COMANDOS SQL

I. COMANDOS DDL(DATA DEFINITION LANGUAGE)

CREATE DATABASE – criar um banco de dados

CREATE DATABASE *nome_banco_de_dados*

DROP DATABASE – excluir um banco de dados

DROP DATABASE *nome_banco_de_dados*

USE – habilitar o uso de um banco de dados

USE *nome_banco_de_dados*

CREATE TABLE – criar uma tabela (no exemplo abaixo os colchetes ([ ]) significam que é opcional)

CREATE TABLE *nome_tabela*
(
nome_coluna1 tipo_dado [**NOT NULL**] [**IDENTITY (1,1)**] **PRIMARY KEY**,
nome_coluna2 tipo_dado [**NOT NULL**],
nome_coluna3 tipo_dado [**NOT NULL**] [**FOREIGN KEY REFERENCES** *nome_tabela2* (*nome_coluna_PK_tabela2*)],
....
nome_colunaN tipo_dado [**NOT NULL**]
);

DROP TABLE - excluir uma tabela

DROP TABLE *nome_tabela*

II. COMANDOS DML(DATA MANIPULATION LANGUAGE)

INSERT INTO – inserir dados em uma tabela

INSERT INTO *nome_tabela (nome_coluna1, nome_coluna2, ..., nome_colunaN)*
VALUES (*valor1, valor2, valor3, ..., valorN*);

SELECT – escolher colunas (campos) da tabela para apresentar na query (consulta)

SELECT *nome_coluna1, nome_coluna2, ..., nome_colunaN*
FROM *nome_tabela*

SELECT *
FROM *nome_tabela*

SELECT DISTINCT *nome_coluna1, nome_coluna2, ..., nome_colunaN*
FROM *nome_tabela*

WHERE – condição para seleção das linhas da tabela (registros)

SELECT *nome_coluna1, nome_coluna2, ..., nome_colunaN*
FROM *nome_tabela*
WHERE *condição*

Operador	Descrição
=	Igual a
<>	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Operadores AND / OR

SELECT *nome_coluna1, nome_coluna2, ..., nome_colunaN*
FROM *nome_tabela*
WHERE *condição1* **AND** *condição2*

SELECT *nome_coluna1, nome_coluna2, ..., nome_colunaN*
FROM *nome_tabela*
WHERE *condição1* **OR** *condição2*

ORDER BY – ordenar a apresentação dos dados

SELECT *nome_coluna1, nome_coluna2, ..., nome_colunaN*
FROM *nome_tabela*
ORDER BY *nome_coluna* **ASC | DESC**

LIKE – condição para seleção de linhas (registros) da tabela usando textos

SELECT *nome_coluna1, nome_coluna2, ..., nome_colunaN*
FROM *nome_tabela*
WHERE *nome_coluna* **LIKE** *padrão*

"Coringa"	**Descrição**
%	Substitui nenhum ou mais caracteres
_	Substitui exatamente um caractere
[*lista_caracteres*]	Qualquer caractere individual na lista de caracteres (*lista_caracteres*)
[^*lista_caracteres*] ou [!*lista_caracteres*]	Qualquer caractere individual que não esteja na lista de caracteres (*lista_caracteres*)

IN (semelhante ao OR)

SELECT *nome_coluna1, nome_coluna2, ..., nome_colunaN*
FROM *nome_tabela*
WHERE *nome_coluna* **IN** *(valor1, valor2, ..., valor3)*

BETWEEN (semelhante ao AND)

SELECT *nome_coluna1, nome_coluna2, ..., nome_colunaN*
FROM *nome_tabela*
WHERE *nome_coluna* **BETWEEN** *valor1* ***AND*** *valor2*

UPDATE – atualizar valor armazenado na tabela

UPDATE *nome_tabela*
SET *nome_coluna1 = valor1, nome_coluna2 = valor2, ..., nome_colunaN = valorN*
WHERE *condição*

DELETE – excluir dados da tabela

DELETE *nome_tabela*
WHERE *condição*

Funções Agregadoras

SUM() – soma de valores

SELECT SUM (*nome_coluna*)
FROM *nome_tabela*

SELECT SUM (*nome_coluna*) **AS** *apelido_coluna*
FROM *nome_tabela*

COUNT() - contagem de linhas (registros)

SELECT COUNT (*nome_coluna*)
FROM *nome_tabela*

SELECT COUNT (*nome_coluna*) **AS** *apelido_coluna*
FROM *nome_tabela*

AVG() - média aritmética de valores

SELECT AVG (*nome_coluna*)
FROM *nome_tabela*

SELECT AVG (*nome_coluna*) **AS** *apelido_coluna*
FROM *nome_tabela*

MAX () - o maior valor de uma coluna da tabela

SELECT MAX (*nome_coluna*)
FROM *nome_tabela*

SELECT MAX (*nome_coluna*) **AS** *apelido_coluna*
FROM *nome_tabela*

MIN () - o menor valor de uma coluna da tabela

SELECT MIN (*nome_coluna*)
FROM *nome_tabela*

SELECT MIN (*nome_coluna*) **AS** *apelido_coluna*
FROM *nome_tabela*

GROUP BY – agrupar os dados, utilizando uma função agregadora

SELECT *nome_coluna1*, ***função_agregadora*** (*nome_coluna2*)
FROM *nome_tabela*
GROUP BY *nome_coluna1*

SELECT *nome_coluna1*, ***função_agregadora*** (*nome_coluna2*)
FROM *nome_tabela*
GROUP BY *nome_coluna1*
HAVING ***função_agregadora*** (*nome_coluna2*) *condição*

JOIN – ligar duas tabelas, permitindo a apresentação de dados de ambas as tabelas

SELECT *nome_colunaN_tabela1,...., nome_colunaN_tabela2,..., nome_colunaN_tabelaN*
FROM *nome_tabela1*
INNER JOIN *nome_tabela2* **ON** *nome_coluna_PK_tabela1 = nome_coluna_FK_tabela2*

SELECT *nome_colunaN_tabela1,...., nome_colunaN_tabela2,..., nome_colunaN_tabelaN*
FROM *nome_tabela1*
LEFT JOIN *nome_tabela2* **ON** *nome_coluna_PK_tabela1 = nome_coluna_FK_tabela2*

SELECT *nome_colunaN_tabela1,...., nome_colunaN_tabela2,..., nome_colunaN_tabelaN*
FROM *nome_tabela1*
RIGHT JOIN *nome_tabela2* **ON** *nome_coluna_PK_tabela1 = nome_coluna_FK_tabela2*

III. Visão

CREATE VIEW – cria uma tabela virtual (VIEW) baseada em uma tabela existente

CREATE VIEW *nome_view* **AS**
comando SELECT para criar a tabela VIEW

IV. Stored Procedure

CREATE PROCEDURE – cria uma stored procedure

CREATE PROCEDURE *nome_stored_procedure* **@***nome_variável1 tipo_variável1,…* **@***nome_variáveln tipo_variáveln*
AS
BEGIN
Comando SELECT que utiliza nome_variável para criar uma condição (WHERE ou HAVING)
END

EXECUTE – executa uma stored procedure

EXECUTE *nome_stored_procedure @valor_variável1,…, @valor_variáveln*

DROP PROCEDURE – exclui uma stored procedure

DROP PROCEDURE *nome_stored_procedure*