Fundamentos de SQL

**Terminología básica**

En las bases de daros relacionales existe un vocabulario específico con el objetivo de que las personas dedicadas a esta actividad puedan entender de qué se trata cada uno de los elementos involucrados. A continuación, se tienen los siguientes conceptos:

* **Datos**: Unidad mínima de información. No provee un significado particular, simplemente es un ente que requiere de una interpretación para dar lugar a la información.
* **Información**: Es un conjunto de datos que a través del procesamiento adecuado adquieren un significado en un contexto determinado.
* **Conjunto de Datos (Dataset)**: Cuando queremos entender un fenómeno necesitamos recabar la mayor cantidad de datos posible.
  + Almacenamiento: Papel, digital (.csv, .tsv, Excel, emails, formas de Google, etc.)
* **Bases de datos**: Abstracción que incluyen reglas de estructuración de mis conjuntos de datos. Toma mis conjuntos de datos y los estructura según algúna característica que tengan en común, puede ser un tema, situación geográfica o algún elemento que los relacione, además de que pueden estar almacenados de manera tradicional (impresos en papel o en formato digital como un archivo.
* **Sistemas Gestores de Bases de Datos (SGDB Database Management System)**: Software que permite administrar mis bases de datos (almacenarlos, crearlos, actualizarlos, borrarlos, etc.).
  + **Bases de datos Relacionales**: **MySQL**, MariaDB, PostgreSQL, SQLite, etc.
    - **SQL**: Structured Query Language: es un lenguaje que nos permite comunicarnos con una base de datos relacional.
  + **Bases de datos no relacionales**: **MongoDB**.
    - Lenguaje de MongoDB.
* Jerarquía de una base de datos relacional.
  + 1. Bases de datos.
  + 2. Tablas.
  + 3. Filas (registros) y Columnas (Campos).
  + **Celda**: Contiene un dato.
    - **Tipos de datos**:
      * Números enteros.
      * Números decimales.
      * Cadenas de caracteres (text).
      * Fechas.
      * Booleanos (Falso y Verdadero).
      * Categorías.
      * Etc.
    - **Llave primaria**: Identificador único que nos sirve para nombrar una fila en específico.
    - **Llaves foráneas**: Identificador que funciona como referencia a una llave primaria en otra tabla distinta.
* **Tabla**: Una tabla o relación es una entidad con atributos, campos o columnas y tuplas, registros o renglones. Una tabla representa a un número indefinido de elementos descritos con características en común, cada uno en un diferente renglón.
* **Registros**: También llamados tuplas o renglones, se refieren a la descripción de cada ente de la tabla. Son instancias particulares, por ejemplo, si tenemos una tabla Persona, un registro contendrá una persona, o sea, sus datos particulares.
* **Campos**: O atributos, se refieren a las columnas de las tablas y representan una característica particular de cada entidad. Por ejemplo, para una persona podrían haber campos para el nombre, edad, dirección, etc. Es importante recordar que los campos son comunes a todos los registros de una tabla.
* **Dominio**: Se refiere al número de valores que puede obtener una columna, por ejemplo, si tuviéramos un campo para almacenar el estado civil, su dominio sería 2 en el caso de tener los valores soltero o casado.
* **Cardinalidad**: Es el número de registros que tiene una tabla.
* **Grado**: Es el número de campos que tiene una tabla.

**Conexión a una base de datos**

Las bases de datos por lo general tienen cientos y miles de registros.

Para almacenar las bases de datos, por lo general, una computadora personal como una laptop no son suficientes, por lo que se necesita alojar dentro de un servidor.

Un servidor, en pocas palabras, es un equipo de cómputo súper potente que forma parte de una red de computadoras y que provee distintos servicios a otros equipos llamados “clientes”.

A los servidores que almacenan bases de datos se les llama Servidores de Bases de Datos. Para conectarnos a un servidor, necesitamos cuatro cosas esenciales:

* La dirección del servidor (host).
* El usuario con el que nos vamos a conectar.
* Una contraseña de acceso.
* El puerto con el cual realizaremos la conexión.

**Comunicación con la base de datos**

Las bases de datos relacionales tienen su propio lenguaje, con éste nos podemos comunicar con ellas y pedirles que realicen algunas acciones, este lenguaje es llamado SQL y es un estándar para bases de datos relacionales. Cada SGBD tiene su propia implementación de SQL, añadiendo características especiales, pero sigue en general un estándar.

La primera instrucción de SQL a estudiar será USE, que nos ayuda a conectarnos a una base de datos. Su sintaxis es la siguiente:

* USE <nombre\_de\_la\_base>;
* USE inventario;

Para saber qué bases de datos hay en un servidor, se puede usar el comando SHOW:

* SHOW DATABASES;

Todas las instrucciones deben terminar con un punto y coma. Está diseñado de esa manera pues en SQL pueden definirse pequeños procedimientos que ejecutan varios comandos a la vez y esta es la forma en la que el lenguaje detecta donde termina e inicia una instrucción.

**Estructura de una Tabla**

Una vez conectados a la base de datos y dependiendo de cómo esté definida, podremos ver todas las tablas contenidas en la misma, mediante el comando SHOW:

* SHOW TABLES;

Como mencionamos antes, una tabla se conforma de un conjunto de columnas que describen cada registro que se almacena en la misma. Para conocer los campos de una tabla, puede usarse el comando DESCRIBE que nos muestra los campos de esta y el tipo de dato a almacenar.

* DESCRIBE <nombre\_de\_la\_tabla>;
* DESCRIBE movies;

Tabla

Descripción generada automáticamente

En realidad, MySQL arroja más resultados cuando realizamos una operación DESCRIBE. Sin embargo, hasta este punto basta con que analicemos estos dos campos.

El resultado de la descripción, nos indica que la tabla incluye tres campos: id, título y género y nos muestra el tipo de dato que almacena cada uno. Los tipos de datos nos permiten manipular de forma más eficiente los datos almacenados, por ejemplo, si tenemos un campo numérico, esto facilitará que podemos hacer alguna operación aritmética con ella.

En nuestro ejemplo se muestran dos tipos de datos: int y varchar que indican que cada campo almacena números enteros y [cadenas de caracteres](https://es.wikipedia.org/wiki/Cadena_de_caracteres) respectivamente.

**Estructura básica de una consulta**

Ahora que sabemos cómo está definida la estructura de una tabla, podemos obtener los registros que tiene. Para esto, haremos uso de la instrucción SELECT que permite obtener registros de una tabla a partir del nombre de los campos y de la tabla. A continuación, se muestra la estructura de un SELECT en su forma más básica.

* SELECT <campo1>, <campo2>, …  
  FROM <tabla>;

Esta estructura permite seleccionar (algunos autores lo llaman proyectar) los registros. El primer parte **< campo1 >**, **< campo2 >**, **…** indica los campos que se mostrarán, separados por comas. Por otro lado, la parte indica la tabla de la cual tomaremos los datos.

Por ejemplo, supongamos que queremos visualizar los registros almacenados en la columna título de la tabla movies, tendríamos que escribir la instrucción:

* SELECT titulo  
  FROM movies;

Tabla

Descripción generada automáticamente

Este tipo de instrucciones recibe el nombre de “consulta” pues como su nombre lo indica, se realiza una consulta de los registros que hay en la tabla correspondiente.

Observa que los resultados de una consulta siempre se muestran en una tabla, aún cuando sólo contengan un único registro.

Si quisiéramos traer todos los campos de una tabla, pueden reemplazarse los nombres de las columnas por un asterisco, por ejemplo:

* SELECT \*  
  FROM movies;

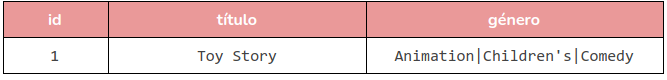
Tabla

Descripción generada automáticamente

También podemos aplicar filtros a los resultados con el fin de facilitar la búsqueda de datos. Por ejemplo, supongamos que queremos los datos de aquellos registros cuyo nombre es “Toy Story”.

Para filtrar resultados se añade la restricción WHERE a la consulta, por ejemplo:

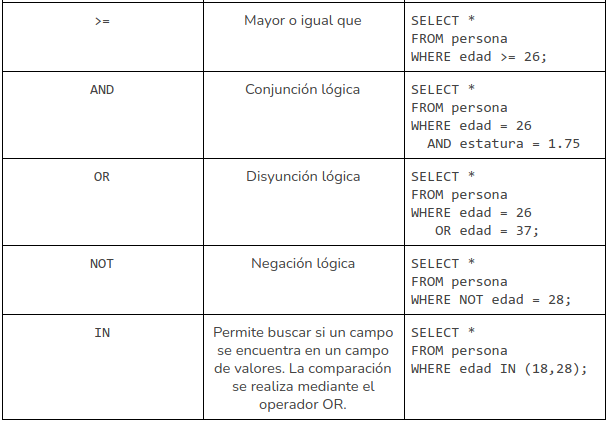
* SELECT \*  
  FROM movies  
  WHERE titulo = ‘Toy Story’;



Con WHERE podemos utilizar operadores relacionales, lógicos y algunos otros de utilidad con el fin de hacer combinaciones de filtros. La siguiente tabla muestra algunos de estos operadores. Es importante que los identifiques y comprendas ya que son los caracteres que te van a permitir realizar operaciones matemáticas, concatenar cadenas y hacer comparaciones.

Tabla

Descripción generada automáticamente



**Ordenamientos y Límites**

Otras restricciones que pueden aplicarse a las consultas son ORDER BY y LIMIT. La primera le indica a la consulta la forma en que debe mostrar los datos, ordenando los mismos por un campo de forma ascendente (ASC) o descendente (DESC). Por defecto, los resultados siempre son ordenados de forma ascendente. La segunda restricción, indica el número de registros máximo que traerá la consulta.

Por ejemplo, la siguiente consulta, ordena los resultados por el campo edad de forma descendente y limita los registros a 3.

* SELECT \*  
  FROM persona  
  WHERE estatura = 1.75  
  ORDER BY edad DESC  
  LIMIT 3;

Por otro lado, las siguientes consultas, ordenan de forma ascendente. Son equivalentes.

* SELECT \*  
  FROM persona  
  WHERE estatura = 1.75  
  ORDER BY edad  
  LIMIT 3;
* SELECT \*  
  FROM persona  
  WHERE estatura = 1.75  
  ORDER BY edad ASC  
  LIMIT 3;

Actividad:

Usando la base de datos tienda, muestra la descripción de las tablas **artículo**, puesto y venta. Por cada tipo de dato que encuentres llena la siguiente tabla (a mano, puedes dibujarla en un cuaderno o donde tu prefieras). Usa la como referencia si no recuerdas como se usa un comando por supuesto puedes preguntarle al experto.

|  |  |
| --- | --- |
| Tipo | Descripción |
| Int | Tipo de valor numérico entero. |
| Double | Tipo de valor numérico con decimales aproximado. |
| decimal | Tipo de valor numérico con decimales exacto. |
| Varchar | Cadena de caracteres. |
| timestamp | Fecha y hora en varchar. |

Links:

<https://github.com/beduExpert/Introduccion-a-Bases-de-Datos-Diciembre-2020/blob/master/Sesion-01/Reto-01/Readme.md>

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

<https://github.com/beduExpert/Introduccion-a-Bases-de-Datos-Diciembre-2020/blob/master/Sesion-01/Reto-01/Readme.md>

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

<https://github.com/beduExpert/Introduccion-a-Bases-de-Datos-Diciembre-2020/blob/master/Sesion-01/Reto-02/Readme.md>

<https://github.com/beduExpert/Introduccion-a-Bases-de-Datos-Diciembre-2020/blob/master/Sesion-01/Reto-03/Readme.md>

<https://github.com/beduExpert/Introduccion-a-Bases-de-Datos-Diciembre-2020/blob/master/Sesion-02/Reto-01/Readme.md>

<https://github.com/beduExpert/Introduccion-a-Bases-de-Datos-Diciembre-2020/blob/master/Sesion-02/Reto-02/Readme.md>

<https://picodotdev.github.io/blog-bitix/2019/06/orden-de-ejecucion-de-las-clausulas-de-las-sentencias-select-de-sql/>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_