

### Data Definition Language (DDL)

**CREATE DATABASE** *db\_name* (field\_1 type\_1 constraint\_1 , ... , field\_n type\_n constraint\_n , any\_extra\_constraints );

**SHOW DATABASES;** Lista databases

**USE** *db\_name;* Cambia de db

**DROP DATABASE** *db\_name;* Elimina db

**SHOW TABLES;** Lista las tablas de la base de datos actual

**SHOW FIELDS FROM** *table;* Lista los campos de la tabla *table*

**DESC** *table;* muestra la estructura de la tabla

**SHOW CREATE TABLE** *table;* muestra el CREATE TABLE de la tabla

**DROP TABLE** *table;* Elimina la tabla

**ALTER TABLE** *table* **ADD** <constraint> Añade una constraint a la tabla *table*

**ALTER TABLE** *table* **ADD COLUMN** <column\_name column\_definition > Añade una columna a la tabla *table*

**ALTER TABLE** *table* **DROP COLUMN** <column\_name> Elimina una columna a la tabla *table*

**INSERT INTO** *table* (col1,...,coln) **VALUES** (val1,...,valn); Inserta un registro en la tabla con los valores especificados para cada campo segun el orden en **VALUES**

### Data Definition Language (DDL) (cont)

**UPDATE** *table* **SET** col1 = val1, ..., **WHERE** *condition;* Actualiza los valores de las columnas a los especificados en los registros que cumplan con las condiciones del WHERE

**DELETE FROM** *table* **WHERE** *condition;* Elimina los registros que cumplen las condiciones del WHERE

permite especificar, administrar y actualizar la base de datos.

### Data Types

**char(n)** String de tamaño fijo n

**varchar(n)** String de tamaño variable, con largo maximo n

**int** nros. entero

**double precision** Nro. de punto flotante de doble precisión.

**json** Objetos JSON

**date** fechas sin componente de tiempo.

**datetime** fechas con componente de tiempo.

**enum('type\_1,...,'type\_n')** lista enumerada de valores permitidos

### OPERACIONES DE CONJUNTO

(query) **UNION [ALL]** (query) Realiza la unión entre las respuestas de ambas queries

(query) **INTERSECT [ALL]** (query) Realiza la intersección entre las respuestas de ambas queries

### OPERACIONES DE CONJUNTO (cont)

(query) **EXCEPT [ALL]** (query) Realiza la resta entre la respuesta de la primera query menos la respuesta de la segunda query

### Expresiones / Funciones utiles

**COUNT(IF(expression, value\_if\_true, value\_if\_false))** Podemos utilizarlo para contar las filas que cumplan la expresion con value\_true = 1 y value\_false = 0

**COALESCE(valor\_nuleable, nuevo\_val)** se usa para reemplazar un valor "-null" retornado en una consulta.

### Constraints

**PRIMARY KEY**(col1, ...,coln) Define a las columnas col\_i como claves primarias (PK) de la tabla. Son **UNIQUE** y **NOT NULL**

**NOT NULL** La columna no puede tener valores nulos

**UNIQUE** La columna no puede tener valores repetidos

**AUTO\_INCREMENT** Genera identificadores unicos para las nuevas filas



### Constraints (cont)

**FOREIGN KEY** Indica que los valores de las columnas col\_1,...,col\_n deben corresponderse con los valores de *table(field)* [ON DELETE option] [ON UPDATE DELETE | ON UPDATE option]: establecen como actuar.

### Data Manipulation Language (DML)

**SELECT** Elige las columnas a devolver en la respuesta (col\_1, ..., col\_n) [AS \*alias]

**SELECT DISTINCT** Devuelve las columnas seleccionadas sin duplicados (col\_1, ..., col\_n)

**FROM** Elige la tabla o tablas en las que buscar los registros, si eligo mas de una tabla se hace producto cartesiano entre ellas

**WHERE** Filtra dejando solo los registros que cumplen con la condicion especificada

**ORDER BY** Ordena según la/s columna/s especificada/s en orden **ASC** o **DESC** y mostrando tantos resultados como indique **LIMIT** [ASC | DESC] [LIMIT int ]

### Funciones de agregación

**MAX()** busca el maximo

**MIN()** busca el minimo

**AVG()** calcula el promedio

**COUNT() | COUNT( 'DISTINCT\*\*')** cuenta las filas (con o sin repeticion)

**SUM()** suma los valores de las filas

Las agregaciones se realizan en los grupos especificados por el **GROUP BY**, A igual valor, mismo grupo. Si se omite el **GROUP BY**, todas las filas se tratan como parte de un solo grupo.

Las columnas en el **SELECT** que no sean agregadas deben aparecer en el **GROUP BY**.

### Consultas Anidadas

#### SET MEMBERSHIP (IN | NOT IN)

**SELECT ... FROM ... WHERE** (columns) **IN** [SUBQUERY] ó [ENUMERATION]

Sirve para filtrar las columnas manteniendo las que coincidan con alguno de los valores resultantes de la subquery o enumeración

**SELECT ... FROM ... WHERE** (columns) **NOT IN** [SUBQUERY] ó [ENUMERATION]

Sirve para excluir las columnas cuyos valores sean la respuesta de la subquery o estén mencionados en la enumeración

**SET COMPARISON (SOME | ALL)** comp = {>, <, >=, <=, =, <>}

### Consultas Anidadas (cont)

**SELECT ... FROM ... WHERE** (columns) comp **SOME** [SUBQUERY]

significa que la condición será verdadera si la operación es verdadera para cualquiera (o alguno) de los valores del rango.

**SELECT ... FROM ... WHERE** (columns) comp **ALL** [SUBQUERY]

significa que la condición será verdadera solo si la operación es verdadera para todos los valores del rango.

#### EMPTY RELATIONS (EXISTS | NOT EXISTS)

**SELECT ... FROM ... WHERE EXISTS** [SUBQUERY]

El operador **EXISTS** devuelve **VERDADERO** si la subconsulta devuelve uno o más registros, Se utiliza para probar la existencia de algún registro en una subconsulta.

**SELECT ... FROM ... WHERE NOT EXISTS** [SUBQUERY]

El operador **NOT EXISTS** devuelve **VERDADERO** si la subconsulta no devuelve ningun registro, Se utiliza para probar la **NO** existencia de algún registro en una subconsulta.

Una subquery es una consulta anidada en un **WHERE** un **FROM** o una **Columna** -> query escalar

### JOINS

**INNER JOIN** FROM A INNER JOIN B ON join\_condition  
 Selecciona todas las filas de las tablas A y B donde la condición del join se satisface.

**LEFT JOIN** FROM A LEFT JOIN B ON join\_condition  
 Selecciona todas las filas de las tablas A y aquellas de la tabla B donde la condición del join se satisface.

**RIGHT JOIN** FROM A RIGHT JOIN B ON join\_condition  
 Selecciona todas las filas de las tablas B y aquellas de la tabla A donde la condición del join se satisface.

**FULL JOIN** FROM A FULL JOIN B ON join\_condition  
 Selecciona todas las filas de las tablas A y B independientemente de si la condición del join se satisface.

### HAVING

**[HAVING where\_condition]** permite filtrar grupos sobre los valores agregados, se chequea luego de que los grupos son computados.

### Seguridad

En SQL standard se incluyen los privilegios **SELECT, INSERT, UPDATE, y DELETE**

**GRANT** <privilege list> on <relation name or view name> to <user/role list>;  
 El comando GRANT es usado para conceder autorización

**REVOKE** <privilege list> on <relation name or view name> from <user/role list>;  
 La sentencia REVOKE se usa para Para anular una autorización

**CREATE ROLE** <rol\_name>;  
 se utiliza para crear un Grupo de privilegios relacionados que se otorgan a usuarios.

**GRANT** <privilege list> on *table* to <rol\_name>;  
 asignación de privilegios a roles

**GRANT UPDATE** <column\_name> on *table* to <rol\_name>;  
 asignación de privilegio de actualización sobre una columna de la tabla a un role

**GRANT** <rol\_name> to <user>;  
 asignación de roles a usuarios



By **MiliC\_**  
[cheatography.com/milic/](https://cheatography.com/milic/)

Not published yet.  
 Last updated 21st January, 2024.  
 Page 3 of 3.

Sponsored by **CrosswordCheats.com**  
 Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>