos rápidamente usando:

```
ALTER TABLE ACTIVATE NOT LOGGED INITIALLY WITH EMPTY TABLE
```

o nuevo con DB2 9.7, use el comando TRUNCATE:

```
TRUNCATE <table name>
```

5. Realizar modificaciones de datos en lotes o grupos. Por ejemplo:

```
DELETE FROM (
    SELECT * FROM tedwas.t1 WHERE c1 = ... FETCH FIRST 3000 ROWS
    ONLY)
```

6. Use las características de concurrencia en las herramientas de movimiento de datos de DB2.
7. Fije el nivel del parámetro LOCKTIMEOUT (el tiempo sugerido está entre 30-120 segundos). No lo deje en su valor predeterminado de -1. Usted también puede usar tiempo de bloqueo basado en una sesión.
8. No regrese más datos de los que son requeridos. Por ejemplo, use la cláusula FETCH FIRST n ROWS ONLY en sentencias SELECT.

Nota:

Para más información sobre las mejores prácticas sobre concurrencia y bloqueo, ver los documentos de Mejores Prácticas disponibles en
<http://www.ibm.com/developerworks/data/bestpractices/>

nuevo en
V9.7

13.10 Resumen

En este capítulo hemos visto cómo mantener la integridad de los datos a través del control transaccional, el acceso concurrente de los usuarios y los niveles de bloqueo. Los diferentes niveles de concurrencia generan situaciones que pueden afectar cómo se accede y administran los datos.

También hemos visto en detalle la posibilidad de fijar los diferentes niveles de aislamiento para tratar con estos temas, y cómo los niveles de aislamiento pueden ser manipulados para proveer la máxima flexibilidad requerida por la aplicación y las necesidades de los datos.

Hemos revisado el escalamiento de bloqueo, espera de bloqueo y control de bloqueo, en conjunto con las situaciones que provocan los abrazos mortales (deadlocks), cómo detectarlos y su resolución.

Finalmente, se mostraron las mejores prácticas e ideas para obtener el mayor beneficio y mejores resultados para cumplir con el nivel de concurrencia necesario.

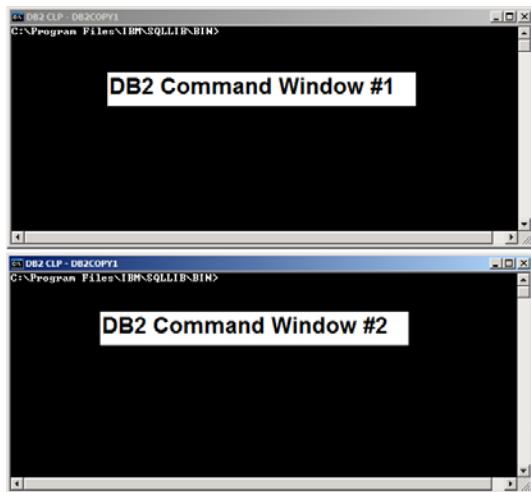
13.11 Ejercicios

En este ejercicio se practicarán los conceptos de concurrencia y bloqueo vistos en el presente capítulo utilizando la Ventana de Comandos de DB2. Esta herramienta usa un nivel de bloqueo de aislamiento CS (estabilidad de cursor) por omisión. Después de ejecutar una instrucción SQL, la Ventana de Comandos invocará automáticamente una confirmación (commit), esto también se lo conoce como auto-confirmación (autocommit). A fines ilustrativos del ejercicio, se utilizará el indicador +c para desactivar el autocommit, en conjunto con la cláusula `WITH <isolation level>` luego de algunas operaciones SQL para sobreescribir el nivel de aislamiento por omisión (CS).

Parte 1: Probando el problema de Lectura Fantasma e aislamiento RR

Procedimiento:

1. Abrir dos Ventanas de Comandos de DB2 como se muestra en la figura siguiente. Llamaremos a la ventana superior "DB2 Command Window #1", y a la ventana inferior "DB2 Command Window #2"



2. Desde DB2 Command Window #1 ejecutar:

```
db2 connect to sample  
db2 +c select * from staff
```

Esta instrucción devolverá 35 registros

3. Desde DB2 Command Window #2 ejecutar:

```
db2 connect to sample  
db2 +c insert into staff (id,name) values (400, 'test')
```

4. Desde DB2 Command Window #1 ejecutar:

```
db2 +c select * from staff
```

Esta instrucción devolverá 35 registros

5. Desde DB2 Command Window #2 ejecutar:

```
db2 commit
```

6. Desde DB2 Command Window #1 ejecutar:

```
db2 +c select * from staff
```

Ahora se recuperan 36 registros!

DB2 Command Window #1 representa una aplicación que abre un cursor o grupo de resultados (`select * from staff`) donde se obtienen 35 registros. Dentro de la misma transacción (debido a que en esta ventana no se ejecuta ninguna instrucción de confirmación (`commit`)), la aplicación abre el mismo cursor, y todavía se ven 35 registros, inclusive luego de que la aplicación en DB2 Command Window #2 inserta (pero no confirmó) un nuevo registro.

Luego, la aplicación DB2 Command Window #2 confirma (commit) el insert, y por tercera vez que la aplicación DB2 Command Window #1 abre el cursor, el resultado devuelve una fila más (lectura fantasma) obteniendo 36 registros. Este ejemplo ilustra el problema de lectura fantasma: Dentro de la misma transacción, abrir el mismo cursor devuelve más filas. Al estar utilizando el nivel de aislamiento CS, como se mencionó anteriormente en este capítulo, no se evitan las lecturas fantasma.

7. Limpiar el registro previamente insertado antes de continuar con los siguientes pasos:

Desde DB2 Command Window #1:

```
db2 rollback
```

Desde DB2 Command Window #2:

```
db2 delete from staff where id = 400
```

```
db2 select * from staff
```

Se recuperan 35 registros nuevamente.

8. Ahora veamos si un nivel de aislamiento RR puede prevenir el problema de lectura fantasma.

Desde DB2 Command Window #1 ejecutar:

```
db2 connect to sample
```

```
db2 +c select * from staff with RR
```

Se recuperan 35 registros

Desde DB2 Command Window #2:

```
db2 connect to sample
```

```
db2 +c insert into staff (id,name) values (400, 'test')
```

Esta sentencia queda esperando, lo cual es el comportamiento esperado.

Como la cláusula WITH RR se agrega a la sentencia SELECT en DB2 Command Window #1, este nivel de aislamiento previene que un INSERT en una fila cambie el resultado. Este ejemplo muestra que el nivel de aislamiento RR (lectura repetible) evite el problema de lectura fantasma.

9. Liberar antes de seguir con la parte 2 de este ejercicio:

Desde DB2 Command Window #2:

```
Ctrl-C (para interrumpir)
```

Cerrar la ventana

Desde DB2 Command Window #1:

```
db2 rollback
Cerrar la ventana
```

Parte 2: Probar los niveles actualmente confirmada (CC) y lectura no confirmada (UR)

Procedimiento 1: Analizar el comportamiento de aislamiento CS sin actualmente confirmada (CC)

1. Abrir una Ventana de Comando de DB2 e invocar las siguientes sentencias:

```
db2 connect to sample
db2 select * from staff
```

Revisar los contenidos de la tabla **STAFF**, buscar específicamente el **ID** de valor '10'. La correspondiente columna **NAME** tiene el valor **Sanders**. Cerrar la ventana.

2. Actualmente confirmada está por omisión en las nuevas bases de datos. Verificar que esté habilitada para la base de datos **SAMPLE**:

```
db2 get db cfg for sample
```

Buscar la linea cerca del final de la salida, donde dice:

Currently Committed	(CUR_COMMIT) = ON
---------------------	---------------------

Si el valor está en ON, cambiarlo a OFF primero para que se pueda analizar el comportamiento del nivel de aislamiento CS como si fuese una versión anterior a DB2 9.7:

```
db2 update db cfg for sample using CUR_COMMIT off
db2 force applications all
```

(Agregar la opción **force** asegura que no haya conexiones de forma tal que CUR_COMMIT haga efecto en la próxima conexión)

Confirmar que CUR_COMMIT se encuentra deshabilitado. Debería decir:

Currently Committed	(CUR_COMMIT) = DISABLED
---------------------	---------------------------

3. Abrir dos Ventanas de Comando de DB2 como en la parte 1, de forma tal de que una esté por encima de la otra. Veamos cómo el nivel de aislamiento trabaja sin actualmente confirmada, cuando un UPDATE (escritura) y un SELECT (lectura) tiene interés sobre la misma fila. Notar que no necesitamos poner "WITH CS" después de las sentencias (debido a que es el valor por omisión).

Desde DB2 Command window #1 (escritura):

```
db2 connect to sample
```

```
db2 +c update staff set name = 'Chong' where id = 10
```

Desde DB2 Command window #2 (lectura):

```
db2 connect to sample  
db2 +c select * from staff
```

Este SELECT se queda esperando para que el bloqueo exclusivo (X) sea liberado por la aplicación DB2 Command Window #1. Como se puede ver, el comportamiento por omisión previo a DB2 9.7 permite menos concurrencia

Desde DB2 Command window #2:

```
CTRL-C (oprimir estas teclas para interrumpir)  
Cerrar la ventana
```

Desde DB2 Command window #1:

```
db2 rollback  
Cerrar la ventana
```

Procedimiento 2: Analizar el comportamiento del nivel de aislamiento CS con actualmente confirmada

1. Establecer a ON actualmente confirmada:

Abrir una Ventana de Comando DB2 e invocar:

```
db2 update db cfg for sample using CUR_COMMIT on  
db2 force applications all
```

Cerrar la ventana.

2. Abrir dos Ventanas de Comando DB2 como se hizo en la parte 1 de forma tal que una quede encima de otra. Luego ejecutar lo siguiente:

Desde DB2 Command window #1:

```
db2 connect to sample  
db2 +c update staff set name = 'Chong' where id = 10
```

Desde DB2 Command window #2:

```
db2 connect to sample  
db2 +c select * from staff
```

Ahora esta sentencia SELECT funciona! .No se queda esperando, y el valor que se muestra es *Sanders*, el cual es el valor actualmente confirmado.

Desde DB2 Command window #1:

```
db2 rollback  
Cerrar la ventana.
```

Desde DB2 Command window #2:

```
db2 rollback  
Cerrar la ventana.
```

Procedimiento 3: Analizar el comportamiento del nivel de aislamiento UR

1. Abrir dos Ventanas de Comando DB2 como en la parte 1, una ventana encima de la otra. Luego, invocar lo siguiente:

Desde DB2 Command window #1:

```
db2 connect to sample  
db2 +c update staff set name = 'Chong' where id = 10
```

Desde DB2 Command window #2:

```
db2 connect to sample  
db2 +c select * from staff with UR
```

Este SELECT funciona, pero hay que observer que el valor recuperado es *Chong*, el cual no está confirmado.

Desde DB2 Command window #1:

```
db2 rollback  
Cerrar la ventana.
```

Desde DB2 Command window #2:

```
db2 rollback  
Cerrar la ventana.
```


PARTE III – APRENDIENDO DB2: DESARROLLO DE APLICACIONES

14

Capítulo 14 – Introducción al Desarrollo de Aplicaciones en DB2

IBM DB2 es un poderoso software de servidor de datos para administrar tanto datos relacionales como XML. Ofrece flexibilidad no solamente a administradores de bases de datos, sino también a desarrolladores. Sin importar el lenguaje que se utilice para desarrollar los programas, DB2 provee los manejadores, adaptadores, y extensiones que se necesitan para trabajar con las bases de datos como parte de la aplicación. Es más, con DB2 Express-C, se pueden desarrollar las aplicaciones sin costo alguno, sin limitación de tamaño de base de datos, y con el mismo nivel de soporte al lenguaje de programación como las otras versiones de DB2. La idea es desarrollar por única vez utilizando DB2 Express-C y ejecutar en cualquier edición de DB2 sin que se requiera modificar la aplicación.

Nota:

Esta sección es solamente una introducción al desarrollo de aplicaciones en DB2. Se están desarrollando una serie de más de 25 libros disponibles online, gratuitos, como parte de la **Serie de Libros del Campus de DB2**, incluyendo un libro específico al desarrollo de aplicaciones en DB2. Otros libros de la serie incluyen temas que no están relacionados con DB2, tales como Java, PHP, Ruby on Rails (Ruby sobre Ruedas), Python, Perl, Web 2.0, SOA, Eclipse, desarrollo de Código Abierto (Open Source), Computación en la Nube (Cloud Computing) y mucho más!. Otros libros contemplan tecnologías IBM en más detalle, como pureQuery, Data Studio, InfoSphere Data Architect.

14.1 Desarrollo de Aplicaciones DB2: Una visión global

DB2 le ofrece a los desarrolladores de bases de datos, la flexibilidad para tomar ventaja de las características de desarrollo del lado del servidor, tales como procedimientos almacenados y funciones definidas por el usuario, y a su vez utilizar las aplicaciones del lado cliente usando el lenguaje de programación elegido. Esta flexibilidad se ve ilustrada en la *Figura 14.1*.

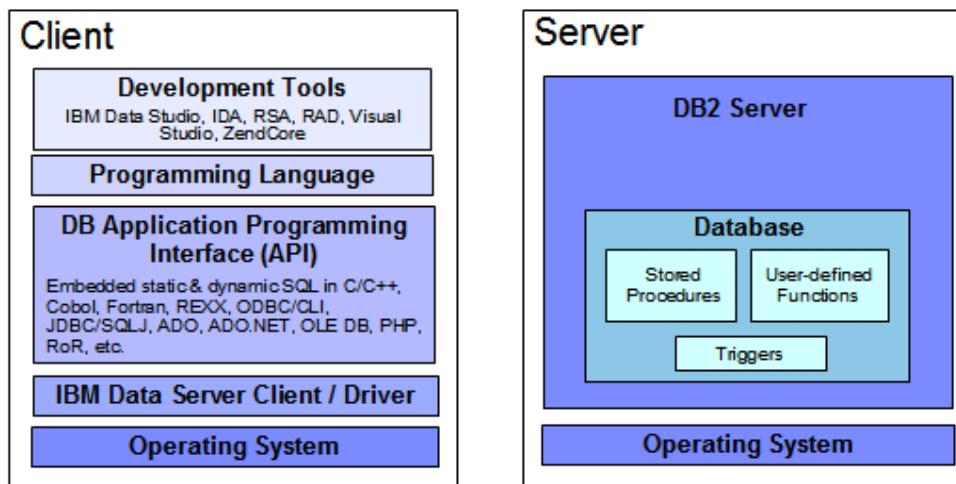


Figura 14.1 - DB2 es para todos: Desarrolladores de Aplicaciones y de Bases de Datos

En la Figura 14.1, la parte izquierda representa una máquina cliente donde un programador de aplicaciones desarrolla y ejecuta un programa. En esta máquina cliente, además del sistema operativo, se instala el Cliente de Servidor de Datos de IBM (IBM Data Server Client) dependiendo del tipo de aplicación que se está desarrollando. Este cliente incluye los manejadores (drivers) de conexión necesarios tales como los de JDBC y los de ODBC/CLI. Estos manejadores se pueden descargar de forma independiente, visitando el sitio Web de IBM DB2 Express-C Web ibm.com/db2/express

Utilizando herramientas de programación tales como IBM Data Studio, InfoSphere Data Architect (IDA), Rational Software Architect (RSA), Rational Application Developer (RAD), y demás, se puede desarrollar la aplicación en el lenguaje de programación deseado. Las bibliotecas API (API libraries) que soportan a estos lenguajes también se incluyen en el Cliente de Servidor de Datos de IBM, de forma tal que cuando hay una conexión al Servidor de DB2, todas las instrucciones del programa se traducen adecuadamente por medio de estas bibliotecas APIs en sentencias SQL o XQuery que son bien entendidas por DB2. La Tabla 1.1 muestra una breve descripción de las herramientas mencionadas anteriormente.

Nombre de la herramienta	Descripción
IBM Data Studio	IBM Data Studio es una herramienta basada en Eclipse que le permite a los usuarios administrar sus servidores de datos y desarrollar procedimientos almacenados, funciones y servicios de Datos Web. IBM Data Studio se detalló en

	capítulos anteriores del presente libro.
InfoSphere Data Architect (IDA)	IDA es una herramienta de modelado. Ayuda a construir tanto el diseño físico como lógico de una base de datos.
Rational Software Architect (RSA)	RSA es una herramienta para la ingeniería de software, basada en Eclipse, que ayuda al desarrollo de diagramas UML.
Rational Application Developer (RAD)	RAD es una herramienta para desarrollos de aplicaciones rápidos basada en Eclipse destinado a programadores de software.
Visual Studio	Microsoft Visual Studio es una herramienta IDE que permite desarrollar aplicaciones en la plataforma de Windows utilizando la tecnología de Microsoft.
ZendCore	Anteriormente llamada <i>ZendCore para IBM</i> , esta es una herramienta sin costo destinada a desarrollar aplicaciones PHP.

Tabla 14.1 – Herramientas que pueden asistirlo a desarrollar aplicaciones con DB2

En el lado derecho de la *Figura 14.1*, se ilustra un servidor DB2 que contiene una base de datos. Dentro de esta base de datos, hay procedimientos almacenados, funciones definidas por el usuario y disparadores (triggers). Se describen todos estos objetos con más detalle en las próximas secciones.

14.2 Desarrollo del lado del Servidor

El desarrollo del lado del Servidor en DB2 implica que los objetos de aplicación están desarrollados y almacenados en una base de datos DB2. Los siguientes objetos de aplicación se discutirán más adelante en esta sección:

- Procedimientos Almacenados (Stored Procedures)
- Funciones definidas por el Usuario (User-defined Functions) (UDFs)
- Disparadores (Triggers)

14.2.1 Procedimientos Almacenados

Un **procedimiento almacenado** es un objeto de aplicación de base de datos que puede encapsular sentencias SQL y lógica de negocio. Mantener parte de la lógica de la aplicación del lado de la base de datos provee mejoras en la performance debido a que la cantidad de tráfico de red entre la aplicación y la base de datos se reduce drásticamente.

Adicionalmente, los procedimientos almacenados proveen una ubicación centralizada para almacenar el código, de forma tal que otras aplicaciones puedan reutilizar los mismos procedimientos almacenados. Para invocar un procedimiento almacenado, hay que utilizar la sentencia **CALL**. En DB2 se pueden desarrollar procedimientos almacenados en varios lenguajes, incluyendo SQL PL, Java, C/C++, CLR, OLE, y COBOL. A continuación se muestra un simple ejemplo de cómo crear e invocar un procedimiento almacenado SQL PL desde la Ventana de Comandos de DB2 o en Linux:

```
db2 create procedure P1 begin end  
db2 call P1
```

En el ejemplo, el procedimiento P1 no tiene contenido, lo cual provoca que no realice nada. El ejemplo muestra lo facil que es crear un procedimiento almacenado. Para desarrollar procedimientos almacenados con lógica más compleja, recomendamos el uso de IBM Data Studio, el cual incluye la opción de rastrear el código (debugger).

14.2.2 Funciones definidas por el Usuario

Una **función definida por el Usuario (User-defined function) (UDF)** es un objeto de aplicación de la base de datos que permite a los usuarios extender el lenguaje SQL con su propia lógica. Una función siempre devuelve un valor o valores generalmente como resultado de la lógica de negocio incluída en la función. Para invocar una función, se efectua dentro de una sentencia SQL, o con la función **values**. En DB2 se pueden desarrollar UDFs en varios lenguajes, incluyendo SQL PL, Java, C/C++, OLE, CLR.

Este simple ejemplo muestra cómo crear e invocar una función definida por el Usuario (UDF) desde la Ventana de Comandos de DB2 o desde Linuxl:

```
db2 create function F1() returns integer begin return 1000; end  
db2 values F1
```

En el ejemplo, la función F1 es una función que devuelve el valor entero 1000. La sentencia **VALUES** puede ser utilizada para invocar la función. Al igual que con los procedimientos almacenados, recomendamos crear funciones a través del IBM Data Studio.

14.2.3 Disparadores (Triggers)

Un **disparador** es un objeto que automáticamente realice una operación sobre una tabla o vista. Una acción sobre un objeto donde se definió un disparador, provoca que el disparador se ejecute. Un disparador generalmente no es considerado un objeto de aplicación; por lo tanto, los desarrolladores de base de datos generalmente no codifican los disparadores, sino que lo hacen los administradores de bases de datos. Debido a que se requiere cierta habilidad para codificar, hemos incluido disparadores en esta sección. Debajo se muestra un ejemplo de un disparador:

```
create trigger myvalidate no cascade before insert on T1
    referencing NEW as N
    for each row
    begin atomic
        set (N.myxmlcol) = XMLVALIDATE(N.myxmlcol
            according to xmlschema id myxmlschema);
    end
```

En este ejemplo, el disparador se ejecuta antes que se efectúe la operación de INSERT (before INSERT) sobre la tabla T1. El disparador inserta el valor (el cual es un documento XML), e invocará a la función XMLVALIDATE para validar este documento con un esquema. El Capítulo 15, *DB2 pureXML* detalla más acerca de XML y esquemas XML.

14.3 Desarrollo del lado del Cliente

Como el nombre lo sugiere, en un desarrollo del lado del cliente, los programadores de las aplicaciones codifican en un cliente y luego se conectan y acceden a la base de datos DB2 utilizando las interfaces de programa de aplicación (APIs) que vienen incluídas con DB2. En esta sección se discute lo siguiente:

- SQL embebido
- SQL Dinámico y Estático
- CLI y ODBC
- JDBC, SQLJ y pureQuery
- OLE DB
- ADO.NET
- PHP
- Ruby on Rails (Ruby sobre Ruedas)
- Perl
- Python

14.3.1 SQL embebido

Las aplicaciones de SQL embebidas son aplicaciones donde el SQL está incorporado dentro de un lenguaje anfitrión, como ser C, C++, o COBOL. La aplicación que tiene el SQL embebido puede incluir SQL estático o dinámico como se describirá en la próxima sección. La Figura 14.2 muestra cómo se construye una aplicación con SQL embebido.

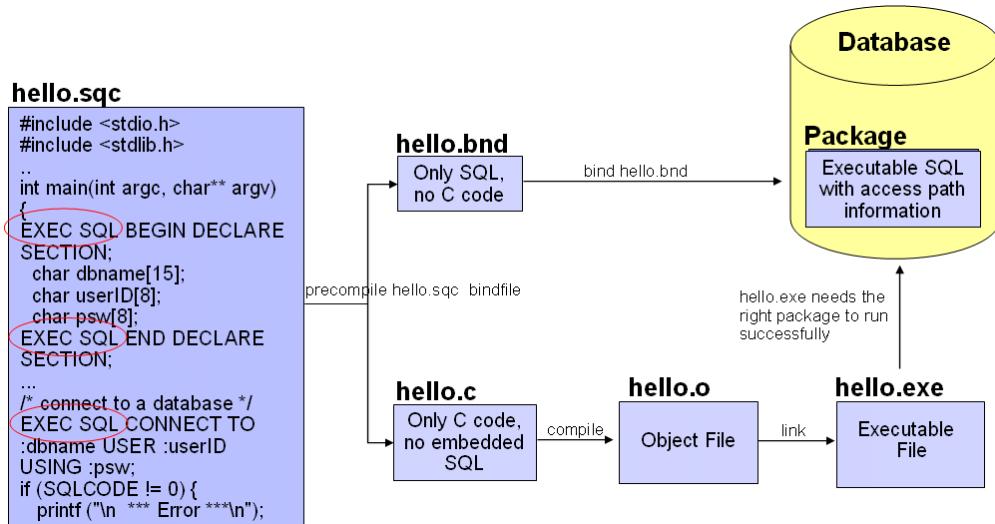


Figura 14.2 – Construyendo aplicaciones con SQL embebido

En la figura, el programa en C **hello.sqc** contiene SQL embebido. La API de SQL para el lenguaje C utiliza EXEC SQL (resaltado en la *Figura 14.2*) para permitir un proceso de precompilación que distingue entre las sentencias SQL embebidas y el código C actual. También se puede ver en el código de **hello.sqc** algunas variables prefijadas con dos puntos (:), como es el caso de **:dbname**, **:userID**, y **:psw**. Estas se denominan variables anfitrión (host variables). Estas variables pertenecen al lenguaje anfitrión y se refieren en las sentencias SQL embebidas.

La invocación del comando **precompile** (también conocido como el comando **prep**) en conjunto con la opción **bindfile**, genera dos archivos, el archivo de enlace (bind) **hello.bnd**, que sólo contiene las sentencias SQL y el archivo **hello.c**, que contiene sólo código en lenguaje C (lenguaje anfitrión). El archivo de enlace será compilado usando el comando **bind** para obtener un **paquete** que se guardará en la base de datos. Un paquete incluye el SQL compilado/ejecutable y el camino de acceso (plan de ejecución) que DB2 seguirá para obtener los datos. Para invocar el comando **bind**, debe existir una conexión a la base de datos. Al final de la figura, se compila y enlaza el archivo **hello.c** como se hace con cualquier programa desarrollado en C. El archivo ejecutable resultante, **hello.exe**, tiene que tener concordancia con el paquete almacenado en la base de datos para que se pueda ejecutar satisfactoriamente.

14.3.2 SQL Estático vs. SQL Dinámico

Las sentencias de **SQL Estático** corresponden a aquellas cuya estructura SQL se resuelve en su totalidad al momento de precompilación. Por ejemplo:

```
SELECT lastname, salary FROM employee
```

En este ejemplo, los nombres de las columnas (`lastname, salary`) y de la tabla (`employee`) que hace referencia la sentencia, se conocen al momento de precompilación. El siguiente ejemplo también es una sentencia estática de SQL:

```
SELECT lastname, salary FROM employee WHERE firstnme = :fname
```

En este segundo ejemplo, una variable anfitriona `:fname` se usa como parte de una sentencia SQL embebida. Aunque el valor de la variable anfitriona se desconoce hasta el momento de la ejecución, su tipo de dato es desconocido para el programa, y todos los otros objetos (nombre de columnas, nombre de tablas) sí lo son. DB2 usa estimaciones para estas variables anfitrionas con el objetivo de calcular el plan de acceso antes de tiempo, por lo tanto, este caso es todavía considerado SQL estático.

Se precompilan, enlazan, y compilan sentencias SQL ejecutadas estáticamente antes de ejecutar la aplicación. El SQL Estático es ideal para aquellas bases de datos donde las estadísticas no cambian mucho. A continuación, veremos el siguiente ejemplo:

```
SELECT ?, ? FROM ?
```

Aquí, los nombres de las columnas y de la tabla referenciadas en la sentencia no se conocen hasta el momento de ejecución. Por lo tanto, el plan de acceso se calcula sólamente cuando se ejecuta el programa y usando las estadísticas disponibles en ese momento. Estos tipos de sentencias son consideradas sentencias SQL Dinámicas.

Algunas interfaces de programación (APIs), como JDBC y ODBC, siempre usan SQL dinámico independientemente de si la sentencia SQL incluye objetos conocidos o no. Por ejemplo, la sentencia `SELECT lastname, salary FROM employee` tiene todos los nombres de las columnas y de las tablas definidas de antemano, pero a través de JDBC o ODBC, las sentencias no son precompiladas. Todos los planes de acceso para las sentencias se calculan en tiempo de ejecución.

En general, dos sentencias se utilizan para tratar una sentencia SQL como dinámica:

- **PREPARE:** Esta instrucción prepara o compila la sentencia SQL calculando el plan de acceso que se utilizará para obtener los datos
- **EXECUTE:** Esta sentencia ejecuta el SQL en cuestión

Como alternativa, también se puede ejecutar **PREPARE** y **EXECUTE** en una sentencia sola: **EXECUTE IMMEDIATE**

El Listado 14.1 muestra un ejemplo de un SQL dinámico embebido en un programa en C, preparado y ejecutado.

```
strcpy(hVStmtDyn, "SELECT name FROM emp WHERE dept = ?");  
PREPARE StmtDyn FROM :hVStmtDyn;  
EXECUTE StmtDyn USING 1;  
EXECUTE StmtDyn USING 2;
```

Listado 14.1 – Una sentencia SQL dinámica embebida en C utilizando PREPARE and EXECUTE

El Listado 14.2 muestra el mismo ejemplo que el Listado 14.1, pero usando la sentencia EXECUTE IMMEDIATELY

```
EXECUTE IMMEDIATELY SELECT name from EMP where dept = 1
EXECUTE IMMEDIATELY SELECT name from EMP where dept = 2
```

Listado 14.2 – Una sentencia SQL dinámica embebida en C utilizando EXECUTE IMMEDIATELY

En muchos lenguajes de programación dinámicos tales como PHP o Ruby on Rails, donde las sentencias SQL se ejecutan dinámicamente, los programadores tienden a escribir las mismas sentencias SQL con diferentes valores en los campos como muestra el siguiente ejemplo:

```
SELECT lastname, salary FROM employee where firsttnme = 'Raul'
SELECT lastname, salary FROM employee where firsttnme = 'Jin'
...

```

En este ejemplo, las sentencias son idénticas exceptuando el valor de la columna `firsttnme`. DB2 considera estas instrucciones dinámicas SQL como diferentes, y por lo tanto, las prepara y ejecuta de forma independiente. El procesamiento necesario para preparar la misma sentencia varias veces puede provocar degradación en el rendimiento de la aplicación, por lo tanto antes de DB2 9.7, la recomendación era codificar las sentencias de la siguiente manera:

```
SELECT lastname, salary FROM employee where firsttnme = ?
```

El signo de pregunta (?) en la sentencia se muestra como un *indicador de parámetro*. Usando los indicadores de parámetros, el programa puede preparar la instrucción solamente una vez, y luego invocar las sentencias `EXECUTE` donde se le pasa al indicador de parámetros los diferentes valores deseados.

new in
V9.7

En DB2 9.7 se incorpora una nueva tecnología llamada **concentrador de sentencia**, donde todas las instrucciones idénticas excepto por los valores de los campos, son automáticamente agrupadas en una sola sentencia con indicadores de parámetros, y luego se realizan las sentencias `EXECUTE` con los respectivos valores. El concentrador de sentencia tiene la inteligencia para determinar cuándo no agrupar ciertas sentencias, por ejemplo, cuando se agregan cláusulas para influenciar al optimizador de DB2.

Respecto del rendimiento, el SQL Estático tendrá un mayor rendimiento que el SQL Dinámico debido a que el plan de acceso del SQL estático se efectúa en tiempo de precompilación y no en tiempo de ejecución. Sin embargo, para ambientes donde ocurre mucha actividad de operaciones de `INSERT` y de `DELETE`, las estadísticas calculadas en tiempo de precompilación podrían no estar actualizadas, lo que indica que el plan de acceso del SQL estático podría no ser óptimo. En este caso, el SQL dinámico puede ser

una mejor elección, asumiendo que el comando RUNSTATS se ejecuta con frecuencia para recolectar estadísticas.

Nota:

Muchos usuarios piensan que el SQL embebido es solamente estático. En realidad, puede ser ambos, estático ó dinámico.

14.3.3 CLI y ODBC

La *Interface de Nivel de Llamada (Call Level Interface) (CLI)* fue originalmente desarrollada por X/Open Company y el SQL Access Group. Fue una especificación para una interface SQL invocable con el propósito de desarrollar aplicaciones C/C++ portables sin importar el proveedor del motor de base de datos (RDBMS). Basada en un borrador preliminar de X/Open Call Level Interface, Microsoft desarrolló **Conectividad Abierta de Bases de Datos (Open Database Connectivity) (ODBC)**, y más tarde, el ISO CLI International Standard, aceptó la mayoría de la especificación de X/Open Call Level Interface. DB2 CLI se basa en ambos: ODBC y el International Standard para SQL/CLI como se muestra en la Figura 14.3.

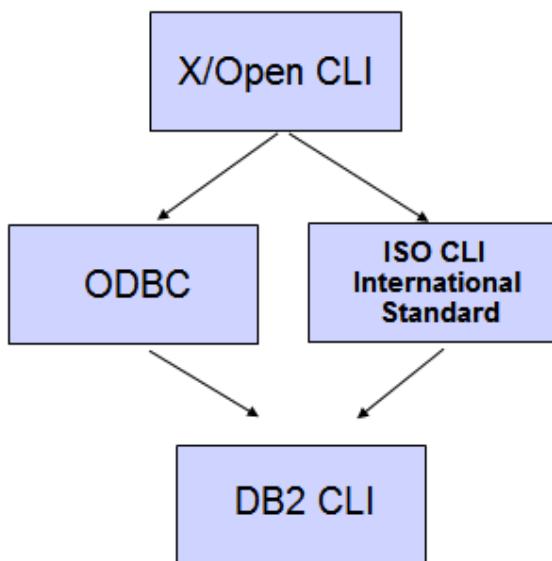


Figura 14.3 - DB2 CLI se basa en ODBC y el Estándar Internacional ISO CLI

DB2 CLI cumple con ODBC 3.51 y se puede usar como ODBC cuando se carga con un Administrador de Manejadores de ODBC. La Figura 14.4 puede ayudar a indicar el soporte de DB2 CLI para ODBC.

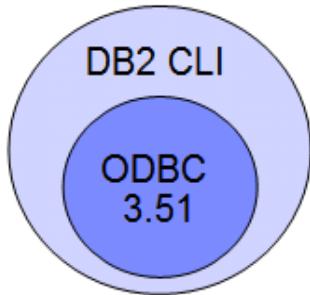


Figura 14.4 - DB2 CLI cumple con el ODBC 3.51

CLI/ODBC tiene las siguientes características:

- El código es fácilmente portable entre varios proveedores de sistemas de gestión de bases de datos (RDBMS)
- A diferencia del SQL embebido, no existe necesidad de un precompilador ni de variables anfitrionas
- Ejecuta SQL dinámico
- Es muy popular

Para [ejecutar](#) una aplicación CLI/ODBC, todo lo que se necesita es el manejador DB2 CLI. Este manejador se instala desde cualquiera de los siguientes clientes y manejadores que pueden ser descargados y utilizados de forma gratuita en www.ibm.com/db2/express:

- IBM Data Server Client
- IBM Data Server Runtime Client
- IBM Data Server Driver for ODBC and CLI

Para [desarrollar](#) una aplicación CLI/ODBC se necesita el manejador DB2 CLI y sus bibliotecas correspondientes. Estas se encuentran dentro del IBM Data Server Client.

En el siguiente ejemplo se puede comprender de mejor forma cómo instalar el manejador DB2 CLI para las aplicaciones. La *Figura 14.5* muestra tres diferentes máquinas, una en Indonesia, la otra en Brasil, y otra en Canadá.

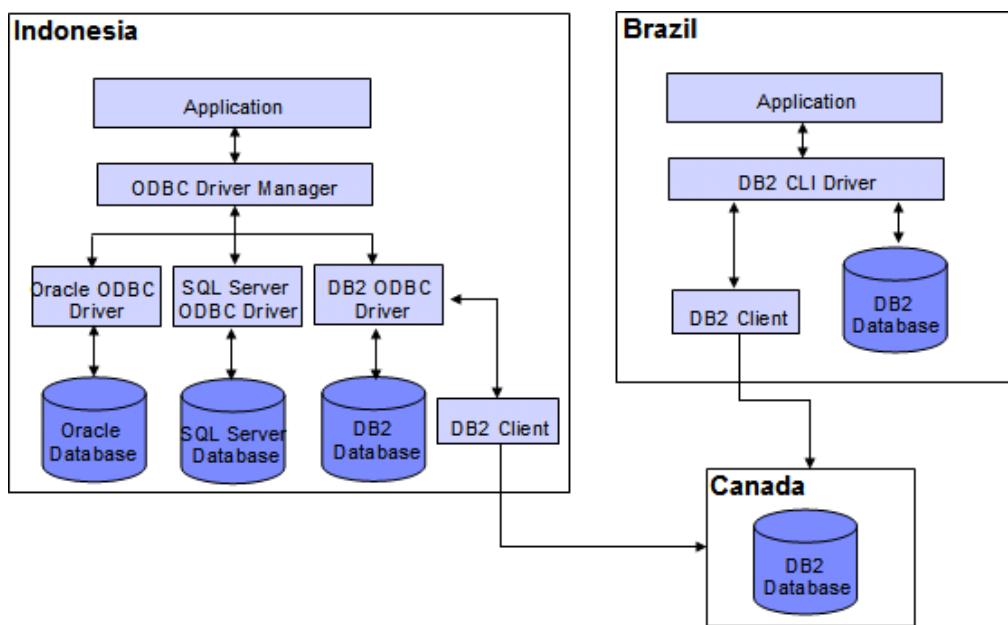


Figura 14.5 – Escenario de ejemplo de DB2 CLI/ODBC

Esta figura muestra dos situaciones:

En el lado izquierdo, la máquina de Indonesia está ejecutando una *aplicación ODBC* la cual podría trabajar con cualquier sistema de gestión de base de datos, como ser Oracle, Microsoft SQL Server o DB2. El Administrador del Manejador de ODBC (ODBC Driver Manager) cargará el manejaror ODBC adecuado dependiendo de la base de datos a la cual se accede. En este caso, donde la aplicación accede a una base de datos sita en Canadá, la conexión necesita ir a través del Cliente DB2 el cual tiene los componentes necesarios para conectarse remotamente.

En el lado derecho, una *aplicación CLI* se ejecuta en una máquina en Brazil. Es una aplicación CLI debido a que puede estar utilizando funciones que no están disponibles en ODBC, y también porque la aplicación solamente trabajará para una base de datos DB2. La aplicación CLI pasará a través del Manejador CLI de DB2 (DB2 CLI Driver). La aplicación puede conectarse a la base de datos local de DB2 en Brazil. Cuando necesite conectarse a la base remota de Canadá, lo hará a través del cliente DB2.

Para finalizar esta sección, se mostrará una comparativa entre una aplicación CLI/ODBC y una aplicación en lenguaje C con SQL dinámico embebido. La Figura 14.6 ilustra esta comparativa.

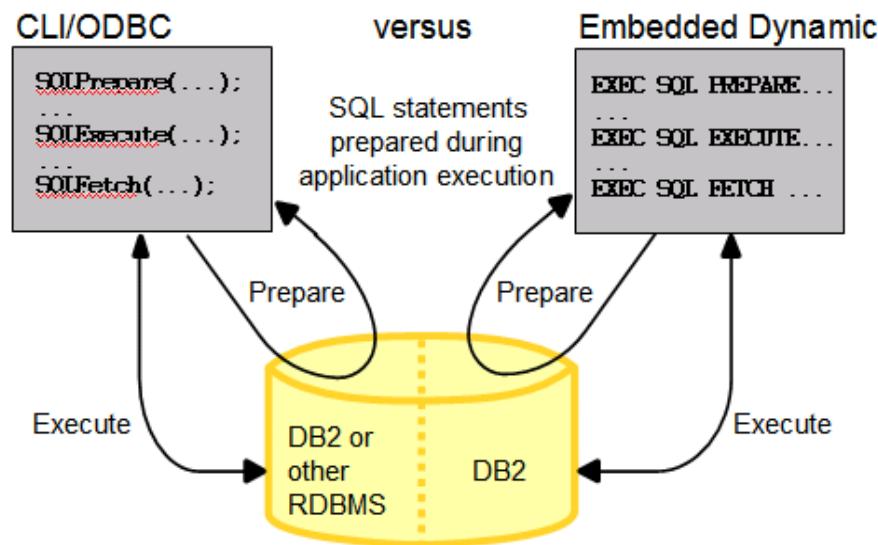


Figura 14.6 – Una aplicación CLI/ODBC versus una aplicación C con SQL dinámico embebido

Como se muestra en la *Figura 14.6*, la única diferencia entre CLI/ODBC vs. SQL dinámico embebido en C es que el código de CLI/ODBC es portable y puede acceder a otros sistemas de gestión de base de datos con el sólo hecho de cambiar la cadena de conexión, mientras que en la versión del SQL dinámico embebido en C, se codifican elementos específicos para DB2. También es de notar la diferencia en la forma de invocar las funciones **PREPARE** y **EXECUTE**.

14.3.4 JDBC, SQLJ y pureQuery

La **Conectividad a la base de Datos con Java (Java Database Connectivity) (JDBC)** es una API de programación Java que estandariza las formas para trabajar y acceder a las bases de datos. El código JDBC es fácilmente portable entre diferentes proveedores de sistemas de gestión de base de datos. El único cambio requerido en el código es la elección del manejador JDBC y la cadena de conexión. JDBC es muy popular y usa sólo SQL dinámico.

SQLJ es el estándar para el SQL embebido en programas Java. Es principalmente utilizado con SQL estático, aunque puede inter-operar con JDBC como se muestra en la *Figura 14.7*. Aunque generalmente es más compacto que los programas JDBC y provee mejor rendimiento, no ha sido ampliamente aceptado. Los programas SQLJ deben correr a través de un preprocesador (el traductor SQLJ) antes de ser compilados.

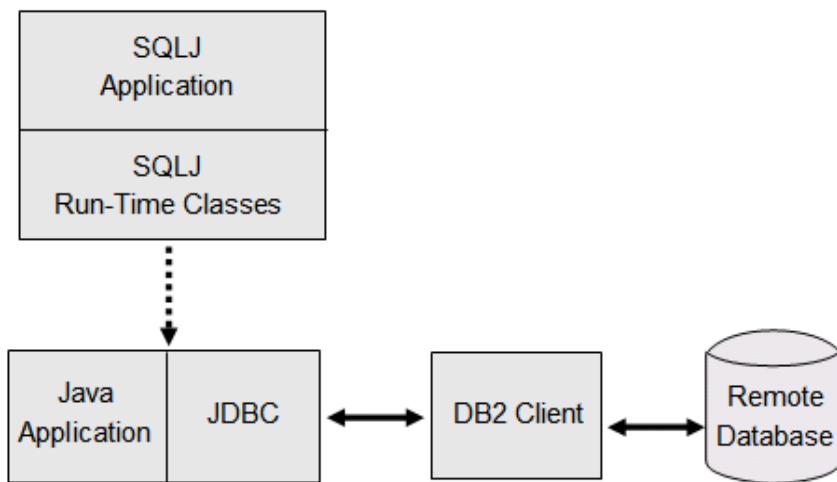


Figura 14.7 – Relación entre las aplicaciones SQLJ y JDBC

En la *Figura 14.7*, un cliente DB2 puede o no necesitarse dependiendo del tipo de manejador de JDBC utilizado como se discutió en esta sección.

pureQuery es un conector (plug-in) de IBM basado en Eclipse para administrar datos relacionales como objetos. pureQuery está disponible desde 2007; puede generar código de forma automática para establecer un mapeo objeto-relacional (ORM) entre código orientado a objetos y los objetos de base de datos relacionales. Se comienza creando un proyecto Java con Optim Development Studio (ODS), conectándose a una base de datos DB2, y luego permitiendo a ODS descubrir todos los objetos de la base de datos. A través de la interfaz ODS se puede elegir una tabla y generar el código pureQuery, el cual transforma todas las entidades de la tabla relacional en un objeto Java. El código se genera para crear las sentencias SQL relevantes y encapsular en objetos Java estas sentencias. Los objetos Java generados y las sentencias SQL incluidas, pueden ser personalizadas luego en cualquier momento. Con pureQuery, se puede decidir ejecutar el SQL de forma estática ó dinámica. pureQuery soporta tanto Java como .NET.

14.3.4.1 Manejadores (drivers) JDBC y SQJ

Aunque existen varios tipos de manejadores JDBC, como el tipo 1, el 2, el 3 y el 4; el tipo 1 y el 3 generalmente no se usan, y el soporte de DB2 de estos tipos está discontinuado. Para el tipo 2, hay dos manejadores que se describirán más adelante, pero uno de ellos también está obsoleto.

El tipo 2 y el 4 los soporta DB2, como se muestra en la *Tabla 14.2*. Los manejadores tipo 2 tienen que tener el cliente DB2 instalado, ya que el manejador lo usa para establecer la comunicación con la base de datos. El tipo 4 es un cliente Java puro, por lo que no es necesario un cliente DB2, sino que el manejador se debe instalar en la máquina donde se está ejecutando la aplicación.

Tipo de Manejador	Nombre del Manejador	Empaquetado como	Soporta	Mínimo nivel de SDK requerido para Java
Tipo 2	Manejador DB2 JDBC de Tipo 2 para Linux, UNIX y Windows (Obsoleto*)	db2java.zip	JDBC 1.2 y JDBC 2.0	1.4.2
Tipo 2 y Tipo 4	Manejador de IBM Data Server para JDBC y SQLJ	db2jcc.jar y sqlj.zip	JDBC 3.0 compliant	1.4.2
		db2jcc4.jar y sqlj4.zip	JDBC 4.0 y earlier	6

Tabla 14.2 – Manejadores DB2 JDBC y SQLJ

* Obsoleto significa que todavía se soporta, pero que no recibe más mejoras

Como se mencionó antes y se mostró en la *Tabla 14.2*, el Tipo 2 se provee con dos manejadores diferentes; sin embargo, el Manejador Tipo 2 de DB2 JDBC para Linux, UNIX y Windows, con el nombre de archivo db2java.zip es obsoleto.

Cuando se instala un servidor DB2, un cliente DB2 o el Manejador de IBM Data Server para JDBC y SQLJ, los archivos db2jcc.jar y sqlj.zip que cumplen con JDBC 3.0 son agregados automáticamente al directorio de classes (classpath).

14.3.5 OLE DB

La **Base de Datos de Objetos Enlazados y Embebidos (Object Linking and Embedding, Database) (OLE DB)** es un conjunto de interfaces que provee acceso a los datos almacenados en diversas fuentes. Fue diseñado en reemplazo del ODBC, pero extendió su soporte a una variedad más amplia de fuentes, incluyendo bases de datos no relacionales, tales como bases de datos orientadas a objetos y hojas de cálculo. OLE DB se implementa por medio de la tecnología Modelo de Objeto Componente (Component Object Model) (COM).

Los clientes OLE DB pueden acceder a una base de datos DB2 con el Proveedor IBM OLE DB para DB2. Este proveedor tiene las siguientes características:

- Nombre de Proveedor: IBMDADB2
- Soporta nivel 0 de la especificación del proveedor de OLE DB, incluyendo algunas interfaces adicionales del nivel 1
- Cumple con la Versión 2.7 o posterior de la especificación de Microsoft OLE DB

- Se debe instalar un Cliente del Servidor de Datos IBM (IBM Data Server Client) en conjunto con los Componentes de Acceso y Datos Microsoft (Microsoft Data Access Components) (MDAC)
- Si IBMDADB2 no está explícitamente especificado, el manejador de Microsoft OLE DB (MSDASQL) será el utilizado por omisión. MSDASQL le permite a los clientes utilizar OLE DB para acceder a orígenes de datos que no son de Microsoft SQL por medio del manejador ODBC, pero no garantiza total funcionalidad del manejador OLE DB.

14.3.6 ADO.NET

El **Marco de Trabajo .NET (.NET Framework)** es el reemplazo de Microsoft para la tecnología COM mencionada anteriormente. Por medio del Marco de Trabajo .NET, se pueden codificar aplicaciones .NET en más de cuarenta lenguajes de programación diferentes; siendo los más populares C# y Visual Basic .NET.

La biblioteca de clase del .NET Framework, provee los bloques de construcción necesarios para desarrollar aplicaciones .NET. Esta biblioteca de clases es un lenguaje agnóstico que provee interfaces para el sistema operativo y para los servicios de aplicación. La aplicación .NET (sin importar el lenguaje) compila en un Lenguaje Intermedio (IL), una especie de código de byte (bytecode).

El Lenguaje Común de Ejecución (Common Language Runtime) (CLR) es el corazón del Marco de Trabajo .NET, se compila el código IL al momento y luego se ejecuta. Al ejecutar el código IL compilado, el CLR activa objetos, verifica su acceso de seguridad, reserva memoria, los ejecuta, y limpia la memoria una vez que la ejecución finalizó.

Como una analogía a la forma de trabajar de Java, en Java, un programa puede correr en múltiples plataformas con mínima o sin modificación alguna: un lenguaje, múltiples plataformas. En .NET, un programa escrito en cualquiera de casi cuarenta lenguajes que soporta puede ejecutarse en una sola plataforma, Windows, con mínima o sin intervención: múltiples lenguajes, pero una plataforma.

ADO.NET trata de proveer el soporte de acceso a los datos en el Marco de Trabajo .NET. ADO.NET soporta tanto acceso con conexión como sin conexión. El componente clave del acceso a los datos sin conexión en ADO.NET es la clase **DataSet**, que son instancias que actúan como una porción memoria de la base de datos que reside en la memoria total de la aplicación.

Tanto como para el acceso con conexión como el sin conexión, sus aplicaciones usan base de datos a través de lo que se conoce como **proveedor de datos (data provider)**. Varios productos de base de datos incluyen sus propios proveedores de datos .NET, incluyendo DB2 para Windows.

Un proveedor de datos .NET consta de implementaciones de las siguientes clases básicas:

- Connect (Conexión): establece y administra una conexión de base de datos.
- Command (Comando): ejecuta una sentencia SQL contra la base de datos.
- DataReader (Lector de Datos): lee y retorna los datos de un conjunto de resultados desde una base de datos.
- DataAdapter (Adaptador de Datos): enlaza una instancia de un Conjunto de Datos (DataSet) a una base de datos. A través de una instancia del Adaptador de Datos, el Conjunto de Datos puede leer y escribir datos de la tabla de una base de datos.

En la *Tabla 14.3* se muestran tres proveedores de datos que pueden trabajar con DB2

Proveedor de Datos	Características
Proveedor de Datos ODBC .NET (no recomendado)	<ul style="list-style-type: none"> ▪ Se hacen llamadas ODBC hacia un origen de datos DB2 por medio del manejador DB2 CLI. ▪ Tiene el mismo soporte y las mismas restricciones que el manejador de DB2 CLI ▪ Se puede usar con el Marco de Trabajo .NET Version 1.1, 2.0, o 3.0.
Proveedor de Datos OLE DB .NET (no recomendado)	<ul style="list-style-type: none"> ▪ Utiliza el Manejador de IBM DB2 OLE DB (IBMDADB2). ▪ Tiene el mismo soporte y las mismas restricciones que el manejador DB2 OLE DB ▪ Se puede utilizar sólamente con el Marco de Trabajo .NET Version 1.1, 2.0, o 3.0.
Proveedor de Datos DB2 .NET (recomendado)	<ul style="list-style-type: none"> ▪ Amplía el soporte de DB2 para la interface ADO.NET. ▪ El proveedor administrado por DB2 implementa el mismo conjunto de clases y métodos estándar de ADO.NET ▪ Está definido bajo el espacio de nombres IBM.DATA.DB2. ▪ Se puede obtener descargando cualquiera de los siguientes programas: <ul style="list-style-type: none"> - Manejador de Servidores de Datos (Data

	<p>Server Driver) para ODBC, CLI, y .NET</p> <ul style="list-style-type: none"> - Cliente de Ejecución del Servidor de Datos IBM - Servidor de Datos DB2
--	--

Tabla 14.3 – Proveedores de datos ADO.NET

14.3.7 PHP

Preprocesador de Hipertexto (Hypertext Preprocessor) (PHP) es un lenguaje de código abierto, independiente de la plataforma y diseñado para el desarrollo de aplicaciones Web. Se puede embeber dentro de código HTML, y comúnmente se ejecuta en un servidor Web que recibe el código PHP y crea las páginas Web como salida.

PHP es un lenguaje modular. Se pueden usar extensiones para personalizar la funcionalidad disponible. Algunas de las extensiones PHP más comunes son aquellas utilizadas para acceder a las bases de datos. IBM soporta acceso a las bases de datos a través de dos extensiones:

- **ibm_db2**: La extensión ibm_db2 brinda una interface de programación de aplicación procedural, para crear, leer, actualizar y escribir operaciones de bases de datos además de extender el acceso hacia el diccionario de datos (metadata) de la base de datos. Se puede compilar en PHP 4 o en PHP 5.
- **pdo_ibm**: El manejador pdo_ibm sirve para los Objetos de Datos PHP (PHP Data Objects) (PDO). Ofrece acceso a las bases de datos DB2 a través de la interface de base de datos estándar orientada a objetos incorporada en PHP 5.1. Se puede compilar directamente contra las bibliotecas de DB2.

Las extensiones PHP y los manejadores están disponibles de forma gratuita en el repositorio PECL en <http://pecl.php.net/>, o con el Cliente del IBM Data Server. Ambos, ibm_db2 y pdo_ibm, se basan en la Capa IBM DB2 CLI

IBM también ha acompañado a Zend Technologies Inc. para soportar ZendCore, que es un conjunto de herramientas sin costo para hacer desarrollos con PHP y DB2 Express-C. El ZendCore incluye bibliotecas PHP, servidor Web Apache, y DB2 Express-C. Para descargar ZendCore, ir a <http://www.ibm.com/software/data/info/zendcore>

14.3.8 Ruby on Rails (Ruby sobre Ruedas)

Ruby es un lenguaje de código abierto orientado a objetos. Rails es un ambiente de trabajo creado utilizando Ruby. Ruby on Rails (RoR) es un medio ideal para desarrollar aplicaciones basadas en web con bases de datos. Esta nueva tecnología se centra en la arquitectura Modelo, Vista, Controlador (Model, View, Controller) (MVC) y sigue los principios del desarrollo ágil de software.

Rails no requiere ningún formato de archivo especial o ambientes de desarrollos integrados (IDEs); se puede comenzar con un editor de texto. Sin embargo, varios IDEs están disponibles con Rails, tales como RadRails, que es un ambiente Rails para Eclipse. Para más información sobre RadRails, visitar <http://www.radrails.org/>.

DB2 soporta Ruby 1.8.5 y Ruby on Rails 1.2.1, así como nuevas versiones de ambos. El IBM_DB incluye el manejador de Ruby y el adaptador de Rails, que permiten trabajar con DB2 y se basan en la capa CLI. IBM_DB se debe instalar con un Cliente IBM Data Server.

14.3.9 Perl

Perl es un lenguaje de programación interpretado muy conocido, que está disponible de forma gratuita para varios sistemas operativos. Utiliza SQL dinámico, y es ideal para aplicaciones desarrolladas por prototipos.

Perl provee un módulo estándar denominado Interface de Base de Datos (Database Interface) (DBI) para acceder a diferentes bases de datos. Está disponible en <http://www.perl.com>. Este módulo “le habla” a los manejadores (drivers) desde diferentes proveedores de base de datos. En el caso de DB2, el encargado de esta tarea es el manejador DBD::DB2, el cual está disponible en <http://www.ibm.com/software/data/db2/perl>.

14.3.10 Python

Python es un lenguaje dinámico generalmente usado para scripts. Hace hincapié en la lectura de código y soporta varios paradigmas de programación, incluyendo procedural, orientado a objetos, orientado a aspectos, meta, y programación funcional. Python es ideal para el desarrollo rápido de aplicaciones.

La Tabla 14.4 muestra las extensiones que están disponibles para acceder a las bases de datos DB2 desde una aplicación Python.

Extensión	Descripción
ibm_db	Definido por IBM Provee el mejor soporte para características avanzadas. Permite ejecutar consultas SQL, llamar a procedimientos almacenados, usar pureXML, y acceder a la información del diccionario de datos (metadata).
ibm_db_dbi	Implementa la Especificación v2.0 de la API de Python de base de datos (Python Database API Specification v2.0). No ofrece algunas características avanzadas que el API de ibm_db soporta.

	Si se tiene una aplicación con un manejador que soporta Python Database API Specification v2.0, fácilmente se puede cambiar a ibm_db. Las APIs ibm_db y ibm_db_dbi vienen empaquetadas juntas.
ibm_db_sa	Soporta SQLAlchemy, un conjunto de herramientas SQL de Python de código abierto y con mapeo objeto-relacional.

Tabla 14.4 - IBM Data Server – Extensiones de Python

14.4 XML y DB2 pureXML

El lenguaje de Marcas Extensible (Extensible Markup Language) (XML) es la tecnología que está por debajo de las herramientas y técnicas de la Web 2.0, al igual que *La Arquitectura Orientada al Servicio (Service Oriented Architecture) (SOA)*. IBM ha reconocido desde hace tiempo la importancia del XML, por lo que ha hecho grandes inversiones para entregar tecnología pureXML – una tecnología que provee mejor soporte para almacenar documentos XML en DB2.

Incluído en 2006, DB2 9 es un servidor de datos híbrido: permite almacenamiento nativo de datos relacionales así como de datos jerárquicos. Mientras versiones previas de DB2 y otros servidores de datos del mercado podían almacenar documentos XML, el método usado por DB2 9 ha mejorado el rendimiento y la flexibilidad. Con la tecnología de pureXML de DB2 9, los documentos XML se guardan internamente de forma jerárquica, como un árbol; por lo tanto, trabajar con documentos XML es una mejora sustancial. Las últimas versiones de DB2, como DB2 9.5 y DB2 9.7, han ido más allá y han mejorado el soporte para pureXML. *El Capítulo 15, pureXML en DB2* se dedica a esta temática en detalle.

14.5 Servicios Web (Web Services)

Como una simple definición, se puede pensar que un servicio Web es una función que se invoca a través de la red, donde no se necesita saber el lenguaje de programación utilizado para desarrollarlo, tampoco el sistema operativo ni la ubicación donde esta función se ejecutará. Los servicios Web permiten que una aplicación intercambie datos con otra por medio de los protocolos estándar basados en XML. Esto se muestra en la *Figura 14.8*.

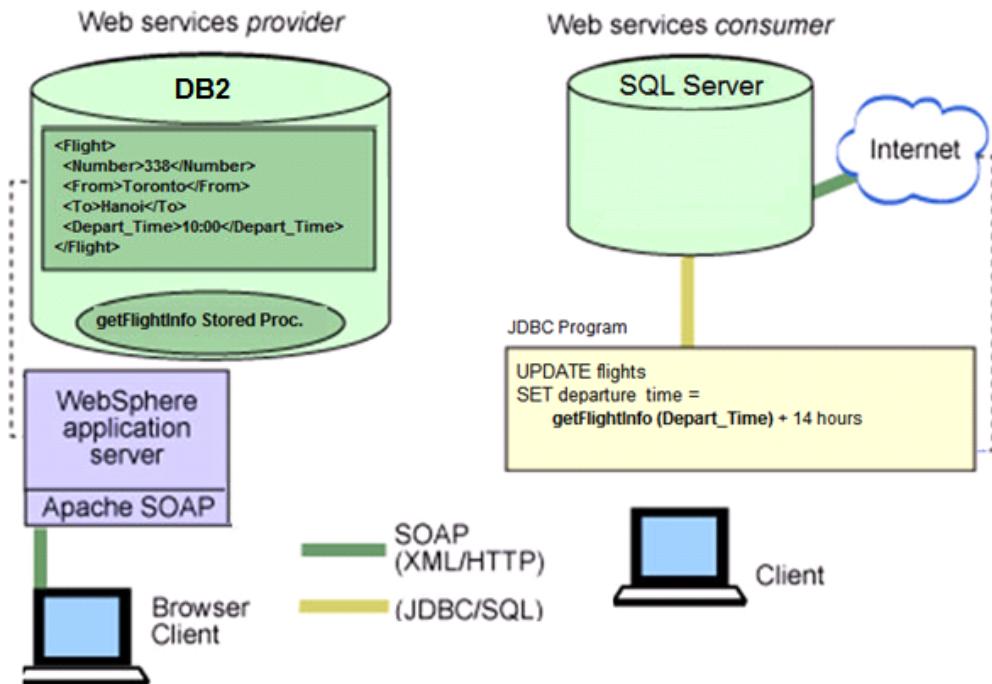


Figura 14.8 – Ejemplo de cómo trabaja un servicio Web

En la figura, digamos que el lado izquierdo representa el sistema de una aerolínea ficticia, Air Atlantis, la cual utiliza DB2 en Linux, y guarda la información de sus vuelos en formato XML en la base de datos DB2. En el lado derecho, tenemos un sistema de otra aerolínea, Air Discovery, la cual utiliza SQL Server sobre Windows. Ahora digamos que Air Atlantis y Air Discovery firman un acuerdo como socios donde las dos compañías desean compartir la información de horarios de vuelos y precios con el objetivo de coordinar sus vuelos. Compartir información entre las dos compañías puede ser un desafío ya que las dos compañías usan sistemas operativos diferentes (Linux, Windows), y distintos servidores de datos (DB2, SQL Server). Cuando Air Atlantis cambia el horario de un vuelo de Toronto a Beijing, ¿Cómo puede Air Discovery de forma automática, ajustar su agenda de vuelos para la conexión de un vuelo desde Beijing a Shanghai? La respuesta es: servicios Web (Web services). Air Atlantis puede exponer algo de la información de sus vuelos creando un **servicio Web de Datos (Data Web service)** que devuelve la salida de un procedimiento almacenado, `getFlightInfo`, con información del vuelo de la base de datos DB2. Un **servicio Web de Datos (Data Web service)** es un servicio Web basado en información de la base de datos. Cuando este servicio Web de Datos se implanta en un servidor de aplicaciones como por ejemplo en WebSphere Application Server; un cliente o socio como Air Discovery puede, muy fácilmente, usar un navegador para acceder a la información de vuelos de Air Atlantis. En este ejemplo, Air Atlantis se comporta como un proveedor de servicios Web, habiendo desarrollado y puesto a disposición estos servicios.

Web, mientras que Air Discovery se comporta como un consumidor de estos servicios, que los usa y consume.

Air Discovery también puede invocar el servicio Web desde su propia aplicación JDBC de forma tal que puede realizar el cálculo de los datos de su base de datos SQL Server. Por ejemplo, si un vuelo desde Toronto a Beijing dura en promedio 12 horas, Air Discovery puede calcular el horario de salida del vuelo de conexión desde Beijing a Shanghai sumándole al tiempo de salida del vuelo Air Atlantis desde Toronto a Beijing la duración del vuelo mas unas horas de estadía del avión. La cantidad de horas de estadía del avión se pueden guardar en la base de datos SQL Server en el sistema de Air Discovery, y la ecuación a usar en la aplicación JDBC se vería así:

Air Discovery Flight Departure time (Beijing to Shanghai)	=	Air Atlantis Flight Departure time (Toronto to Beijing)	+	Air Atlantis Flight Duration (Toronto to Beijing)	+	Air Atlantis Flight Buffer time (Toronto to Beijing)
				12 hours average		2 hours

Si Air Atlantis cambia el horario de salida de su vuelo, esta información se le comunica automáticamente al sistema de Air Discovery cuando invoca el servicio Web.

14.6 APIs Administrativas

DB2 proporciona una amplia variedad de APIs administrativas que los desarrolladores pueden usar para construir sus propias utilidades o herramientas. Por ejemplo, para crear una base de datos se puede invocar a la **API sqlecrea**; para iniciar una instancia, sería la **API db2InstanceStart**; o para importar los datos a una tabla, la **API db2Import**. La lista completa se encuentra disponible en el Centro de Información de DB2 (DB2 Information Center), cuya URL se encuentra en la sección *Recursos* al final de este libro.

14.7 Otros desarrollos

Algunos usuarios de DB2 Express-C también pueden interactuar con productos de terceros tales como MS Excel y MS Access para crear formularios sencillos que se conectan a DB2. En esta sección describimos cómo trabajar con estos productos y DB2 Express-C.

DB2 Express-C está disponible también en Mac OS X, de forma tal que se puede usar DB2 de forma nativa para desarrollar aplicaciones de base de datos en una Mac. Esto puede resultar atractivo especialmente para la comunidad RoR (Ruby on Rails) que se ha apegado tanto a la plataforma Mac.

14.7.1 Trabajando con Microsoft Access y Microsoft Excel

Microsoft Excel y Microsoft Access son dos herramientas populares que se utilizan para generar reportes, crear formularios y desarrollar aplicaciones simples que proveen algo de

inteligencia de negocios a los datos. DB2 interactúa muy fácilmente con estas herramientas. Un DBA puede almacenar los datos de la compañía en un servidor DB2 seguro, y los usuarios habituales de Access o Excel pueden acceder a estos datos y generar reportes. Esta situación se ilustra en la *Figura 14.9*.

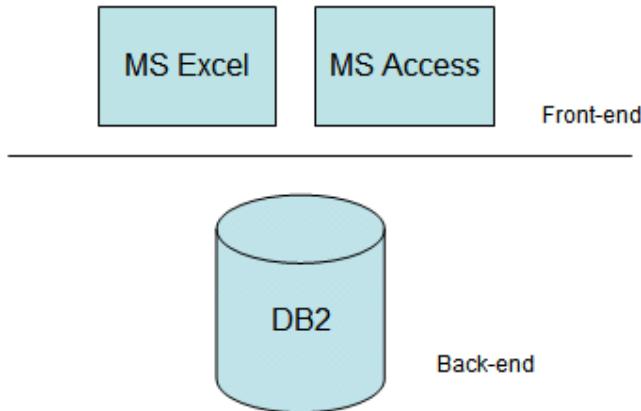


Figura 14.9 – Trabajando con Excel, Access y DB2

En la figura, Excel y Access se pueden usar para desarrollar una aplicación cliente (front-end), mientras DB2 se encarga de la seguridad de los datos, la confiabilidad y el rendimiento de la aplicación y del servidor (back-end). Con todos los datos centralizados en DB2, se creará un modelo de almacenamiento de datos simplificado.

En el caso de Excel, la forma más fácil de acceder a los datos de DB2 es a través de un manejador OLE DB, por ejemplo el Proveedor IBM OLE DB para DB2. Este viene incluido cuando se instala la herramienta gratuita IBM Data Server Client, que puede ser descargada del sitio de DB2 Express-C en www.ibm.com/db2/express. Una vez instalado, se selecciona el origen de datos con el proveedor OLE DB adecuado desde el menú de MS Excel. Elegir *Datos* → *Importar Datos Externos* → *Importar Datos*. Los próximos pasos están documentados en el artículo *IBM® DB2® Universal Database™ and the Microsoft® Excel Application Developer... for Beginners* [1]. Ver la sección *Referencias* para más detalles.

En el caso de Microsoft Access, tener también instalado lo siguiente:

- El cliente IBM Data Server, ó
- IBM Data Server Driver para ODBC, CLI y .Net, ó
- IBM Data Server Driver para ODBC y CLI

El IBM Data Server Driver para ODBC, CLI y .Net, y el IBM Data Server Driver para ODBC y CLI se conoce también con el nombre IBM DB2 ODBC Driver, el cual es el mismo que el manejador DB2 CLI. Este es el manejador a utilizar para conectarse desde Access a DB2. Luego de que el manejador se instala, hay que crear un proyecto Access 2007 y

elegir la opción *base de datos ODBC* disponible dentro de la pestaña *Datos Externos* en la banda de *Herramientas de Tablas*. Los próximos pasos están documentados en el artículo *DB2 9 and Microsoft Access 2007 Part 1: Getting the Data...[2]*. Cuando se utilizan **tablas enlazadas** en Microsoft Access, los datos están disponibles a los usuarios de Access 2007, pero los datos residen en el servidor de datos DB2.

Para versiones de Access anteriores a 2007, la instalación es un poco diferente, por lo que se puede revisar el artículo *Use Microsoft Access to interact with your DB2 data [3]*. Ver la sección de *Referencias* para más detalles.

14.8 Herramientas de Desarrollo

Microsoft Visual Studio y Eclipse son dos de los más populares Ambientes de Desarrollo Integrados (Integrated Development Environments) (IDEs) utilizado por los desarrolladores de hoy. Ambos IDEs trabajan bien con DB2.

Por el lado de Microsoft Visual Studio, DB2 provee un agregado (Add-in) de Visual Studio de forma tal que luego de instalado, las herramientas IBM se agregan a los menúes de Visual Studio. De esta forma, un desarrollador no necesita cambiar a otras herramientas para trabajar con las bases de datos de DB2. Se puede descargar el agregado (Add-in) de Visual Studio desde el sitio web de DB2 Express-C en www.ibm.com/db2/express.

Con respecto a Eclipse, IBM ha lanzado IBM Data Studio, una herramienta gratis basada en Eclipse que permite desarrollar SQL, scripts XQuery, procedimientos almacenados, funciones definidas por el usuario (UDFs) y servicios Web. Debido a que la plataforma está basada en Eclipse, muchos desarrolladores pueden utilizar su conocimiento previo para trabajar con esta herramienta que seguramente les resulte familiar.

14.9 Programas de Ejemplo

Para ayudar a saber cómo programar en diferentes lenguajes usando DB2 como el servidor de datos, puede revisar las aplicaciones de ejemplo que vienen con la instalación del servidor DB2 en el directorio SQLLIB\samples. La Figura 14.10 debajo, muestra algunos programas de ejemplo provistos con DB2 en una plataforma Windows.

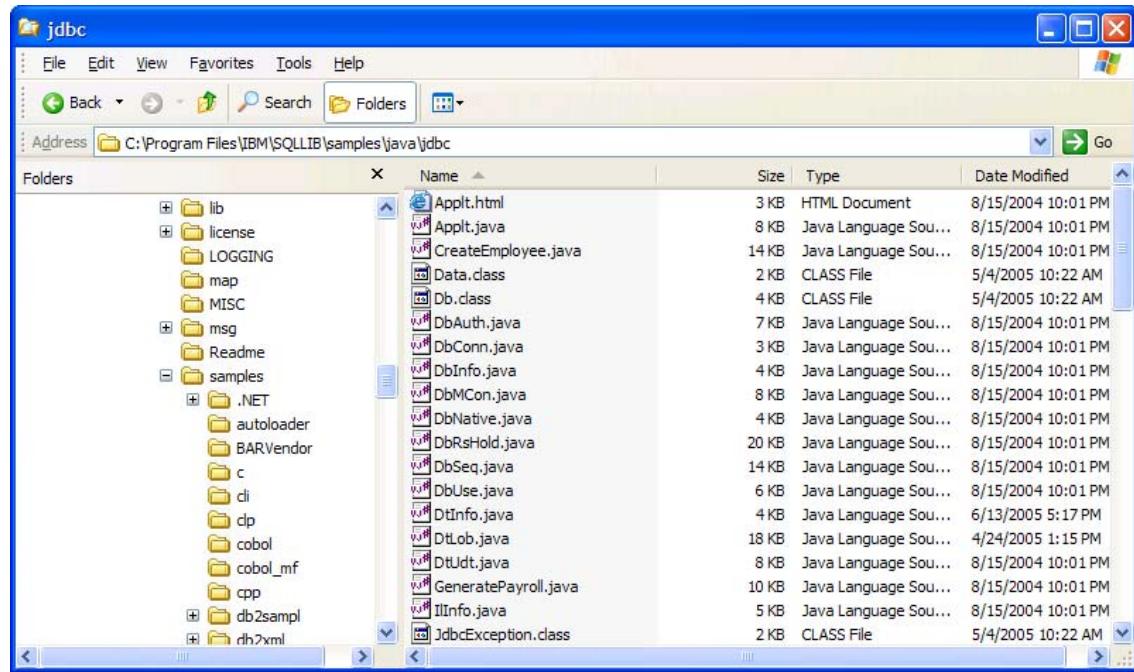


Figura 14.10 – Programas de Ejemplo que vienen con DB2

14.10 Resumen

En este capítulo, hemos visto cómo DB2 brinda la flexibilidad necesaria para programar aplicaciones de base de datos ya sea del lado de las aplicaciones del cliente con conexiones al servidor de datos DB2 como dentro de la base de datos misma.

La sección de la aplicación del lado del servidor, incluyó procedimientos almacenados, funciones definidas por el usuario (UDFs) y disparadores.

Del lado del cliente, se discutió la variedad de interfaces de programación y los métodos permitidos para el desarrollo de aplicaciones con DB2, una vez más, mostrando la notable flexibilidad y capacidad de DB2 como un servidor de base de datos.

15

Capítulo 15 – DB2 pureXML

En este capítulo discutiremos pureXML, la nueva tecnología disponible a partir de DB2 9 para soportar el almacenamiento nativo de datos XML. Muchos de los ejemplos y conceptos discutidos en este capítulo han sido tomados de los libros rojos (redbooks) de IBM: *DB2 9: pureXML overview and fast start*. Vea la sección Recursos para más información. La Figura 15.1 esboza cual sección de DB2 “Big Picture” vamos a discutir en este capítulo.

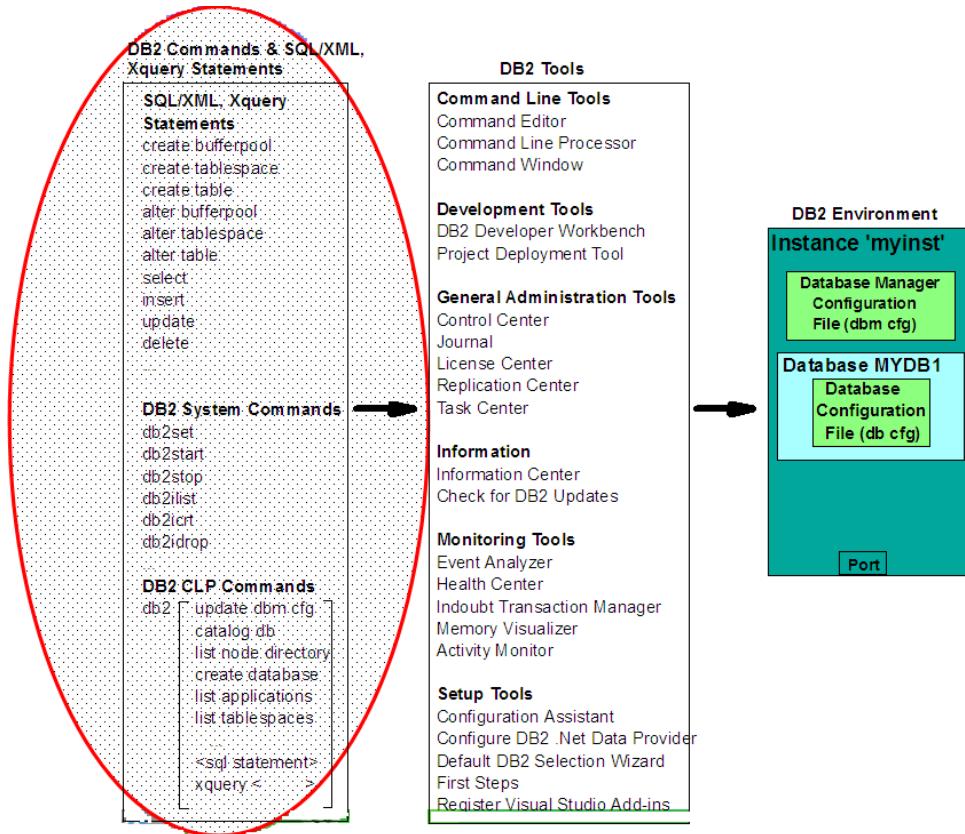


Figura 15.1 – DB2 Visión General: Comandos DB2, SQL/XML y XQuery

Nota:

Para mas información acerca de pureXML, vea este video:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4382>

15.1 Uso de XML en bases de datos

Los documentos XML pueden ser almacenados en archivos de texto, repositorios XML, o base de datos. Hay dos razones principales por las que muchas compañías proponen almacenarlos en una base de datos:

- La gestión de largos volúmenes de datos XML es un problema de base de datos. Datos en XML son datos como cualquier otros, solo que en diferente formato. Los mismos criterios para almacenar datos relacionales en las bases de datos se aplican a datos XML: Las bases de datos ofrecen búsquedas y recuperaciones eficientes, soporte robusto para persistencia de datos, back up y recuperación, soporte de transacción, rendimiento y escalabilidad.
- Integración: Al almacenar datos relacionales y documentos XML juntos, usted puede integrar los datos XML con datos relacionales existentes. Por otro lado, los datos relacionales pueden ser publicados como XML y viceversa. A través de la integración, las bases de datos pueden soportar aplicaciones Web, SOA, y servicios Web.

15.2 Bases de datos XML

Hay dos tipos de bases de datos para almacenar datos XML:

- Bases de datos XML-enabled
- Bases de datos Native XML

15.2.1 Bases de datos XML-enabled

Una base de datos XML-enabled utiliza un modelo relacional para el centro de datos del modelo de almacenamiento. Esto requiere un mapeo entre el modelo de datos XML (jerárquico) y el modelo de datos relacional o bien almacenar los datos XML como un objeto carácter largo. Si bien esto puede considerarse como una “vieja” tecnología, aun esta siendo usado por muchos proveedores de bases de datos. La figura 15.2 explica con más detalle las dos opciones para bases de datos XML-enabled.

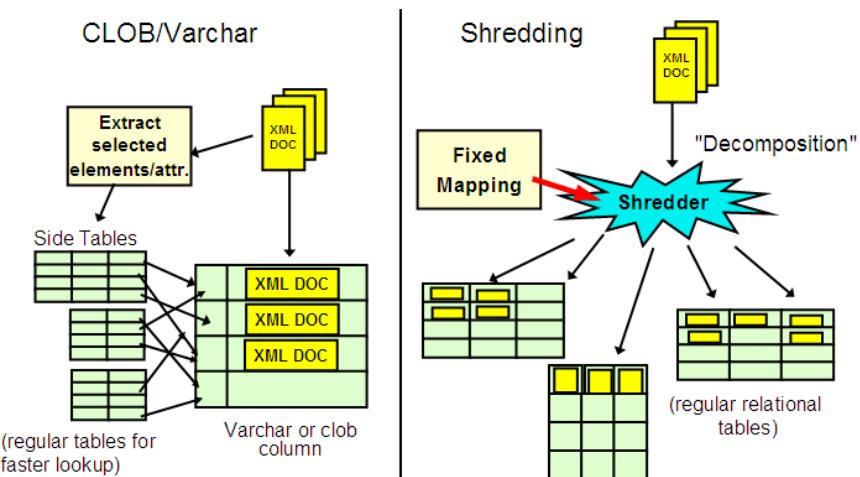


Figura 15.2 – Dos opciones para almacenar XML en una base de datos XML-enabled

El lado izquierdo de la figura 15.2 muestra el método “CLOB/Varchar” de almacenamiento de documentos XML en una base de datos. Usando este método, un documento XML es almacenado como una cadena no parseada tanto en base de datos CLOB como Varchar. Si el documento XML es almacenado como cadena, cuando se quiere recuperar parte del documento XML, su programa tiene que recuperar la cadena, y efectuar el parseo para encontrar lo que desea. Este método no es muy flexible.

La otra opción para bases de datos XML-enabled se llama Trituración o descomposición y esta ilustrada en el lado derecho de la figura 15.2. Usando este método, un documento XML entero es desmenuzado en partes más pequeñas que son almacenadas en tablas. Usando este método, el modelo jerárquico de un documento XML es forzado dentro del modelo relacional. Esto tampoco es bueno para la flexibilidad: un cambio en el documento XML no es fácilmente propagable en las correspondientes tablas y muchas otras tablas pueden necesitar ser creadas. Este método tampoco es bueno para el rendimiento: si necesita obtener el documento XML original de vuelta, tendrá que realizar una costosa operación de SQL, lo que aumenta aun más cuando hay más tablas unidas.

15.2.2 Bases de datos Nativa XML

Las bases de datos Nativa XML usan el modelo de datos XML jerárquico para almacenar y procesar XML internamente. El formato de almacenamiento es el mismo que el formato de proceso: no existe mapeo para el modelo relacional, y los documentos XML no son almacenados como imágenes. Cuando las declaraciones XPath o XQuery son usadas, son procesadas en forma nativa por el motor, y no se convierten a SQL. Esta es la razón por la que las bases de datos son conocidas como bases de datos XML “nativa”. DB2 comenzando con la versión 9 es actualmente el único servidor comercial que provee este soporte.

15.3 XML en DB2

La figura 15.3 describe a continuación como datos relacionales y datos jerárquicos (documentos XML) son almacenados en DB2 comenzando con la versión 9. En la figura, asumimos que la tabla dept esta definida como sigue:

```
CREATE TABLE dept (deptID CHAR(8),..., deptdoc XML);
```

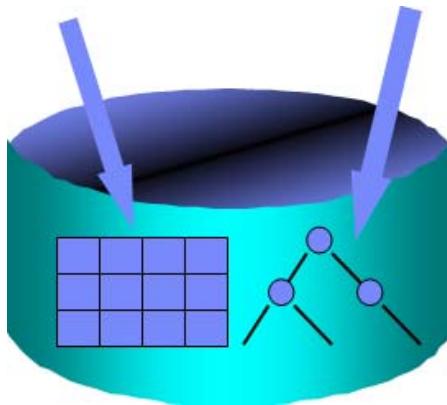


Figura 15.3 – XML en DB2

Nótese que la definición de la tabla usa un nuevo tipo de dato, XML, para la columna deptdoc. La flecha de la izquierda en la figura indica la columna relacional deptID almacenada en formato relacional (tabla), mientras que la columna XML deptdoc esta almacenada en formato jerárquico parseado.

La figura 15.4 ilustra que comenzando con DB2 9, hay ahora cuatro nuevos caminos para acceder a los datos.

- Uso de SQL para acceder a datos relacionales
- Uso de SQL con extensión XML para acceder a datos XML
- Uso de XQuery para acceder a datos XML
- Uso de XQuery para acceder a datos relacionales

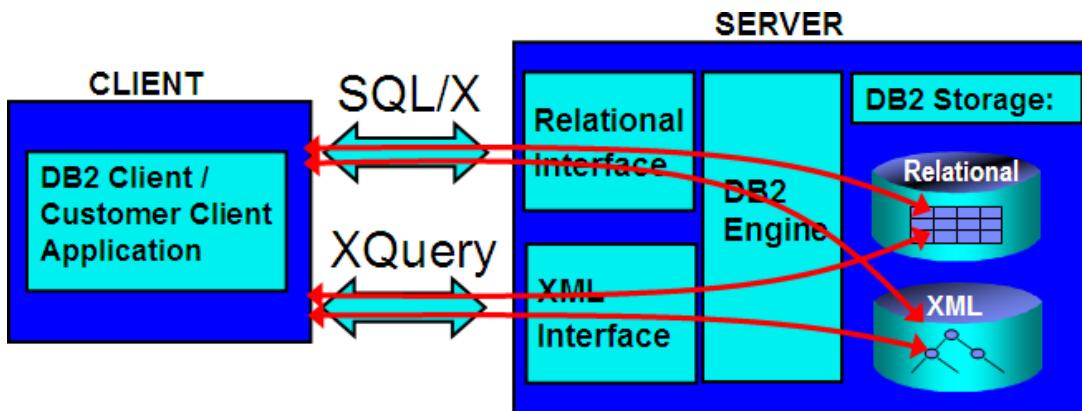


Figura 15.4 – Cuatro caminos para acceder a los datos en DB2

Por lo tanto, mientras una persona con habilidad en SQL puede ver a DB2 como un RDBMS de clase mundial que también soporta XML, una persona con habilidad en XML vería a DB2 como un repositorio de XML de clase mundial que también soporta SQL.

Nótese que IBM usa el término **pureXML** en lugar de “XML nativo” para describir esta tecnología. Mientras que otros proveedores aun utilizan la vieja tecnología de CLOB/varchar o trituración para almacenar documentos XML, ellos llaman a esas viejas tecnologías “XML nativo”. Para evitar confusiones, IBM decidió usar el término pureXML, y registra este nombre de manera que ninguna otra base de datos o proveedor pueda utilizar el mismo termino para referirse a diferencias tecnologías. En DB2 9.1, el soporte pureXML fue proporcionado solo cuando la base de datos era creada como base de datos Unicode. Esta restricción ha sido levantada con DB2 9.5, donde se puede usar pureXML en bases de datos Unicode o no unicode.

15.3.1 Ventajas tecnológicas de pureXML

Muchas ventajas son proporcionadas por el uso de la tecnología pureXML.

1. Puede perfectamente aprovechar su inversión relacional, dado que los documentos XML son almacenados en las columnas de las tablas usando el nuevo tipo de datos XML.
2. Puede reducir la complejidad del código. Por ejemplo, la figura 15.5 ilustra un script PHP escrito usando y sin usar pureXML. Usando XML (la caja verde) las líneas de código son reducidas. Esto no solo significa que el código es menos complejo, el rendimiento general se mejora ya que hay menos líneas para analizar y mantener en el código.

```

<?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 xmlicustomer
(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,

```



```

$cid) // 
db2_execute($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->("pcode-zip");
$custPhone = (string) $dom->phone;

$stmt = db2_prepare($conn, "INSERT INTO sqlicustomer
(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);

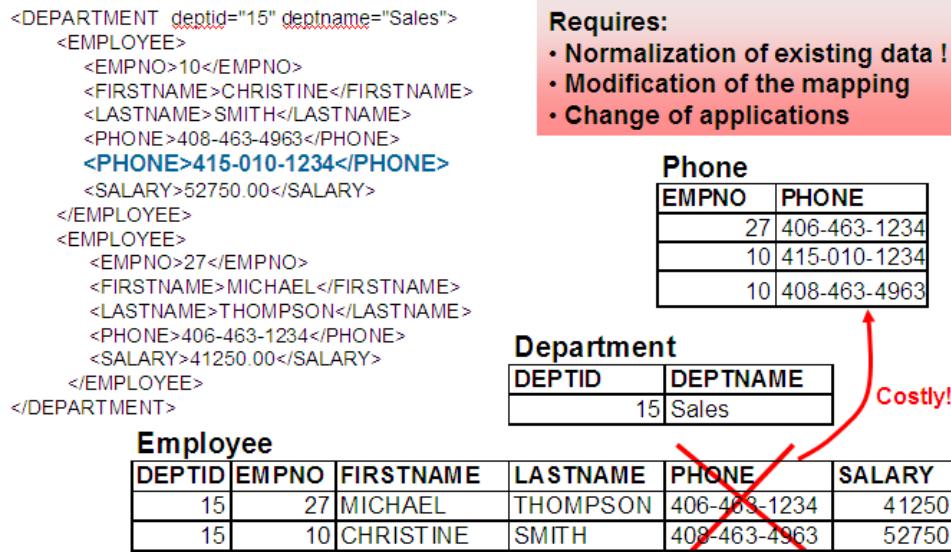
$prodID = (string) $dom["pid"];

```

Figura 15.5 – Complejidad de código con y sin pureXML

3. Los cambios en su esquema son más fáciles usando tecnología pureXML. La figura 15.6 ilustra un ejemplo de este aumento de flexibilidad. En la figura, suponga que tiene una base de datos consistente de las tablas Empleado y Departamento. Típicamente con una base de datos no-XML, si su gerente le pidió almacenar no solo un numero telefónico por empleado (el numero telefónico del hogar), sino también un segundo numero telefónico (un numero de teléfono móvil), entonces puede agregar una columna extra a la tabla Empleado y almacenar el numero de teléfono móvil en esta nueva columna. Sin embargo, este método estaría en contra de las reglas de normalización de las bases de datos relacionales. Si desea conservar estas normas, debería crear una nueva tabla de teléfono y mover toda la información telefónica a esta nueva tabla. Puede añadir a continuación los teléfonos móviles también. Crear una nueva tabla “teléfono” es costoso, no solo por las grandes cantidades de datos pre-existentes que necesitan ser movidos, sino también porque todas las SQL en su aplicación deberían ser cambiadas para apuntar a la nueva tabla.

En la parte izquierda de la figura, mostramos como esto puede hacerse usando XML. Si la empleada “Cristina” tiene un número de teléfono móvil, una nueva etiqueta se puede añadir a esta información. Si el empleado “Michael” no posee teléfono móvil, solamente lo dejamos como está.

**Figura 15.6 – Mayor flexibilidad de datos usando XML**

4. Puede mejorar el rendimiento de su aplicación XML. Pruebas realizadas usando tecnología pureXML mostraron enormes mejoras en el rendimiento de varias aplicaciones XML. La figura 15.7 muestra el resultado de la prueba para una compañía que pasa a pureXML desde viejas tecnologías. La columna del medio muestra el resultado usando el viejo método trabajando con XML, y la tercera columna muestra el resultado usando XML.

Tarea	Otra BD relacional	DB2 con pureXML
Desarrollo de búsqueda y recuperación de procesos de negocios	CLOB: 8 hrs Shred: 2 hrs	30 min.
Líneas relativas a código de E/S	100	35 (65% reducción)
Añadir a campo de esquema	1 semana	5 min.
Consultas	24 - 36 hrs	20 seg. - 10 min

Figura 15.7 – Aumento de rendimiento utilizando tecnología pureXML

15.3.2 XPath básico

Xpath es un lenguaje que puede ser usado para consultar documentos XML. La figura 15.8 muestra un documento XML, y la figura 15.9 ilustra el mismo documento representado en formato parseo-Jerárquico (también llamado "hoja" o "nodo"). Nosotros vamos a usar el formato Parseo-Jerárquico para explicar como funciona XPath.

```
<dept bldg="101">
    <employee id="901">
        <name>John Doe</name>
        <phone>408 555 1212</phone>
        <office>344</office>
    </employee>
    <employee id="902">
        - - -
    </employee>
```

Figura 15.8 – Un documento XML

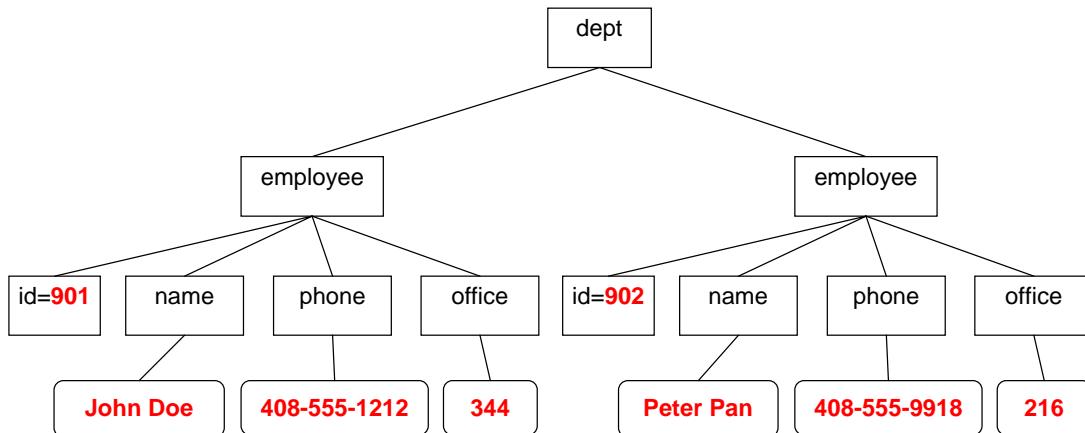


Figure 15.9 – Representación Parseo-Jerárquica del documento XML en la Fig. 15.8

Un rápido camino para aprender XPath es compararlo con el comando change directory (CD) en MS-DOS o Linux/UNIX. Con el uso del comando CD puede recorrer el árbol del directorio como sigue:

CD /directorio1/directorio2/...

De manera similar, en XPath usted usa barras para ir de un elemento a otro dentro de un documento XML. Por ejemplo, usando el documento en la figura 15.9 en XPath puede recuperar los nombres de todos los empleados usando esta consulta:

```
/dept/employee/name
```

Expresiones XPath

Las expresiones XPath utilizan caminos totalmente calificados para especificar elementos y/o atributos. Un signo “@” se usa para especificar un atributo. Para recuperar solo el valor (nodo texto) de un elemento, usar la función “text()”. La tabla 15.1 muestra consultas XPath y los correspondientes resultados usando el documento XML de la figura 15.9

XPath	Resultado
/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

Tabla 15.1 – Ejemplo de expresiones XPath

Comodines XPath

Existen dos comodines principales en XPath:

- “*” coincide con cualquier final de nombre
- “//” es el comodín “descendente o propio”

La tabla 15.2 proporciona mas ejemplos usando documentos XML de la figura 15.6

XPath	Resultados
/dept/employee/*/text()	John Doe 408 555 1212 344 Peter Pan

	408 555 9918 216
/dept/*/@id	901 902
//name/text()	Peter Pan John Doe
/dept//phone	<phone>408 555 1212</phone> <phone>408 555 9918</phone>

Tabla 15.2 – Ejemplo de comodines XPath

Predicados XPath

Los predicados están encerrados entre corchetes []. Como una analogía, puede pensar en ellos como el equivalente a la cláusula WHERE en SQL. Por ejemplo [@id="902"] puede ser leído como “WHERE atributo id es igual a 902”. Pueden haber múltiples predicados en una expresión XPath. Para especificar un prediccado posicional, use [n] que significa que el enésimo niño sería seleccionado. La tabla 15.3 proporciona más ejemplos.

XPath	Resultados
/dept/employee[@id="902"]/name	<name>Peter Pan</name>
/dept[@bldg="101"]/employee[office>"300"]/name	<name>John Doe</name>
//employee[office="344" OR office="216"]/@id	901 902
/dept/employee[2]/@id	902

Tabla 15.3 – Ejemplos de predicados XPath

XPath: El eje padre.

Al igual que en MS-DOS o Linux/UNIX, puede usar un “.” (Punto) para indicar en la expresión, que usted se refiere al contexto actual, y “..” (Punto punto) para referirse al contexto padre. La tabla 15.4 proporciona más ejemplos.

XPath	Resultados
/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

Tabla 15.4 – Eje padre en XPath

15.3.3 XQuery: Definición

XQuery es un lenguaje de consulta creado para XML. XQuery soporta expresiones de rutas para navegar la estructura jerárquica de XML. De hecho, XPath es un subconjunto de XQuery; por lo que todo lo aprendido anteriormente sobre XPath se aplica también a XQuery. XQuery soporta datos con tipo y datos sin tipo. XQuery carece de valores nulos porque los documentos XML omiten la falta de datos o los datos desconocidos. XQuery devuelve secuencias de datos XML. Es importante notar que las expresiones de XQuery y XPath distinguen entre mayúsculas y minúsculas.

XQuery soporta la expresión FLWOR. Si usamos SQL para analogía, es equivalente a la expresión SELECT-FROM-WHERE. La sección siguiente describe FLWOR con más detalle.

XQuery: La expresión FLWOR

Significado de FLWOR:

- FOR (Para): itera a través de una secuencia, une una variable a los elementos.
- LET (Permitir): Une una variable a una secuencia.
- WHERE (Donde): elimina los elementos de la iteración.
- ORDER (Orden): reordena los elementos de la iteración.
- RETURN (retorno): Construye resultados de consultas.

Es una expresión que permite la manipulación de documentos XML, permitiendo devolver otra expresión. Por ejemplo, suponga que tiene una tabla con esta definición.

```
CREATE TABLE dept(deptID CHAR(8),deptdoc XML);
```

Y el siguiente documento XML es insertado en la columna 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>
```

Entonces las siguiente sentencia XQuery utilizando la expresión FLWOR se podría ejecutar:

```
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>
```

Esto devolvería lo siguiente:

```
<EmpList bldg="101">
<name>
    John Doe
</name>
<name>
    Peter Pan
</name>
</EmpList>
```

15.3.4 Inserción de documentos XML

La Inserción de documentos XML en una base de datos DB2 se puede realizar utilizando la sentencia INSERT SQL, o la utilidad IMPORT. XQuery no puede utilizarse para este fin ya que esto aun no ha sido definido en el standard.

Vamos a examinar el siguiente script, que puede ser ejecutado desde el comando DB2 en los shell de Windows o Linux usando la siguiente declaración:

```
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),
    contact    xml
);
;

-- (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)
;
```

Tenga en cuenta que este archivo de secuencia de comandos y archivos relacionados son proporcionados en el archivo zip **expressc_book_quicklabs.zip** que acompaña a este libro. Siga a lo largo la descripción de cada línea del script.

1. Eliminar la base de datos “mydb”. Esto se hace normalmente en los archivos script para realizar una limpieza antes de crear nuevos objetos de la base de datos. Si “mydb” no existía, recibirá un mensaje de error, pero es correcto.
2. Crear la base de datos “mydb” utilizando los códigos UTF-8. Se requiere una base de datos UNICODE para soportar pureXML, por lo que este paso es necesario para crear la base de datos como una base de datos UNICODE. Si esta usando la version 9.5, crear la base de datos como UNICODE es opcional.
3. Conectar a la base datos "mydb" recién creada. Esto es necesario para crear objetos dentro de la base de datos.
4. Crear la tabla “ítems”. Tenga en cuenta que la ultima columna de la tabla (columna “comentarios”) esta definida como una columna XML usando el nuevo tipo de dato XML.
5. Creamos la tabla “clientes”. Tenga en cuenta que la ultima columna en la tabla (columna “contacto”) también esta definida con el nuevo tipo de datos XML.
6. Usando esta sentencia SQL INSERT puede insertar un documento XML en una columna XML. En la sentencia INSERT usted puede pasar al documento XML como una cadena encerrada entre comillas simples.
7. Usando el comando IMPORT, puede insertar o importar varios documentos XML junto a los datos relacionales hacia la base de datos. En (7) esta importando el dato desde el archivo clients.del (un archivo ascii delimitado), y también indica donde se encuentra el dato XML referenciado por el archivo clients.del (para este ejemplo, en D:/Raul)

Veamos con mas detalle el archivo clients.del, pero primero, veamos el contenido del directorio D:/Raul (figura 15.10)

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

Figura 15.10 - Contenido del directorio D:/Raul con documentos XML

Estos son los contenidos del archivo 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' />
```

En el archivo clients.del, “XDS FIL=” es usado para apuntar a un archivo documento XML.

La figura 15.11 muestra el Centro de Control luego de ejecutar el script de arriba.

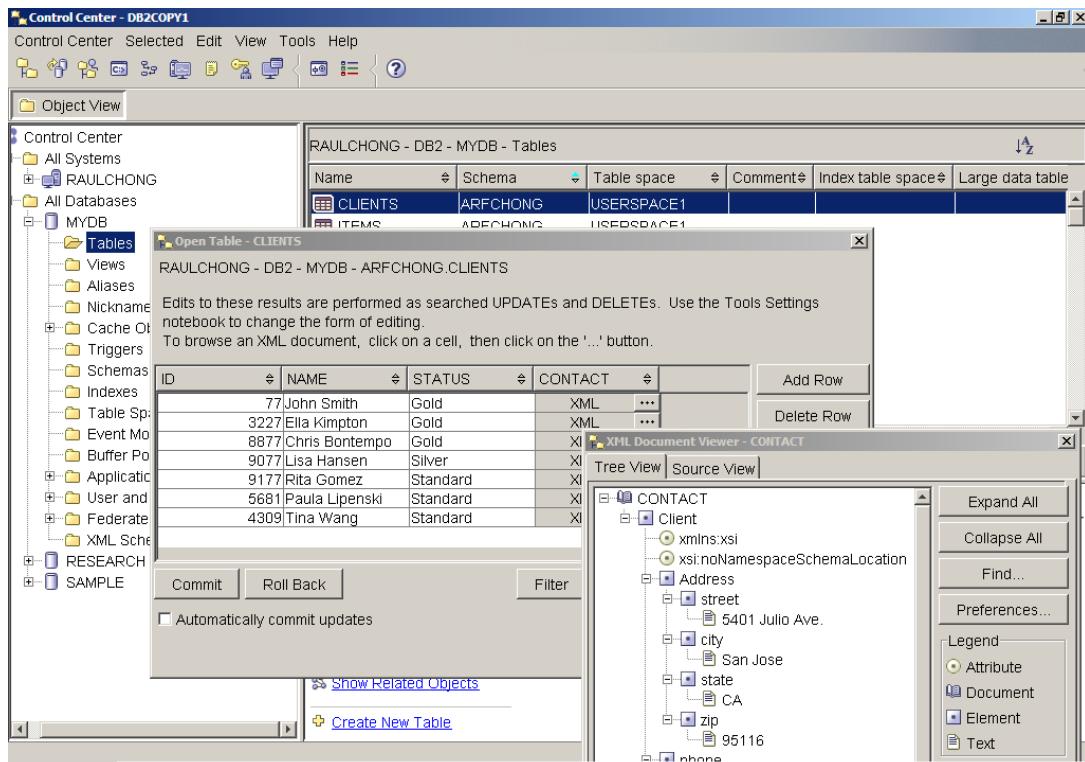


Figura 15.11 –El Centro de control después de ejecutar table_creation.txt

Tenga en cuenta que en la figura se muestra el contenido de la tabla CLIENTS. La última columna “Contact” es una columna XML. Al hacer clic en el botón con tres puntos, se abre otra ventana que mostrara el contenido del documento XML. Esto se muestra en la esquina inferior derecha en la figura 15.11

15.3.5 Consulta de datos XML

Hay dos caminos para consultar datos XML en DB2:

- Usar SQL con extensión XML (SQL/XML)
- Usar XQuery

En ambos casos, DB2 sigue las normas internacionales.

Consulta de datos XML con SQL/XML

El uso de sentencias SQL permite trabajar con filas y columnas. Una sentencia SQL puede ser usada para trabajar con todo el documento XML; pero si se deseara obtener solo parte del documento XML, entonces se necesitaría usar SQL con extensión XML (SQL/XML).

La tabla 15.5 describe algunas de las funciones SQL/XML disponibles con el estándar SQL 2006.

Nombre de la función	Descripción
XMLPARSE	Parseo de carácter dato binario de objeto largo, produce valores XML
XMLSERIALIZE	Convierte un valor XML en carácter o dato binario de objeto largo.
XMLVALIDATE	Valida el valor XML contra un esquema XML y anota el tipo de valor XML
XMLEXISTS	Determina si una XQuery retorna un resultado (es decir, una secuencia de uno o mas ítems)
XMLQUERY	Ejecuta una XQuery y devuelve la secuencia resultado.
XMLTABLE	Ejecuta una XQuery devuelve el resultado secuencia como una tabla relacional(si es posible)
XMLCAST	Castea desde o hacia un tipo XML

Tabla 15.5 – Funciones SQL/XML

Los siguientes ejemplos pueden ser probados usando la base de datos “mydb” creada anteriormente.

Ejemplo 1

Este es un ejemplo de una consulta. Imagine que necesita localizar los nombres de todos los clientes que viven en un determinado código postal. La tabla “clients” almacena las direcciones de los clientes, incluyendo código postal en una columna XML. Usando XMLEXISTS, puede buscar en la columna XML el código postal deseado y luego limitar el resultado retornado como corresponda.

```
SELECT name FROM clients
WHERE xmlexists(
    '$c/Client/Address[zip="95116"]'
    passing clients.contact as "c"
)
```

La primera línea es una cláusula SQL que especifica que necesita recuperar la información en la columna “name” de la tabla “clients”

La cláusula WHERE invoca a la función XMLEXISTS, especificando la expresión XPath que induce a DB2 a navegar hasta el elemento “zip” y chequear el valor 95116.

La cláusula “\$c/Client/Address” indica la ruta dentro de la jerarquía del documento XML donde DB2 puede localizar el elemento “zip”. El signo dólar (\$) se utiliza para especificar

una variable; por lo que “c” es una variable. Esta variable se define por esta línea: passing clients.contact as “c”. Aquí, “clients” es el nombre de la tabla y “contact” es el nombre de la columna con un dato de tipo XML. En otras palabras, estamos pasando el documento XML a la variable “c”.

DB2 inspecciona los datos XML contenidos en la columna “contact”, navega desde la raíz, nodo “Client”, al nodo “Address” y luego al nodo “zip” y finalmente determina si el cliente vive en el código postal buscado. La función XMLEXISTS evalúa por “verdadero (true)” y DB2 retorna el nombre del cliente asociado con esa fila.

En DB2 9.5, las consultas pueden simplificarse como sigue:

```
SELECT name FROM clients
WHERE xmlexists(
    '$CONTACT/Client/Address[zip="95116"]'
)
```

Una variable con el mismo nombre que la columna de tipo XML es creada automáticamente por DB2. En el ejemplo anterior, la variable CONTACT es creada automáticamente por DB2. Este nombre coincide con la columna CONTACT de tipo XML.

Ejemplo 2

Vamos a estudiar la forma de resolver el problema de como crear un informe con lista de direcciones e-mail de clientes de estado “Gold”. La siguiente consulta podría ejecutar:

```
SELECT xmlquery('$c/Client/email' passing contact as "c")
    FROM clients
    WHERE status = 'Gold'
```

La primera línea indica que queremos retornar la dirección de e-mail que es un elemento del documento XML (no una columna relacional). Como en el ejemplo anterior, “\$c” es una variable que contiene el documento XML. En este ejemplo usamos la función XMLQUERY que puede ser usada después de SELECT, mientras la función XMLEXISTS puede ser usada después de una cláusula WHERE.

Ejemplo 3

Puede haber situaciones en la que le gustaría presentar datos XML como tablas. Esto es posible con la función XMLTABLE como se muestra en el ejemplo debajo.

```
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
```

La primera línea especifica las columnas que deben incluirse en el conjunto de resultados. Las columnas con el prefijo “t” están basadas en valores de elementos XML.

La segunda línea invoca la función XMLTABLE para especificar la columna XML que contiene el dato buscado (“i.comments”) y la ruta dentro de la columna de documentos XML donde los elementos de interés están localizados.

La cláusula “columns”, que abarca desde la línea 4 a la 6, identifican los elementos específicos XML que van a ser mapeados a la salida de la columna en el conjunto resultado SQL especificado en la línea 1. Parte de este mapeo implica especificar en el tipo de dato que los valores de los elementos XML serán convertidos. En este ejemplo todo dato XML es convertido en el tradicional tipo de datos SQL.

Ejemplo 4

Ahora vamos a estudiar un ejemplo sencillo donde usted incluye una expresión XQuery FLWOR dentro de una función XMLQUERY SQL/XML

```
SELECT name, xmlquery(
  'for $e in $c/Client/email[1] return $e'
  passing contact as "c"
)
FROM clients
WHERE status = 'Gold'
```

La primera línea especifica que los nombres de los clientes y la salida desde la función XMLQUERY estarán incluidos en el conjunto de resultados. La segunda línea indica el primer sub-elemento “email” del elemento “client”. Sera retornado. La tercera línea identifica la fuente del dato XML (columna “contact”). La cuarta línea nos dice que esta columna proviene de la tabla “clients”; y la quinta línea indica que solo los clientes “Gold” son de interés.

Ejemplo 5

Este ejemplo muestra nuevamente la función XMLQUERY que toma una expresión XQuery FLWOR, pero note que esta vez retornamos no solo XML, sino también HTML.

```
SELECT xmlquery('for $e in $c/Client/email[1]/text()
  return <p>{$e}</p>'
  passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

La cláusula return de XQuery permite transformar la salida XML como necesite. Usando la función text() en la primer línea, indica que solo la representación del texto de la primera dirección de email de clientes calificados son de interés. La segunda línea especifica que esta información estará encerrada por las etiquetas de párrafo HTML.

Ejemplo 6

El siguiente ejemplo utiliza la función XMLEMENT para crear una serie de elementos ítems, cada uno contiene valores para sub-elementos ID, nombre de marca, y la unidad de mantenimiento de existencias (stock keeping unit (SKU)) obtenidos de las correspondientes columnas en la tabla “ítems”. Básicamente puede usar la función XMLEMENT cuando desee convertir dato relacional a dato XML.

```
SELECT
    xmlelement (name "item", itemname),
    xmlelement (name "id", id),
    xmlelement (name "brand", brandname),
    xmlelement (name "sku", sku)
FROM items
WHERE srp < 100
```

La consulta anterior provee una salida como esta:

```
<item>
    <id>4272</id>
    <brand>Classy</brand>
    <sku>981140</sku>
</item>
...
<item>
    <id>1193</id>
    <brand>Natural</brand>
    <sku>557813</sku>
</item>
```

Consulta de datos XML con XQuery

En la sección anterior, vimos como consultar datos XML utilizando SQL con extensiones XML. SQL fue siempre el método de consulta primario, y XPath estaba embebido dentro de SQL. En esta sección, estudiaremos como consultar datos XML utilizando XQuery. Esta vez, XQuery será el método primario de consulta, y en algunos casos, usaremos SQL

embebido dentro de XQuery (usando la función “db2-fn:sqlquery”). Cuando usamos XQuery, invocaremos algunas funciones, y utilizaremos también la expresión FLWOR.

Ejemplo 1

Una simple XQuery para retornar dato del contacto cliente.

```
xquery db2-fn:xmlcolumn('CLIENTS.CONTACT')
```

Siempre prefije cualquier expresión XQuery con el comando “xquery” así DB2 sabrá que tiene que utilizar XQuery, de lo contrario DB2 asumirá que esta tratando de ejecutar una expresión SQL. La función **db2-fn:xmlcolumn** es una función que recupera documentos XML de la columna especificada por parámetro. Es equivalente a la siguiente sentencia SQL, que recupera el contenido de la columna completa:

```
SELECT contact FROM clients
```

Ejemplo 2

En este ejemplo, usamos la expresión FLWOR para recuperar el dato del fax del cliente.

```
xquery
for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/fax
return $y
```

La primera línea invoca a el parseador XQuery. La segunda línea instruye a DB2 a iterar a través del sub-elemento fax contenido en la columna CLIENTS.CONTACT. Cada elemento fax esta rodeado por la variable \$y. La tercera línea indica que por cada iteración, el valor “\$y” es retornado.

La salida de esta consulta es similar a lo siguiente (puede incluir el nombre del espacio, o namespace, por defecto, pero no lo mostramos, de lo contrario esta salida seria más difícil de leer, ya que podría abarcar varias líneas):

```
<fax>4081112222</fax>
<fax>5559998888</fax>
```

Ejemplo 3

El siguiente ejemplo consulta datos XML y retorna el resultado como HTML.

```
xquery
<ul> {
  for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/Address
  order by $y/zip
  return <li>{$y}</li>
}
```

```
</ul>
```

La salida HTML se ve así:

```
<ul>
<li>
<address>
  <street>9407 Los Gatos Blvd.</street>
  <city>Los Gatos</city>
  <state>ca</state>
  <zip>95302</zip>
</address>
</li>
<address>
  <street>4209 El Camino Real</street>
  <city>Mountain View</city>
  <state>CA</state>
  <zip>95302</zip>
</address>
</li>
...
</ul>
```

Ejemplo 4

El siguiente ejemplo muestra como embeder SQL dentro de XQuery usando la función db2-fn:sqlquery. La función db2-fn:sqlquery ejecuta una consulta SQL y retorna solo el dato XML seleccionado. La consulta SQL que paso a db2-fn:sqlquery solo debe retornar datos de tipo XML. Este dato XML puede ser procesado mas aun por XQuery

```
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>
)
```

En el ejemplo, la consulta SQL filtra líneas basadas en la condición que la columna “srp” tiene un valor mayor a 100. De estas líneas filtradas, seleccionara la columna “comments”, que es la columna de tipo XML. Luego XQuery (o XPath) es aplicado para ir a los sub-elementos.

Nota:

DB2 no distingue mayúsculas de minúsculas y trata a todos los nombres de tablas y columnas en mayúsculas mientras XQuery si distingue estas. Las funciones que hemos discutido hasta ahora, son funciones de XQuery, así que todo nombre de tablas y columnas deben ser pasados en mayúsculas. Si se utilizaran minúsculas puede resultar en un error de un objeto indefinido.

15.3.6 Unir con SQL/XML

Esta sección describe como llevar a cabo operaciones JOIN entre dos columnas XML de diferentes tablas, o entre una columna XML y una columna relacional. Suponga que tiene creada dos tablas con estas sentencias:

```
CREATE TABLE dept (unitID CHAR(8), deptdoc XML)

CREATE TABLE unit (unitID CHAR(8) primary key not null,
                   name     CHAR(20),
                   manager  VARCHAR(20),
                   ...
)
```

Puede realizar una operación JOIN de dos maneras:

Método 1:

```
SELECT u.unitID
      FROM dept d, unit u
     WHERE XMLEXISTS (
        '$e//employee[name = $m]'
       passing d.deptdoc as "e", u.manager as "m")
```

En la línea 3 de esta sentencia se muestra que la operación JOIN se produce entre el elemento “name” que es sub-elemento de la columna XML “deptdoc” en la tabla “dept”, y la columna relacional “manager” en la tabla “unit”.

Método 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))
```

En este modo alternativo, la columna relacional se encuentra en el lado izquierdo del JOIN. Si la columna relacional esta en el lado izquierdo de la ecuación, un índice relacional será utilizado en vez de un índice XML.

15.3.7 Unir con XQuery

Suponga que las siguientes tablas han sido creadas:

```
CREATE TABLE dept(unitID CHAR(8), deptdoc XML)
CREATE TABLE project(projectDoc XML)
```

Si usamos SQL/XML, la consulta se vería como sigue:

```
SELECT XMLQUERY (
    '$d/dept/employee' passing d.deptdoc as "d")
  FROM dept d, project p
 WHERE XMLEXISTS (
    '$e/dept[@deptID=$p/project/deptID]'
      passing d.deptdoc as "e", p.projectDoc as "p")
```

La consulta equivalente usando XQuery seria:

```
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
```

Este segundo método es mas fácil de interpretar – la variable “\$dept” tiene el documento XML de la columna XML “deptdoc” en la tabla “dept”. La variable “\$proj” tiene el documento XML de la columna XML “projectdoc” en la tabla “project”. Luego la línea 4 lleva a cabo la operación JOIN entre un atributo del primer documento XML y un elemento del segundo documento XML.

15.3.8 Operaciones de actualización y eliminación

Operaciones de actualización y eliminación en datos XML se pueden hacer de dos maneras:

- Usando sentencias SQL UPDATE y DELETE
- Usando la expresion TRANSFORM

En la primera forma, usando sentencias SQL UPDATE y DELETE, la actualización o la eliminación se produce a nivel documento; es decir, todo el documento XML se sustituye

con la actualización. Por ejemplo, si en el ejemplo siguiente todo lo que nos gustaría cambiar es el elemento <state>, en realidad todo el documento XML se sustituye.

```
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>
            <fax>4087776666</fax>
            <email>newemail@someplace.com</email>
        </Client>')
)
WHERE id = 3227
```

Para la segunda forma, puede actualizar sub-documentos usando la expresión TRANSFORM, la cual es un poco más eficiente. Esto le permite reemplazar, insertar, eliminar o renombrar nodos en un documento XML. También puede cambiar el valor de un nodo sin sustituir el nodo en sí, normalmente para cambiar un elemento o valor de un atributo, que es un tipo de actualización muy común. Esto es nuevo con DB2 9.5.

La expresión TRANSFORM es parte del lenguaje XQuery, puede usarla en cualquier lugar donde usa XQuery, por ejemplo en una expresión FLWOR o en la función XMLQUERY en una sentencia SQL/XML. El uso más típico es en sentencias SQL UPDATE para modificar un documento XML es una columna XML.

Esta es la sintaxis de la expresión TRANSFORM:

```
>>-transform--| copy clause |--| modify clause |--| return clause
|-><

copy clause
-----.
      V
      |
| --copy---$VariableName---:---CopySourceExpression-----|
modify clause
```

```
|--modify--ModifyExpression-----|
```

```
return clause
```

```
|--return--ReturnExpression-----|
```

La cláusula `copy` se usa para asignar a una variable el documento XML que quiere procesar. En la cláusula `modify`, puede invocar una expresión `insert`, `delete`, `rename` o `replace`. Estas expresiones le permiten actualizar el documento XML. Por ejemplo, si quiere añadir nuevos nodos al documento, debería usar la expresión `insert`, para eliminar nodos desde un documento XML, use la expresión `delete`, para renombrar un elemento o atributo en el documento XML, use la expresión `rename`, y para reemplazar un nodo existente con un nuevo nodo o secuencia de nodos, usa la expresión `replace`. El valor reemplazado de la expresión puede ser usado solamente para cambiar el valor de un elemento o atributo. La cláusula `return` retorna el resultado de una expresión `transform`.

A continuación hay un ejemplo de una sentencia UPDATE usando la expresión `TRANSFORM`.

```
(1)-- UPDATE customers
(2)-- SET contactinfo = xmlquery( 'declare default element namespace
(3)--                               "http://posample.org" ;
(4)--     transform
(5)--     copy $newinfo := $c
(6)--         modify do insert <email2>my2email.gm.com</email2>
(7)--             as last into $newinfo/customerinfo
(8)--     return $newinfo' passing contactinfo as "c")
(9)-- WHERE id = 100
```

En el ejemplo anterior, las líneas (1), (2), y (9) son parte de la sintaxis SQL UPDATE. En la línea (2) la función XMLQUERY es invocada, y llama a la expresión transform en la línea (4). El bloque de la expresión transform va desde la línea (4) a la línea (8), y es usada para insertar un nuevo nodo dentro del documento XML que contiene el elemento `email2`. Tenga en cuenta que la actualización de los elementos en un documento XML a través de una vista no esta soportado.

Eliminar documentos XML completos desde tablas es tan simple como cuando se utiliza la sentencia SELECT en SQL/XML. Use la sentencia SQL DELETE y especifique cualquier predicado WHERE necesario.

15.3.9 Indexación XML

En un documento XML, la indexación puede ser creada para elementos, atributos, o para valores (nodos de texto). Abajo hay algunos ejemplos. Suponga que la siguiente tabla esta creada.

```
CREATE TABLE customer(info XML)
```

Y suponga que este es uno de los documentos XML almacenados:

```
<customerinfo Cid="1004">
  <name>Matt Foreman</name>
  <addr country="Canada">
    <street>1596 Baseline</street>
    <city>Toronto</city>
    <state>Ontario</state>
    <pcode>M3Z-5H9</pcode>
  </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) Esta sentencia crea un índice en el atributo "Cid"

```
CREATE UNIQUE INDEX idx1 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/@Cid'
  AS sql DOUBLE
```

- 2) Esta sentencia crea un índice en el elemento "name"

```
CREATE INDEX idx2 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/name'
  AS sql VARCHAR(40)
```

- 3) Esta sentencia crea un índice en todos los elementos "name"

```
CREATE INDEX idx3 ON customer(info)
  GENERATE KEY USING
  xmlpattern '//name'
  AS sql VARCHAR(40);
```

- 4) Esta sentencia crea un índice en todos los nodos de texto (todos los valores). Esto no es recomendable, ya que sería demasiado costoso mantener el índice en una operación de actualización, eliminación o inserción, y el índice sería demasiado grande.

```
CREATE INDEX idx4 ON customer(info)
  GENERATE KEY USING
    xmppattern '//text()'
  AS sql VARCHAR(40);
```

15.4 Trabajando con Esquemas XML

DB2 permite insertar un documento XML en la base de datos en caso de que esté bien formado. Si no es así, se retornará un error al momento de la inserción. Por otro lado, DB2 no obliga a validar un documento XML. Si se desea que un documento XML sea validado, se tienen varias alternativas, las cuales se presentan en este sección.

15.4.1 Registrando los Esquemas XML

Los Esquemas XML se almacenan en las bases de datos de DB2 dentro de lo que se denomina repositorio de Esquema de XML. Para agregar un Esquema XML a un repositorio, se utiliza el comando REGISTER XMLSCHEMA.

Por ejemplo, digamos que se tiene un documento XML almacenado en el archivo `order.xml` como se muestra en la *Figura 15.10*

```
<?xml version="1.0" encoding="UTF-8"?>
<po:PurchaseOrder xmlns:po="http://www.test.com/po">
  <Header>
    <Id>1</Id>
    <date>2004-01-29</date>
    <description>purchase order</description>
    <value>20</value>
    <status>shipped</status>
  </Header>
  <Items>
    <Item>
      <ItemDescription color="red" weight="5">
        <Name>Widget C</Name>
        <SKU>1</SKU>
        <Price>30</Price>
        <Comment>no comment</Comment>
      </ItemDescription>
      <NumberOrdered>1</NumberOrdered>
    </Item>
  </Items>
  <Customer type="regualar">
    <Name>Manoj K Sardana</Name>
    <Address>ring road, bangalore</Address>
    <Phone>918051055109</Phone>
    <email>m.sardana@in.ibm.com</email>
  </Customer>
</po:PurchaseOrder>
```

Figura 15.10 – El archivo order.xml que contiene el documento XML

Ahora, digamos que se tiene un documento de Esquema XML almacenado en el archivo order.xsd como se muestra en la *Figura 15.11*

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.test.com/po"
  xmlns:po="http://www.test.com/po"
  xmlns:head="http://www.test.com/header"
  xmlns:prod = "http://www.test.com/product"
  xmlns:cust = "http://www.test.com/customer">

  <xsd:import namespace="http://www.test.com/product" schemaLocation="product.xsd" />
  <xsd:import namespace="http://www.test.com/customer" schemaLocation="customer.xsd" />
  <xsd:import namespace="http://www.test.com/header" schemaLocation="header.xsd" />
  <xsd:complexType name="itemType">
    <xsd:sequence>
      <xsd:element name="Item" minOccurs="1" maxOccurs="unbounded" >
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="ItemDescription" type="prod:prodType" />
            <xsd:element name="NumberOrdered" type="xsd:integer" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="potype">
    <xsd:sequence>
      <xsd:element name="Header" type="head:headerType" />
      <xsd:element name="Items" type="po:itemType" />
      <xsd:element name="Customer" type="cust:customerType" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="PurchaseOrder" type="po:potype" />
</xsd:schema>

```

Figura 15.11 – El archivo order.xsd que contiene un esquema XML

En este documento de Esquema XML hemos resaltado con una elipse lo siguiente::

- <xsd:schema>: Indica que es un documento de Esquema XML
- <xsd:import ...>: Importamos otros archivos xsd (otros Esquemas XML) que serían parte de un Esquema XML más grande.
- minOccurs="1": Un ejemplo de una “regla” de un Esquema XML, donde decimos que el elemento **Item** debería aparecer por lo menos una vez, en otras palabras, debería haber al menos un elemento **Item**.

Luego, para registrar un Esquema XML en la base de datos, se podría utilizar un script similar al que se muestra en el *Listado 15.33* debajo. Este script incluye comentarios que hacen que se explique por si solo.

```

-- CONNECT TO THE DATABASE
CONNECT TO SAMPLE;

-- REGISTER THE MAIN XML SCHEMA
REGISTER XMLSCHEMA http://www.test.com/order FROM D:\example3\order.xsd AS

```

```
order;

-- ADD XML SCHEMA DOCUMENT TO MAIN SCHEMA
ADD XMLSCHEMA DOCUMENT TO order ADD http://www.test.com/header FROM
D:\example3\header.xsd;

-- ADD XML SCHEMA DOCUMENT TO MAIN SCHEMA
ADD XMLSCHEMA DOCUMENT TO order ADD http://www.test.com/product FROM
D:\example3\product.xsd;

-- ADD XML SCHEMA DOCUMENT TO MAIN SCHEMA
ADD XMLSCHEMA DOCUMENT TO order ADD http://www.test.com/customer FROM
D:\example3\customer.xsd;

-- COMPLETE THE SCHEMA REGISTRATION
COMPLETE XMLSCHEMA order;
```

Listado 15.33 – Un script de ejemplo que muestra los pasos necesarios para registrar un esquema XML

Para revisar esta información, se puede utilizar la sentencia SELECT para consultar las tablas del Catálogo como lo muestra el *Listado 15.34* debajo.

```
SELECT CAST(OBJECTSCHEMA AS VARCHAR(15)), CAST(OBJECTNAME AS VARCHAR(15))
  FROM syscat.xsrobjects
 WHERE OBJECTNAME='ORDER' ;
```

Listado 15.34 – Recuperando la información del esquema XML a partir de las tablas del catálogo de DB2

15.4.2 Validación del Esquema XML

Una vez que los Esquemas XML han sido registrados en DB2, se pueden validar los documentos XML de dos formas diferentes:

- Mediante la función XMLVALIDATE durante la operación INSERT
- Mediante un BEFORE Trigger (Disparador de base de datos anterior a un evento)

La Figura 15.12 muestra un ejemplo donde el documento XML de la *Figura 15.10* se valida en base al Esquema XML que se muestra en la *Figura 15.11*.

```

DROP TABLE t1;
CREATE TABLE t1 (po xml);

INSERT INTO t1 VALUES(xmlvalidate(xmlparse(document('<?xml version="1.0" encoding="UTF-8"?>
<po:PurchaseOrder xmlns:po="http://www.test.com/po">
<Header>
<Id>1</Id>
<date>2004-01-29</date>
<description>purchase order</description>
<value>20</value>
<status>shipped</status>
</Header>
<Items>
<Item>
<ItemDescription color="red" weight="5">
<Name>Widget C</Name>
<SKU>1</SKU>
<Price>30</Price>
<Comment>no comment</Comment>
</ItemDescription>
<NumberOrdered>1</NumberOrdered>
</Item>
</Items>
<Customer type="regular">
<Name>Manoj K Sardana</Name>
<Address>ring road, bangalore</Address>
<Phone>918051055109</Phone>
<email>msardana@in.ibm.com</email>
</Customer>
</po:PurchaseOrder>')) ACCORDING TO XMLSCHEMA ID order));

```

Figura 15.12 – Validación de un Esquema XML por medio de XMLVALIDATE

Para probar si un documento XML ha sido validado, puede utilizar el predicado “ES VALIDADO” (“IS VALIDATED”) en una restricción tipo CHECK (CHECK constraint).

Se pueden validar los documentos XML en una columna utilizando esquemas XML diferentes. Esto es importante para facilitar la migración de la versión 1 a la versión 2 de un esquema XML. En la misma columna XML, también se pueden encontrar documentos XML sin ningún tipo de validación. Esto resulta útil si los documentos se recuperan a partir de fuentes confiables o no confiables, donde solamente las no confiables requieren validación de esquema XML.

15.4.3 Otro soporte para XML

Pequeños documentos XML pueden ahora incluirse dentro de la tabla base. Esto significa que los datos XML sean almacenados en el mismo lugar que los datos relacionales, y por lo tanto, pueden tomar ventaja de los mismos mecanismos de compresión que cualquie

dato relacional. Documentos XML más grandes se almacenan en un objeto interno separado, el cual también puede ser comprimido.

DB2 también soporta evolución de Esquema XML. Esto significa que si el Esquema XML cambia, se puede actualizar fácilmente con el comando UPDATE XMLSCHEMA. Si los cambios al Esquema XML son muy drásticos, se pueden obtener algunos errores.

En DB2, la descomposición o “desmembramiento” de un XML también se soporta. Este es el “viejo” método para almacenar XML en las bases de datos, y es lo que otros proveedores usan para almacenar XML. DB2 todavía soporta este método si se desea utilizar; sin embargo, recomendamos usar pureXML. DB2 también soporta el XML Extender (extensiones al XML), utilizando el antiguo método para almacenar XML, pero este extender dejará de recibir mejoras.

nuevo en
V9.7

Con DB2 9.7 todos los beneficios de pureXML han sido extendidos a las particiones de base de datos comúnmente utilizadas para almacenes de datos (data warehouses). La Característica de Particionamiento de Bases de Datos (Database Partitioning Feature) (DPF) se ofrece con la Edición Empresarial de DB2 (DB2 Enterprise Edition).

15.6 Resumen

Este capítulo presentó una introducción al XML y a la tecnología pureXML. El uso del documento XML cada vez está creciendo más debido al advenimiento de la Web 2.0 , sus herramientas y técnicas, así como la Arquitectura Orientada al Servicio (SOA). Guardando los documentos XML en una base de datos DB2, se puede tomar ventaja de la seguridad, el rendimiento, y la flexibilidad de codificación utilizando pureXML. pureXML es una tecnología que permite almacenar documentos XML en un formato validado jerárquicamente, como un árbol, y esto se realiza en la base de datos al momento de la inserción. No hay necesidad de validar el documento XML para construir un árbol antes del procesamiento. El árbol para el documento XML ya ha sido construido y guardado en la base de datos. Como agregado, la tecnología pureXML utiliza un motor de XML nativo que entiende XQuery, por lo tanto, no hay necesidad de relacionar XQuery a SQL, lo cual es lo que hacen otros productos de sistemas de gestión de bases de datos.

El capítulo también mostró cómo insertar, borrar, actualizar y consultar documentos XML con SQL/XML y XQuery. También se explicaron los índices XML, el Esquema XML, y otras características tales como compresión y evolución del Esquema XML.

15.7 Ejercicios

A través de este capítulo, se han visto varios ejemplos de SQL/XML y sintaxis XQuery y se han incluido al Editor de Comando DB2 y al IBM Data Studio. En este ejercicio, se pondrá a prueba el conocimiento adquirido de SQL/XML y XQuery mientras se gana experiencia con estas herramientas. Utilizaremos la base de datos `mydb` que se creó con el script del archivo `table_creation.txt`, el cual se explicó anteriormente en este capítulo (*Listado 15.5*).

Procedimiento

1. Crear la base de datos **mydb** y cargar los datos XML, como se discutió anteriormente en el capítulo. El archivo `table_creation.txt` se incluye en **Expressc_book_exercises_9.7.zip**, que viene con el este libro, en la carpeta del Capítulo 2. Ejecutar el script del archivo `table_creation.txt` desde la Ventana de Comando DB2 o desde Linux como se muestra a continuación:

```
db2 -tvf table_creation.txt
```

2. Si el script falla en alguno de los pasos, intente descifrar el problema revisando los mensajes de error. Un problema típico cuando se ejecuta el script es que se pueda requerir cambiar los directorios donde residen los archivos ya que pueden tener otra ubicación. En todo momento se puede eliminar la base de datos y volver a comenzar ejecutando el siguiente comando desde la Ventana de Comando de DB2 o desde Linux:

```
db2 drop database mydb
```

3. Si la sentencia anterior de eliminado de la base de datos **mydb** devuelve un error debido a que hay conexiones activas, ejecutar primero el siguiente comando:

```
db2 force applications all
```

4. Luego de ejecutar satisfactoriamente el script, usar el Centro de Control de DB2 o el IBM Data Studio para verificar que las tablas **items** y **clients** se creen y que contengan 4 y 7 filas respectivamente.
5. Con la base de datos **mydb** creada y con las dos tablas completas, ahora puede conectarse a la base de datos y realizar las consultas que se muestran en los *Listados 15.7 al 15.19*

A

Apéndice A – Resolución de problemas

Este apéndice trata de como resolver problemas que pueden encontrarse al trabajar con DB2. La *Figura A.1* nos da una vista sucinta de las acciones a tomar cuando surge un problema.

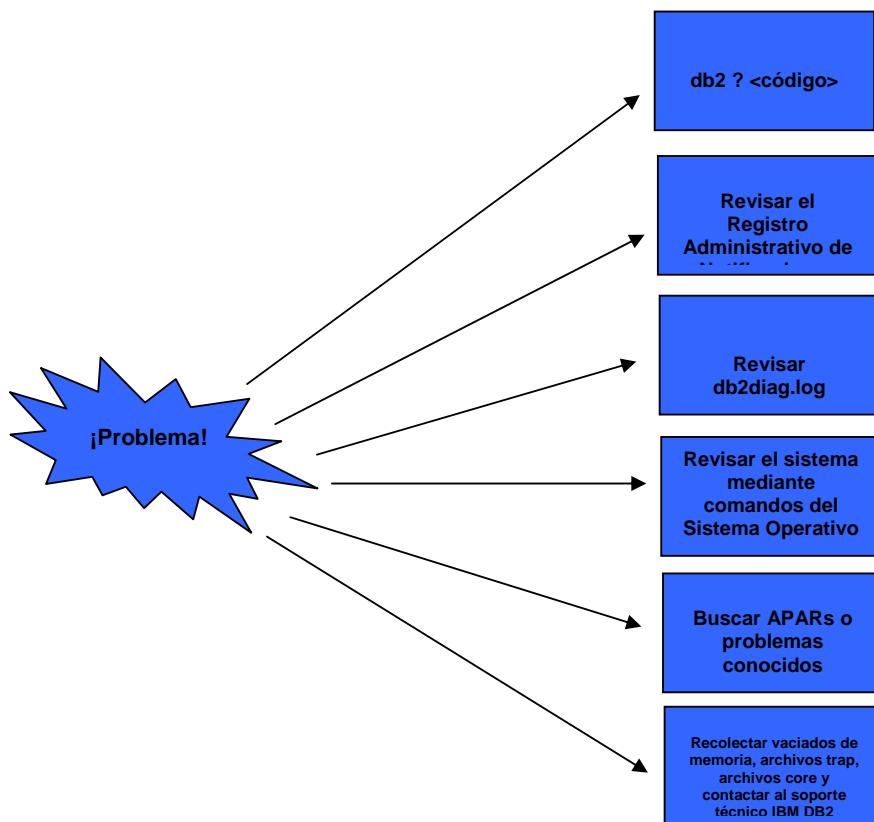


Figura A.1 – Resolución de problemas – Visión general

Nota:

Para mayor información acerca de la resolución de problemas, vea este video (en inglés):
<http://www.channeldb2.com/video/video/show?id=807741:Video:4462>

A.1 Como encontrar más información sobre los códigos de error

Para obtener mayor información sobre un código de error, entre el código con un signo de interrogación (?) delante, en la zona de entrada del *Editor de comandos* y dé clic en el botón de *Ejecutar*, tal como se ve en la *Figura A.2*.

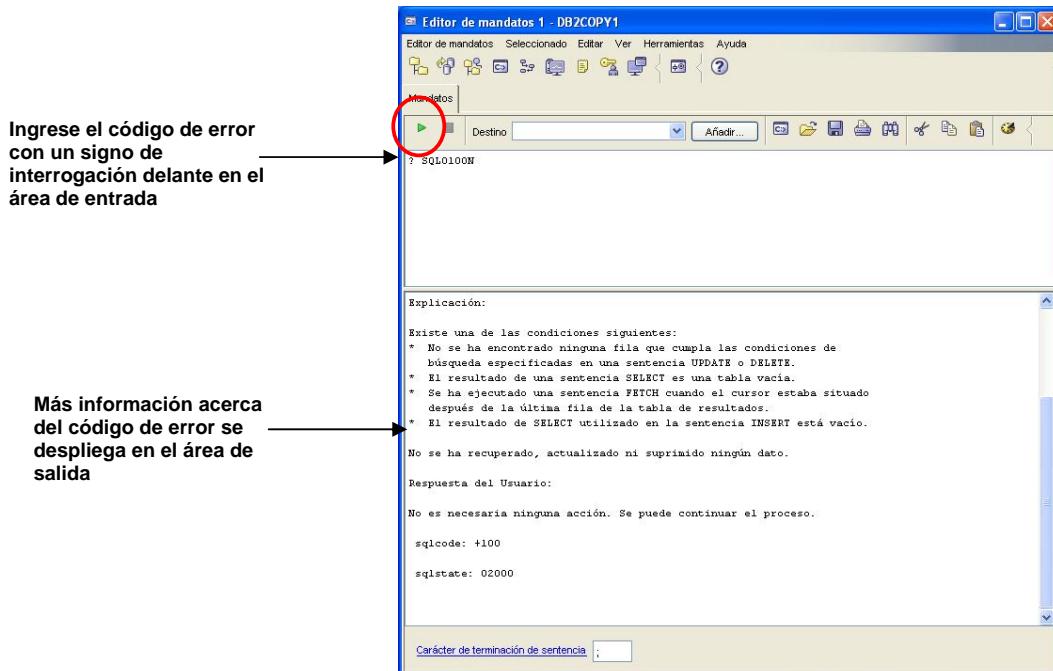


Figura A.2 – Como encontrar más información sobre los códigos de error de DB2

El signo de interrogación (?) invoca el comando de ayuda de DB2. Abajo se ven varios ejemplos de como invocarlo si se recibiera, por ejemplo, el código de error SQL “-104”. Todos estos ejemplos son equivalentes.

```
db2 ? SQL0104N
db2 ? SQL104N
db2 ? SQL-0104
db2 ? SQL-104
db2 ? SQL-104N
```

A.2 SQLCODE y SQLSTATE

Un SQLCODE es un código que se recibe cada vez que se ejecuta una sentencia SQL. A continuación un resumen del significado de sus valores:

- SQLCODE = 0; la sentencia se ejecutó satisfactoriamente
- SQLCODE > 0; la sentencia se ejecutó satisfactoriamente, pero dio un aviso
- SQLCODE < 0; la sentencia terminó mal y retornó un error

El SQLSTATE es una cadena de cinco caracteres que cumple con el estándar ISO/ANSI SQL92. Los primeros dos caracteres son conocidos como la clase del SQLSTATE:

- Si la clase es 00 significa que la sentencia terminó satisfactoriamente.
- Si la clase es 01 implica un aviso.
- Si la clase es 02 corresponde a una condición no encontrada.
- Todas las otras clases se consideran errores.

A.3 El registro administrativo de notificaciones de DB2

El registro administrativo de notificaciones DB2 provee información de diagnóstico acerca de los errores en el punto de falla. En las plataformas Linux y UNIX, el registro administrativo de notificaciones es un archivo de texto llamado <nombre_de_la_instancia>.nfy (por ej. "db2inst.nfy"). En Windows, todos los mensajes administrativos de notificación se graban en el registro de eventos de Windows.

El parámetro de configuración DBM **notifylevel** permite a los administradores especificar el nivel de información a registrar:

- 0 – No se captura ningún mensaje de notificación (no se recomienda)
- 1 – Errores fatales o irrecuperables
- 2 – Requiere acción inmediata
- 3 – Información importante que no requiere acción inmediata (por omisión)
- 4 – Mensajes informativos

A.4 db2diag.log

El db2diag.log provee información más detallada que el registro administrativo de notificaciones de DB2. Normalmente es utilizado por el soporte técnico DB2 de IBM o por DBAs experimentados. En el db2diag.log se incluye la siguiente información:

- El sector del código DB2 desde donde se reportó el error.
- Identificadores de aplicación que permiten parear las entradas relativas a una aplicación entre los db2diag.logs de servidores y clientes.
- Mensaje de diagnóstico (que comienza con "DIA") explicando la razón del error.
- Cualquier otro dato de soporte, tal como la estructura de datos SQLCA y punteros a la ubicación de cualquier vaciado de memoria o archivos trap.

En Windows (fuera de Vista), el db2diag.log está ubicado por omisión bajo el directorio:

```
C:\Documents and Settings\All Users\Application  
Data\IBM\DB2\DB2COPY1\<nombre_de_la_instancia>
```

En Windows Vista, el db2diag.log está ubicado por omisión bajo el directorio:

```
C:\ProgramData\IBM\DB2\DB2COPY1\<nombre_de_la_instancia>
```

On Linux/UNIX, the db2diag.log is located by default under the directory:

```
/home/<propietario_de_la_instancia>/sqlllib/db2dump
```

El nivel de detalle del texto de diagnóstico es determinado por el parámetro de configuración del dbm cfg, DIAGLEVEL. El rango de valores es de 0 a 4, donde 0 es el menos detallado y el 4 tiene el mayor detalle. El nivel por omisión es 3.

A.5 Trazadores CLI

Para aplicaciones CLI, Java, PHP, y *Ruby on Rails*, se puede activar una facilidad trazadora CLI para encontrar errores en las aplicaciones. Esto se hace introduciendo algunos cambios en el archivo db2cli.ini en el servidor donde corre la aplicación. En el *Listado A.1* se ven las entradas más habituales del archivo db2cli.ini

```
[common]  
trace=0  
tracerefreshinterval=300  
tracepathname=/path/to/writeable/directory  
traceflush=1
```

Listado A.1 – Entradas en el archivo db2cli.ini para activar el trazador CLI

También existe un trazador de bajo nivel (db2trc), pero en general solo es útil para el soporte técnico de DB2.

A.6 Defectos y correcciones de DB2

Hay veces en que un problema que hayamos encontrado pueda tener como origen un defecto en DB2. IBM publica regularmente paquetes de correcciones (*fix packs*) que contienen correcciones para defectos (APARs). La documentación del paquete de correcciones contiene una lista de las correcciones contenidas en el paquete. Cuando se desarrollan aplicaciones nuevas, es siempre recomendable hacerlo usando el último paquete de correcciones disponible, de forma de beneficiarse de las últimas correcciones. The fix pack documentation contains a list of the fixes contained in the fix pack. Para saber cual es la versión y el nivel de paquete de correcciones con el que se está trabajando, en el Centro de control seleccione la opción Acerca de en el menú de Ayuda; o en la Ventana de mandatos escriba **db2level**. Tenga en cuenta que los paquetes de correcciones y el soporte técnico oficial de IBM para DB2 no están disponible para DB2 Express-C, en DB2 Express-C, las correcciones están incorporadas en la imagen misma.

B

Apéndice B – Referencias y recursos

B.1 Referencias

- [1] ZIKOPOULOS, P. *IBM® DB2® Universal Database™ and the Microsoft® Excel Application Developer... for Beginners*, dbazine.com article, April 2005 <http://www.dbazine.com/db2/db2-disarticles/zikopoulos15>
- [2] ZIKOPOULOS, P. *DB2 9 and Microsoft Access 2007 Part 1: Getting the Data...*, Database Journal article, May 2008 <http://www.databasejournal.com/features/db2/article.php/3741221>
- [3] BHOGAL, K. *Use Microsoft Access to interact with your DB2 data*, developerWorks article, May 2006. <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0605bhogal/>
- [4] SARACCO, C. et all. IBM Redbook *DB2 9: pureXML overview and fast start* July 2006. <http://www.redbooks.ibm.com/abstracts/sg247298.html>

B.2 Sítios web:

1. Sitio web de DB2 Express-C: www.ibm.com/db2/express
En este sitio podrá descargar la imagen de servidores DB2 Express-C, clientes DB2, controladores DB2, manuales, tener acceso al *blog* del grupo, lista de correos, inscripción, etc.
2. Forum de DB2 Express: www.ibm.com/developerworks/forums/dw_forum.jsp?forum=805&cat=19
Use el forum para hacer consultas técnicas cuando no ha sido capaz de encontrar la respuesta en los manuales por sí mismo(a).
3. Centro de información DB2
El Centro de información provee acceso a los manuales en línea (en inglés) y es la fuente de información más actualizada y al día. Para cada versión de DB2 hay un correspondiente Centro de información DB2:
 - DB2 9.1: <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>
 - DB2 9.5: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp>

- DB2 9.7: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>
4. Manuales de DB2 9.7 traducidos a varios idiomas (Entre ellos el español – En formato PDF):
https://www-304.ibm.com/support/docview.wss?rs=71&uid=swg27015149&wv=1#es_main
 5. developerWorks: <http://www.ibm.com/developerworks/db2>

Este sitio web es un excelente recurso para desarrolladores y DBAs dando acceso a los más recientes artículos, guías de aprendizaje, etc. en forma gratuita.
 6. alphaWorks: <http://www.alphaworks.ibm.com/>

En este sitio web se encuentra en forma directa la tecnología emergente de IBM proveniente de los centros de investigación de IBM.
 7. planetDB2: www.planetDB2.com

Conjunto de *blogs* de varios autores que escriben sobre DB2.
 8. Soporte técnico DB2: http://www.ibm.com/software/data/db2/support/db2_9/

Aquí podrá buscar información acerca de defectos, reportes de problemas (APARs) y otra información técnica.
 9. ChannelDB2: <http://www.ChannelDB2.com/>

ChannelDB2 es una red social para la comunidad DB2. Allí se encuentran videos relacionados con DB2, demostraciones, *podcasts*, *blogs*, discusiones, recursos, etc. para Linux, UNIX, Windows, z/OS, y i5/OS.

B.3 Libros

1. Redbook gratis: 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, August 2006 - SG24-7301-00
<http://www.redbooks.ibm.com/abstracts/sg247301.html?Open>
2. Redbook gratis: DB2 pureXML Guide
Whei-Jen Chen, Art Sammartino, Dobromir Goutev, Felicity Hendricks, Ippei Komi, Ming-Pang Wei, Rav Ahuja, Matthias Nicola. August 2007
<http://www.redbooks.ibm.com/abstracts/sg247315.html?Open>
3. Redbook gratis: Developing PHP Applications for IBM Data Servers.
Whei-Jen Chen, Holger Kirstein, Daniel Krook, Kiran H Nair, Piotr Pietrzak May 2006 - SG24-7218-00
<http://www.redbooks.ibm.com/abstracts/sg247218.html?Open>
4. Understanding DB2 – Learning Visually with Examples V9.5
Raul F. Chong, et all. January 2008

ISBN-10: 0131580183

5. DB2® SQL PL: Essential Guide for DB2® UDB on Linux™, UNIX®, Windows™, i5/OS™, and z/OS®, 2nd Edition

Zamil Janmohamed, Clara Liu, Drew Bradstock, Raul Chong, Michael Gao, Fraser McArthur, Paul Yip

ISBN: 0-13-100772-6

6. DB2 9: pureXML overview and fast start

Cynthia M. Saracco, Don Chamberlin, Rav Ahuja June 2006 SG24-7298

<http://www.redbooks.ibm.com/abstracts/sq247298.html?Open>

7. Information on Demand - Introduction to DB2 9 New Features

Paul Zikopoulos, George Baklarz, Chris Eaton, Leon Katsnelson

ISBN-10: 0071487832

ISBN-13: 978-0071487832

B.4 emails de contacto

Buzón general de DB2 Express-C (Para preguntas de tipo administrativo): db2x@ca.ibm.com

Buzón general del programa DB2 on Campus: db2univ@ca.ibm.com



Conocer DB2 no podría ser más fácil. Lea este libro y:

- **Descubra el DB2 utilizando la versión gratuita, DB2 Express-C**
- **Comprenda la arquitectura, herramientas, y seguridad de DB2**
- **Aprenda como administrar bases de datos DB2**
- **Escriba SQL, XQuery, procedimientos almacenados**
- **Desarrolle aplicaciones para DB2**
- **Practique con los ejercicios que se incluyen**

DB2 Express-C de IBM es la versión sin costo del servidor de base de datos DB2 para manejar datos relacionales, como datos XML con facilidad. Sin costo significa que DB2 Express-C es gratis para bajar, gratis para desarrollar aplicaciones, gratis para desplegar en producción, y gratis para distribuir con su aplicativo. DB2 no pone límites artificiales en el tamaño de la base de datos, el número de bases de datos, o el número de usuarios.

DB2 Express-C corre en sistemas Windows y Linux e incluye los drivers de muchos lenguajes de programación tales como C/C++, Java, .NET, PHP, Perl, y Ruby. Opcionalmente, a un costo bajo puede comprar una suscripción de soporte técnico y otras características. Si requiere de mayor escalabilidad o funcionalidades mas avanzadas, usted puede sin problemas desplegar las aplicaciones que creó usando DB2 Express-C a otras ediciones de DB2 tal como el DB2 Workgroup o el DB2 Enterprise.

Esta edición gratis de DB2 es ideal para desarrolladores, consultores, distribuidores, administradores de base de datos (DBAs), estudiantes, o cualquiera que tenga la intención de desarrollar, probar, desplegar, o distribuir aplicaciones de bases de datos. Únete a la comunidad de usuarios de DB2 Express-C que crece cada día y prueba el DB2 Express-C. Empieza a descubrir como puedes crear la siguiente generación de aplicativos y desarrolle soluciones innovadoras.

Para más detalles, o para descargar el DB2 Express-C, visite:

ibm.com/db2/express

Para socializar con la comunidad de DB2, vea los videos y blogs en:

Channeldb2.com

Este libro es parte de la serie "DB2 on Campus". Visite db2university.com para más información.



Precio: 24.99 dólares