

Bases de Datos I

Unidad 1: Introducción a las bases de datos

La **arquitectura cliente/servidor** es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina **cliente** al proceso que inicia el diálogo o solicita los recursos. Se denomina **servidor** a un conjunto de Hardware y Software que responde a los requerimientos de un cliente. Los Servidores proporcionan un servicio al cliente y devuelven los resultados. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD).

Los **Motores de Bases de Datos**, son programas específicos, dedicados a servir de intermediarios entre las bases de datos y las aplicaciones que las utilizan como función principal, cada uno de estos cumple una tarea específica, que van desde crear la Base de Datos hasta administrar el uso y acceso a esta.

Ejemplos de gestores de base más conocidos: MySQL, PostgreSQL, Sqlite que son libres y por otro lado tenemos IBM DB2, IBM Informix, Ms SQL Server, MS Access, Oracle, Sybase que son propietarios.

SQL (Structured Query Language) es un lenguaje de programación estándar e interactivo para la obtención de información desde una base de datos y para actualizarla. Aunque SQL es a la vez un ANSI y una norma ISO, muchos productos de bases de datos soportan SQL con extensiones propietarias al lenguaje estándar.

Las sentencias SQL se dividen en dos categorías; **Lenguaje de definición de datos**, data definition language (**DDL**) y **Lenguaje de manipulación de datos**, data manipulation language (**DML**).

Las sentencias DDL se utilizan para crear y modificar la estructura de las tablas así como otros objetos de la base de datos:

- CREATE - para crear objetos en la base de datos.
- ALTER - modifica la estructura de la base de datos.
- DROP - borra objetos de la base de datos.
- TRUNCATE - elimina todos los registros de la tabla, incluyendo todos los espacios asignados a los registros.

Las sentencias de lenguaje de manipulación de datos (DML) son utilizadas para gestionar datos dentro de los schemas. Algunos ejemplos:

- SELECT - para obtener datos de una base de datos.
- INSERT - para insertar datos a una tabla.
- UPDATE - para modificar datos existentes dentro de una tabla.
- DELETE - elimina todos los registros de la tabla; no borra los espacios asignados a los registros.

Un **Schema** es el conjunto de objetos que pertenecen a un espacio, este va a contener objetos como (tablas, índices, vistas, procedimientos almacenados, triggers).

Se define **Tabla** a una estructura que modela datos, la misma se asemeja a la vista general de un programa de Hoja de cálculo. Las tablas se componen de dos estructuras:

Campo: corresponde al nombre de la columna. Debe ser único y además de tener un tipo de dato asociado.

Registro: corresponde a cada fila que compone la tabla. Allí se componen los datos y los registros.

En la definición de cada campo, debe existir un nombre único, con su tipo de dato correspondiente. Esto es útil a la hora de manejar varios campos en la tabla, ya que cada nombre de campo debe ser distinto entre sí. A los campos se les puede asignar, además, propiedades especiales que afectan a los registros insertados. El campo puede ser definido como índice o autoincrementable, lo cual permite que los datos de ese campo cambien solos o sea el principal indicar a la hora de ordenar los datos contenidos.

La **clave primaria** es un campo, o grupo de campos que identifica en forma única un registro. Ningún otro registro puede tener la misma clave primaria, esta se utiliza para distinguir un registro con el fin de que se pueda tener acceso a ellos, organizarlos y manipularlos.

Una **clave foránea** (o Foreign key FK) es una limitación referencial entre dos tablas. La clave foránea identifica una columna o grupo de columnas en una tabla (tabla hija o referendo) que se refiere a una columna o grupo de columnas en otra tabla (tabla padre o referenciada). Las columnas en la tabla referendo deben ser la clave primaria u otra clave candidata en la tabla referenciada.

Dado un conjunto de relaciones en el que participan dos o más conjuntos de entidades, la correspondencia de cardinalidad indica el número de entidades con las que puede estar relacionada una entidad dada.

Dado un conjunto de relaciones binarias y los conjuntos de entidades A y B, la correspondencia de cardinalidades puede ser:

- **Uno a Uno:** (1:1) Un registro de una entidad A se relaciona con solo un registro en una entidad B. (ejemplo dos entidades, profesor y departamento, con llaves primarias, codigo_profesor y jefe_depto respectivamente, un profesor solo puede ser jefe de un departamento y un departamento solo puede tener un jefe).
- **Uno a varios:** (1:N) Un registro en una entidad en A se relaciona con cero o muchos registros en una entidad B. Pero los registros de B solamente se relacionan con un registro en A. (ejemplo: dos entidades, vendedor y ventas, con llaves primarias, código_vendedor y venta, respectivamente, un vendedor puede tener muchas ventas pero una venta solo puede tener un vendedor).
- **Varios a Uno:** (N:1) Una entidad en A se relaciona exclusivamente con una entidad en B. Pero una entidad en B se puede relacionar con 0 o muchas entidades en A (ejemplo empleado-area de trabajo).
- **Varios a Varios:** (N:M) Una entidad en A se puede relacionar con 0 o con muchas entidades en B y viceversa (ejemplo asociaciones-ciudadanos, donde muchos ciudadanos pueden pertenecer a una misma asociación, y cada ciudadano puede pertenecer a muchas asociaciones distintas).