DDL: Data definition language

CREATE

CREATE DATABASE BaseDeDatos

CREATE TABLE tabla (COLUMN\_NAME DATATYPES[,....]);

ALTER

ALTER TABLE tabla ADD (col1 tipoDato, colN tipoDato); > add a new column

ALTER TABLE MODIFY(COLUMN DEFINITION....); > modify existing column

DROP

DROP TABLE tabla ;

TRUNCATE: to delete all the rows from the table and free the space containing the table.

TRUNCATE TABLE tabla;

RENAME:

DML: Data Manipulation Language

INSERT:

"INSERT INTO *tabla* (*col1, col2*) VALUES (@*col1*, @*col2*);” no pido ID

"INSERT INTO TABLA VALUES (@col1, @col2);” //se indican los valores en orden

UPDATE

"UPDATE *tabla* SET *col1*=@*col1*, *col2*=@*col2* WHERE id=@id;" pido nuevos valores y ID

DELETE

"DELETE FROM *table”* > borra la tabla

"DELETE FROM *tabla* WHERE id=@id;" pido solo ID

DCL: Data Control Language

Grant: used to give user access privileges to a database.

GRANT SELECT, UPDATE ON MY\_TABLE TO SOME\_USER, ANOTHER\_USER;

Revoke: to take back permissions from the user

REVOKE SELECT, UPDATE ON MY\_TABLE FROM USER1, USER2;

TCL: Transaction Control Language

COMMIT: to save all the transactions to the database

ROLLBACK: to undo transactions that have not already been saved to the database.

SAVEPOINT: to roll the transaction back to a certain point without rolling back the entire transaction

DQL: Data Query Language

SELECT

"SELECT *\**  FROM *tabla* [WHERE id=@id];"

"SELECT *col1, col2* FROM *tabla* [WHERE id=@id];"

Para obtener el ID generado en un Insert:

"INSERT INTO *TABLA* (*col1, col2*) VALUES (@*col1*, @*col2*); SELECT CAST(scope\_identity() AS int);”

int id = (int)command.ExecuteScalar; > en lugar de ExecuteNonQuery

Condiciones:

WHERE col1 LIKE ‘%valor%’ //like > contiene %> comodín

Clave secundaria:

SELECT tablaA.col1, tablaB.col1 FROM tablaA,tablaB WHERE tablaB.col1 LIKE ‘a%’ AND tablaA.ClaveSecundaria=tablaB.ID;

Operaciones matemáticas;

SELECT *col1-1* FROM *tabla* Operadores: +,-,\*,/ (result c/decimales), DIV(result entero), MOD(resto)

SELECT MIN(*col1)* FROM *tabla* Valores agrupados: MIN, MAX, SUM, AVG, COUNT

"SELECT COUNT(\*) FROM dbo.region"; // + Int32 count = (Int32) cmd.ExecuteScalar();

Grupos/subtotales:

SELECT COUNT(\*),col1 FROM table GROUP BY col1; //return cantidad de repeticiones p/c/valor de la col1

SELECT COUNT(\*),col1 FROM table GROUP BY col1 HAVING col1<10; //filtra el resultado