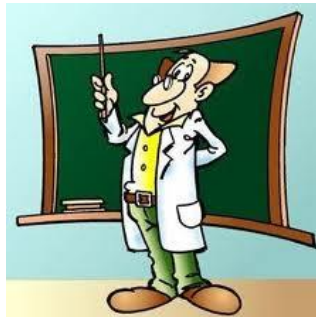




Unidade 22 – MySQL Views



Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP
aparecidovfreitas@gmail.com

Bibliografia

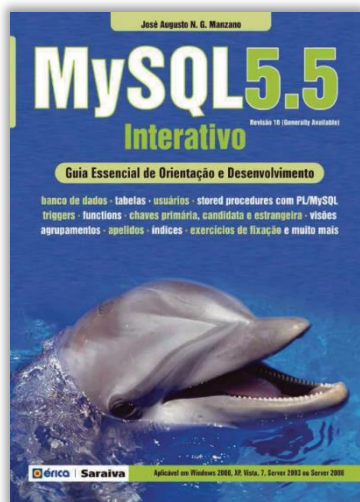
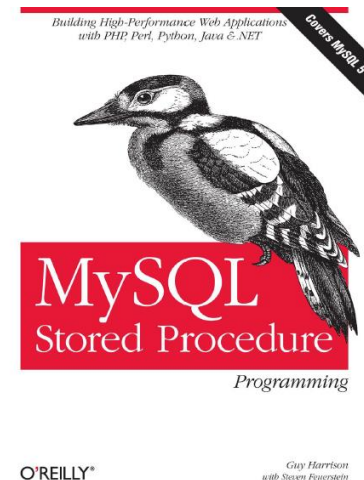
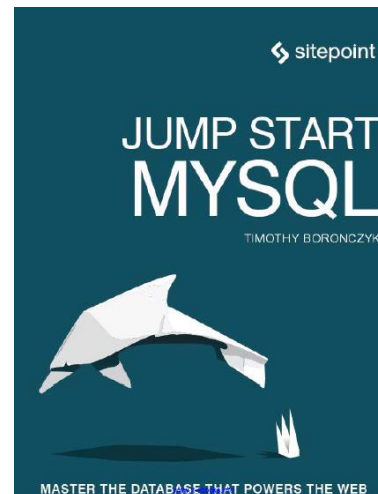
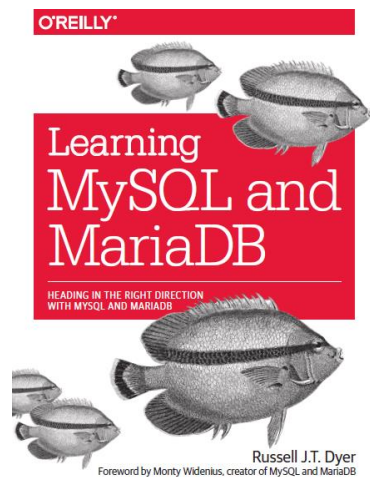
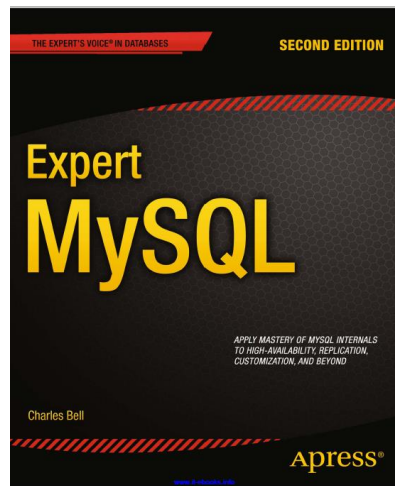
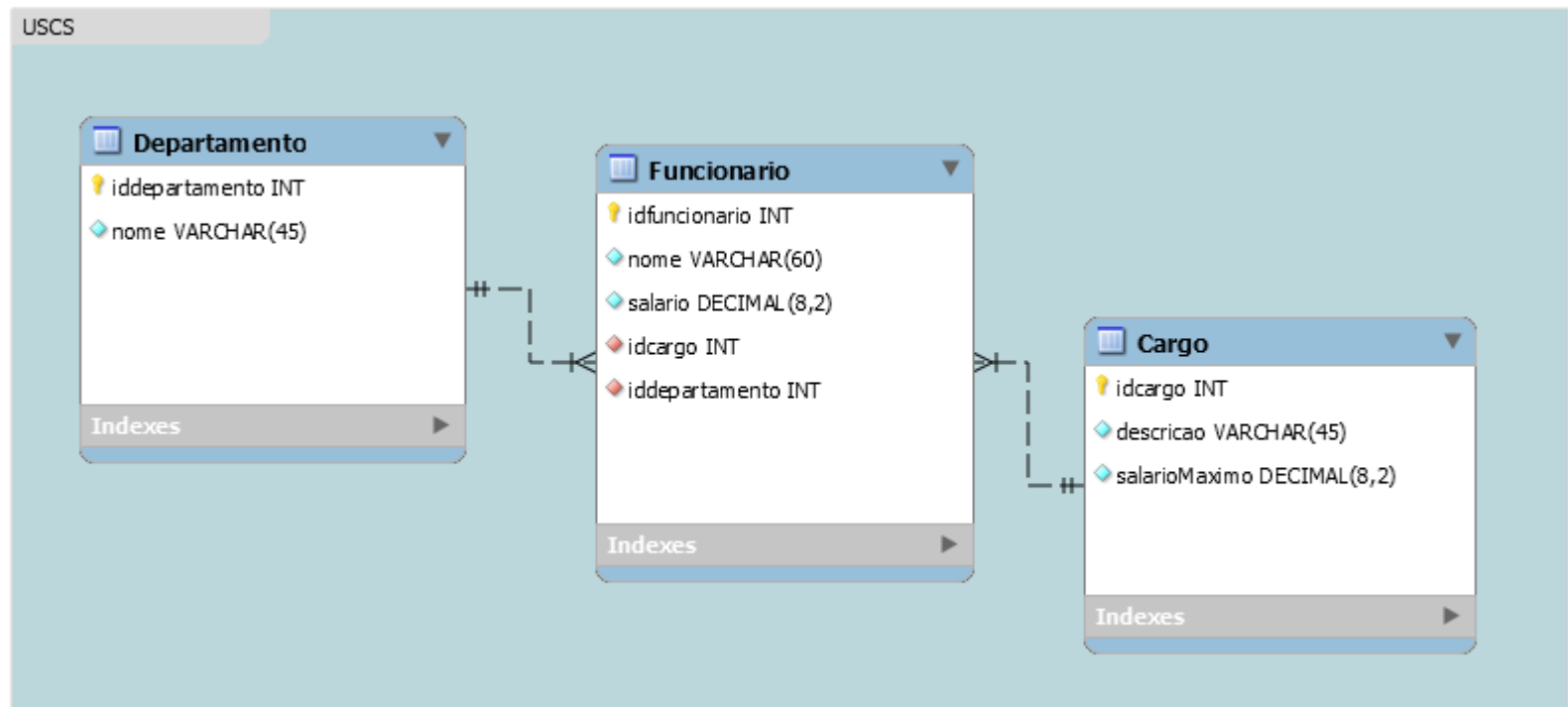


Diagrama Entidade Relacionamento

Este é o diagrama entidade relacionamento utilizado para essa unidade!



Views

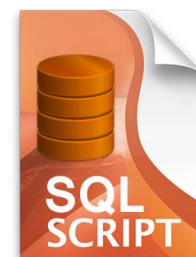
```
CREATE DATABASE funcionario;

USE funcionario;

CREATE TABLE Cargo (
    idcargo INT(11) NOT NULL AUTO_INCREMENT,
    descricao VARCHAR(45) NOT NULL,
    salarioMaximo DECIMAL(8,2) NOT NULL,
    PRIMARY KEY (idcargo));

CREATE TABLE Departamento (
    iddepartamento INT(11) NOT NULL AUTO_INCREMENT,
    nome VARCHAR(45) NOT NULL,
    PRIMARY KEY (iddepartamento));

CREATE TABLE Funcionario (
    idfuncionario INT(11) NOT NULL AUTO_INCREMENT,
    nome VARCHAR(60) NOT NULL,
    salario DECIMAL(8,2) NOT NULL,
    idcargo INT(11) NOT NULL,
    iddepartamento INT(11) NOT NULL,
    PRIMARY KEY (idfuncionario),
    INDEX fk_Funcionario_Cargo_idx (idcargo ASC),
    INDEX fk_Funcionario_Departamento1_idx (iddepartamento ASC),
    CONSTRAINT fk_Funcionario_Cargo FOREIGN KEY (idcargo)
        REFERENCES Cargo (idcargo),
    CONSTRAINT fk_Funcionario_Departamento1 FOREIGN KEY (iddepartamento)
        REFERENCES Departamento (iddepartamento));
```

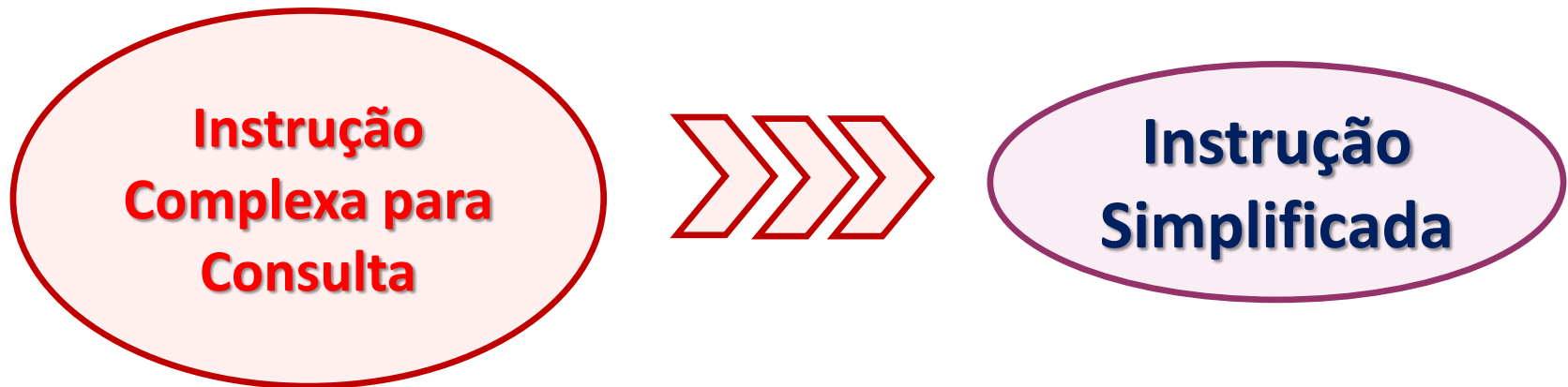


FuncionarioCriacaoDB.sql

Views

Definição

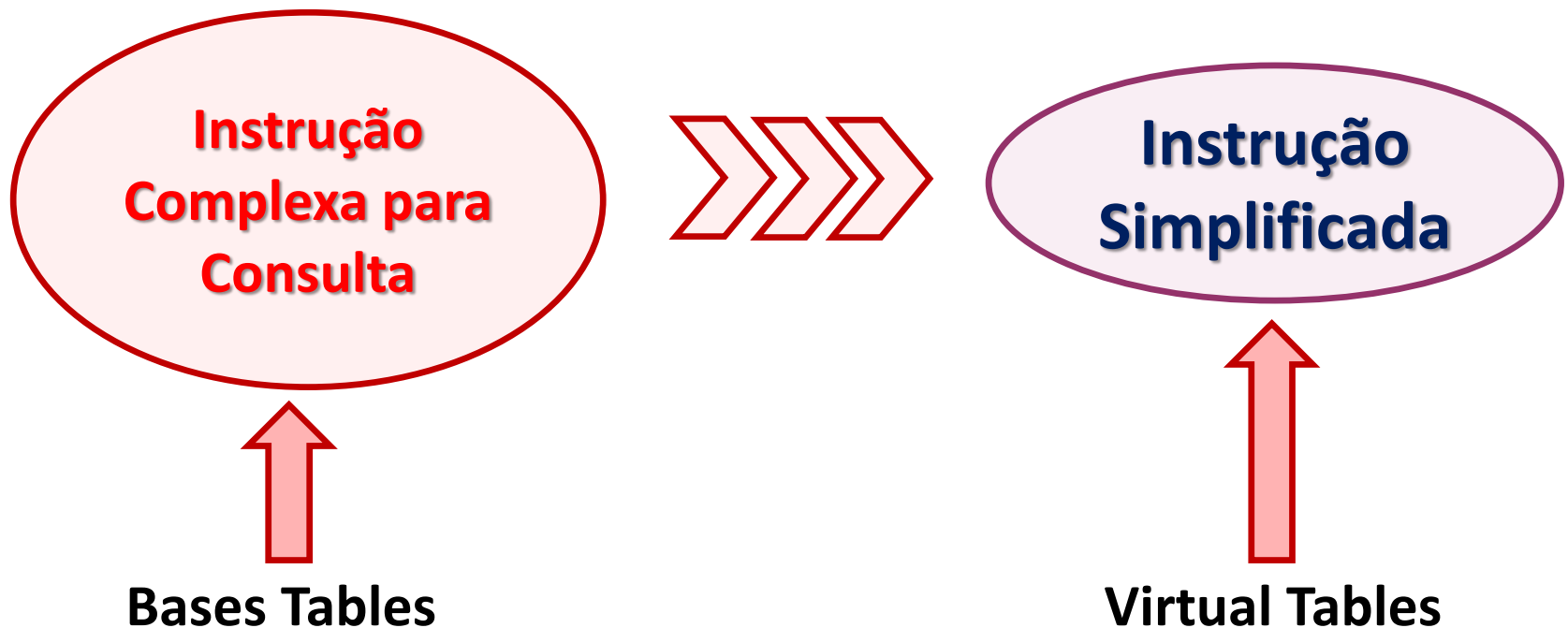
Objeto que pertence a um banco de dados, definida e baseada em uma declaração **SELECT**, retornando uma determinada visualização de dados de uma ou mais tabelas.



Views

Definição

Esses objetos são denominados “**virtual tables**”, formadas a partir de outras tabelas que por sua vez são chamadas de “**based tables**” ou ainda outras Views.



Views

Definição

Com base em configurações, as Views são atualizáveis e podem ser alvos de declaração INSERT, UPDATE e DELETE, que na verdade modificam sua(s) *“based tables”*.



Views

- ✓ Uma View pode ser utilizada, por exemplo, para retornar um valor baseado em um identificador de registro (**WHERE**);
- ✓ Pode ser utilizada para promover restrições em dados para aumentar a segurança dos mesmos e definir políticas de acesso em nível de tabela e coluna.
- ✓ Podem ser configurados para mostrar colunas diferentes para diferentes usuários do banco de dados;
- ✓ Em MySQL, as views são materializadas no momento da consulta.

Views

- ✓ Visualizar dados corresponde à criação de tabelas virtuais, a partir de uma tabela real existente;
- ✓ As tabelas virtuais se parecem com as reais, mas não são as tabelas reais em si; apenas uma composição em forma de consulta predefinida a partir de uma tabela real.
- ✓ As tabelas reais possuem os dados cadastrados e por esta razão ocupam espaço em disco, já as virtuais possuem apenas as **referências** de acesso à consulta das tabelas reais, e por assim dizer, não ocupam espaço em disco.
- ✓ As visualizações agilizam as operações de consulta, uma vez que concentram em cada tabela virtual os campos que realmente interessam.
- ✓ As visões podem ocorrer a partir de uma única tabela, de múltiplas tabelas ou mesmo a partir de outras visões já constituídas.

Views

Para definir Views em um banco de dados, utilize a declaração **CREATE VIEW**, a qual tem a seguinte sintaxe:

```
CREATE [OR REPLACE]
    VIEW view_name [(column_list)]
    AS select_statement
```

Views

Relação de Tabelas

Como sabemos o comando **SHOW TABLES** é responsável por exibir as tabelas de um determinado banco de dados.

```
SHOW TABLES;
```

	Tables_in_funcionario_views
►	cargo
	departamento
	funcionario

Criação de Views

Todos os itens entre [] são opcionais, logo iremos definir uma view de forma mais simples possível.

```
CREATE OR REPLACE VIEW vw_CargosAltoValor_001
AS SELECT c.idCargo, c.descricao, c.salarioMaximo
FROM cargo c
WHERE c.salarioMaximo >= 6000;
```

```
CREATE [OR REPLACE] [ALGORITHM = algorithm_type]
VIEW view_name [(column_list)]
AS select_statement
[WITH [CASCADED | LOCAL] CHECK OPTION]
```

SHOW TABLES;

	Tables_in_funcionario_views
►	cargo
	departamento
	funcionario
	vw_cargosaltovalor_001

Views

Criação de Views

```
CREATE OR REPLACE VIEW vw_CargosAltoValor_001
AS SELECT c.idCargo, c.descricao, c.salarioMaximo
FROM cargo c
WHERE c.salarioMaximo >= 6000;
```

SHOW TABLES;

	Tables_in_funcionario_views
►	cargo
	departamento
	funcionario
	vw_cargosaltovalor_001

Views

Representação de uma View

Vamos comparar o resultado de uma consulta simples entre a tabela base e a tabela virtual.

```
SELECT * FROM cargo;
```

	idcargo	descricao	salarioMaximo
▶	8	Estagiário (Penúltimo Ano)	1479.00
	9	Estagiário (Último Ano)	1636.00
	7	Técnico de Informática	2438.00
	6	Analista de Suporte Técnico	4564.00
	4	Analista de Segurança de Sistemas	5827.00
	3	Analista de Sistemas	6882.00
	2	Administrador de Rede	7087.00
	5	Coordenador de Suporte Técnico	8981.00
	1	Coordenador de TI	10954.00

```
SELECT * FROM vw_CargosAltoValor_001;
```

	idCargo	descricao	salarioMaximo
▶	3	Analista de Sistemas	6882.00
	2	Administrador de Rede	7087.00
	5	Coordenador de Suporte Técnico	8981.00
	1	Coordenador de TI	10954.00

Views

Utilização de uma View

Em princípio todos os recursos disponíveis para manipulação de uma tabela também estão disponíveis em uma View.

```
SELECT vw1.idCargo, vw1.descricao, vw1.salarioMaximo
FROM vw_CargosAltoValor_001 vw1
WHERE vw1.salarioMaximo >= 8500;
```

	idCargo	descricao	salarioMaximo
▶	1	Coordenador de TI	10954.00
	5	Coordenador de Suporte Técnico	8981.00

Views

Inserção através de Views

Verifique as seguintes instruções. Observe que ao executar estas instruções nenhum erro é apresentado.

```
INSERT INTO vw_CargosAltoValor_001  
(idcargo, descricao, salarioMaximo)  
VALUES (10, 'Especialista - SCRUM', 7500);
```

```
INSERT INTO vw_CargosAltoValor_001  
(idcargo, descricao, salarioMaximo)  
VALUES (11, 'Especialista - XP', 5800);
```


Views

Inserção através de Views

Porém ao consultar os registros da nossa View o registro 11 não é exibido. Por quê?

```
SELECT * FROM vw_CargosAltoValor_001;
```

	idCargo	descricao	salarioMaximo
▶	1	Coordenador de TI	10954.00
	2	Administrador de Rede	7087.00
	3	Analista de Sistemas	6882.00
	5	Coordenador de Suporte Técnico	8981.00
	10	Especialista - SCRUM	7500.00



Views

Inserção através de Views

Já na tabela base o registro é localizado.

```
SELECT * FROM cargo;
```

	idcargo	descricao	salarioMaximo
►	1	Coordenador de TI	10954.00
	2	Administrador de Rede	7087.00
	3	Analista de Sistemas	6882.00
	4	Analista de Segurança de Sistemas	5827.00
	5	Coordenador de Suporte Técnico	8981.00
	6	Analista de Suporte Técnico	4564.00
	7	Técnico de Informática	2438.00
	8	Estagiário (Penúltimo Ano)	1479.00
	9	Estagiário (Último Ano)	1636.00
	10	Especialista - SCRUM	7500.00
	11	Especialista - XP	5800.00



Views

Atualização através de Views

Verifique a seguinte instrução:

```
UPDATE vw_CargosAltoValor_001  
SET salarioMaximo = 4750  
WHERE idcargo = 11;  
/*Porque o registro não foi atualizado?*/
```

Views

Atualização e Deleção

Justifique cada uma das instruções e seus resultados.

```
UPDATE vw_CargosAltoValor_001  
SET salarioMaximo = 7890  
WHERE idcargo = 10;
```

→ Ok

```
DELETE FROM vw_CargosAltoValor_001  
WHERE idcargo = 10;
```

→ Ok

```
DELETE FROM vw_CargosAltoValor_001  
WHERE idcargo = 11;
```

→ Exclusão não realizada

```
DELETE FROM cargo  
WHERE idcargo = 11;
```

→ Ok

Views

Exclusão de uma View

Similar a outras estruturas a VIEW deve ser excluída através do comando DROP.

```
DROP VIEW vw_CargosAltoValor_001;
```

Views

Exercício

Desenvolva uma view com nome igual a **vw_FuncDepart1e2** que relacione o código do funcionário, nome do funcionário, código do departamento e nome do departamento de todos os funcionários dos departamentos identificados pelo código 1 e 2.

Views

Exercício *** RESOLUÇÃO ***

Desenvolva uma view com nome igual a **vw_FuncDepart1e2** que relacione o código do funcionário, nome do funcionário, código do departamento e nome do departamento de todos os funcionários dos departamentos identificados pelo código 1 e 2.

```
CREATE VIEW vw_FuncDepart1e2
AS
    SELECT f.idfuncionario, f.nome,
           f.iddepartamento, d.nome NomeDepartamento
    FROM funcionario f
    INNER JOIN departamento d USING(iddepartamento)
    WHERE f.iddepartamento IN (1,2);
```