SQL

Maneja DCL, DDL y DML

DDL -> define tablas y relaciones y bases de datos

CREATE -> crea una base de datos, tabla, vistas

CREATE TABLE cliente(

clie\_codigo char(6) NOT NULL primary key,

clie\_razon\_social char(100) NULL

ALTER -> modifica la estructura, es decir agregar o sacar columnas de la tabla

Alter table cliente add clie\_provincia Int null;

Alter table cliente drop column clie\_provincia;

Alter table cliente alter column clie\_provincia char(10) null;

DROP -> elimina una base de datos o tabla

Es un lenguaje tipado

A white text with black text

AI-generated content may be incorrect.

Tipos de datos para **fecha y hora**

* Datetime -> almacena fecha al milisegundo
* smallDateTime -> almacena fecha al minuto, usa la mitad de memoria que Datetime
* timestamp -> marca registro con fecha de inserción o actualización. Este se actualiza con actualización o inserción de datos. Genera registros de auditoria mediante la implementación de triggers.

Tipos de datos para **char**

Char (n) -> string de longitud fija

Varchar(n) -> tiene un limite de n pero solo ocupa lo que le escribo yo

Text(n) -> representa un varchar(max)

Nchar y nvarchar -> son iguales a char y varchar pero están en formato Unicode

Tipo de dato BLOB (binary large object)

Binary(n) -> almacena datos binarios de longitud fija

Varbinary(n) -> almacena datos binarios de longitud variable

Primero creo todas las tablas, luego les agrego las primary key y por ultimo las foreign key

Constraint -> restricciones sobre la primary o foreign key

Check -> me permite poner restricciones a mano sobre valores de tabla

Unique -> es igual a una primary key pero permite un solo null. Es decir no permite repetidos pero si NULL

Default -> si esta en NULL, ingreso el valor predeterminado

Autoincremental -> se autosuma y no lo controlo. Le pongo una semilla. A la primera fila le pongo la semilla y de ahí en mas en las demás filas se incrementa solo

Lenguaje DML

Es un sublenguaje que maneja sql

SELECT, esta sentencia se utiliza para realizar consultas sobre los datos

INSERT, con esta instrucción podemos insertar los valores en una tabla

UPDATE, sirve para modificar los valores de una o varias filas de una tabla

DELETE, se utiliza para eliminar filas de una tabla

Insert -> es un comando unitario, cada vez que quiero agregar una fila hago insert

Con especificación de columna: en la columna n inserto el valor n. Puedo saltarme valores si lo necesito

INSERT INTO tabla (col1, col2, …, coln)

Values (valor1, valor2, …, valorn)

Sin especificacion de columnas: aca tengo que poner todos los valores no puedo saltarme ninguno

INSERT table values(val1, …, val n)

Update -> modifica un conjunto de filas, entonces no es unitario.

UPDATE tabla SET col1 = val1, col2 = val2

Where condicion

UPDATE table SET col1 = val1, col2 = val2

Delete -> filas de tabla

DELETE FROM tabla

WHERE condición

DELETE tabla

WHERE condición

Select -> permite consultar filas de una o mas tablas. Afecta a conjuntos. Devuelve un recordset o dataset, es decir, un pedazo de la tabla.

SELECT col1, col2, …, col n // esta también es obligatoria

FROM tabla // esta es obligatoria, lo demás es opcional

WHERE condición

GROUP BY col1, col2, …, col n

HAVING condicion

ORDER BY col1, col2, …, col n

Con SELECT digo que columnas quiero ver, con FROM digo de que tabla. Si pongo

SELECT \* elige todas las columnas.

Puedo seleccionar de mas de una tabla con FROM y hace el producto cartesiano de ambas, es decir pares ordenados de columnas de ambas tablas.

Condición WHERE

WHERE cond1(=, >< ) AND cond2 OR cond3 NOT cond4 LIKE ‘string\_%’ BETWEEN valor1 AND valor2

LIKE -> \_ y % son calificadores. % es cualquier cosa, es decir, si pongo A% me da lo quie empieza por A. El \_ es una posición que puede tener cualquier cosa

GROUP BY -> unifica entradas de tabla o fila en una sola. Agrupa por determinado campo o columna

SELECT col1, col2, ….coln

FROM tabla

GROUP BY col1, col2

Puedo agregar datos a las agrupaciones con

* + COUNT() -> cuantas filas agrupe en una sola (es decir cuantos atributos repetían registro por el cual agrupo)
  + SUM() -> sumo los campos que agruop
  + MIN() -> la mas pequeña del campo
  + MAX() -> la mas grande del campo
  + AVG() -> promedio del campo por el que agrupo

Having -> si lo uso si o si debo usar un GROUP BY. Es la condición que deben cumplir las filas para ser devueltas en la consulta luego que fueron agrupadas por la cláusula GROUP BY

ORDER BY -> ordena las filas por la condición que se le meto adentro