Data Engineering

Filipe Nascimento
fgvn@cesar.school

C.E.S.A.R

C . E . S . A . R

Pessoas impulsionando inovação.

Inovação impulsionando negócios.

Everton Dias
etgdb@cesar.org.br

Janaína Branco
jcb@cesar.org.br

SQL

Neste curso iremos abordar desde os fundamentos do que é ser um ENGENHEIRO DE DADOS, conhecer técnicas, ferramentas e compartilhar casos de uso e projetos.

CONCEITOS

SQL

SQL ou Structured Query Language é uma linguagem de programação criada no início dos anos 70 para lidar com os bancos de dados. Sua praticidade e facilidade permitem ao engenheiro de dados escrever códigos que criem, recuperem, modifiquem e eliminem estruturas de dados e registros em um banco de dados.

Através de um simples comando "CREATE" é possível criar bancos de dados ou tabelas ou mesmo em um "SELECT" recuperar informações de uma ou mais tabelas.

Muito embora a linguagem SQL tenha surgido na IBM, surgiram vários dialetos e derivações o que juntamente com a sua ampla utilização adoção levou a necessidade da criação de um padrão para a linguagem - a SQL ANSI- realizada pelo American National Standards Institute em 1986 e a ISO 1987.

SQL

ACID – é o acrônimo do conceito das quatro propriedades de transação de um banco de dados, são elas :

- **Atomicidade:** Em uma transação envolvendo duas ou mais partes de informações discretas, ou a transação será executada totalmente ou não será executada, garantindo assim que as transações sejam atômicas.
- **Consistência:** A transação cria um novo estado válido dos dados ou em caso de falha retorna todos os dados ao seu estado antes que a transação foi iniciada.
- **Isolamento:** Uma transação em andamento mas ainda não validada deve permanecer isolada de qualquer outra operação, ou seja, garantimos que a transação não será interferida por nenhuma outra transação concorrente.
- **Durabilidade:** Dados validados são registados pelo sistema de tal forma que mesmo no caso de uma falha e/ou reinício do sistema, os dados estão disponíveis em seu estado correto.

SQL

'Transação é uma sequência de operações como uma única unidade lógica de trabalho dentro de um SGBD, podendo ser qualquer alteração realizada.'

SQL

SQL

A linguagem SQL se subdivide nos seguintes subconjuntos básicos de comandos:

- DDL
- DML
- DQL
- DTL
- DCL

SQL - DDL

DDL - Data Definition Language ou Linguagem de Definição de Dados

Um comando pertencente ao subconjunto DDL permite ao engenheiro definir novas estruturas sejam tabelas, bancos de dados. Além do poder de criação e alteração, também é possível remover as estruturas por meio do comando DROP.

Exemplos:

CREATE DATABASE	ALTER DATABASE	DROP DATABASE
CREATE TABLE	ALTER TABLE	DROP TABLE
CREATE VIEW	ALTER VIEW	DROP VIEW

SQL - DML

DML - Data Manipulation Language ou Linguagem de Manipulação de Dados

A DML trata do subconjunto que agrupa as funcionalidades de inclusão, consultas, alterações e exclusões de dados presentes nas tabelas do banco de dados.

As DML's podem ainda ser subclassificadas como DMLs Procedurais ou Declarativas.

Exemplos:

INSERT

UPDATE

DELETE

SQL - DQL

DQL - Data Query Language ou Linguagem de Consulta de Dados

A DQL trata do subconjunto que agrupa as funcionalidades de seleção dos dados, sendo o grupo mais utilizado da linguagem SQL. Através do comando SELECT é possível construir nossas consultas e utilizá-las em combinações até com comandos DML e DDL buscando trazer o resultado esperado.

Exemplos:

```
SELECT * FROM ALUNOS;
```

SQL - DQL

Existem diversas opções de cláusulas e operadores que podemos utilizar de maneira combinada para obtermos mais flexibilidade e poder sobre nossas consultas são eles:

- FROM – Utilizada para especificar a tabela, que se vai selecionar os registros.
- WHERE – Utilizada para especificar as condições que devem reunir os registros que serão selecionados.
- GROUP BY – Utilizada para separar os registros selecionados em grupos específicos.
- HAVING – Utilizada para expressar a condição que deve satisfazer cada grupo.
- ORDER BY – Utilizada para ordenar os registros selecionados com uma ordem específica.
- DISTINCT – Utilizada para selecionar dados sem repetição.
- UNION – combina os resultados de duas consultas SQL em uma única tabela para todas as linhas correspondentes.

SQL - DQL

Existem diversas opções de cláusulas e operadores que podemos utilizar de maneira combinada para obtermos mais flexibilidade e poder sobre nossas consultas são eles:

Cláusulas

- FROM – Utilizada para especificar a tabela, que se vai selecionar os registros.
- WHERE – Utilizada para especificar as condições que devem reunir os registros que serão selecionados.
- GROUP BY – Utilizada para separar os registros selecionados em grupos específicos.
- HAVING – Utilizada para expressar a condição que deve satisfazer cada grupo.
- ORDER BY – Utilizada para ordenar os registros selecionados com uma ordem específica.
- DISTINCT – Utilizada para selecionar dados sem repetição.
- UNION – combina os resultados de duas consultas SQL em uma única tabela para todas as linhas correspondentes.

SQL - DQL

Operadores Lógicos

- **AND – E lógico.** Avalia as condições e devolve um valor verdadeiro caso ambos sejam corretos.
- **OR – OU lógico.** Avalia as condições e devolve um valor verdadeiro se algum for correto.
- **NOT – Negação lógica.** Devolve o valor contrário da expressão.

Operadores Relacionais

- **<** - **menor que**
- **>** - **maior que**
- **<=** - **menor ou igual a**
- **>=** - **maior ou igual a**
- **=** - **igual**
- **<>** - **diferente**
- **between** - **Valor dentro de um espaço fechado;**
- **like** - **Comando para comparação por proximidade**
- **in** - **Busca em uma lista**

SQL - DQL

Funções de Agregação

As funções de agregação, como os exemplos abaixo, são usadas dentro de uma cláusula SELECT em grupos de registros para devolver um único valor que se aplica a um grupo de registros:

- AVG – Utilizada para calcular a média dos valores de um campo determinado.
- COUNT – Utilizada para devolver o número de registros da seleção.
- SUM – Utilizada para devolver a soma de todos os valores de um campo determinado.
- MAX – Utilizada para devolver o valor mais alto de um campo especificado.
- MIN – Utilizada para devolver o valor mais baixo de um campo especificado.
- STDDEV - Utilizada para funções estatísticas de desvio padrão
- VARIANCE - Utilizada para funções estatísticas de variância

PRÁTICA

Exercício

1. Suba as instâncias do Postgres e do MySQL no arquivo docker-compose.yml disponibilizado no classroom.
2. Dentro do conteiner do postgres acesse o console do sgbd utilizando o comando :
   a. $psql -U postgres
3. Dentro do conteiner do MariaDB acesse o console do SGBD utilizando o comando:
   a. $mysql -u root -p
4. Execute os comandos DDL e DML apresentados no arquivo disponibilizado no classroom.
5. Escreva um "Gerador de Massas" em shell para sua estrutura criada na aula;
   a. Foram fornecidos os seguintes arquivos, nomes.txt, datas.txt, com base neles crie um gerador de script para popular seus bancos de dados;
   b. Após a criação destes scripts, inclua-os junto em um volume em seu arquivo compose e execute-os dentro de seu conteiner;

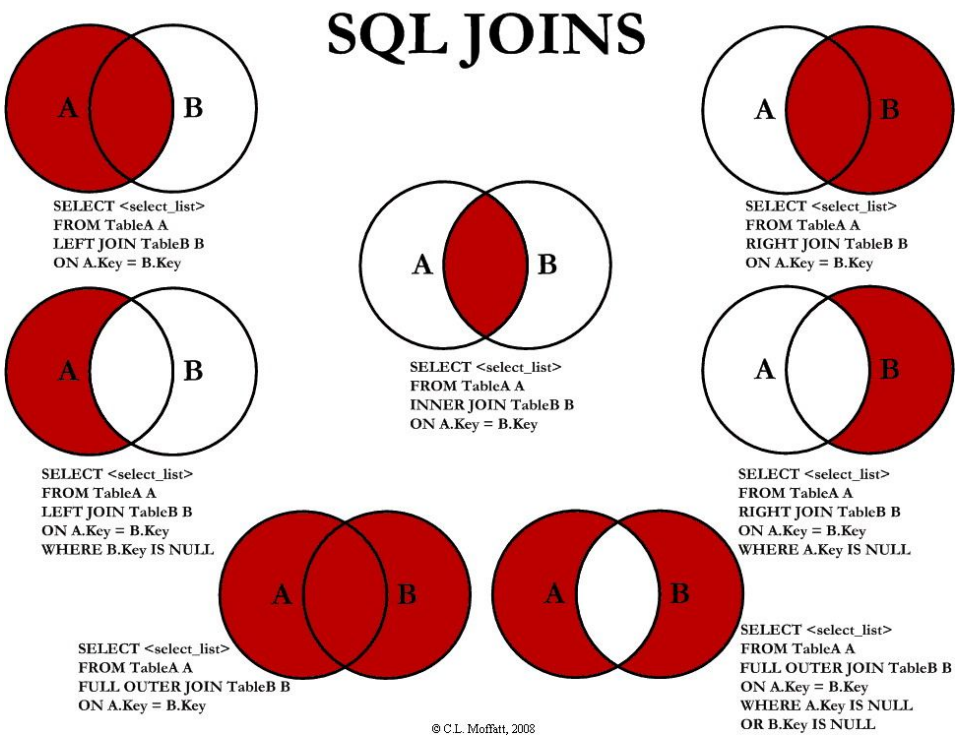
SQL - DQL

JOIN

A cláusula join é uma das mais importantes, pois permite realizar a junção de uma ou mais tabelas em suas pesquisas, podendo ser combinada de maneira a realizar operações de conjuntos às relações.

```
SELECT      PESSOAS.NOME_PESSOA AS NOME,
            BAIRRO.NOME_BAIRRO AS BAIRRO
FROM PESSOAS INNER JOIN ENDERECO
ON PESSOAS.CODIGO_ENDERECO = ENDERECO.CODIGO_ENDERECO
INNER JOIN BAIRRO
ON ENDERECO.CODIGO_BAIRRO = BAIRRO.CODIGO_BAIRRO
```

SQL - DQL



AMANHÃ TEM A PARTE 2!