

# Introducción a SQL

Sitio: Agencia de Habilidades para el Futuro

Curso: Administración de Base de Datos 1° D

Libro: Introducción a SQL

Imprimido por: RODRIGO PINTO

Día: miércoles, 27 de noviembre de 2024, 08:12

# Tabla de contenidos

## 1. Conceptos

## 2. Sintaxis

### 2.1. InnoDB

### 2.2. DML (lenguaje de manipulación de datos)

## 3. Álgebra relacional migrada a SQL

## Conceptos de SQL

Dada su fuerte conexión con la teoría del **modelo relacional**, SQL es un lenguaje de alto nivel orientado a conjuntos de registros. Esto implica que un solo **comando SQL** puede equivaler a decenas o cientos de líneas de código que se tendrían que utilizar en un lenguaje de más bajo nivel orientado a registros.

SQL permite lograr con mayor rapidez y facilidad la definición y manipulación de los objetos de base de datos, haciendo posible alcanzar una mayor eficiencia y productividad en el desarrollo.

Para el almacenamiento y operaciones de manipulación de los datos, el modelo relacional establece el uso de relación y variable de relación; mientras tanto SQL hace uso de las tablas como elemento fundamental.

Posee comandos referentes a:

- Un **lenguaje de definición de datos** o DDL que permite:
  - La creación de la estructura o esquema de base de datos.
  - La modificación de dicha estructura.
- Un **lenguaje de manipulación de datos** o DML que hace posible:
  - La inserción de datos en tablas.
  - Las consultas sobre los datos de estas tablas.
  - La actualización de los datos contenidos en estas tablas.
  - La eliminación de los registros de estas tablas.
  - Peticiones de información más complejas que incluyen JOINS y subconsultas.

- **Integridad:** el DDL incluye también comandos para especificar las restricciones de integridad que el DBMS debe hacer cumplir.
  - Definición de **vistas** o tablas derivadas.
- **Funciones de agrupamiento:** que permiten hacer cálculos de resúmenes.
- **Control de transacciones:** como unidad de trabajo lógica, unidad de recuperación y mecanismo de concurrencia.
- **Autorización:** incluye comandos para otorgar los privilegios de acceso a las tablas, **vistas** y otros elementos de base de datos.
- **SQL incorporado y dinámico:** esto quiere decir que se puede incorporar comandos SQL en lenguajes de programación como C++, PHP, Java, etc.

Un **lenguaje de control de datos** o DCL que hace posible:

- Controlar el acceso a los datos contenidos en la base de datos.

**Observemos la siguiente tabla comparativa para comprender mejor las diferencias y usos.**

Observemos la siguiente tabla comparativa para comprender mejor las diferencias y usos.	
DDL	<ul style="list-style-type: none"><li>• Definir, modificar y eliminar esquemas de relaciones.</li><li>• Crear índices, definir <b>vistas</b> y especificar restricciones de integridad.</li><li>• Ejemplos: CREATE, ALTER, DROP.</li></ul>
DML	<ul style="list-style-type: none"><li>• Consulta, actualizar y eliminar los elemento creados con el DDL como tablas, índices.</li><li>• Basado en el algebra relacional</li><li>• Ejemplos: SELECT, INSERT,UPDATE, DELETE.</li></ul>

## DCL

- Definir permisos de acceso a la base de datos
- Ejemplo: GRANT, REVOKE.

## Sintaxis en SQL - DDL (lenguaje de definición de datos)

DDL => lenguaje de definición de datos

Administrar un modelo físico de datos SQL presenta tres cláusulas básicas: **CREATE**, **DROP** y **ALTER**. Los mismos se corresponden con crear, borrar o modificar el esquema existente.

- Para **generar una BD** la sentencia SQL es:

```
create database nombre_BD;
```

- Para **eliminar una BD** completa la sentencia es:

```
drop database nombre_BD;
```

- Para **generar una tabla en una BD la instrucción es "Create Table"**. Se debe especificar el nombre de la tabla, cada uno de los atributos con el tipo de dominio, la clave primaria y, en caso de tener, las claves foráneas o extranjeras.

```
create table Socio(  
  CodSocio int,  
  DNI varchar (10),  
  Nombre varchar (60),  
  Apellido varchar (60),  
  Direccion varchar (30),  
  Tel varchar (15),  
  constraint pk_persona primary key (CodSocio)  
)engine=innodb;
```

En este caso algunos de los atributos son:

- **CodSocio** con dominio entero, no puede ser nulo porque está declarado como PK.
- **Nombre** con dominio **varchar (caracteres)** y con longitud máxima 60.

## Sintaxis en SQL - InnoDB



### ¿Qué es InnoDB?

Es un mecanismo de almacenamiento de datos de código abierto para la base de **datos MySQL**, incluido como formato de tabla estándar en todas las distribuciones de **MySQL** a partir de las versiones 4.0. Su característica principal es que soporta **transacciones** de tipo **ACID** y bloqueo de registros e integridad referencial. **InnoDB** ofrece una fiabilidad y consistencia muy superior a **MyISAM**, la anterior tecnología de tablas de **MySQL**, si bien el mejor rendimiento de uno u otro formato dependerán de la aplicación específica.

- Para eliminar una tabla de la BD la sentencia es:

```
drop table nombre_tabla;
```

- Para modificar una tabla del modelo la sentencia es:

```
alter table nombre_tabla .....
```



Esta sentencia que observamos en la imagen debe indicar, además que tipo de modificación se desea realizar sobre la tabla; se puede agregar, modificar o borrar atributos, índices o restricciones de integridad.

- Si se desea agregar una FK:

```
ALTER TABLE nombre_tabla ADD FOREIGN(atributo)REFERENCES Tabla(Atributo);
```

- Si desea crear un atributo:

```
ALTER TABLE nombre_tabla ADD column [Atributo] [Tipo de datos];
```

## Sintaxis en SQL - DML (lenguaje de manipulación de datos)

La estructura básica de SQL tiene el siguiente formato:

```
SELECT    lista_de_Atributos
FROM      lista_de_Tablas
WHERE     predicado ;
```

**Veamos que significan cada uno de estos ítems:**

**lista\_de\_Atributos** indica los nombres de los atributos que serán presentados en el resultado; estos atributos deben estar contenidos en las tablas indicadas en la consulta.

**lista\_de\_Tablas** indica las tablas de la BD necesarios para resolver la consulta; sobre las tablas indicadas de la lista se realiza el producto cartesiano.

**Predicado** indica que condición deben cumplir las tuplas de las tablas para estar en el resultado final de la consulta.



## Ejemplo de álgebra relacional migrado a SQL

En la siguiente tabla podemos observar como es trabajar con alguna de las operaciones que vimos en **álgebra relacional**, pero utilizando los conceptos vistos en **SQL**.

Álgebra relacional	SQL
$\Pi_{Atri1, Atri2, Atri3} (\sigma_{Atri4=valor}(tabla1 \times tabla2))$	<b>SELECT</b> Atr1,Atri2,Atri 3  <b>FROM</b> tabla1, tabla2  <b>WHERE</b> Atri4=Valor  <b>AND</b> tabla1.atributo=tabla2.atributo  (atributo es el atributo en común)
<b>SELECT</b> equivale a la operación de <b>proyección</b> .  <b>FROM</b> equivale al <b>producto cartesiano</b> .  <b>WHERE</b> equivale a la <b>selección</b> .	

