



Sistemas de Gerenciamento de Banco de Dados

Diogo Bortolini

Estruturas (Conjuntos) de Linguagens SQL

DDL – Linguagem de Definição de Dados: criar novas tabelas e alterar ou excluir elementos associados. Comandos: CREATE – ALTER - DROP

DML – Linguagem de Manipulação de Dados: alterações dos dados no registro. Comandos: INSERT - UPDATE - DELETE

DCL – Linguagem de Controle de Dados: autorização de acesso ou manipulação dos dados. Comandos: GRANT - REVOKE

DTL – Linguagem de Transação de Dados: controlam as transações. Comandos: BEGIN WORK (START TRANSACTION) – COMMIT - ROLLBACK

DQL – Linguagem de Consulta de Dados: consulta dos dados. Comando: SELECT

DQL

DDL – Linguagem de Definição de Dados: criar novas tabelas e alterar ou excluir elementos associados. Comandos: CREATE – ALTER - DROP

DML – Linguagem de Manipulação de Dados: alterações dos dados no registro. Comandos: INSERT - UPDATE - DELETE

DCL – Linguagem de Controle de Dados: autorização de acesso ou manipulação dos dados. Comandos: GRANT - REVOKE

DTL – Linguagem de Transação de Dados: controlam as transações. Comandos: BEGIN WORK (START TRANSACTION) – COMMIT - ROLLBACK

DQL – Linguagem de Consulta de Dados: consulta dos dados.
Comando: SELECT

UNION e UNION ALL

Combina os resultados de duas ou mais consultas em um único conjunto e resultados. O número e a ordem das colunas são idênticas em todas as consultas e o tipo do dado precisa ser compatível.

A diferença entre UNION e UNION ALL é que o primeiro faz um SELECT DISTINCT Antes de apresentar o resultado, ou seja, não apresentar linhas duplicadas.

O UNION ALL não faz SELECT DISTINCT, ou seja, apresenta duplicados.

UNION

SELECT colunas **FROM** tabela1
UNION
SELECT colunas **FROM** tabela2 ;

SELECT colunas **FROM** tabela1
UNION ALL
SELECT colunas **FROM** tabela2 ;

UNION

```
SELECT primeiro_nome, ultimo_nome FROM cliente  
UNION ALL
```

```
SELECT primeiro_nome, ultimo_nome FROM ator;
```

```
SELECT primeiro_nome, ultimo_nome FROM cliente  
UNION
```

```
SELECT primeiro_nome, ultimo_nome FROM ator;
```


UNION

```
SELECT funcionario_id, primeiro_nome, ultimo_nome FROM funcionarios  
WHERE depto = "producao" OR cidade = "Joinville" OR setor = "A2";
```

VS

```
SELECT funcionario_id, primeiro_nome, ultimo_nome FROM funcionarios WHERE depto = "producao"  
UNION ALL  
SELECT funcionario_id, primeiro_nome, ultimo_nome FROM funcionarios WHERE cidade = "Joinville"  
UNION ALL  
SELECT funcionario_id, primeiro_nome, ultimo_nome FROM funcionarios WHERE setor = "A2";
```

OPERADOR IN

IN depois da cláusula WHERE, impõe a condição dos valores que correspondem à lista de valores fornecida na palavra-chave IN.

IN ajuda a reduzir o número de cláusulas OR.

```
SELECT pais_id, pais  
FROM pais  
WHERE pais IN ('Afghanistan', 'Bangladesh', 'China');
```


Referências

- RODRIGUEZ, F. T. **Um pouco sobre a história dos bancos de dados**. Save Point 2017.
- GOMES, Daniella. **Modelagem de dados: 1:N ou N:N?** Disponível em: <https://www.devmedia.com.br/modelagem-1-n-ou-n-n/38894>
- FERREIRA, Nickerson Fonseca. **Normalização SQL**. IFRN, 2019
- DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campus, 2004.
- RAMAKRISHNAN, Raghu. **Sistemas de gerenciamento de banco de dados**. São Paulo: McGraw Hill, 2009.

Referências

- MILANI, ANDRÉ. **Construindo Aplicações Web com PHP e Mysql**. Novatec, 2010.
- SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. Rio de Janeiro: Elsevier, 2006.
- MACHADO, Felipe Nery R.; ABREU, M. **Projeto de Banco de Dados: Uma visão prática**. 15ª ed. São Paulo: Érica, 2008.
- <https://dev.mysql.com/doc/refman/8.0/en/storage-engines.html>