

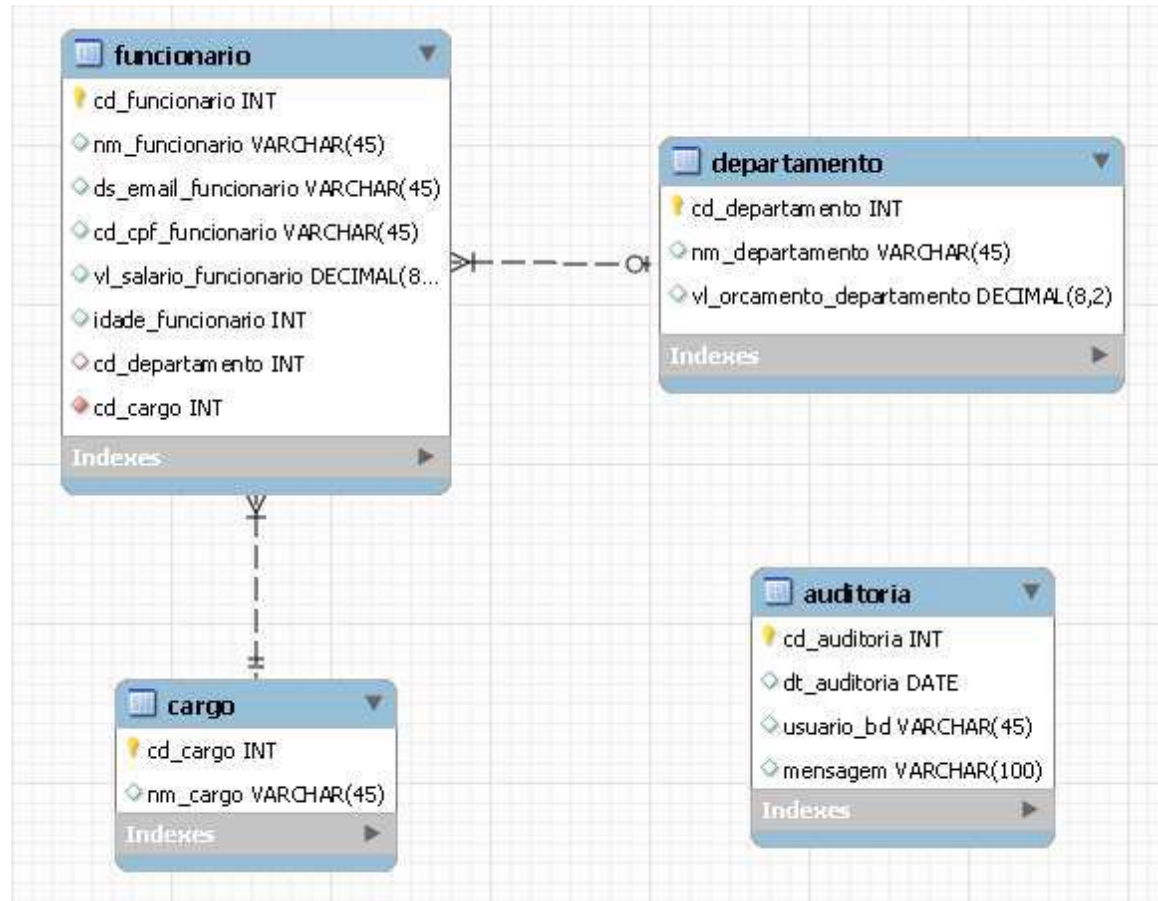
Linguagem de Programação de Banco de Dados

Transações

Carlos & Genivaldo

UNIP – Cidade Universitária

- Para o laboratório vamos implementar o seguinte banco de dados:



```
CREATE DATABASE UNIP_LPBD;  
USE UNIP_LPBD;  
SET SQL_SAFE_UPDATES = 0; -- Usar somente para MySQL/MySQL Workbench
```

```
-- -----  
-- Table unip_lpbd.departamento  
-- -----  
CREATE TABLE IF NOT EXISTS unip_lpbd.departamento (  
    cd_departamento INT NOT NULL AUTO_INCREMENT,  
    nm_departamento VARCHAR(45) NULL,  
    vl_orcamento_departamento DECIMAL(8,2) NULL,  
    dt_criacao DATETIME NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (cd_departamento))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table unip_lpbd.cargo  
-- -----  
CREATE TABLE IF NOT EXISTS unip_lpbd.cargo (  
    cd_cargo INT NOT NULL AUTO_INCREMENT,  
    nm_cargo VARCHAR(45) NULL,  
    dt_criacao DATETIME NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (cd_cargo))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table unip_lpbd.funcionario  
-- -----  
CREATE TABLE IF NOT EXISTS unip_lpbd.funcionario (  
    cd_funcionario INT NOT NULL AUTO_INCREMENT,  
    nm_funcionario VARCHAR(45) NULL,  
    ds_email_funcionario VARCHAR(45) NULL,  
    cd_cpf_funcionario VARCHAR(45) NULL,  
    vl_salario_funcionario DECIMAL(8,2) NULL,  
    idade_funcionario INT NULL,  
    cd_departamento INT NULL,  
    cd_cargo INT NOT NULL,  
    dt_criacao DATETIME NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (cd_funcionario),  
    UNIQUE INDEX cd_cpf_funcionario_UNIQUE (cd_cpf_funcionario ASC),  
    INDEX fk_funcionario_departamento_idx (cd_departamento ASC),  
    INDEX fk_funcionario_cargo1_idx (cd_cargo ASC),  
    CONSTRAINT fk_funcionario_departamento  
        FOREIGN KEY (cd_departamento)  
        REFERENCES unip_lpbd.departamento (cd_departamento)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
    CONSTRAINT fk_funcionario_cargo1  
        FOREIGN KEY (cd_cargo)  
        REFERENCES unip_lpbd.cargo (cd_cargo)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table unip_lpbd.auditoria  
-- -----  
CREATE TABLE IF NOT EXISTS unip_lpbd.auditoria (  
    cd_auditoria INT NOT NULL AUTO_INCREMENT,  
    dt_auditoria DATE NULL,  
    usuario_bd VARCHAR(45) NULL,  
    mensagem VARCHAR(100) NULL,  
    PRIMARY KEY (cd_auditoria))  
ENGINE = InnoDB;
```

```
INSERT INTO departamento (nm_departamento, vl_orcamento_departamento)
VALUES
('Desenvolvimento', 100000),
('Produto', 200000),
('Serviços', 100000),
('Diretoria', 500000);

INSERT INTO cargo (nm_cargo)
VALUES
('Analista de Sistemas'),
('Vendedor Técnico'),
('Motorista'),
('VP');

INSERT INTO funcionario (nm_funcionario, ds_email_funcionario, cd_cpf_funcionario, vl_salario_funcionario,
idade_funcionario, cd_departamento, cd_cargo)
VALUES
('Mario', 'mario.quinello@docente.unip.br', 11111111111, '20000', 44, 1, 1),
('Fernando', 'fernando@fernando.com.br', 22222222222, '10000', 24, 3, 1),
('Otavio', 'otavio@otavio.com.br', 33333333333, '30000', 35, 4, 2),
('Lauro', 'lauro@lauro.com.br', 44444444444, '30000', 30, 4, 3),
('Gumerindo', 'gumerindo@gumerindo.com.br', 55555555555, '20000', 50, 1, 3),
('Roberto', 'roberto@roberto.com.br', 66666666666, '20000', 50, 1, 4),
('Claudio', 'claudio@claudio.com.br', 77777777777, '20000', 50, 1, 4);
```


Para que seja possível utilizar o conceito de transações, no MySQL é preciso desabilitar o Commit Automático (autocommit); vamos verificar qual o status deste recurso:

```
SELECT @@AUTOCOMMIT;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	@@AUTOCOMMIT			
1				

O valor “1” indica que o recurso está ativo; vamos desativá-lo utilizando o seguinte SQL:

```
SET @@AUTOCOMMIT = 0;
```

Verifique novamente se o Commit Automático está desabilitado agora:

```
SELECT @@AUTOCOMMIT;
```







Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	@@AUTOCOMMIT			
	0			

Nosso primeiro teste será através de uma transaction para garantir o rollback de um UPDATE na tabela funcionário;

```
START TRANSACTION;  
UPDATE funcionario SET vl_salario_funcionario = '12000' WHERE cd_funcionario =1;
```

Neste momento estamos com uma transação em execução; verifique se a alteração anterior foi acatada na tabela funcionário:

```
SELECT * FROM funcionario WHERE cd_funcionario =1;
```

Result Grid									
Filter Rows: <input type="text"/>									
Edit:   									
Export/Import:  									
Wrap Cell Content: 									
	cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao
	1	Mario	mario.quinello@docent...	11111111111	12000.00	44	1	1	2020-09-23 ...
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Para desfazer este UPDATE execute o comando abaixo:

```
ROLLBACK;
```

Result Grid									
Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:			
cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao	
1	Mario	mario.quinello@docent...	11111111111	20000.00	44	1	1	2020-09-23 ...	
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Agora vamos repetir o processo, porém, desta vez vamos persistindo a alteração através do COMMIT:

```
START TRANSACTION;  
UPDATE funcionario SET vl_salario_funcionario = '12000' WHERE cd_funcionario =1;  
COMMIT;
```

Neste momento a transação foi finalizada e a alteração esta salva no banco de dados:

```
SELECT * FROM funcionario WHERE cd_funcionario =1;
```









	cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao
1	1	Mario	mario.quinello@docent...	11111111111	12000.00	44	1	1	2020-09-23 ...
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Perceba que mesmo que você tente executar o ROLLBACK, esta informação não é mais passível de retorno:

```
ROLLBACK;
```

```
SELECT * FROM funcionario WHERE cd_funcionario =1;
```

Result Grid									
Filter Rows: <input type="text"/>									
Edit:    Export/Import:   Wrap Cell Content: 									
	cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao
	1	Mario	mario.quinello@docent...	11111111111	12000.00	44	1	1	2020-09-23 ...
+	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Agora vamos inserir um novo funcionário em nossa tabela dentro de uma transaction:

```
START TRANSACTION;
INSERT INTO funcionario (nm_funcionario, ds_email_funcionario, cd_cpf_funcionario, vl_salario_funcionario,
idade_funcionario, cd_departamento, cd_cargo)
VALUES
('Rodolfo', 'rodolfo@rodolfo.unip.br', 88888888888, '20000', 54, 1, 1);
```

Verifique se o registro foi inserido na tabela funcionário:

```
SELECT * FROM funcionario WHERE nm_funcionario like 'Rodolfo%';
```

cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao
8	Rodolfo	rodolfo@rodolfo.unip.br	88888888888	20000.00	54	1	1	2020-09-23 ...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Vamos fazer o ROLLBACK desta informação e verificar o resultado deste procedimento:

```
ROLLBACK;
```

```
SELECT * FROM funcionario WHERE nm_funcionario like 'Rodolfo%';
```



The screenshot shows a database application interface with a toolbar and a result grid. The toolbar includes buttons for 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The result grid has 10 columns: cd_funcionario, nm_funcionario, ds_email_funcionario, cd_cpf_funcionario, vl_salario_funcionario, idade_funcionario, cd_departamento, cd_cargo, and dt_criacao. All cells in the first row are NULL.

cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Repita o processo anterior, porém, agora faça o COMMIT da transação (INSERT):

```
START TRANSACTION;  
INSERT INTO funcionario (nm_funcionario, ds_email_funcionario, cd_cpf_funcionario, vl_salario_funcionario,  
idade_funcionario, cd_departamento, cd_cargo)  
VALUES  
('Rodolfo', 'rodolfo@rodolfo.unip.br', 88888888888, '20000', 54, 1, 1);  
COMMIT;
```

Verifique se o registro foi inserido na tabela funcionário:

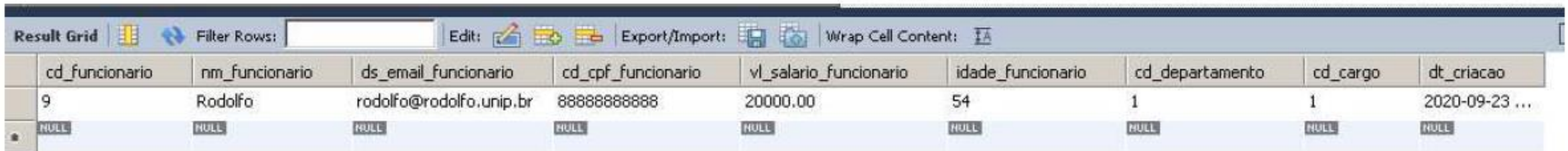
```
SELECT * FROM funcionario WHERE nm_funcionario like 'Rodolfo%';
```

Result Grid									
Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:			
cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao	
9	Rodolfo	rodolfo@rodolfo.unip.br	888888888888	20000.00	54	1	1	2020-09-23 ...	
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Tente retornar o INSERT efetuando um ROLLBACK; o registro está persistido e não é mais passível de retorno:

```
ROLLBACK;
```

```
SELECT * FROM funcionario WHERE nm_funcionario like 'Rodolfo%';
```



cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao
9	Rodolfo	rodolfo@rodolfo.unip.br	88888888888	20000.00	54	1	1	2020-09-23 ...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Agora vamos utilizar uma transaction para uma função DELETE; execute o comando SQL abaixo e verifique seu resultado:

```
START TRANSACTION;  
DELETE FROM funcionario WHERE nm_funcionario like 'Rodolfo%';
```

```
SELECT * FROM funcionario WHERE nm_funcionario like 'Rodolfo%';
```



The screenshot shows a database application interface with a toolbar and a result grid. The toolbar includes buttons for 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The result grid has 9 columns: cd_funcionario, nm_funcionario, ds_email_funcionario, cd_cpf_funcionario, vl_salario_funcionario, idade_funcionario, cd_departamento, cd_cargo, and dt_criacao. The first row of data shows all NULL values.

cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

O registro do funcionário Rodolfo foi removido; vamos tentar retornar este comando através do ROLLBACK:

```
ROLLBACK;
```

Perceba que o registro é devolvido a tabela:

Result Grid									
Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:			
cd_funcionario	nm_funcionario	ds_email_funcionario	cd_cpf_funcionario	vl_salario_funcionario	idade_funcionario	cd_departamento	cd_cargo	dt_criacao	
9	Rodolfo	rodolfo@rodolfo.unip.br	888888888888	20000.00	54	1	1	2020-09-23 ...	
• NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Com base no modelo utilizado neste tutorial, reporte o resultado dos exercícios abaixo:

1. “Em uma transação, remova o campo `idade_funcionario` da tabela `funcionário`; feito isso, faça o rollback do procedimento e informe os resultados verificados.”
2. “Efetue um truncate na tabela `funcionarios` (deletar todos os registros armazenados); feito isso, efetue um rollback do procedimento e informe os resultados verificados.”

1. “Em uma transação, remova o campo idade_funcionario da tabela funcionario; feito isso, faça o rollback do procedimento e informe os resultados verificados.”

```
-- Exercício 1
SELECT * FROM funcionario;

START TRANSACTION;
ALTER TABLE funcionario DROP COLUMN idade_funcionario;

-- Verificando se o registro do funcionario foi inserido
SELECT * FROM funcionario;

-- Efetuar o Rollback da transação anterior
ROLLBACK;

-- Verificando se o registro do funcionario foi inserido
SELECT * FROM funcionario;
```


2. “Efetue um truncate na tabela funcionarios (deletar todos os registros armazenados); feito isso, efetue um rollback do procedimento e informe os resultados verificados.”

```
-- Exercício 2
START TRANSACTION;
TRUNCATE TABLE funcionario;

-- Verificando se o registro do funcionario foi inserido
SELECT * FROM funcionario;

-- Efetuar o Rollback da transação anterior
ROLLBACK;

-- Verificando se o registro do funcionario foi inserido
SELECT * FROM funcionario;
```

OBRIGADO