Fundamentos de SQL — Structured Query Language

Bases de datos

De ahora en adelante base de datos = bd.
Son un conjunto de datos almacenados en forma ordenada, que nos permiten hacer búsquedas y encontrar información fácilmente.
Los datos se organizan en forma de tablas, con sus respectivas columnas y filas.
Cada dato se llama registro y tiene una identificación única e irrepetible llamada primary key, que es el ID.
El lenguaje SQL se utiliza en sistemas de gestión de bases de datos como MySQL, SQL Server, MS Access, Oracle, Sybase, Informix y Postgres, entre otros.

Tipos de datos

Para una lista completa ir a https://www.w3schools.com/sql/sql_datatypes.asp
INTEGER: número entero largo.
REAL: número entero simple.
FLOAT: número entero doble.
CHAR: texto con una cantidad definida de caracteres que debe ser exacta. No puede ser ni menos ni más que la cantidad de caracteres que se define. Por ejemplo el dato con CHAR(10) debe tener exactamente 10 caracteres. El máximo es 255 caracteres.
VARCHAR: texto que no puede superar una cierta cantidad de caracteres definida por nosotros. Se define escribiendo la cantidad de caracteres entre paréntesis. Por ejemplo VARCHAR(40) debe tener hasta 40 caracteres inclusive.
BINARY: binario, define si algo es true o false.
DATE: Fecha. Fomato YYYY-MM-DD, donde YYYY es año, MM es mes y DD es día. Por ejemplo: 2018-06-31.
DATETIME: Fecha y hora. Formato YYYY-MM-DD HH:MI:SS, donde H es hora, MI son los minutos y SS son los segundos. Por ejemplo: 2018-06-31 12:25:33.

Operadores

Para una lista completa ir a https://www.w3schools.com/sql/sql_operators.asp
Se pueden usar diferentes operadores para buscar o comprarar datos.
Los operadores lógicos comprueban la veracidad de alguna condición. Al igual que los operadores de comparación, devuelven el tipo de datos Boolean con el valor TRUE, FALSE o UNKNOWN.
Comparadores: =, <, >, <=, >=, <> (distinto).
Lógicos:
AND (si ambas expresiones booleanas son true)
OR (si cualquiera de las dos expresiones booleanas es true)
NOT (invierte el valor de cualquier otro operador booleano)
LIKE (si lo que busco coincide con un patrón determinado)
BETWEEN (si lo que busco está dentro de un intervalo de datos determinado)
IN (si lo que busco es igual a un dato dentro de una lista de expresiones)
ALL (si el conjunto completo de comparaciones es true)
ANY (si cualquier miembro del conjunto de comparaciones es true)

Restricciones o constraints

Para una lista completa ir a https://www.w3schools.com/sql/sql_constraints.asp
Las restricciones se utilizan para especificar reglas para los datos de una tabla.
PRIMARY KEY: si el dato tiene una clave primaria es único e irrepetible. No puede haber dos datos con primary key en una misma tabla. Generlamente se usan para los id.
UNIQUE: identifica todos los datos únicos e irrepetibles dentro de una tabla. Se pueden tener varios datos con la restricción UNIQUE dentro de una misma tabla.

NOT NULL: ningún dato de la columna con esta restricción puede estar vacío.
FOREIGN KEY: se usa para linkear o relacionar dos tablas distintas. Los datos con foreign keys se relacionan con los datos de otra tabla que tienen clave primaria. La tabla con primary key es la tabla padre y la que tiene la foreign key es la tabla hijo.
Ejemplo: ir a https://www.w3schools.com/sql/sql_foreignkey.asp

Comandos básicos

Cada vez que quiero que un comando se ejecute, escribo primero el comando y luego escribo GO.

CREATE : crea la base de datos o las tablas de una base de datos ya creada. Cuando se crea la tabla se especifica el tipo de datos que se ingresarán y el modificador. La creación de la tabla se hace junto a sus columnas.
CREATE DATABASE nombredelabd
CREATE TABLE nombretabla(
NombreFila CHAR o VARCHAR(cantidad de caracteres) INTEGER(tipo de dato. Si es un string no se pone nada) PRIMARY KEY (el modificador, si corresponde. Se asocia al dato que se convertirá en el id) NOT NULL (quiere decir que el dato no puede quedar vacío; si es un dato opcional como comentarios o observaciones, no se pone nada)
)
Ejemplo:
CREATE TABLE MisProductos(
NombreProducto VARCHAR(30) NOT NULL,
CodigoProducto INTEGER PRIMARY KEY NOT NULL,
PrecioUnitario FLOAT NOT NULL,
Categoria VARCHAR(30) NOT NULL,
Vencimiento DATETIME NOT NULL,
Observaciones VARCHAR(100),
)
CUIDADO: poner las comas después de cada categoría porque sino dará error el código y habrá que crear la tabla nuevamente.

USE: nos conecta a la base de datos para empezar a trabajar con ella.
USE nombredelabd

INSERT INTO: inserta un nuevo registro (o fila) en una tabla de la bd.
INSERT INTO nombretabla
VALUES (‘dato1’, ‘dato2’);
Ejemplo:
INSERT INTO ContactosDeTrabajo
VALUES ('Matias Gallardo', 'Alberdi 676', '4568890', 'Edesur');
VALUES ('Juana Gomez', 'Rivadavia 45', '4556699', 'Fibertel');
También se pueden insertar datos sólo en columnas específicas de la tabla.
Ejemplo:
INSERT INTO ContactosDeTrabajo (NombreApellido, Direccion)
VALUES ('Juana Gomez', 'Rivadavia 45');

UPDATE: actualiza o modifica datos de una tabla de la bd.
UPDATE nombretabla SET nombrecolumna = datonuevo
WHERE nombrecolumna = nombredatoexistente
Ejemplo:
UPDATE MisProductos SET NombreProducto = 'Fideos tirabuzon' WHERE Categoria = 'Pastas';
CUIDADO: si no ponemos el WHERE luego del UPDATE se modificarán todos los registros de la tabla.

DELETE: borra un registro (fila) entero de la tabla de una bd.
DELETE FROM nombretabla WHERE nombrecolumna = nombredatoborrar
Ejemplo:
DELETE FROM MisProductos WHERE NombreProducto = 'Fideos tirabuzon';
Esto borra el registro (fila) entero. La primary key asignada a ese registro no se puede volver a usar. No se pueden eliminar ni modificar columnas una vez creada la base de datos.
CUIDADO: si no ponemos el WHERE luego del DELETE se borrará toda la tabla.

SELECT: selecciona o busca datos. Los datos seleccionados son guardados en una tabla llamada result-set.
SELECT * FROM MisProductos – muestra toda la tabla
SELECT * FROM MisProductos WHERE NombreProducto = 'Fideos tirabuzon'; - busca un dato específico
SELECT * FROM MisProductos ORDER BY NombreProducto; - muestra los datos de manera ordenada en forma ascendente. Si quiere mostrarlos de manera descendente, agrego la línea DESC (ORDER BY NombreProducto DESC;).
SELECT DISTINCT Categoria FROM MisProductos; - muestra sólo los datos que no están repetidos. Podemos tener varios registros con una misma categoría. En este caso sólo mostrará un registro por cada categoría.
SELECT COUNT (DISTINCT Categoria) FROM MisProductos; - muestra el número de categorías que hay (sólo la cantidad, no lista los nombres de cada categoría).

DROP : borra toda la base de datos o toda una tabla. Cuidado: no pregunta antes de borrar. Lo borrado no se puede recuperar.
DROP DATABASE Almacen;
DROP TABLE MisProductos;

Tablas: acciones

Agregar una columna
Siempre se agregan al extremo final de la tabla.
ALTER TABLE MisProductos
ADD nombrecolumna tipodedato modificador
Ejemplo:
ADD Marca VARCHAR(30) NOT NULL

Eliminar una columna
ALTER TABLE MisProductos
DROP COLUMN nombrecolumna

Modificar una columna
ALTER TABLE MisProductos
ALTER COLUMN nombrecolumna tipodedato modificador

Resumen de todos estos comandos en
https://www.w3schools.com/sql/sql_quickref.asp