

Sistema Gestor de Base de Datos relacional: DB2 (y III)

Siguiendo con la línea iniciada hace dos meses, este mes le toca el turno al sistema gestor Relacional, y para ello se ha elegido el que cuenta con mayor número de licencias instaladas en los grandes centros de procesos de datos de España, el DB2 de IBM.

Introducción:

Este sistema de gestión es un sistema basado en el **modelo relacional**. Esto quiere decir que las relaciones entre los distintos ficheros (o *tablas*, usando la terminología relacional) se establece mediante los propios valores de los campos, no mediante apuntadores como se hace en los modelos jerárquicos, en red, o basados en listas invertidas, vistos anteriormente.

Para poder llevar a cabo su trabajo, este gestor mantiene un *catálogo* con la información que necesita. Este catálogo está formado por las **30 tablas** que se enumeran en el cuadro de la **figura 1** (se incluye al final del artículo) junto con una breve descripción. La **figura 2** muestra un fragmento del modelo de datos con las relaciones existentes entre las distintas tablas que componen dicho catálogo.

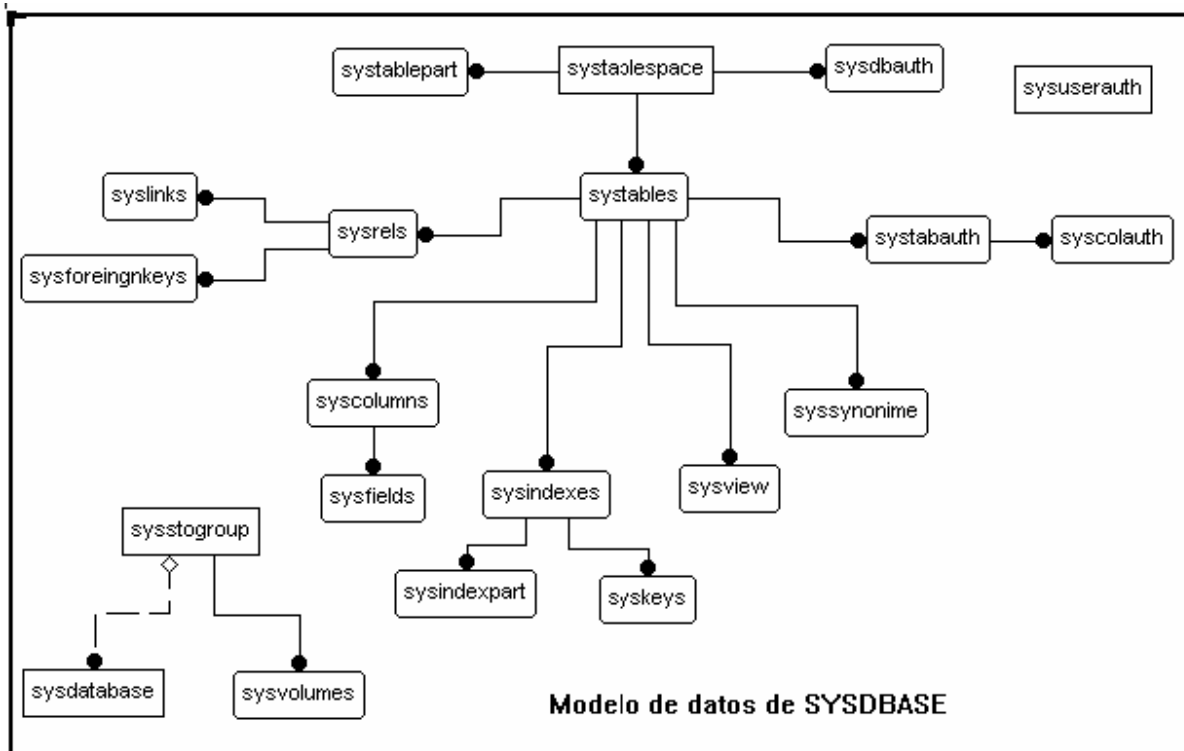


Figura 2

Terminología:

Una de las particularidades de este modelo es el uso de una terminología especial, por eso en este apartado se enumeran aquellos términos más comunes del entorno relacional.

Tabla o relación: Es un conjunto o lista de elementos homogéneos que puede representarse mediante una matriz de dos dimensiones, en la que cada una de las columnas establece los atributos o características definibles a cada uno de los elementos de la tabla, y por lo tanto cada fila contiene todas las características de una ocurrencia.

Grado de una tabla es el número de atributos definidos en una tabla.

Dominio de un atributo: Conjunto de valores que podrán usarse como contenido del atributo especificado.

Atributo o columna: cada campo de un registro,

Fila o tupla : cada uno de los registros de una tabla.

Entidad: elemento del *modelo de datos* que se implementa físicamente mediante una tabla en la base de datos.

Claves

Un concepto importante, y básico para este gestor, es que cada una de las ocurrencias (registro) de una tabla debe estar perfectamente identificada mediante al menos un atributo (campo) único, pues a la hora de recuperar o actualizar cada ocurrencia, solo podremos referenciarla con ese valor.

Las claves, como se puede intuir, juegan un papel muy importante en este modelo, ya que van a facilitar la recuperación de los datos, debiéndose diferenciar los siguientes tipos de clave:

Clave candidata: Atributo o conjunto de atributos que permiten identificar cada ocurrencia de una tabla de forma unívoca. Ejemplo: Cod_empleado y DNI_empleado.

Clave principal: La clave elegida como tal entre las distintas claves candidatas. Se suele representar por PK (primary Key). Ejemplo: COD_empleado

Clave Externa : Son aquellos atributos de una tabla que son clave Primaria en otra (FK:Foreign Key). De esta forma es como se establecen las relaciones, *migrando* la clave primaria de una tabla como claves externas de todas aquellas con las que se establezca la relación. Ejemplo: Cod_provincia.

Índices

A la hora de implementar físicamente el modelo de datos relacional, y con objeto de aumentar la eficiencia con la que DB2 debe acceder a los datos, se pueden definir índices físicos, los cuales son como cualquier índice de cualquier sistema de ficheros, estando cada entrada compuesta por el valor de la clave simbólica y un identificador de registro o RID, siendo este RID una numeración interna que identifica de modo único cada fila para DB2.

Físicamente el RID está compuesto de los siguientes elementos

- *Numero de página*: (3 Bytes) siendo la página, el espacio físico de la tabla donde se almacena la fila (registro).
- *Identificador (ID)* (1 Byte) el número de la entrada de la Tabla de localización de registros (TLR) de la página referenciada en los tres primeros Bytes.

El número de índices que se pueden definir sobre una tabla depende principalmente del modo en el que se van a procesar las filas en ella contenidas, ya que los índices reducen las operaciones de I/O en los procesos de consulta, pero los aumentan cada vez que se actualiza una columna que forma parte del índice, o cuando se borran o añaden filas.

Dentro de este apartado, se debe resaltar el hecho de que en DB2 no es necesario definir índices, ya que en este caso, el gestor, para acceder a los datos de una tabla, realiza un barrido de las páginas del espacio definido para esa tabla. Por regla general, las tablas que ocupen menos de 4 páginas (1 página = 4k) no suelen tener índices definidos, ya que, aunque los tengan, DB2 al decidir el medio que usará para recuperar la información, optará por el barrido físico de la tabla. En cambio, las de tamaño superior, suelen tener asociado un índice, que se denomina *cluster*, y que se suele corresponder con la clave primaria de la tabla.

Cuando existe un índice cluster, DB2 intentará almacenar las filas en las páginas de la tabla, en la misma secuencia que se encuentren las entradas en el índice cluster, es decir, las agrupará físicamente en base al valor del índice cluster.

Arquitectura:

Toda la información gestionada por DB2, el propio gestor la agrupa en subsistemas, entendiendo por tal, un completo, independiente y autónomo entorno DB2, lo cual quiere decir que:

- Tiene su propio catálogo
- Puede instalarse en la misma CPU que otro subsistema, o en CPUs separadas.
- Puede tener un interface con otros subsistemas vía DDF (Distributed Data Facility)

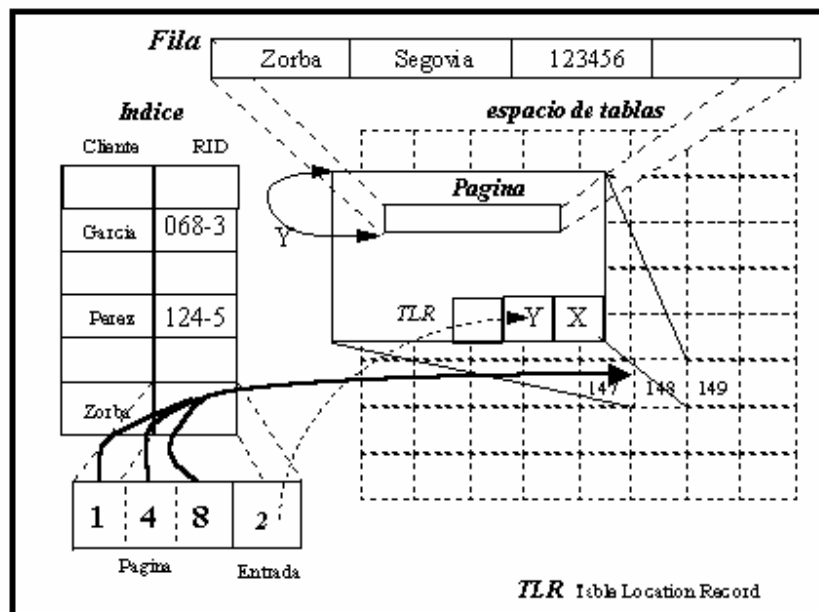


Figura 3

La figura 3 muestra de forma esquemática todos los elementos de la arquitectura de la Base de datos. Físicamente las distintas ocurrencias de una tabla se localizan en las páginas que tiene asignadas la tabla a la que pertenece dentro de su TABLESPACE. Cada una de estas páginas dispone en las últimas posiciones de la misma, de una tabla (la TLR) en la que cada entrada contiene los desplazamientos relativos a principio de página de cada una de las ocurrencias almacenadas en dicha página.

Por otra parte, se ha querido mostrar en la misma figura la composición de un índice, y cómo se puede localizar mediante el RID un registro.

El lenguaje relacional:

Este es el segundo de los componentes principales de un modelo relacional, ya que permite la recuperación/actualización de los datos mediante el uso de un lenguaje preciso y estructurado, el **SQL** (Structured Query Language), el cual además se ajusta a los estándares ISO y ANSI.

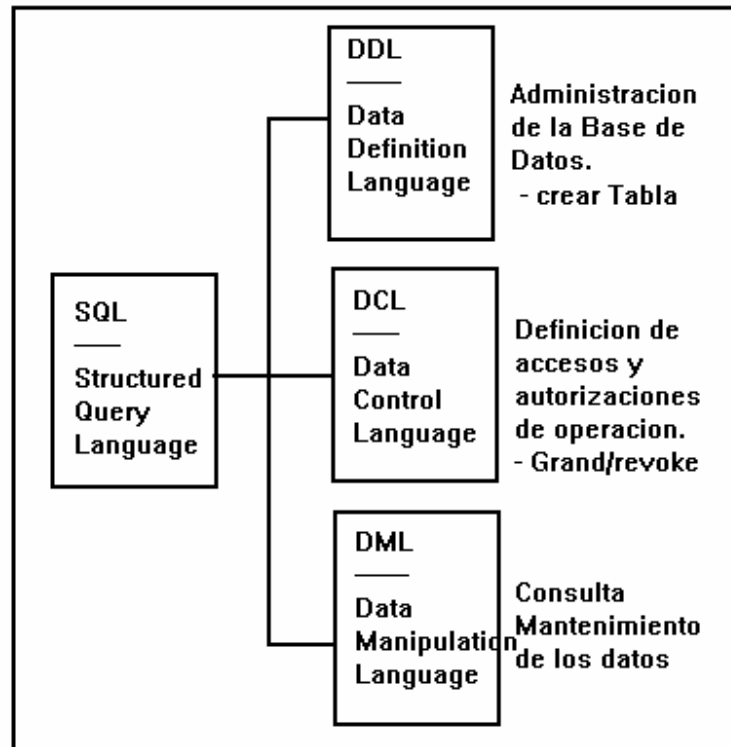


Figura 4

La figura 4 muestra los subconjuntos en los que se pueden agrupar los distintos comandos de este lenguaje, así como el uso al que se destinan, siendo, desde el punto de vista del programador, el SQL/DM el que mas importancia tiene, ya que es el que tendrá que manejar mas frecuentemente para acceder a los datos.

El criterio usado para este acceso es:

- **QUE** columnas se quiere recuperar
- **DESDE** qué tabla o vista
- **DONDE** va a colocar la información leída
- en qué **ORDEN** va a proporcionar los datos.

```

SELECT cod_cliente, nom_cliente, tel_cliente
FROM tab_cliente, his_cliente
WHERE apel_cliente="perez"

```

Ejemplo :
Recuperar los datos de código, nombre y telefono de la tabla de CLIENTES y de la tabal HISTORICO de clientes, correspondientes al cliente que tiene por primer apellido PEREZ .

Figura 5

Como se puede apreciar en el ejemplo de la figura 5, básicamente, la idea que predomina en SQL, es la de especificar qué el lo que se quiere, y no especificar cómo debe hacerlo el gestor.

BÁSICAMENTE, LA IDEA QUE PREDOMINA EN SQL, ES QUE SE ESPECIFICA QUÉ ES LO QUE SE QUIERE, PERO NO SE ESPECIFICA CÓMO

Esto obliga a introducir algunos conceptos nuevos en el proceso de compilación de los programas, quedando éste como se muestra en la figura 6 que se acompaña.

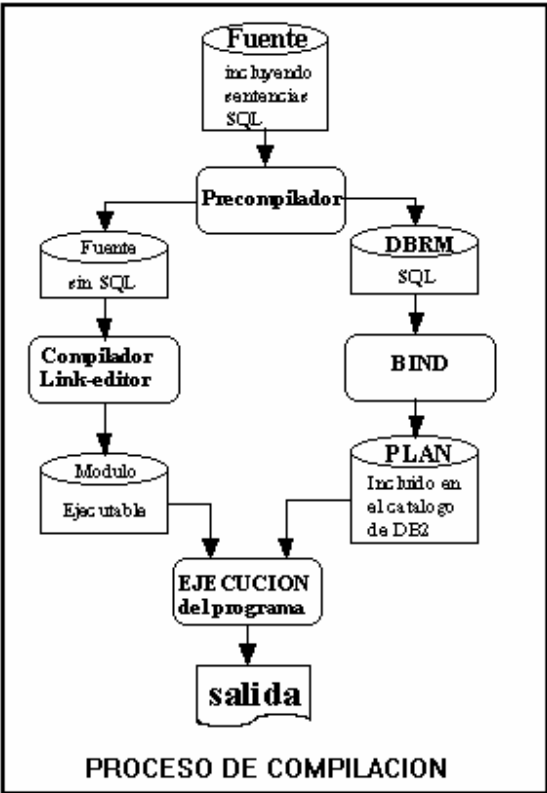


Figura 6

Así, un programa escrito en cualquier lenguaje convencional de programación que quiera usar datos almacenados

en DB2, deberá incluir sentencias SQL enmarcándolas entre las palabras reservadas **EXEC** y **END-EXEC** con las que poder decir al gestor, qué información es la que desea recuperar. El primer paso por tanto, para compilar el programa, debe consistir en eliminar esas sentencias que no son del lenguaje en el que se escribe el programa, y que por lo tanto, no entenderá el compilador. En este proceso de precompilación pasan a ser comentarios las sentencias SQL, siendo sustituidas por sentencias CALL que invocan a determinados módulos. Así mismo, en este primer paso, se extraen a otro fichero todas las sentencias SQL, para su posterior tratamiento, constituyendo los módulos denominados DBRM.

En el siguiente paso, se compila el programa principal, pues ya todas las sentencias son del lenguaje conocido por el compilador, obteniéndose un modulo objeto, para convertirse posteriormente después de pasar el proceso del montador, en modulo ejecutable.

Por otra parte, a los módulos DBRM que se obtuvieron en el proceso de precompilación, se les debe aplicar el proceso de BIND a fin de establecer la estrategia de acceso a los datos que seguirá DB2 cuando sea invocado. Esta estrategia, o elección del camino a seguir para la ejecución de una sentencia SQL, se toma en base a las estadísticas que lleva el propio gestor, tales como número de páginas que ocupa la tabla, índices definidos, etc. El resultado de esta estrategia es almacenada por DB2 en una de sus tablas de gestión, y es lo que constituye un PLAN.

De esta forma, es la combinación Modulo-ejecutable/Plan lo que permite la ejecución de un programa que trabaje contra tablas DB2.

Integridad referencial:

Con objeto de asegurar la integridad referencial, cuando se selecciona una fila para modificarla, se bloquea no solo el registro seleccionado, sino toda la página en la que se encuentra, pues es a este nivel en el que se encuentran unos indicadores dispuestos para tal efecto, como ya se expondrá en próximos artículos.

Hay que tener en cuenta que no es suficiente con identificar las claves primarias de una entidad con la externa de otra, para asegurar la integridad, sino que hay un conjunto de reglas que se tienen que aplicar a cada relación cuando se insertan o actualizan filas con claves externas o cuando se borran claves primarias. Así, cuando se intenta borrar una fila de una clave primaria, las reglas de integridad referencial definen lo que debe ocurrir en el resto de tablas que contienen como clave externa el valor borrado, debiéndose aplicar una de las siguiente reglas:

- *Borrado restringido*: No se permite borrar la fila de la clave primaria.
- *Borrado con neutralización*: Se ponen a nulo todos los valores de la clave externa que son iguales a la clave primaria que se quiere borrar.
- *Borrado en cascada*: Todas las filas de la clave externa con un valor igual a la clave primaria de la fila que se borra, se borran también.

Reglas similares deben aplicarse en inserción, pero en este caso, es el valor especificado como clave externa el que debe existir en la columna de la clave primaria de la tabla padre. Y, en actualización, el nuevo valor especificado en una clave externa tiene que ser igual a un valor existente en la tabla padre, o tener valor nulo.

Seguridad en DB2:

Por último, se quiere resaltar otro tema muy importante en DB2. Es el tema de las autorizaciones. IBM tomó desde el principio la decisión de que cualquier objeto creado por DB2 debería ser controlado por DB2. Mucho se podría hablar sobre este tema, pero con objeto de no dejarle en el tintero simplemente se enuncian las tres fuentes de autorización que hay en DB2:

- *Creando Objetos DB2*: Como creador de un objeto, se tienen ciertos privilegios sobre dicho objeto, sin necesidad de haber sido autorizado.
- *Autorización administrativa*: Son grupos de privilegios que se unen a entornos de trabajo de DB2.

- El comando *GRANT*: esta sentencia SQL/DC se usa para asignar privilegios.

En fin, como ya se ha dicho anteriormente, cada uno de los puntos enunciados en estos tres artículos, pueden ser la introducción de un tema monográfico mucho mas amplio, pero la idea que movió al autor de estos artículos fue la de proporcionar al desarrollador una visión general de los SGDB desde la perspectiva de sistemas, pues este conocimiento facilita en muchas ocasiones el desarrollo de programas.

Jose Maria Peco

Figura 1:

NOMBRE	CONTENIDO
	<u>Informacion sobre tablas y vistas:</u>
SYSTABLES	1 fila por cada tabla del sistema.
SYSVTREE	1 fila por cada vista (los primeros 4000 bytes)
SYSVLTREE	1 fila por cada vista (los restantes bytes)
SYSVIEWS	1 o mas filas por cada vista
SYSCOLUMNS	1 fila por cada columna de tabla o vista
SYSFIELDS	1 fila por cada columna que tenga asociado un procedimiento en FLDPROC
SYSSYNONYMS	1 fila por cada sinonimo de una tabla o vista.
	<u>Informacion sobre espacios para tablas y sistema</u>
SYSTOGROUP	1 fila por cada grupo de almacenamiento.
SYSVOLUMNES	1 fila por cada volumen de cada grupo de almacenamiento.
SYSDATABASE	1 fila por cada base de datos, excepto para DSNDB01
SYSTABLESPACE	1 fila por cada tablespace.
SYSTABLEPART	1 fila por cada tablespace no particionado; y 1 fila por cada particion de los particionados.
SYSINDEXES	1 fila por cada indice, incluyendo los del catalogo.
SYSINDEXPART	1 fila por cada indice no particionado; y 1 fila por cada particion de lso particionados.
SYSKEYS	1 fila por cada columna clave de un indice.
	<u>Informacion sobre PLANES de aplicacion</u>
SYSPLAN	1 fila por cada plan de aplicacion
SYSDBRM	1 fila por cada DBRM de la aplicacion
SYSSTMT	1 o mas filas por cada sentencia SQL de cada DBRM
	<u>Informacion sobre dependencias</u>
SYSPLANDEP	Registra las dependencias de los planes respecto a tablas, vistas, sinonimos, espacios para tablas e indices.
SYSVIEWDEP	Registra las dependencias de las vistas con respecto a las tablas y a otras vistas.
	<u>Informacion sobre autorizaciones</u>
SYSTABAUTH	Privilegios de los usuarios sobre tablas y vistas
SYSCOLAUTH	Privilegios de actualizacion de los usuarios sobre las distintas columnas de las tablas y vistas.
SYSDBAUTH	Privilegios de los usuarios sobre bases de datos.

SYSRESAUTH	Privilegios de los usuarios sobre los pools de buffers, grupos de almacenamiento, y espacios para tablas.
SYSPLANAUTH	Privilegios de los usuarios sobre planes de aplicacion
SYSUSERAUTH	Privilegios de los usuarios sobre los recursos del sistema

Informacion de control interno

SYSFOREINGKEYS	1 fila por cada columna de clave externa dentro de las tablas del catalogo
SYSLINKS	1 fila por cada LINK (informacion interna del catalogo)
SYSRELS	1 fila por cada LINK entre las tablas del catalogo.
