*SQL*

Database

Criar uma nova base de dados – CREATE DATABASE *nomedabd;*

Eliminar uma base de dados existente – DROP DATABASE *nomedabd;*

Tables

Criar uma nova tabela – CREATE TABLE *nome\_tabela (coluna1 datatype, coluna2 datatype, ...);*

Eliminar uma tabela existente – DROP TABLE *nome\_tabela;*

Eliminar os dados dentro de uma tabela – TRUNCATE TABLE *nome\_tabela;*

Inserir dados numa tabela – INSERT INTO *nome\_tabela (coluna1, coluna2, ...)* VALUES *(valor1, valor2, ...);*

Atualizar dados numa tabela – UPDATE *nome\_tabela* SET *coluna1 = novo\_valor WHERE coluna (>, < ou =) valor [AND, OR coluna (>, < ou =) valor];*

**ALTER**

* Adicionar uma coluna à tabela – ALTER TABLE *nome\_tabela* ADD *nome\_coluna datatype;*
* Eliminar uma coluna à tabela- ALTER TABLE *nome\_tabela* DROP COLUMN *nome\_coluna;*
* Alterar o datatype de uma coluna – ALTER TABLE *nome\_tabela* ALTER COLUMN *nome\_coluna datatype;*

Constraints

Constraints são usados como regras para o tipo de dados que pode ser inserido numa tabela.

Exemplo: CREATE TABLE *nome\_tabela (coluna1 datatype constraint*, *coluna2 datatype constraint,...);*

**NOT NULL –** Faz com que uma coluna não possa aceitar valores nulos.

CREATE TABLE Persons (

ID int NOT NULL,

...  
);

**UNIQUE –** Garante que uma tabela ou conjunto de tabelas têm valores diferentes.

CREATE TABLE Persons (

ID int NOT NULL UNIQUE,

LastName varchar(255) NOT NULL,

...  
);

**PRIMARY KEY –** Chave primária de uma tabela, que pode consistir em um ou vários campos, mas que não pode ter valores nulos.

*Usando o CREATE TABLE:*

CREATE TABLE Persons (

ID int NOT NULL PRIMARY KEY,

...  
);

*Usando o ALTER TABLE:*

ALTER TABLE Persons ADD PRIMARY KEY (ID);

**FOREIGN KEY –** Chave usada para ligar 2 tabelas. Consiste num campo de uma tabela que se refere á primary key de outra tabela.

*Usando o CREATE TABLE:*

CREATE TABLE Orders (

...

...

PersonID int FOREIGN KEY REFERENCES Persons(ID)  
);

*Usando o ALTER TABLE:*

ALTER TABLE Orders ADD FOREIGN KEY (PersonID) REFERENCES Persons(ID);

**CHECK –** Utilizado para limitar os valores que se podem colocar numa coluna.

*Usando o CREATE TABLE:*

CREATE TABLE Persons (

...

...

Age int CHECK (Age>=18)  
);

*Usando o ALTER TABLE:*

ALTER TABLE Persons ADD CHECK (Age>=18);

**DEFAULT –** Utilizado para providenciar um valor padrão para uma tabela, caso mais nenhum valor seja especificado.

*Usando o CREATE TABLE:*

CREATE TABLE Persons (

...

...

City varchar(255) DEFAULT ‘Sandness’  
);

*Usando o ALTER TABLE:*

ALTER TABLE Persons ALTER COLUMN City SET DEFAULT ‘Sandness’

**AUTO INCREMENT –** Automaticamente incrementa um valor na coluna sempre que há um novo registo. NOTA: IDENTITY(valor *inicial* **,** *valor a incrementar por cada novo registo).*

CREATE TABLE Persons (

ID int IDENTITY(1,1) PRIMARY KEY,

...  
);

SELECT

**ALL –** Devolve todos os dados de uma tabela.

SELECT \* FROM *nome\_tabela;*

**DISTINCT –** Devolve valores diferentes numa tabela.

SELECT DISTINCT *coluna1* FROM *nome\_tabela;*

**TOP** (top\_valor) **-** Devolve o número de linhas especificado.

SELECT TOP(*número)* FROM *nome\_tabela;*

**PERCENT –** Devolve o número de linhas especificado, baseado na percentagem.

SELECT TOP(*número)* PERCENTFROM *nome\_tabela;*

**WHERE -** Condição utilizada para filtrar dados.

SELECT *coluna1, coluna2, ...* FROM *nome\_tabela* WHERE *condição;*

**OPERADORES AND, OR E NOT**

**AND -** SELECT *coluna1, coluna2, ...* FROM *nome\_tabela* WHERE *condição1 AND condição2;*

**OR -** SELECT *coluna1, coluna2, ...* FROM *nome\_tabela* WHERE *condição1 OR condição2;*

**NOT -** SELECT *coluna1, coluna2, ...* FROM *nome\_tabela* WHERE NOT *condição;*

**GROUP BY –** Ordena os dados de acordo com uma coluna.

SELECT *coluna1, coluna2, ...* FROM *nome\_tabela* GROUP BY *coluna;*

**HAVING –** Devolve os dados de acordo com uma coluna, onde os mesmos têm de obedecer a uma condição.

SELECT *coluna1, coluna2, ...* FROM *nome\_tabela* GROUP BY *coluna* HAVING *condição;*

**ORDER BY –** Devolve os dados ordenados de forma ascendente ou descendente.

SELECT *coluna1, coluna2, ...* FROM *nome\_tabela* ORDER BY *coluna1, coluna2, ...* ASC**|**DSC

**FUNÇÕES COUNT, AVG E SUM**

**COUNT –** Devolve o número de linhas que são iguais ao critério utilizado.

SELECT COUNT(*coluna1)* FROM *nome\_tabela* WHERE *condição;*

**AVG –** Devolve a média de valores de uma coluna numérica.

SELECT AVG(*coluna1)* FROM *nome\_tabela* WHERE *condição;*

**SUM –** Devolve a soma de valores de uma coluna numérica.

SELECT SUM(*coluna1)* FROM *nome\_tabela* WHERE *condição;*