Manipulação de banco de dados

Professor
Fernando Martins Medeiros

SQL

- Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional)
- Desenvolvido originalmente no início dos anos 70 nos laboratórios da IBM
- Linguagem padrão entre os bancos de dados relacionais por sua simplicidade e facilidade de uso;

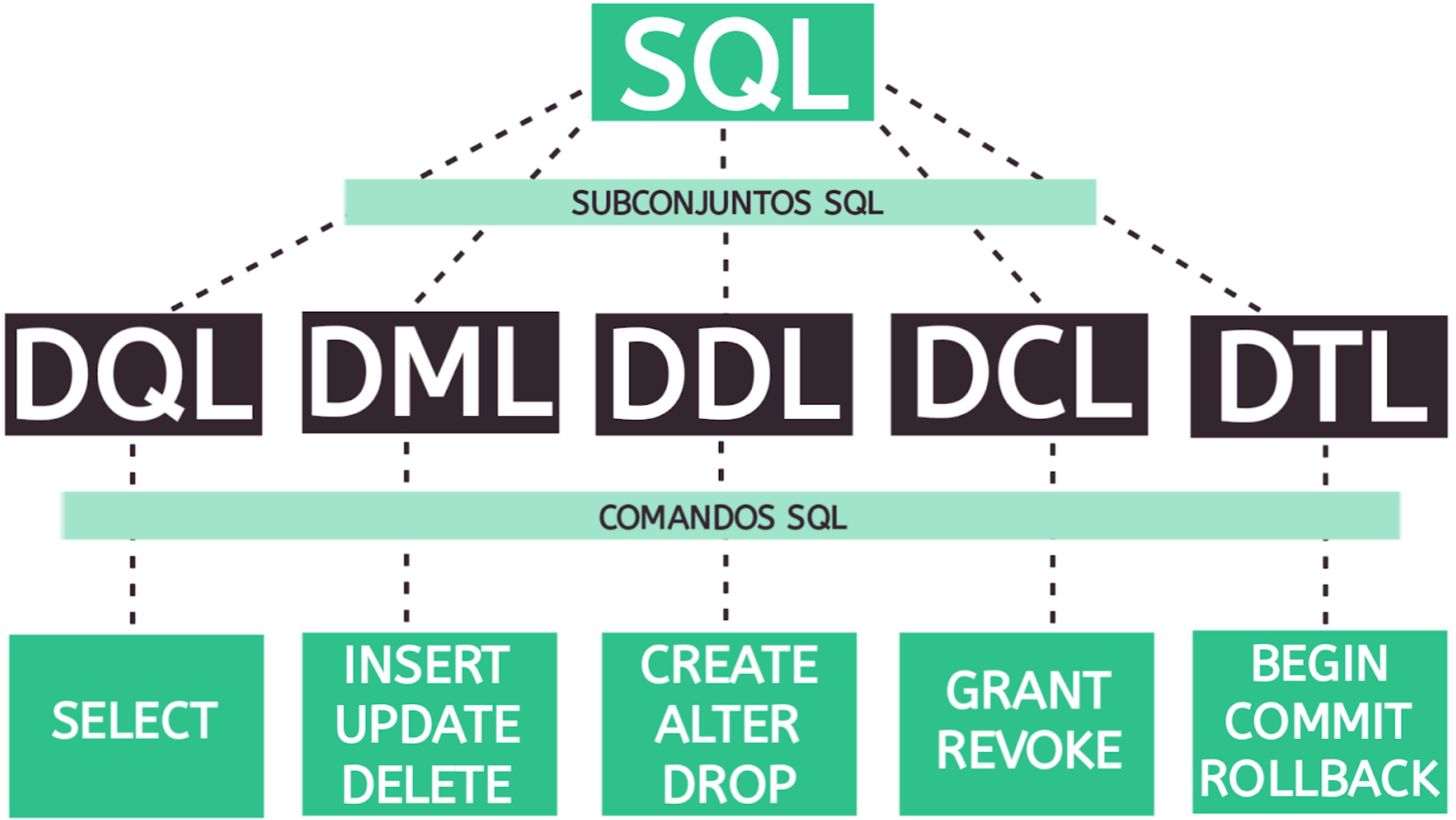
Organização da SQL

- DQL - Linguagem de Consulta de Dados - Define o comando utilizado para que possamos consultar (SELECT) os dados armazenados no banco;
- DML - Linguagem de Manipulação de Dados - Define os comandos utilizados para manipulação de dados no banco (INSERT, UPDATE e DELETE);
- DDL - Linguagem de Definição de Dados - Define os comandos utilizados para criação (CREATE) de tabelas, views, índices, atualização dessas estruturas (ALTER), assim como a remoção (DROP);
- DCL - Linguagem de Controle de Dados - Define os comandos utilizados para controlar o acesso aos dados do banco, adicionando (GRANT) e removendo (REVOKE) permissões de acesso;
- DTL - Linguagem de Transação de Dados - Define os comandos utilizados para gerenciar as transações executadas no banco de dados, como iniciar (BEGIN) uma transação, confirmá-la (COMMIT) ou desfazê-la (ROLLBACK).

Organização da SQL

- DQL - Linguagem de Consulta de Dados - Define o comando utilizado para que possamos consultar (SELECT) os dados armazenados no banco;
- DML - Linguagem de Manipulação de Dados - Define os comandos utilizados para manipulação de dados no banco (INSERT, UPDATE e DELETE);
- DDL - Linguagem de Definição de Dados - Define os comandos utilizados para criação (CREATE) de tabelas, views, índices, atualização dessas estruturas (ALTER), assim como a remoção (DROP);
- DCL - Linguagem de Controle de Dados - Define os comandos utilizados para controlar o acesso aos dados do banco, adicionando (GRANT) e removendo (REVOKE) permissões de acesso;
- DTL - Linguagem de Transação de Dados - Define os comandos utilizados para gerenciar as transações executadas no banco de dados, como iniciar (BEGIN) uma transação, confirmá-la (COMMIT) ou desfazê-la (ROLLBACK).

Organização da SQL



DQL - Linguagem de Consulta de Dados

Função	Comandos SQL	Descrição do comando	Exemplo
consultas	SELECT	O Select é o principal comando usado em SQL para realizar consultas a dados pertencentes a uma tabela.	`Select * From Pessoa;`

```sql
SELECT id, nome, idade FROM pessoa WHERE nome = 'FERNANDO';
```

DML - Linguagem de Manipulação de Dados

função	comandos SQL	descrição do comando	exemplo
inclusões	INSERT	é usada para inserir um registro (formalmente uma tupla) a uma tabela existente.	`INSERT INTO Pessoa (id, nome, sexo) VALUE;`
alterações	UPDATE	para mudar os valores de dados em uma ou mais linhas da tabela existente.	UPDATE Pessoa SET data_nascimento = '11/09/1985' WHERE id_pessoa = 7
exclusões	DELETE	permite remover linhas existentes de uma tabela.	DELETE FROM pessoa WHERE id_pessoa = 7

```sql
INSERT INTO marca (id, nome, id_tipo) VALUES (1,'CHEVROLET',1);
UPDATE marca SET nome = 'VOLKSWAGEN' WHERE id = 1;
DELETE FROM marca WHERE id = 1;
```

DDL - Linguagem de Definição de Dados

Uma DDL permite ao utilizador definir tabelas novas e elementos associados. A maioria dos bancos de dados de SQL comerciais tem extensões proprietárias no DDL.

Os comandos básicos da DDL são poucos:

- CREATE: cria um objeto (uma Tabela, por exemplo) dentro da base de dados.
- DROP: apaga um objeto do banco de dados.

Alguns sistemas de banco de dados usam o comando ALTER, que permite ao usuário alterar um objeto, por exemplo, adicionando uma coluna a uma tabela existente.

Outros comandos DDL:

- CREATE TABLE
- CREATE INDEX
- CREATE VIEW
- ALTER TABLE
- ALTER INDEX
- DROP INDEX
- DROP VIEW

DDL - Linguagem de Definição de Dados

```sql
CREATE TABLE `tipo` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(45) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `nome_UNIQUE` (`nome`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

CREATE TABLE `marca` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(255) NOT NULL,
  `id_tipo` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_marca_tipo_idx` (`id_tipo`),
  CONSTRAINT `fk_marca_tipo` FOREIGN KEY (`id_tipo`)
  REFERENCES `tipo` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=535 DEFAULT CHARSET=utf8;

CREATE TABLE `modelo` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(255) NOT NULL,
  `id_marca` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_modelo_marca1_idx` (`id_marca`),
  CONSTRAINT `fk_modelo_marca1` FOREIGN KEY (`id_marca`)
  REFERENCES `marca` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=5029 DEFAULT CHARSET=utf8;
```

DCL - Linguagem de Controle de Dados

DCL controla os aspectos de autorização de dados e licenças de usuários para controlar quem tem acesso para ver ou manipular dados dentro do banco de dados.

Duas palavras-chaves da DCL:

- GRANT - autoriza ao usuário executar ou setar operações.
- REVOKE - remove ou restringe a capacidade de um usuário de executar operações.

```
GRANT SELECT, UPDATE, INSERT
ON classicmodels.*
TO rfc@localhost;
```

```
REVOKE INSERT, UPDATE
ON classicmodels.*
FROM rfc@localhost;
```

```
SHOW GRANTS FOR rfc@localhost;
```

Grants for rfc@localhost
GRANT USAGE ON *.* TO `rfc`@`localhost`
GRANT SELECT, INSERT, UPDATE ON `classicmodels`.* TO `rfc`@`localhost`

Cláusulas

As cláusulas são condições de modificação utilizadas para definir os dados que deseja selecionar ou modificar em uma consulta:

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 especifica.
DISTINCT – Utilizada para selecionar dados sem repetição.
UNION – combina os resultados de duas consultas SQL em uma única tabela para todas as linhas correspondentes.

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

O SQL possui operadores relacionais, que são usados para realizar comparações entre valores, em estruturas de controle.

Operador	Descrição	Exemplos	
<	Menor	SELECT * FROM informacao.tabela WHERE idade < 18;	Seleciona todos os registros na "tabela" que possuem o campo "idade" com valores **menores** que 18.
>	Maior	SELECT * FROM informacao.tabela WHERE idade > 18;	Seleciona todos os registros na "tabela" que possuem o campo "idade" com valores **maiores** que 18.
<=	Menor ou igual	SELECT * FROM informacao.tabela WHERE idade <= 18;	Seleciona todos os registros na "tabela" que possuem o campo "idade" com valores **menores ou iguais à** 18.
>=	Maior ou igual	SELECT * FROM informacao.tabela WHERE idade >= 18;	Seleciona todos os registros na "tabela" que possuem o campo "idade" com valores **maiores ou iguais à** 18.
=	Igual	SELECT * FROM informacao.tabela WHERE idade = 18;	Seleciona todos os registros na "tabela" que possuem o campo "idade" com valores **exatamente iguais à** 18.
<>	Diferente	SELECT * FROM informacao.tabela WHERE idade <> 18;	Seleciona todos os registros na "tabela" que possuem o campo "idade" com valores que **são diferentes de** 18.

BETWEEN – Utilizado para especificar valores dentro de um intervalo fechado.
LIKE – Utilizado na comparação de um modelo e para especificar registros de um banco de dados. "Like" + extensão % significa buscar todos resultados com o mesmo início da extensão.
IN - Utilizado para verificar se o valor procurado está dentro de um« »a lista. Ex.: valor IN (1,2,3,4).

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