

## Programação e Desenvolvimento de Banco de Dados

### Recursos avançados e automação de processos

Prof. Dr. Gilberto Fernandes Jr.

1

- Unidade de Ensino: 4
- Competência da Unidade: Conhecer e compreender a automação de processos em banco de dados
- Resumo: Saber elaborar script SQL para automação de tarefas em tabelas.
- Palavras-chave: visão, índice, transação, procedimento, função.
- Título da Teleaula: Recursos avançados e automação de processos
- Teleaula nº: 4

2

### Contextualização

- Visões e índices
- Controle transacional
- Procedimentos e funções

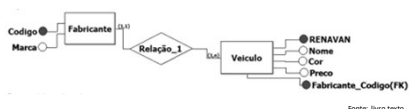
3

### Visões e índices em bancos de dados

4

### Introdução

- Necessidade de técnicas que proporcionem um maior aproveitamento dos recursos disponíveis
- Considere o exemplo:



5

### Visões (VIEW)

- O recurso SQL para gerar visões é uma alternativa para visualizar os dados de uma ou mais tabelas de um BD (**tabela virtual**)
- A técnica de VIEW encapsula uma seleção de dados (SELECT)
- Torna as consultas mais rápidas
- Quando há alterações, o SGBD atualiza a VIEW automaticamente.

6

## Visões (VIEW)

- Sintaxe e operações com uma VIEW:

```
CREATE VIEW [nome_da_VIEW] AS
SELECT [coluna]
FROM [tabela]
WHERE [condições];
```

```
SELECT * FROM [nome_da_VIEW];
```

```
DROP VIEW [nome_da_VIEW];
```

7

## Visões (VIEW)

- Exemplo:

```
CREATE VIEW v_select1 AS
SELECT veiculo.nome as "Veiculo", fabricante.
marca as "Marca", veiculo.cor as "Cor", veiculo.
preco as "Valor"
FROM veiculo INNER JOIN fabricante
WHERE veiculo.fabricante_Codigo = fabricante.
Codigo AND veiculo.preco <= 50000;
```

```
mysql> show tables;
+-----+
| Tables_in_car |
+-----+
| fabricante    |
| v_select1     |
| veiculo       |
+-----+
3 rows in set (0.05 sec)
```

Fonte: livro texto

8

## Visões (VIEW) - Vantagens

- Economia de tempo: diminuição na carga de criação de comandos SELECT
- Velocidade de acesso: devido às VIEWS estarem pré armazenadas
- Ocultação da complexidade: o usuário não necessita saber dos campos, nem das seleções.

9

## Índice (INDEX)

- Problema do *Table Scan*: tempo de verificação tende a ser muito grande.
- A utilização dos **índices** é opcional para a seleção de dados → estruturas redundantes.
- Não era admitido até a versão SQL:1999

10

## Índice (INDEX)

- Declarar um índice, no desenvolvimento da tabela:

```
CREATE TABLE [nomeDaTabela] (
  Campo1 tipo(tamanho),
  Campo2 tipo(tamanho),
  INDEX(Campo1));
```

- Declarar um índice em tabela existente no BD:

```
CREATE INDEX [nomeDoIndice]
ON [nomeDaTabela](Campo);
```

11

- Exemplo: 

```
CREATE INDEX idx_Renavam ON veiculo(RENAVAM);
```

```
SHOW INDEX FROM veiculo;
```

| Table   | Non_unique | Key_name          | Seq_in_index | Column_name       |
|---------|------------|-------------------|--------------|-------------------|
| veiculo | 0          | PRIMARY           | 1            | RENAVAM           |
| veiculo | 1          | fabricante_Codigo | 1            | fabricante_Codigo |
| veiculo | 1          | idx_Renavam       | 1            | RENAVAM           |

Fonte: livro texto

12

### Utilizar um índice

- Sintaxe para utilizar um índice:

```
SELECT [coluna] FROM [nomeDaTabela]
USE INDEX (nomeDoÍndice)
WHERE [condições];
```

- Exemplo:

```
SELECT nome AS "Veículo", cor AS "Cor",
Preço AS "Valor" FROM veiculo
USE INDEX(idx_Renavam)
WHERE preco <= 50000;
```

13

### FULLTEXT em banco de dados relacional

- Buscar um trecho dentro de várias *strings*

```
ALTER TABLE [nome_tabela] ADD FULLTEXT
(nome_da_coluna);
```

- Buscar palavras dentro de longos textos

```
SELECT [coluna] FROM nome_da_tabela
WHERE MATCH(coluna)
AGAINST('palavra_desejada');
```

14

## Otimizando consultas em um banco de dados

15

### Descrição da SP

- Você trabalha na prefeitura de um município voltado para o turismo.
- Devido ao movimento no entorno do pier ser sempre grande, há um banco de dados para gerenciamento e controle dos passeios, escunas e barqueiros.

16

### Descrição da SP

- MAS, os funcionários do órgão regulador relataram lentidão ao gerarem consultas para o relatório dos passeios contendo: nome da escuna, destino, horários e data.
- Foi solicitado que você desenvolvesse uma solução para o problema relatado.**

17

## Controle Transacional

18

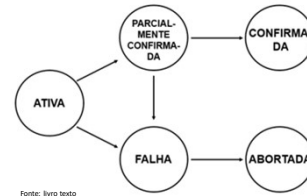
## Controle de Transação

- Controles de transação auxiliam nas tarefas para garantir a integridade do BD
  - pontos de salvamento, ou cancelar uma alteração
- Propriedades das transações (**ACID**):
  - A**tomicidade, **C**onsistência, **I**solamento e **D**urabilidade.

19

## Estados de uma transação

- Diagrama de estado transacional



20

## Comando COMMIT

- Quando uma transação se completa, é considerada CONFIRMADA (*committed*)
- AUTO COMMIT**: COMMIT em modo automático.
  - SET AUTOCOMMIT=0; para modo manual

21

- Exemplo:

```
UPDATE Medicacao SET EnfermeiroCoren = 22222
WHERE Id = 1;
```

Após um logout/login do sistema **SEM** COMMIT

```
mysql> select * from Medicacao where Id = 1;
```

| Id | Data       | Hora     | PacienteNum | RemedioCod | EnfermeiroCoren |
|----|------------|----------|-------------|------------|-----------------|
| 1  | 2018-07-01 | 05:00:00 | 1003        | 104        | 22222           |

Fonte: livro texto

```
mysql> select * from Medicacao where Id = 1;
```

| Id | Data       | Hora     | PacienteNum | RemedioCod | EnfermeiroCoren |
|----|------------|----------|-------------|------------|-----------------|
| 1  | 2018-07-01 | 05:00:00 | 1003        | 104        | 11111           |

Fonte: livro texto

COMMIT; Após um logout/login do sistema **COM** COMMIT

| Id | Data       | Hora     | PacienteNum | RemedioCod | EnfermeiroCoren |
|----|------------|----------|-------------|------------|-----------------|
| 1  | 2018-07-01 | 05:00:00 | 1003        | 104        | 22222           |

Fonte: livro texto

22

## Comando ROLLBACK

- Reverter transações em um banco
- Instruções DDL de criação e exclusão de banco de dados, ou ainda, as alterações, exclusões e criação de tabelas **não** admitem o uso do ROLLBACK.
- Para retornar a determinado ponto com o ROLLBACK, utiliza-se o SAVEPOINT!

23

## Comando ROLLBACK

- Sintaxe para criar pontos de restauração:

```
SAVEPOINT [nomeDoPonto];
```

- Para utilizar esse ponto:

```
ROLLBACK TO SAVEPOINT [nomeDoPonto];
```

- Para os controles **SAVEPOINT** e **ROLLBACK** funcionarem → SET AUTOCOMMIT = 0

24

## Resolução de SP

- Você está ligado a um projeto para gerenciamento e controle dos passeios de escunas e os barqueiros, a fim de se garantir a segurança dos turistas.
- Você já implementou uma visão (VIEW), fazendo com que o tempo de consulta diminuísse.

25

## Resolução de SP

- Ao alterar o nome de um destino, por uma falha de operação, todos os nomes foram alterados de forma errada:

```
mysql> select * from destino;
+----+-----+
| Id | Nome |
+----+-----+
| 1  | Pequena ilha do Mar |
| 2  | Pequena ilha do Mar |
| 3  | Pequena ilha do Mar |
| 4  | Pequena ilha do Mar |
| 5  | Pequena ilha do Mar |
| 6  | Pequena ilha do Mar |
| 7  | Pequena ilha do Mar |
| 8  | Pequena ilha do Mar |
+----+-----+
8 rows in set (0.00 sec)
```

Fonte: livro texto

26

## Descrição da SP: suas tarefas!

- Alteração do COMMIT para que as alterações não sejam gravadas automaticamente;
- Criação de um ponto de restauração no banco;
- Teste para gerar o mesmo erro;
- Utilização teste do ponto de restauração criado;
- Gravar as alterações feitas;
- Criar um novo ponto de restauração.
- **Vamos resolver a SP no MySQL Workbench!**

27

## Funções e procedimentos

28

## Funções (FUNCTION)

- Sintaxe para criar:

```
CREATE FUNCTION nome_da_funcao (x tipo, y tipo)
RETURNS tipo
RETURN (função);
```

- Sintaxe para utilizar:

```
SELECT nome_da_funcao (parâmetro x, parâmetro y)
FROM nome_da_tabela
WHERE nome_da_coluna (condição);
```

29

Exemplo:

```
CREATE FUNCTION fn_media(x DECIMAL(3,1), y DECIMAL(3,1))
RETURNS DECIMAL(3,1)
RETURN (x * 0.4) + (y * 0.6);
```

```
SELECT Aluno.Nome, disciplina.Nome AS "Disciplina",
       fn_media(NotaP1, NotaP2) AS "Média Final"
FROM Notas INNER JOIN Aluno
ON Notas.AlunoRA = Aluno.RA
INNER JOIN Disciplina
ON Notas.DisciplinaId = Disciplina.Id
WHERE fn_media(NotaP1, NotaP2) >= 4.0
AND fn_media(NotaP1, NotaP2) <= 6.9;
```

30

- Exemplo: saída...

| Nome    | Disciplina              | Média Final |
|---------|-------------------------|-------------|
| Aluno_A | Banco de dados          | 6.1         |
| Aluno_B | Banco de dados          | 6.1         |
| Aluno_D | Banco de dados          | 4.1         |
| Aluno_F | Banco de dados          | 6.9         |
| Aluno_A | Programação estruturada | 6.9         |
| Aluno_C | Programação estruturada | 4.7         |
| Aluno_E | Programação estruturada | 6.9         |
| Aluno_F | Programação estruturada | 5.7         |
| Aluno_A | Redes de computadores   | 6.7         |
| Aluno_B | Redes de computadores   | 5.9         |
| Aluno_C | Redes de computadores   | 4.9         |
| Aluno_D | Redes de computadores   | 5.0         |
| Aluno_E | Redes de computadores   | 5.5         |
| Aluno_A | LFA                     | 5.9         |
| Aluno_C | LFA                     | 6.7         |
| Aluno_D | LFA                     | 6.6         |

Fonte: livro texto

31

## Funções (FUNCTION): outros comandos

- exibir todas as funções desenvolvidas

```
SHOW FUNCTION STATUS;
```

- exibir a estrutura de uma função

```
SHOW CREATE FUNCTION nome_da_funcao;
```

- excluir uma função

```
DROP FUNCTION nome_da_funcao;
```

32

## Procedimento armazenado (PROCEDURE)

- Permitir armazenar procedimentos como seleção de dados, exclusão de registros, alteração de dados, entre outras funções.

```
CREATE PROCEDURE nome_do_procedure (var_nome
tipo)
    Declarações.
```

```
SHOW PROCEDURE STATUS;
```

```
DROP PROCEDURE nome_do_procedure;
```

33

- Exemplo:
- Considere a tabela

```
mysql> select * from Notas;
```

| AlunoId | DisciplinaId | NotaP1 | NotaP2 |
|---------|--------------|--------|--------|
| 1234    | 1            | 7.0    | 5.5    |
| 1235    | 1            | 7.0    | 5.5    |
| 1236    | 1            | 6.0    | 8.5    |
| 1237    | 1            | 5.0    | 3.5    |
| 1238    | 1            | 2.5    | 3.5    |
| 1239    | 1            | 9.0    | 5.5    |
| 1234    | 2            | 6.0    | 7.5    |
| 1235    | 2            | 6.5    | 8.5    |
| 1236    | 2            | 5.0    | 4.5    |
| 1237    | 2            | 7.5    | 6.5    |
| 1238    | 2            | 6.0    | 5.5    |
| 1239    | 2            | 8.5    | 5.5    |
| 1234    | 3            | 7.5    | 3.5    |
| 1235    | 3            | 7.0    | 3.5    |
| 1236    | 3            | 2.0    | 7.0    |
| 1237    | 3            | 2.5    | 9.5    |
| 1238    | 3            | 5.0    | 6.5    |
| 1239    | 3            | 7.5    | 7.5    |
| 1234    | 4            | 7.0    | 2.5    |
| 1235    | 4            | 6.0    | 7.0    |
| 1236    | 4            | 4.5    | 2.5    |
| 1237    | 4            | 2.0    | 2.5    |

Fonte: livro texto

34

- Exemplo: Calcular a média geral de todos os alunos que estão de exame em cada uma das disciplinas

```
CREATE PROCEDURE proc_MediaExame (var_DisciplinaId
int)
    SELECT AVG(fn_media(NotaP1, NotaP2)) AS "Média
Exame"
    FROM Notas
    WHERE DisciplinaId = var_DisciplinaId
        AND fn_media(NotaP1, NotaP2) >= 4.0
        AND fn_media(NotaP1, NotaP2) <= 6.9;
```

35

- Exemplo: resultado...

```
CALL proc_MediaExame(1);
```

```
CALL proc_MediaExame(2);
```

| Média Exame |
|-------------|
| 5.00000     |

| Média Exame |
|-------------|
| 6.05000     |

Fonte: livro texto

```
CALL proc_MediaExame(3);
```

```
CALL proc_MediaExame(4);
```

| Média Exame           |
|-----------------------|
| 5.60000               |
| row in set (0.02 sec) |

| Média Exame |
|-------------|
| 6.40000     |

Fonte: livro texto

36

## Resolução de SP

- Você trabalha na prefeitura de uma cidade litorânea, e seu projeto atual envolve o controle de passeios de barcos para as ilhas próximas à cidade.
- Criar um ponto de venda de passagens para os passeios.

| Field     | Type    | Null | Key | Default | Extra          |
|-----------|---------|------|-----|---------|----------------|
| Numero    | int(11) | NO   | PRI | NULL    | auto_increment |
| DestinoId | int(11) | NO   | MUL | NULL    |                |
| Embarque  | date    | NO   |     | NULL    |                |
| Qtd       | int(11) | NO   |     | NULL    |                |

Fonte: livro texto

37

## Resolução de SP

- Alter table destino add column Valor decimal(5,2);
- Insert

```
mysql> select * from destino;
```

| Id | Nome                | Valor  |
|----|---------------------|--------|
| 1  | Ilha Dourada        | 100.00 |
| 2  | Ilha D'areia fina   | 120.00 |
| 3  | Ilha Encantada      | 80.00  |
| 4  | Ilha dos Ventos     | 90.00  |
| 5  | Pequena Ilha do Mar | 100.00 |
| 6  | Ilha Torta          | 150.00 |
| 7  | Ilha dos Sonhos     | 120.00 |
| 8  | Ilha do Sono        | 180.00 |

Fonte: livro texto

38

## Descrição da SP

- Desenvolver uma solução no banco de dados para digitar o numero da venda já efetuada e retornar o valor que devera ser pago, respeitando o desconto de 30% em baixa temporada.

39

## Recapitulando

40

## Recapitulando

- Visões e índices
  - VIEW e INDEX
- Controle transacional
  - COMMIT, ROLLBACK e SAVEPOINT
- Procedimentos e funções
  - FUNCTION e PROCEDURE (STORED PROCEDURE)

41