

4. SQL

Este capítulo³ contiene todo lo relativo a las características de las bases de datos, se menciona la clasificación de los diversos modelos que existen para manejar bases de datos. Se explica de una forma detallada el modelo de base de datos relacional, además de explicar algunos términos como **DBMS** y **RDBMS**. Posteriormente se expone lo que es SQL (en inglés, *Structured Query Language*) o lenguaje de consulta estructurado, que es un tipo de **RDBMS**. Por último, presentaré los comandos básicos de SQL y cómo se utilizaron para la creación y población del sistema desarrollado en esta tesis.

Base de datos relacionales

Antes de abordar directamente lo que son las bases de datos relacionales y específicamente SQL, hay que mencionar que éstas no son el único tipo de base de datos que existe. A lo largo de la historia se han ido creando diversos modelos o métodos para manejar la información contenida en las bases de datos. Estos modelos, en donde se guarda la información, por lo general no son algo tangible. Más bien son algoritmos y conceptos matemáticos que también nos permiten almacenar y recuperar información.

³ La información presentada en este capítulo se obtuvo de revisar las siguientes fuentes: Momjian Bruce, *PostgreSQL: Introduction and Concepts*, Addison Wesley 2001; Douglas Korry, *PostgreSQL*, Sams Publishing; Coles Michael, *Pro T-SQL 2008 Programmer's Guide*, 2008; y PostgreSQL 8.3.1 Documentation. Adicionalmente, se consultaron las siguientes páginas:

- <http://www.elsevier.com/wiki/SQL>
- http://www.htmlpoint.com/sql/sql_08.htm
- http://es.wikipedia.org/wiki/Base_de_datos
- <http://www.scribd.com/doc/13664065/007-Bases-de-Datos>
- <http://arquitecturaestefany.blogspot.com/>

Los más comunes son:

- Bases de datos jerárquicos
- Bases de datos reticulares
- Bases de datos de red
- Bases de datos multidimensionales
- Bases de datos relacionales
- Base de datos por objetos

Los dos primeros modelos ya no son muy usados. La mayoría de las bases de datos que se usan en la actualidad pertenecen al tipo relacional. La razón de la popularidad de este tipo es ofrecer la posibilidad de construir sistemas simples y efectivos para representar y manipular los datos. Se basa en relaciones. A un conjunto de datos se les llama tupla. Para explicarlo de una manera más sencilla se podría imaginar que cada relación fuera una tabla que está formada por registros que son las filas o renglones de una tabla, que simbolizarían las tuplas; los campos serían las columnas de la tabla. Como ya se mencionó, todos los modelos de bases de datos se ayudan de algoritmos y conceptos matemáticos. En el modelo relacional se tienen bases teóricas sólidas del álgebra y cálculo relacional, que permite que la información pueda ser recuperada o almacenada fácilmente por medio de comandos llamados “consultas” (*queries*), de forma sistemática y ordenada.

Para manejar y organizar una base de datos se utiliza un **DBMS** (*DataBase Management System*, o sistema de gestión de bases de datos). Estos son un tipo de software que tiene la finalidad de administrar el funcionamiento de la base de

datos, permitiendo almacenar y acceder a la información contenida en dicha base de forma rápida y organizada, ya sean requeridas por un usuario o por alguna aplicación o software que necesite utilizar dicha base para realizar alguna tarea específica. Ya se mencionaron superficialmente algunas características de las bases de datos que existen y en forma general lo que son los **DBMS**, pero debido a que la tesis tiene como finalidad la creación de una base de datos relacional, se abordará más detalladamente lo que son las bases de datos relacionales, y comenzaré describiendo algunas de sus características.

Una base de datos relacional está compuesta por varias tablas. En ella no pueden existir dos tablas con el mismo nombre, y cada tabla es a su vez, como se dijo arriba, un conjunto de registros, filas o tuplas. Cada uno de estos registros consta de varias columnas, campos o atributos.

Algunas de las restricciones que existen en la creación de este tipo de bases de datos, son que en este tipo no existen dos columnas que se nombren de la misma manera en una misma tabla, los datos que se encuentran almacenados en una columna deben ser del mismo tipo, y todas las filas de una misma tabla tienen el mismo número de columnas. Es importante considerar todas estas restricciones para evitar problemas en el momento de realizar el llenado o poblado de la base de datos.

Para obtener la información que está contenida dentro de una base de datos se utilizan los llamados manejadores de bases de datos relacionales, como ya se mencionó, los que a su vez utilizan un lenguaje relacional para realizar

dichas tareas. Actualmente para realizar todas las consultas relacionadas con las bases de datos se cuenta con dos lenguajes formales, el álgebra relacional y el cálculo relacional. Todas las consultas pueden realizarse aplicando uno u otro método. Sin embargo, puede existir una mayor dificultad si se efectúa una u otra opción. Esto se debe a la forma en que se realiza la consulta. La diferencia más importante es que el álgebra relacional tiene que especificar detalladamente los pasos a seguir para obtener una respuesta o consulta, mientras que en el cálculo relacional la respuesta se obtiene sin saber cuál fue el procedimiento para obtener dicha respuesta.

Como se puede deducir, los lenguajes que usan el cálculo relacional son de "más alto nivel" o "más declarativos" que los que se basan en el álgebra relacional porque en esta última se va describiendo, aunque sea de una manera parcial, el orden en que se deben seguir las instrucciones. En cambio, en el cálculo relacional el resultado que se desea obtener se introduce a un compilador o intérprete que lo analiza y busca el procedimiento más efectivo para obtener el resultado de la consulta.

Como se puede observar los manejadores de bases de datos son fundamentales para administrar toda la información contenida en ellas, auxiliándose de lenguajes para realizar las consultas. El más utilizado es SQL y debido a que este fue el usado para la creación de los comandos que permitieron la construcción y alimentación de la base de datos descrita en esta tesis, conviene abordar de una manera más detallada sus principales características y algunos de

sus comandos básicos para la creación y población de una base de datos relacional.

Manejadores de base de datos relacionales y SQL

Como se explicó anteriormente, para manejar bases de datos relacionales existen programas especializados. Este software es conocido como **SGBD** (sistema de gestión de base de datos relacional) o, en inglés, **RDBMS** (*Relational database management system*).

El lenguaje más común para construir las consultas de bases de datos relacionales es SQL (como se dijo arriba, *Structured Query Language*). Se trata de un estándar implementado para los principales motores o sistemas de gestión de bases de datos relacionales.

El modelo SQL necesita que las columnas tengan un orden definido (aunque esto no es necesario en un modelo relacional), gracias a esta característica resulta fácil implementarlo a una computadora, ya que la memoria es lineal.

SQL es un lenguaje que se usa para ejecutar acciones que administran la base de datos. Una de sus características es que mediante el manejo del álgebra y/o el cálculo relacional nos permite realizar consultas con el fin de recuperar información de interés de una base de datos (como ya se explicó arriba), así como también hacer cambios sobre la misma de una manera simple y ordenada.

Los sistemas más populares que utilizan SQL son:

- Oracle
- Sybase
- Microsoft SQL Server
- Ingres
- MySQL
- PostgreSQL

Todos los sistemas tienen comandos especiales que nos permiten administrar de una mejor manera la base de datos (agregados al lenguaje). Sin embargo, con los comandos básicos (*Select*, *Insert*, *Update*, *Delete*, *Create* y *Drop*) pueden realizarse la mayoría de las tareas necesarias para administrar y consultar una base de datos.

Otra característica importante que tienen los gestores son que no solamente nos permiten alimentar y consultar datos almacenados en la base de datos, sino que además nos proporcionan herramientas, que garantizan evitar la duplicidad de registros. También garantizan la integridad referencial. Esto quiere decir, que al eliminar un registro se eliminan todos los registros relacionados dependientes entre sí.

Hay que mencionar que a pesar de que las bases de datos relacionales solucionan la mayor parte de la administración de las bases de datos, presentan carencias con datos gráficos (imágenes), multimedia, CAD y sistemas de información geográfica y no se manipulan de forma eficiente los bloques de texto como tipo de dato. Para satisfacer los casos antes mencionados, donde se

presentan ciertas carencias, se crearon las bases de datos orientadas a objetos (**BDOO**) que complementan pero no reemplazan a las bases de datos relacionales.

Creación de bases de datos

Una base de datos consiste en una serie de tablas que tienen relaciones entre sí, que se utilizan para almacenar y obtener información de una manera rápida. La parte fundamental de crear una base de datos consiste en la creación de las tablas que la componen. Sin embargo antes de proceder a la creación de las tablas, normalmente hay que crear la base de datos, lo que a menudo significa definir un espacio de nombres separado para cada conjunto de tablas. De esta manera, para un **SGBD** se pueden administrar diferentes bases de datos independientes al mismo tiempo sin que ocurran conflictos con los nombres que se usan en cada una de éstas. Hay que mencionar que aunque SQL es un estándar, cada **SGBD** que utiliza el lenguaje SQL puede tener un procedimiento propietario para crear una base de datos. Esto quiere decir que además de utilizar los comandos SQL comunes del lenguaje, pueden existir comandos más cortos que tengan la misma función cambiando el nombre de las instrucciones previstas en el estándar. A estos comandos adicionales se les llama agregados al lenguaje.

Por ejemplo, la sintaxis estándar empleada en SQL, que usan todas las **SGBDs** más conocidas, es la siguiente:

```
CREATE DATABASE nombre_base de datos
```

Pero además de utilizar la sintaxis arriba mencionada para la instrucción de crear una base de datos, se puede utilizar un agregado de lenguaje propio de PostgreSQL, que aunque se escribe diferente realiza la misma operación que la del estándar SQL. El número de caracteres es menor, lo que nos ahorra tiempo al momento de ejecutar los comandos.

CREATEDB nombre_base de datos

Una vez creada la base de datos, se pueden crear las tablas que la componen. La instrucción SQL para este fin es:

*CREATE TABLE nombre_tabla (
nombre_columna tipo_columna [cláusula_defecto] [vínculos_de_columna]
[, nombre_columna tipo_columna [cláusula_defecto] [vínculos_de_columna] ...]
[, [vínculo_de_tabla] ...])*

Los tipos de datos de las columnas (*tipo_columna*) SQL se clasifican en 13 tipos de datos primarios y de varios sinónimos válidos reconocidos por dichos tipos de datos. Los tipos de datos primarios son:

<i>Binary</i>	<i>bit</i>	<i>byte</i>	<i>counter</i>
<i>currency</i>	<i>datetime</i>	<i>single</i>	<i>double</i>
<i>short</i>	<i>long</i>	<i>longtext</i>	<i>longbinary</i>
<i>text</i>			

Población de la base de datos

Cuando se habla de "poblar la base de datos", se está hablando de la operación de introducir datos en ella. En una base de datos relacional esto corresponde a la

creación de las líneas o tuplas que componen las tablas que forman la base de datos. Normalmente, la memorización de una información concreta corresponde a la introducción de una o más líneas en una o más tablas de la base de datos.

El orden de las operaciones de llenado no es de una manera desordenada. De hecho, la inserción de las líneas tiene que hacerse de modo que se respeten las relaciones formadas en las tablas. De lo contrario, no se podría recuperar la información de una manera correcta.

Hasta aquí, he presentado algunos conceptos básicos relacionados con SQL, que es el lenguaje que utilizaré para crear las bases de datos a partir del análisis de esquemas XML y la posterior población de dicha base, planeadas en esta tesis. En los siguientes capítulos presentaré el desarrollo que seguí para elaborar el constructor y alimentador de la base de datos, comenzando en el capítulo 5 con la metodología seguida.