



IN

# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

# PY - SQL II

**01** Entendendo os tipos de comandos SQL

**02** Criando tabelas

**03** Alimentando o banco de dados

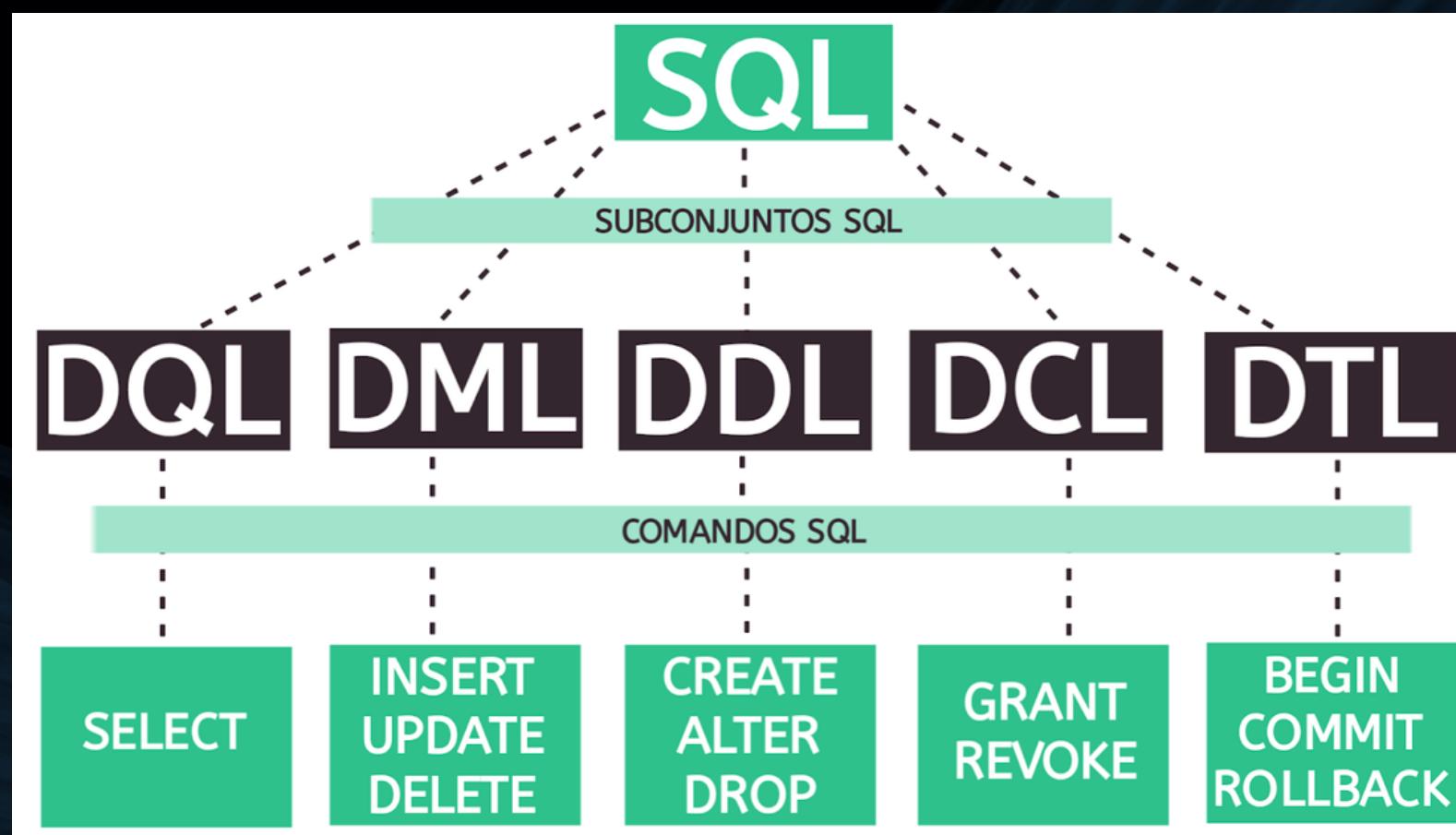
**04** Comando select

**05** Exercício



# PY - SQL II

## Tipos de comandos SQL



Como vimos na aula passada, o SQL é uma linguagem para se trabalhar com banco de dados relacionais.

Com o SQL podemos gerenciar os nossos dados em um banco de dados através das *querys*, ou seja, através das requisições.

Nessa aula vamos entender que temos diferentes grupos de comandos para fazer diferentes requisições ao SGBD.

Cada grupo de comando também será responsável por uma letra do CRUD

# PY - SQL II

01

## CREATE

Criar ou adicionar novas entradas

02

## READ (RETRIEVE)

Ler, recuperar ou ver entradas existentes

03

## UPDATE

Atualizar ou editar entradas existentes

04

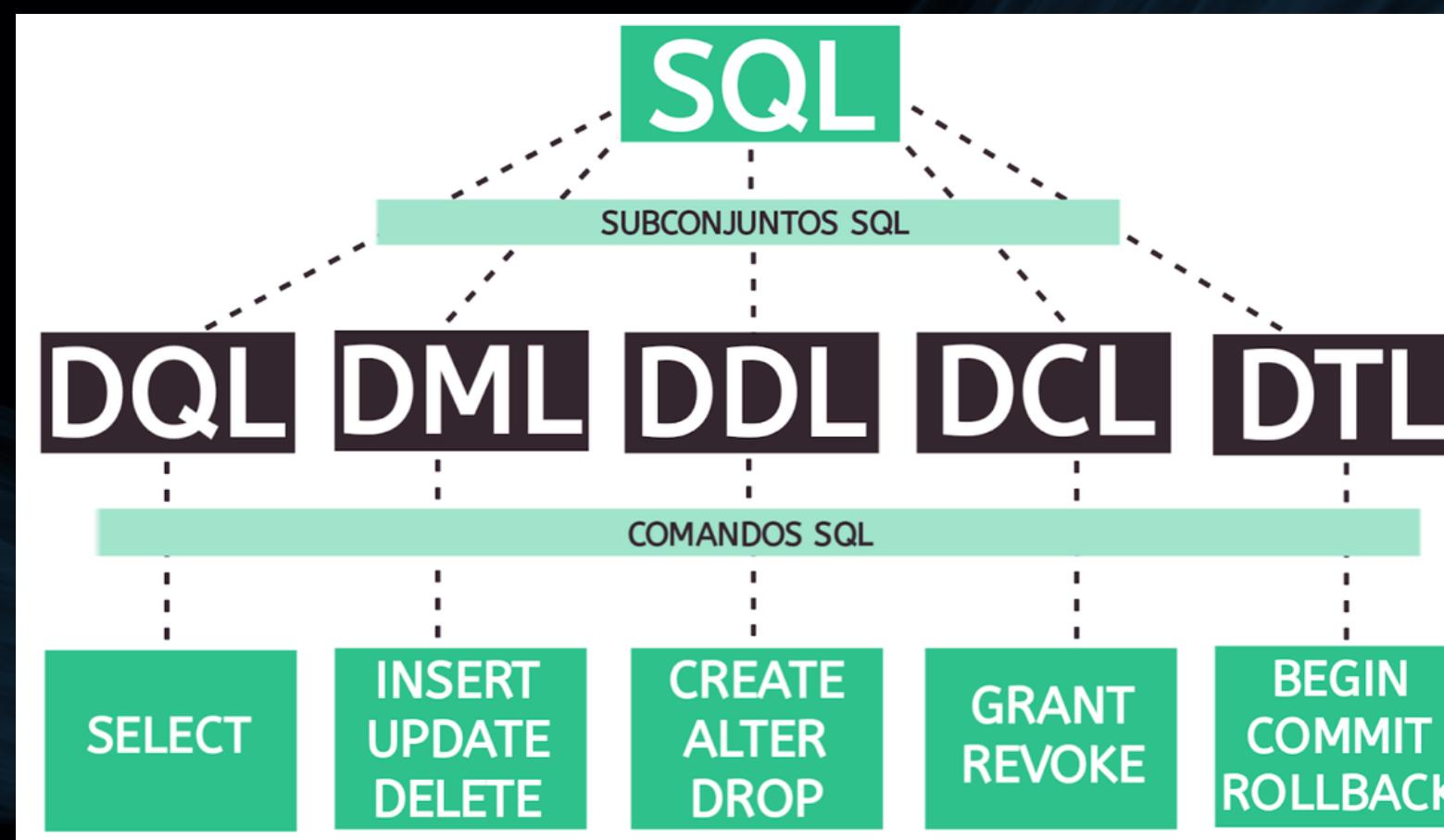
## Delete (Destroy)

Remover entradas existentes

CRUD É O ACRÔNIMO PARA AS 4 OPERAÇÕES BÁSICAS EM UM SISTEMA

# PY - SQL II

## Comandos DDL



Os comandos DDL (Data Definition Language) são estruturais. Eles servem para definir aspectos da estrutura do banco de dados, como criar tabelas, excluir tabelas, alterar tabelas e mais. São mais focados em um aspecto amplo, e menos nos dados necessariamente.

Ou seja, esses comandos servirão para a estrutura do banco de dados e **não para os dados**.

Os comandos DDL normalmente não são usados por um usuário geral, pois, nesse cenário, um usuário seria capaz de mexer na estrutura do banco de dados.

Fazem parte dos comandos DDL: **CREATE, DROP, ALTER, TRUNCATE, COMMENT, RENAME**

# PY - SQL II

## CREATE



```
1 create database if not exists nome_bd
2 /* Sintaxe básica */
3
4 /*
5  O banco será criado se não existir um banco de dados
6  com o mesmo nome
7 */
```

Existem duas instruções CREATE disponíveis no SQL:

- **CREATE DATABASE**
- **CREATE TABLE**

O comando **CREATE DATABASE** irá criar um novo banco de dados

# PY - SQL II

## CREATE



```
1 create table nome_tabela(  
2     coluna_01 tipo_dado(tamanho),  
3     coluna_02 tipo_dado(tamanho),  
4     coluna_03 tipo_dado(tamanho)  
5 );  
6 /* Sintaxe básica */  
7  
8 create table Usuarios(  
9     id int,  
10    nome varchar(40), /* Tamanho máximo de 40 caracteres*/  
11    esta_ativo boolean  
12 );  
13 /* Exemplo */
```

Com o banco de dados criado, precisamos criar as tabelas, para armazenar os dados.

O comando **CREATE TABLE** é usado para criar uma **tabela**.

Sabemos que uma tabela é composta por linhas e colunas, portanto, ao criar tabelas, precisamos fornecer todas as informações ao SQL sobre os nomes das colunas, tipos de dados a serem armazenados nas colunas, tamanho dos dados, etc.

# PY - SQL II

## DROP



```
1 drop table nome_tabela;  
2 drop database nome_bd;
```

O comando **DROP** é utilizado para excluir um banco de dados inteiro ou apenas uma tabela.

A instrução DROP destrói os objetos como um banco de dados, tabela, índice ou visão.

## PY - SQL II

## TRUNCATE



```
1 truncate table nome_tabela;  
2 /*  
3 Ao executar o comando acima, a tabela será truncada  
4 ou seja, os dados serão excluídos, mas a estrutura  
5 permanecerá na memória para operações posteriormente  
6 */
```

A instrução TRUNCATE que é usada para remover os dados de uma tabela para fins de deslocação (vazia para reutilização).

O resultado dessa operação remove rapidamente todos os dados de uma tabela, geralmente ignorando vários mecanismos de imposição de integridade.

A instrução TRUNCATE TABLE é logicamente equivalente à instrução DELETE FROM (sem a cláusula WHERE).



## PY - SQL II

# DROP X TRUNCATE

### DROP X TRUNCATE

- Truncar normalmente é ultrarrapido e é ideal para excluir dados de uma tabela temporária
- Truncar preserva a estrutura da tabela para uso futuro, ao contrário de **DROP TABLE** onde a tabela é excluída com sua estrutura completa.
- A exclusão da tabela ou banco de dados usando a instrução **DROP** não pode ser revertida, portanto, deve ser usada com cautela.

# PY - SQL II

## ALTER

**ALTER TABLE** é usado para adicionar, excluir/descartar ou modificar colunas na tabela existente.

Também é usado para adicionar ou descartar restrições na tabela existente.

Em conjunto com o comando **ALTER TABLE**, temos os comandos **ADD, DROP, MODIFY e RENAME**.

**ADD** é usado para adicionar colunas à tabela existente

**DROP** é usado para excluir colunas da tabela

**MODIFY** é usado para modificar uma coluna

**RENAME** é usado para renomear a tabela



```
1 alter table nome_tabela add (
2     nova_coluna tipo_dado(tamanho),
3     nova_coluna tipo_dado(tamanho)
4 );
5
6 alter table nome_tabela drop column nome_coluna;
7
8 alter table nome_tabela modify nome_coluna tipo_coluna;
```

## PY - SQL II

## RENAME



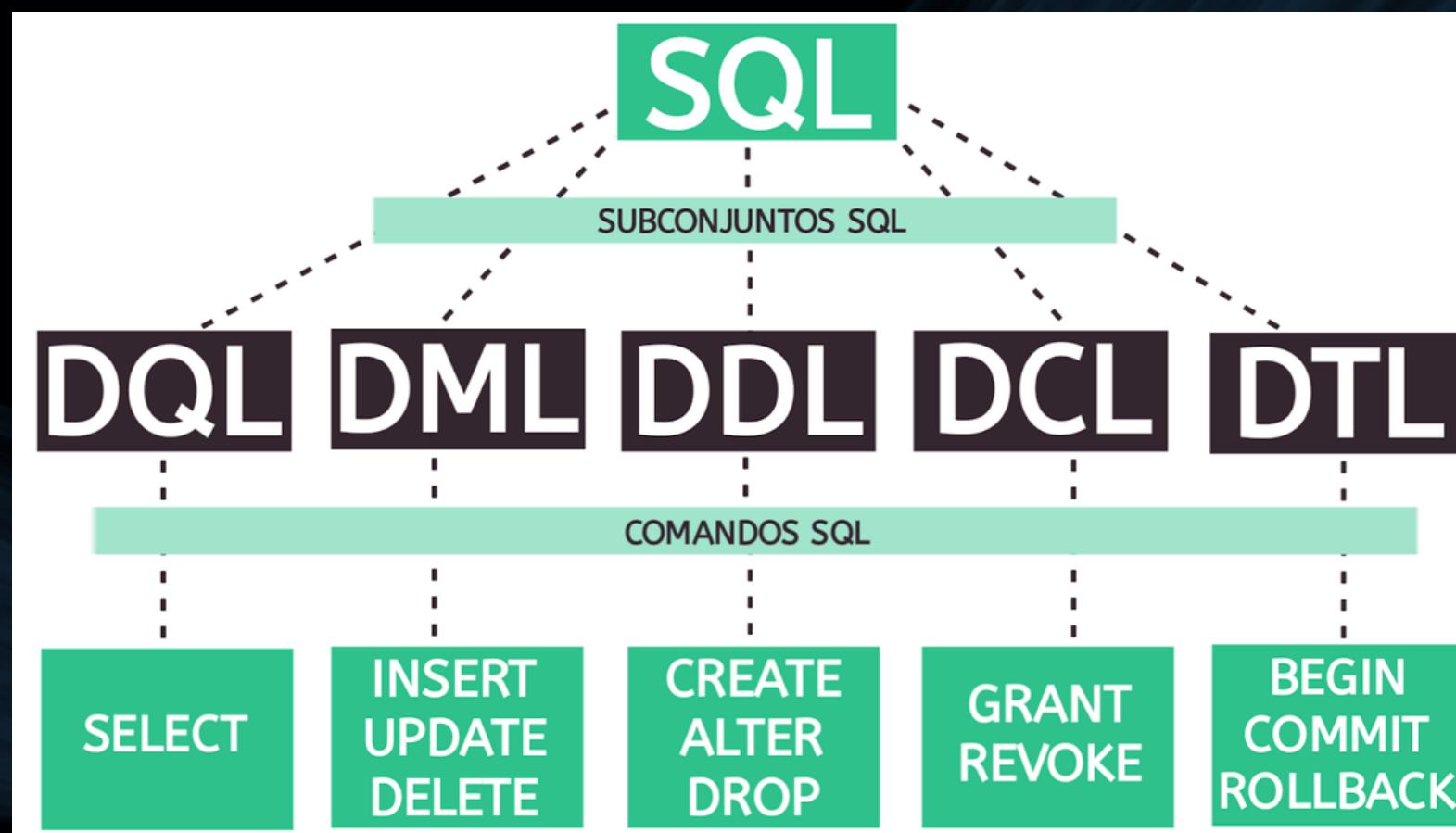
```
1 /* Para os SGBDs Oracle e MySQL*/  
2  
3 alter table nome_tabela rename to novo_nome_tabela;  
4  
5 alter table nome_tabela rename column nome_antigo to nome_novo;  
6  
7 /* Para o SGBD MariaDB*/  
8  
9 alter table nome_tabela change nome_antigo novo_nome tipo_dado;
```

Quando queremos renomear as tabelas usamos o comando **ALTER TABLE** em conjunto com o comando **RENAME**.

A sintaxe difere dependendo do SGBD

# PY - SQL II

## Comandos DQL



Os comandos DQL (Data Query Language) são mais focados na consulta. Ou seja, eles servem para recuperar dados e fazer a leitura ou manipulação deles. Indica onde está o dado que se quer gerenciar em um dado momento.

A instrução **SELECT** é usada para recuperar ou buscar dados de um banco de dados.

Elá permite buscar a tabela inteira ou buscar de acordo com alguma regra especificada.

Os dados retornados são armazenados em uma tabela de resultados. Essa tabela de resultados também é chamada de conjunto de resultados.

Ou seja, com a cláusula **SELECT**, especificamos as colunas que queremos que sejam exibidas no resultado da consulta.

# PY - SQL II

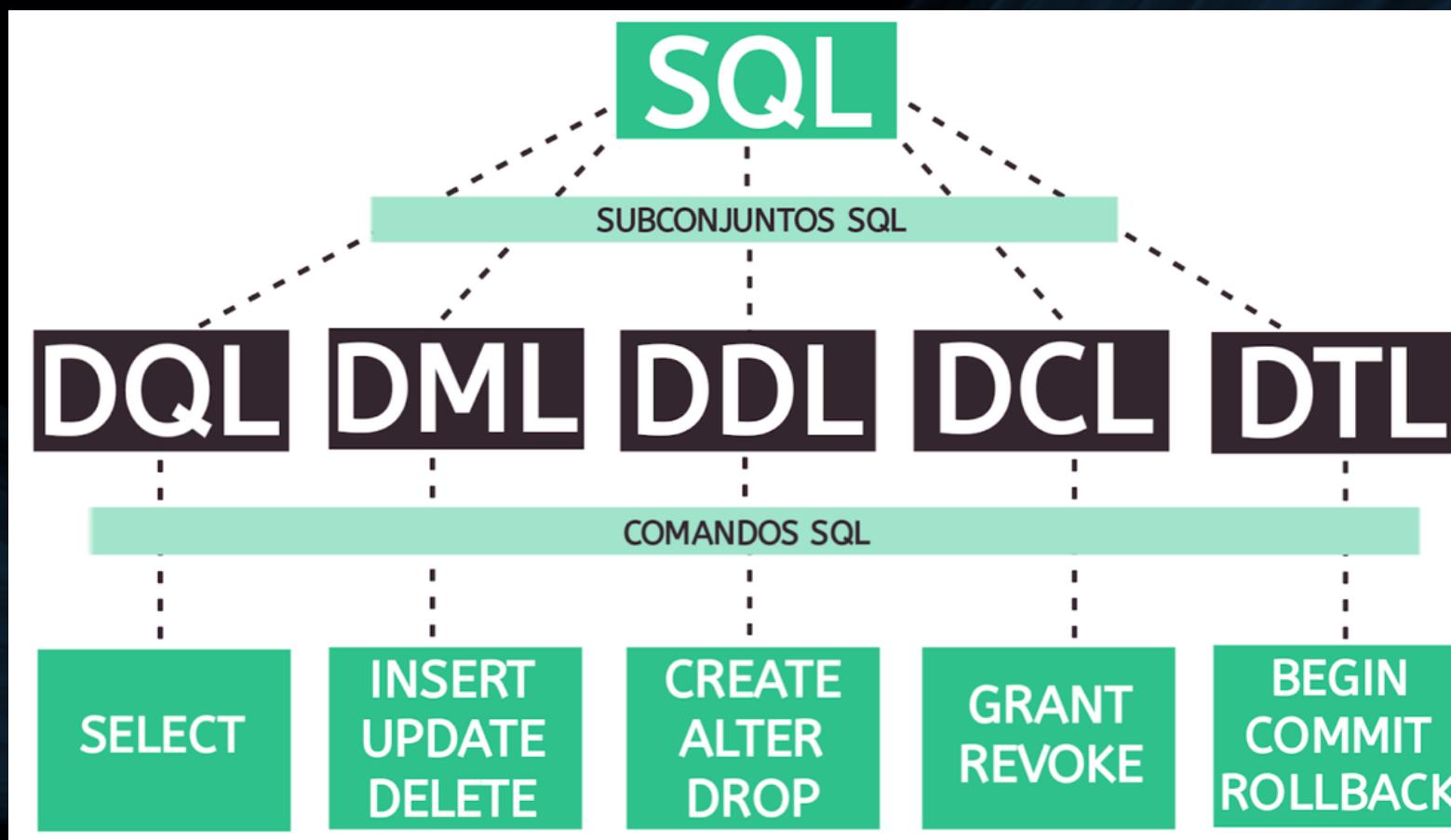
## Comandos DQL



```
1 select coluna_01 from nome_tabela; /* Sintaxe básica */
2
3 select * from nome_tabela; /* Selecionando a tabela inteira */
4
5 select coluna_01, coluna_02 from nome_tabela where condicao;
6 /* Podemos passar determinada condição para a consulta */
7
8 select nome, idade from Usuarios where id = 1; /* Exemplo */
```

# PY - SQL II

## Comandos DML



Os comandos DML (Data Manipulation Language) lidam com a manipulação dos dados presentes no banco de dados.

É nesta categoria que encontramos a maioria das instruções SQL

Comandos utilizados para gerenciar os dados armazenados no banco, permitindo inserir novos dados, alterar dados existentes ou excluir dados armazenados.

- **INSERT:** Usado para criar um novo registro no banco de dados, inserir uma nova linha
- **UPDATE:** Usado para atualizar um registro que está dentro do banco de dados
- **DELETE:** Usado para deletar um registro no banco de dados.

# PY - SQL II

## INSERT

A instrução **INSERT INTO** do SQL é usada para inserir uma nova linha em uma tabela.

Existem duas maneiras de usar a instrução **INSERT INTO**

**Somente valores:** Desta maneira, especificamos apenas os valores a serem inseridos, sem os nomes da colunas.

**Nomes e valores das colunas:** Especificamos as colunas que queremos preencher e seus valores correspondentes



```
1 insert into nome_tabela(valor1, valor2, valor3)
2 /* Passando somente os valores*/
3
4 insert into nome_tabela(coluna1, coluna2, coluna3)
5 values (valor1, valor2, valor3)
6 /* Passando o nome das colunas e os respectivos valores*/
```

# PY - SQL II

## UPDATE



```
1 update nome_tabela  
2 set coluna1 = valor1, coluna2 = valor2  
3 where condicao;  
4 /* Sintaxe básica */  
5  
6 update Usuario set name = 'Pedro' where id = 123;  
7 /* Exemplo */  
8  
9 update Usuario set name = 'Pedro', email = 'pedro@pedro.com'  
10 where id = 123;  
11 /* Exemplo atualizando mais de uma coluna*/
```

A instrução UPDATE em SQL é utilizada para atualizar os dados de uma tabela existente no banco de dados.

Podemos atualizar várias colunas ou apenas uma.

**OBS: Se não utilizarmos a cláusula WHERE, todos os registro sofrerão a mudança**

# PY - SQL II

## DELETE



```
1 delete from nome_tabela where condicao;  
2 /* Sintaxe básica */  
3  
4 delete from Usuario where id = 123;  
5 /* Exemplo */  
6  
7 delete from Usuario  
8 delete * from Usuario  
9 /* Removendo todos os registros */
```

A instrução **DELETE** é usada para excluir algum registro no banco de dados.

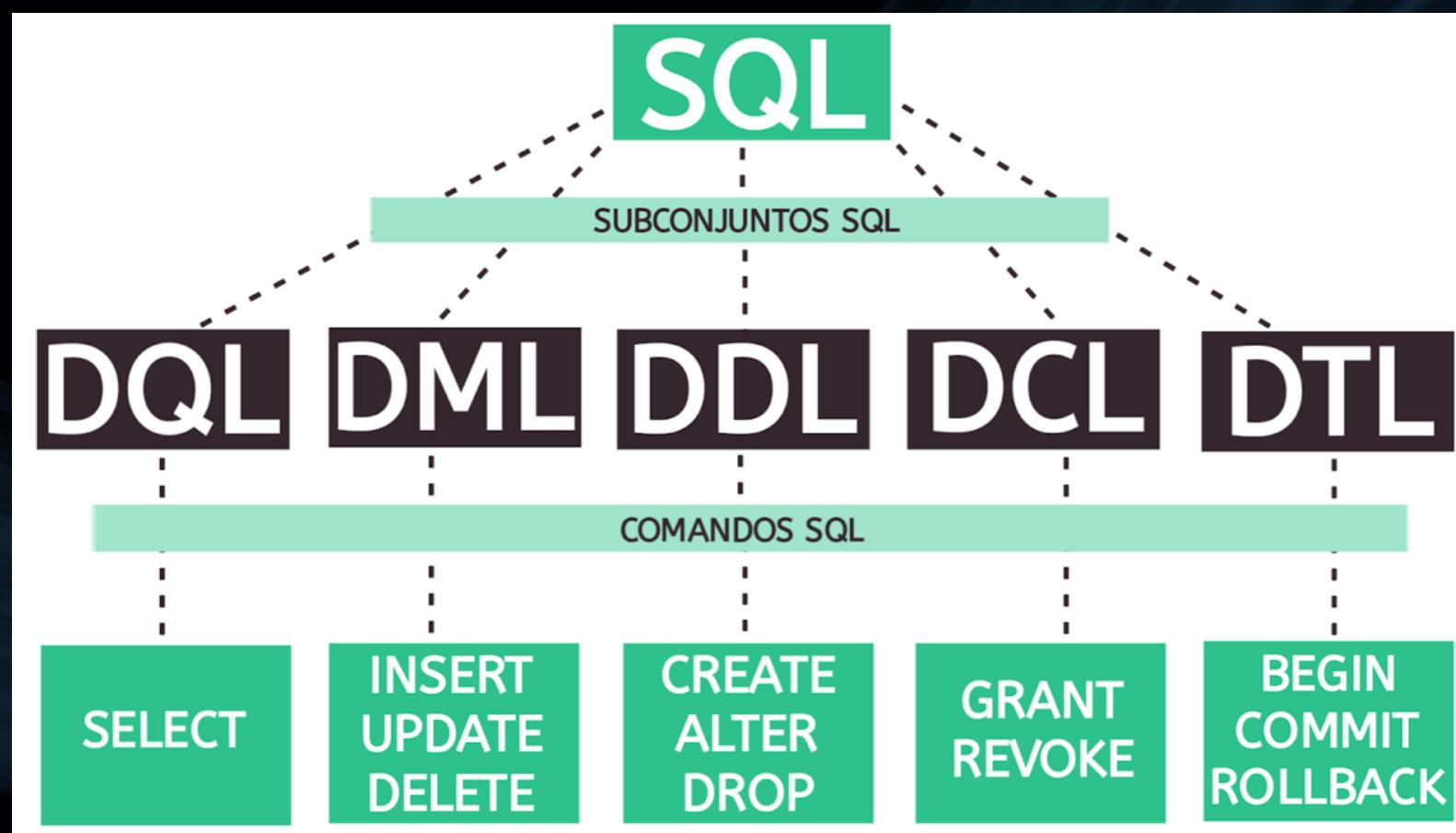
Podemos excluir um registro ou vários registros dependendo da condição que passarmos na cláusula **WHERE**.

OBS: Se omitirmos a cláusula **WHERE**, **todos os registros serão excluídos e a tabela ficará vazia**.

OBS2: **DELETE** é um comando DML, portanto, a operação executada por DELETE pode ser revertida ou desfeita.

# PY - SQL II

## Comandos DTL / TCL



Considerada por muitos, como uma outra categoria de cláusulas SQL, DTL (**Data Transaction Language**) / TCL (**Transactional Control Language**), concentra instruções que lidam com a transação dentro do banco de dados.

Ou seja, são comandos empregados para gerenciar transações no banco de dados. São usados para gerenciar as alterações realizadas por comandos DML executados.

Comandos DTL:

**COMMIT:** Confirma uma transação feita, ou seja, salva transações de forma permanente no banco de dados.

**ROLLBACK:** Restaura o banco ao último estado após um commit que teve êxito, ou seja, esse comando permite reverter uma transação caso ocorra algum erro.

**SET TRANSACTIONAL:** Especifica as características da transação.

# PY - SQL II

## Comandos DCL

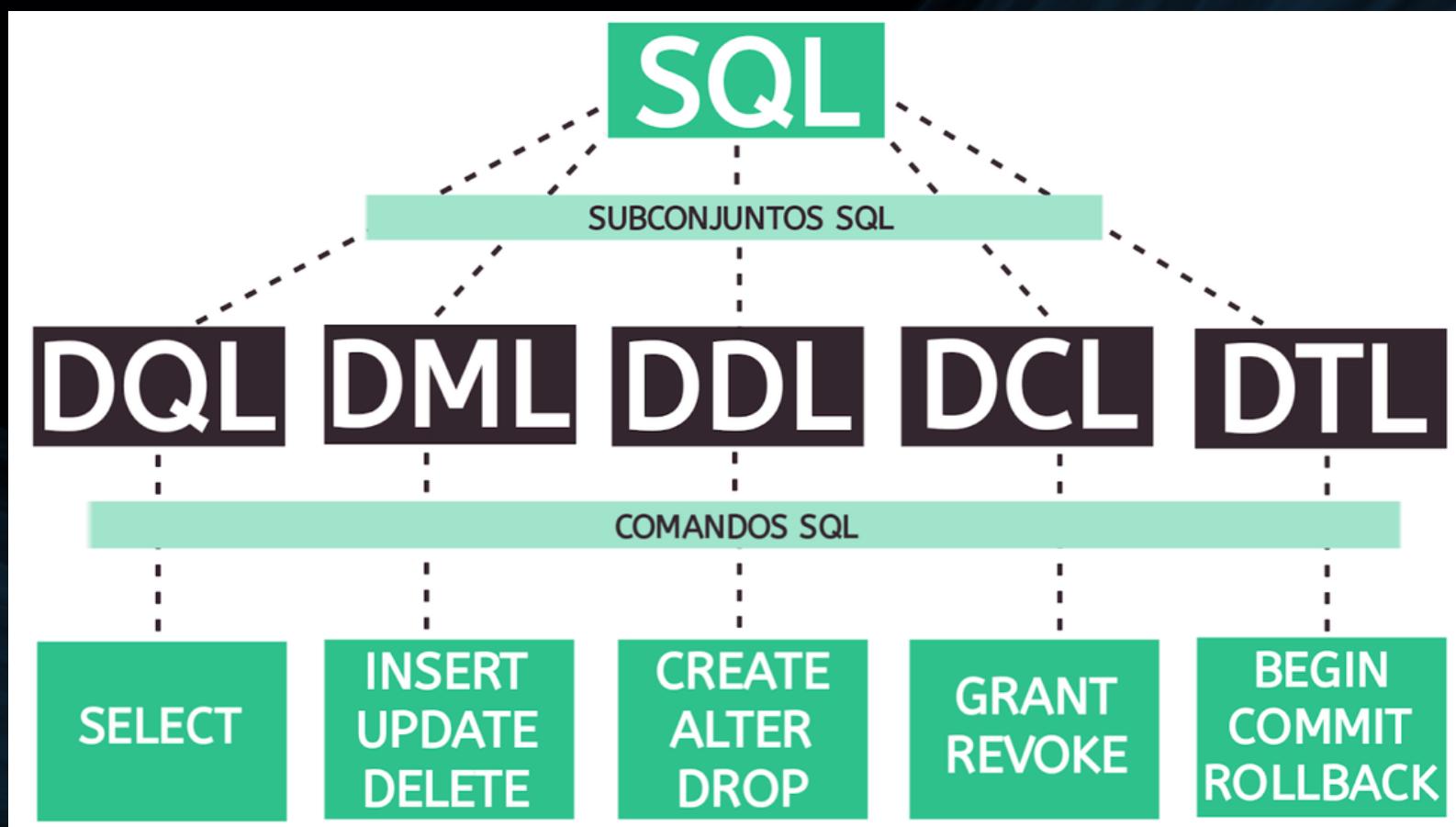
Os comandos DCL (Data Control Language) inclui comandos como **GRANT** e **REVOKE**, que lidam principalmente com os direitos, permissões e outros controles do sistema de banco de dados.

Ou seja, são comandos utilizados para controlar o acesso aos dados armazenados no banco, por meio de permissões de acesso.

**GRANT:** Dá aos usuários privilégios de acesso ao banco de dados.

**REVOKE:** Retira os privilégios de acesso do usuário concedidos pelo comando GRANT.

**ALTER USER:** Permite modificar contas, como a senha de um usuário.



# PY - SQL II

## Mão no código

Crie um banco de dados para um sistema de uma escola, esse banco de dados ficará responsável por persistir os dados sobre alunos, professores, turmas e disciplinas.

Para os alunos é importante que contenha um número de matrícula, o nome, a idade, e o endereço.

Para os professores, deverá conter um número de matrícula, nome, especialidade e endereço.

Para a turma deverá conter um identificador, horário de início e dia de semana.

Para disciplina é importante que contenha um identificador, nome e quantidade de aulas.



# PY - SQL II

## Mão no código

Crie 3 registros de alunos, 3 registros para professores, 2 para turmas e uma disciplina.

Após inserir esses novos registros, apague o registro do primeiro aluno e do segundo professor.

Mude o horário de início das turmas para 2h mais cedo.

Crie uma disciplina nova.





IN

# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

IN PARABÉNS! VOCÊ TERMINOU A AULA 11 DO MÓDULO DE PYTHON