

Aula 5

Banco de dados

Prof. Lucas Rafael Filipak

Conversa Inicial

Banco de dados

- Consulta entre tabelas
- *Data Control Language (DCL)*
- *Transact Control Language (TCL)*
- Índice

Consulta entre tabelas

Cláusula JOIN

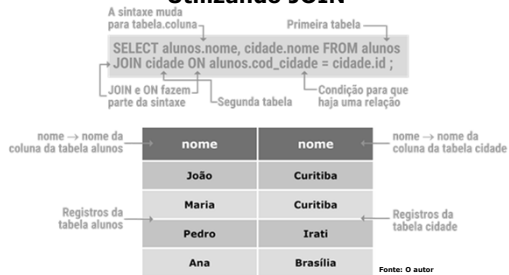
- A cláusula JOIN é utilizada para recuperar dados entre mais de uma tabela. Assim, com o cruzamento de dados entre elas, é possível obter registros mais completos

Exemplos de tabelas

Tabela: alunos			Tabela: cidade	
id	nome	id_cidade	id	nome
1	João	1	1	Curitiba
2	Maria	1	2	Irati
3	Pedro	2	3	Brasília
4	Ana	3	5	Cuiabá
5	Julia	4		

Fonte: O autor

Utilizando JOIN



Utilizando LEFT JOIN

```
SELECT alunos.*, cidade.* FROM alunos LEFT
JOIN cidade ON alunos.cod_cidade = cidade.id ;
```

id	nome	id_cidade	id	nome
1	João	1	1	Curitiba
2	Maria	1	1	Curitiba
3	Pedro	2	2	Irati
4	Ana	3	3	Brasília
5	Julia	4	NULL	NULL

Fonte: O autor

Utilizando RIGHT JOIN

```
SELECT alunos.*, cidade.* FROM alunos RIGHT
JOIN cidade ON alunos.cod_cidade = cidade.id ;
```

id	nome	id_cidade	id	nome
1	João	1	1	Curitiba
2	Maria	1	1	Curitiba
3	Pedro	2	2	Irati
4	Ana	3	3	Brasília
NULL	NULL	NULL	5	Cuiabá

Fonte: O autor

Apelidando colunas

- As colunas podem receber apelidos para facilitar sua identificação

"nome aluno" é o apelido dado à coluna nome da tabela alunos

"cidade" é o apelido dado à coluna nome da tabela cidade

```
SELECT alunos.nome as 'nome aluno', cidade.nome as 'cidade'
FROM alunos JOIN cidade ON alunos.cod_cidade = cidade.id ;
```

nome aluno	cidade
João	Curitiba

Fonte: O autor

Outras funções do SELECT

- Média (AVG)

```
SELECT avg(salario)
FROM funcionarios ;
```

avg (salario)
4475.000000

Fonte: O autor

- GROUP BY

```
SELECT sexo, avg(salario) FROM
funcionarios GROUP BY sexo;
```

sexo	avg(salario)
M	4650.000000
F	4300.000000

Fonte: O autor

■ Soma (SUM)

```
SELECT sum(salario)
FROM funcionarios ;
```

sum(salario)
17900.00

Fonte: O autor

■ Número de registros (COUNT)

```
SELECT count(*) as
'funcionarios' FROM
funcionarios ;
```

funcionarios
4

Fonte: O autor

■ Máximo (MAX) e Mínimo (MIN)

```
SELECT max(salario)
FROM funcionarios ;
```

max(salario)
5300.00

```
SELECT min(salario)
FROM funcionarios ;
```

min(salario)
3300.00

Fonte: O autor

Data Control Language (DCL)

- A DCL, ou Linguagem de Controle de Dados, é uma subcategoria da DML
- É responsável por definir critérios de segurança em relação aos usuários dentro de um banco de dados
- Controla os aspectos de autorização de dados e permissões dos usuários

Criar usuário

- Somente um usuário com permissão para criar um novo usuário pode executar o comando

```
CREATE USER 'chefe' IDENTIFIED BY '123' ;
```

Fonte: O autor

Excluindo usuário

Exemplo:

```
DROP USER 'chefe' ;
```

Fonte: O autor

Permissões do usuário

- GRANT → concede as permissões, autorizando o usuário a executar ou "setar" operações
- REVOKE → retira as permissões, removendo ou restringindo a capacidade de um usuário de executar operações

GRANT

- Para Alves (2014),
 - Direitos: indica os direitos que podem ser concedidos ao usuário
 - nome_tabela: tabela ou visão na qual será aplicada a concessão dos direitos
 - Identificação: a quem os privilégios foram concedidos

Sintaxe

```
GRANT direitos ON nome_tabela TO identificação ;
```

Fonte: O autor

Exemplo

```
GRANT select ON alunos TO chefe ;
```

Fonte: O autor

Lista de permissões

CREATE	criar tabelas ou databases
DROP	deletar tabelas ou databases
DELETE	deletar registros
INSERT	inserir registros
SELECT	selecionar registros
UPDATE	atualizar/modificar registros

Fonte: O autor

```
GRANT select, insert, delete, update ON alunos TO chefe ;
```

Fonte: O autor

Nível de privilégio

,	Privilégio global
db.*	Qualquer tabela do bando db
db.tb	Apenas a tabela tb do banco de dados db. Para especificar apenas algumas colunas de determinada tabela, estas deverão ser listadas ao lado do privilégio (priv {colunas})

Fonte: Duarte, 2006

Exemplos

```
GRANT all privileges ON *.* TO chefe ;
```

Fonte: O autor

```
GRANT all privileges ON sistema.* TO chefe ;
```

Fonte: O autor

REVOKE

- Remove os privilégios de um usuário
- Utiliza a mesma lista de permissões e os mesmos níveis de privilégios do comando GRANT

```
REVOKE select ON alunos FROM chefe ;
```

Fonte: O autor

- Exemplo de remoção de todos os privilégios no banco de dados sistema, do usuário-chefe

```
REVOKE all privileges ON sistema.* FROM chefe ;
```

Fonte: O autor

Transact Control Language (TCL)

Transações

- São um conjunto de operações (comandos) que deve ser executado sem erros, para que a transação se efetive

Exemplo de transação

- Transferência bancária
 - Os dados do destinatário estão corretos
 - Tem dinheiro na conta do remetente
 - Tirar o dinheiro do remetente e passar para o destinatário

Comandos TCL

- **Begin:** indica o início de uma transação
- **Commit:** é o fim da transação, executando as instruções no banco de dados (permanente)
- **Rollback:** é o fim da transação também, mas cancela todas as alterações efetuadas porque algo deu errado

Exemplo de transação

```
BEGIN TRANSACTION;  
  
UPDATE CONTA_CORRENTE  
set saldoConta= saldoConta - @Valor  
where numConta = @contaDe;  
  
UPDATE CONTA_CORRENTE  
set saldoConta= saldoConta + @Valor  
where numConta = @contaPara;  
  
COMMIT;
```

Fonte: O autor

```
BEGIN TRANSACTION 'transferencia'  
INSERT INTO uf VALUES('pr'), ('sc');  
ROLLBACK TRANSACTION 'transferencia';
```

Fonte: O autor

- Quando a instrução é concluída, os dados não são inseridos na tabela "Uf", pois o comando COMMIT não foi utilizado

Índice

Por que utilizar?

- Para facilitar e agilizar as consultas
 - Exemplo: livro
- Os índices têm a função semelhante à dos índices dos livros físicos

Métodos de pesquisa de dados

- Exame nas tabelas → a consulta percorre todos os registros das tabelas e seleciona apenas os verdadeiros
- Índices → percorre a estrutura da árvore do índice, comparando e extraindo somente os registros verdadeiros

Onde criar?

- Não utilizar colunas com uma grande quantidade de dados duplicados ou com pouca variação, como a coluna *sexo*
- O SGBD gasta recursos mantendo os índices sempre atualizados e associados

Índices

■ Simples

```
CREATE INDEX 'NomeFuncionario'  
ON 'funcionarios' (nome);
```

Fonte: O autor

■ Multicoluna

```
CREATE INDEX 'NomeFuncionario'  
ON 'funcionarios' (nome, sobrenome);
```

Fonte: O autor

```
CREATE TABLE funcionarios  
(  
  cod INT(3) auto_increment,  
  nome VARCHAR(30),  
  sobrenome VARCHAR(30),  
  sexo CHAR,  
  salario DECIMAL(6,2),  
  PRIMARY KEY (cod),  
  INDEX NomeFuncionario (sobrenome)  
);
```

Fonte: O autor

- Chaves primárias
- Chaves estrangeiras
- Colunas acessadas por intervalos (BETWEEN)
- Colunas utilizadas em GROUP BY ou ORDER BY

Finalizando

- JOIN
- AVG, SUM, MAX, MIN, COUNT
- DCL: criar usuários, atribuir e retirar permissões de acesso
- TCL: transações (COMMIT e ROLLBACK)
- Índices

Referências

- ALVES, W. P. Banco de dados. São Paulo: Érica, 2014.
- DUARTE, E. Gerenciamento de usuários e controle de acessos do MySQL. 2006. Disponível em: <<https://www.devmedia.com.br/gerenciamento-de-usuarios-e-controle-de-acessos-do-mysql/1898>>. Acesso em: 21 jul. 2018.

