

Banco de Dados – TAREFA T4 – Parte A - Solução

Restrições de Integridade Referencial – MySQL

1. Iniciar o servidor de Banco de Dados **MySQL**.

2. Criar um banco de dados chamado **bd**.

```
CREATE DATABASE bd;
```

```
USE bd;
```

3. No banco de dados **bd**, criar uma tabela chamada **DEPARTAMENTO** com os seguintes atributos:

- **IDDEPTO**, integer, obrigatório.
- **NOMEDEPTO**, texto (30 caracteres), opcional.

```
CREATE TABLE DEPARTAMENTO (  
    IDDEPTO INTEGER NOT NULL ,  
    NOMEDEPTO VARCHAR (30) );
```

4. Executar o comando **MySQL** para descrever a estrutura da tabela **DEPARTAMENTO**:

```
DESCRIBE DEPARTAMENTO;
```

5. Executar o comando **MySQL** para **alterar** a estrutura da tabela **DEPARTAMENTO** e incluir uma regra de integridade de chave primária (**CONSTRAINT**). Empregar o campo **IDDEPTO** como chave primária da tabela. Nomear essa **CONSTRAINT** pelo nome **PK_DEPARTAMENTO**:

```
ALTER TABLE DEPARTAMENTO ADD CONSTRAINT PK_DEPARTAMENTO PRIMARY  
KEY(IDDEPTO);
```

6. Executar o comando **MySQL** para exibir a estrutura da tabela **DEPARTAMENTO**:

```
DESCRIBE DEPARTAMENTO
```

7. Inserir na tabela **DEPARTAMENTO**, as seguintes tuplas:

(10, 'Compras')
(20, 'Engenharia')
(40, 'Vendas')
(55, 'Financeiro')

INSERT INTO DEPARTAMENTO VALUES

**(10, 'Compras'),
(20, 'Engenharia'),
(40, 'Vendas'),
(55, 'Financeiro');**

8. Executar o comando SQL para listar as tuplas da tabela **DEPARTAMENTO**:

SELECT * FROM DEPARTAMENTO;

9. No banco de dados **bd**, criar uma tabela chamada **FUNCIONARIO** com os seguintes atributos:

- **IDFUNC**, integer, obrigatório;
- **NOMEFUNC**, texto (40 caracteres), obrigatório;
- **CPF**, texto (14 caracteres), obrigatório, chave única;
- **IDDEPTO** integer.

CREATE TABLE FUNCIONARIO (

**IDFUNC INTEGER NOT NULL ,
NOMEFUNC VARCHAR (40) NOT NULL,
CPF CHAR(14) NOT NULL UNIQUE,
IDDEPTO INTEGER,**

10. Executar o comando **MySQL** para **alterar** a estrutura da tabela **FUNCIONARIO** e incluir uma regra de integridade de chave primária (**CONSTRAINT**). Empregar o campo **IDFUNC** como chave primária da tabela. Nomear essa **CONSTRAINT** pelo nome **PK_FUNCIONARIO**:

```
ALTER TABLE FUNCIONARIO ADD CONSTRAINT PK_FUNCIONARIO PRIMARY KEY(IDFUNC);
```

11. Executar o comando MySQL para alterar a estrutura da tabela **FUNCIONARIO** e incluir uma regra de integridade de chave estrangeira (**CONSTRAINT**). Empregar o campo **IDDEPTO** como chave estrangeira referenciada da tabela **DEPARTAMENTO**. Nomear essa **CONSTRAINT** pelo nome **FK_FUNCIONARIO**. Especificar que um departamento não deve ser alterado nem deletado caso haja algum funcionário alocado no correspondente departamento;

```
ALTER TABLE FUNCIONARIO ADD CONSTRAINT FK_FUNCIONARIO FOREIGN KEY(IDDEPTO) REFERENCES DEPARTAMENTO(IDDEPTO) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

12. Executar o comando **MySQL** exibir a estrutura da tabela **FUNCIONARIO**:

```
DESCRIBE FUNCIONARIO;
```

13. Inserir as seguintes tuplas na tabela **FUNCIONARIO**:

```
(3456, 'Paulo de Souza Alves', '345.987.123-98',10 )
(9872, 'Jose da Silva', '987.243.098-01',10 )
(1890, 'Pedro Rangel de Souza', '112.872.340-81',NULL)
(4680, 'Angela Silva Medeiros', '567.982.045-27',40 )
```

```
INSERT INTO FUNCIONARIO VALUES
```

```
(3456, 'Paulo de Souza Alves', '345.987.123-98',10 ),
(9872, 'Jose da Silva', '987.243.098-01',10 ),
(1890, 'Pedro Rangel de Souza', '112.872.340-81',NULL),
(4680, 'Angela Silva Medeiros', '567.982.045-27',40 );
```

14. Executar o comando **SQL** para listar as tuplas da tabela **FUNCIONARIO**:

```
SELECT * FROM FUNCIONARIO;
```

15. Executar o comando **SQL** para deletar o Departamento **10**:

```
DELETE FROM DEPARTAMENTO WHERE idDepto = 10;
```

16. Caso tenha havido algum erro, anotar a mensagem e justificar o erro encontrado:

```
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key  
constraint fails ('db'.funcionario', CONSTRAINT;
```

17. Executar o comando **MySQL** para alterar a regra de integridade de chave estrangeira **FK_FUNCIONARIO**. Especificar que ao se deletar um departamento, os funcionários alocados devem ter código de departamento nulo;

```
ALTER TABLE FUNCIONARIO DROP FOREIGN KEY FK_FUNCIONARIO;
```

```
ALTER TABLE FUNCIONARIO ADD CONSTRAINT FK_FUNCIONARIO FOREIGN  
KEY(IDDEPTO) REFERENCES DEPARTAMENTO(IDDEPTO) ON DELETE SET NULL ON  
UPDATE SET NULL;
```

18. Executar o comando **SQL** para deletar o Departamento **10**:

```
DELETE FROM DEPARTAMENTO WHERE idDepto = 10;
```

19. Executar o comando **SQL** visualizar os registros das tabelas **DEPARTAMENTO** e **FUNCIONARIO**;

```
SELECT * FROM FUNCIONARIO;  
SELECT * FROM DEPARTAMENTO;
```

20. Deletar os dados das tabelas **DEPARTAMENTO** e **FUNCIONARIO**.

```
DELETE FROM FUNCIONARIO;  
DELETE FROM DEPARTAMENTO;
```

21. Reinsserir os dados das tabelas **DEPARTAMENTO** e **FUNCIONARIO**.

INSERT INTO DEPARTAMENTO VALUES

**(10, 'Compras'),
(20, 'Engenharia'),
(40, 'Vendas'),
(55, 'Financeiro');**

INSERT INTO FUNCIONARIO VALUES

**(3456, 'Paulo de Souza Alves', '345.987.123-98',10),
(9872, 'Jose da Silva', '987.243.098-01',10),
(1890, 'Pedro Rangel de Souza', '112.872.340-81',NULL),
(4680, 'Angela Silva Medeiros', '567.982.045-27',40);**

22. Executar o comando **MySQL** para alterar a regra de integridade de chave estrangeira **FK_FUNCIONARIO**. Especificar que ao se deletar um departamento no qual haja funcionários alocados, todos os funcionários também serão deletados juntamente com o departamento.

ALTER TABLE FUNCIONARIO DROP FOREIGN KEY FK_FUNCIONARIO;

**ALTER TABLE FUNCIONARIO ADD CONSTRAINT FK_FUNCIONARIO FOREIGN
KEY(IDDEPTO) REFERENCES DEPARTAMENTO(IDDEPTO) ON DELETE CASCADE ON
UPDATE CASCADE;**

23. Executar o comando **SQL** para deletar o Departamento **10**:

DELETE FROM DEPARTAMENTO WHERE idDepto = 10;

24. Executar o comando **SQL** visualizar os registros das tabelas **DEPARTAMENTO** e **FUNCIONARIO**;

**SELECT * FROM FUNCIONARIO;
SELECT * FROM DEPARTAMENTO;**

Banco de Dados – TAREFA T4 – Parte B - Solução

Restrições de Integridade Referencial – Oracle

1. Iniciar o servidor de Banco de Dados **Oracle**.
2. Criar uma tabela chamada **DEPARTAMENTO** com os seguintes atributos:
 - **IDDEPTO**, integer, obrigatório,
 - **NOMEDEPTO**, texto (30 caracteres), opcional.

```
CREATE TABLE DEPARTAMENTO (  
    IDDEPTO INTEGER NOT NULL ,  
    NOMEDEPTO VARCHAR (30) );
```

3. Executar o comando **Oracle** para descrever a estrutura da tabela **DEPARTAMENTO**:
DESCRIBE DEPARTAMENTO;
4. Executar o comando **SQL** para **alterar** a estrutura da tabela **DEPARTAMENTO** e incluir uma regra de integridade de chave primária (**CONSTRAINT**). Empregar o campo **IDDEPTO** como chave primária da tabela. Nomear essa **CONSTRAINT** pelo nome **PK_DEPARTAMENTO**:

```
ALTER TABLE DEPARTAMENTO ADD CONSTRAINT PK_DEPARTAMENTO PRIMARY  
KEY(IDDEPTO);
```

5. Executar o comando **Oracle** para exibir a estrutura da tabela **DEPARTAMENTO**:

```
DESCRIBE DEPARTAMENTO
```

6. Inserir na tabela **DEPARTAMENTO**, as seguintes tuplas:
 - (10, 'Compras')
 - (20, 'Engenharia')
 - (40, 'Vendas')
 - (55, 'Financeiro')

INSERT INTO DEPARTAMENTO VALUES

**(10, 'Compras'),
(20, 'Engenharia'),
(40, 'Vendas'),
(55, 'Financeiro');**

7. Executar o comando **SQL** para listar as tuplas da tabela **DEPARTAMENTO**:

SELECT * FROM DEPARTAMENTO;

8. Criar uma tabela chamada **FUNCIONARIO** com os seguintes atributos:

- **IDFUNC**, integer, obrigatório;
- **NOMEFUNC**, texto (40 caracteres), obrigatório;
- **CPF**, texto (14 caracteres), obrigatório, chave única;
- **IDDEPTO** integer ,

CREATE TABLE FUNCIONARIO (

**IDFUNC INTEGER NOT NULL ,
NOMEFUNC VARCHAR (40) NOT NULL,
CPF CHAR(14) NOT NULL UNIQUE,
IDDEPTO INTEGER,**

9. Executar o comando **SQL** para **alterar** a estrutura da tabela **FUNCIONARIO** e incluir uma regra de integridade de chave primária (**CONSTRAINT**). Empregar o campo **IDFUNC** como chave primária da tabela. Nomear essa **CONSTRAINT** pelo nome **PK_FUNCIONARIO**:

ALTER TABLE FUNCIONARIO ADD CONSTRAINT PK_FUNCIONARIO PRIMARY KEY(IDFUNC);

10. Executar o comando **SQL** para alterar a estrutura da tabela **FUNCIONARIO** e incluir uma regra de integridade de chave estrangeira (**CONSTRAINT**). Empregar o campo **IDDEPTO** como chave estrangeira referenciada da tabela **DEPARTAMENTO**. Nomear essa **CONSTRAINT** pelo nome **FK_FUNCIONARIO**. Especificar que um departamento não deve ser alterado nem deletado caso haja algum funcionário alocado no correspondente departamento;

ALTER TABLE FUNCIONARIO ADD CONSTRAINT FK_FUNCIONARIO FOREIGN KEY(IDDEPTO) REFERENCES DEPARTAMENTO(IDDEPTO);

11. Executar o comando **Oracle** exibir a estrutura da tabela **FUNCIONARIO**:

DESCRIBE FUNCIONARIO;

12. Inserir as seguintes tuplas na tabela **FUNCIONARIO**:

(3456, 'Paulo de Souza Alves', '345.987.123-98',10)
(9872, 'Jose da Silva', '987.243.098-01',10)
(1890, 'Pedro Rangel de Souza', '112.872.340-81',NULL)
(4680, 'Angela Silva Medeiros', '567.982.045-27',40)

INSERT INTO FUNCIONARIO VALUES

**(3456, 'Paulo de Souza Alves', '345.987.123-98',10),
(9872, 'Jose da Silva', '987.243.098-01',10),
(1890, 'Pedro Rangel de Souza', '112.872.340-81',NULL),
(4680, 'Angela Silva Medeiros', '567.982.045-27',40);**

13. Executar o comando **SQL** para listar as tuplas da tabela **FUNCIONARIO**:

SELECT * FROM FUNCIONARIO;

14. Executar o comando **SQL** para deletar o Departamento **10**:

DELETE FROM DEPARTAMENTO WHERE idDepto = 10;

15. Caso tenha havido algum erro, anotar a mensagem e justificar o erro encontrado:

ORA-02292: integrity constraint (SYSTEM.FK_FUNCIONARIO) violated – child record found

16. Executar o comando **Oracle** para alterar a regra de integridade de chave estrangeira **FK_FUNCIONARIO**. Especificar que ao se deletar um departamento, os funcionários a ele alocados deve ter código de departamento nulo.

```
ALTER TABLE FUNCIONARIO DROP CONSTRAINT FK_FUNCIONARIO;
```

```
ALTER TABLE FUNCIONARIO ADD CONSTRAINT FK_FUNCIONARIO FOREIGN  
KEY(IDDEPTO) REFERENCES DEPARTAMENTO(IDDEPTO) ON DELETE SET NULL ON;
```

17. Executar o comando **SQL** para deletar o Departamento **10**:

```
DELETE FROM DEPARTAMENTO WHERE idDepto = 10;
```

18. Executar o comando **SQL** visualizar os registros das tabelas **DEPARTAMENTO** e **FUNCIONARIO**;

```
SELECT * FROM FUNCIONARIO;
```

```
SELECT * FROM DEPARTAMENTO;
```

19. Deletar os dados das tabelas **DEPARTAMENTO** e **FUNCIONARIO**.

```
DELETE FROM FUNCIONARIO;
```

```
DELETE FROM DEPARTAMENTO;
```

20. Reinserir os dados das tabelas **DEPARTAMENTO** e **FUNCIONARIO**.

```
INSERT INTO DEPARTAMENTO VALUES
```

```
(10, 'Compras'),
```

```
(20, 'Engenharia'),
```

```
(40, 'Vendas'),
```

```
(55, 'Financeiro');
```

```
INSERT INTO FUNCIONARIO VALUES
```

```
(3456, 'Paulo de Souza Alves', '345.987.123-98',10 ),
```

```
(9872, 'Jose da Silva', '987.243.098-01',10 ),
```

```
(1890, 'Pedro Rangel de Souza', '112.872.340-81', NULL),  
(4680, 'Angela Silva Medeiros', '567.982.045-27', 40 );
```

21. Executar o comando **Oracle** para alterar a regra de integridade de chave estrangeira **FK_FUNCIONARIO**. Especificar que ao se deletar um departamento no qual haja funcionários alocados, todos os funcionários também serão deletados juntamente com o departamento.

```
ALTER TABLE FUNCIONARIO DROP CONSTRAINT FK_FUNCIONARIO;
```

```
ALTER TABLE FUNCIONARIO ADD CONSTRAINT FK_FUNCIONARIO FOREIGN  
KEY(IDDEPTO) REFERENCES DEPARTAMENTO(IDDEPTO) ON DELETE CASCADE ON  
UPDATE CASCADE;
```

22. Executar o comando **SQL** para deletar o Departamento **10**:

```
DELETE FROM DEPARTAMENTO WHERE idDepto = 10;
```

23. Executar o comando **SQL** visualizar os registros das tabelas **DEPARTAMENTO** e **FUNCIONARIO**;

```
SELECT * FROM FUNCIONARIO;
```

```
SELECT * FROM DEPARTAMENTO;
```

Banco de Dados – TAREFA T4 – Parte C - Solução

Restrições de Integridade Referencial – PostgreSQL

1. Iniciar o servidor de Banco de Dados **PostgreSQL**.

2. Criar um banco de dados chamado **bd**.

```
CREATE DATABASE bd;
```

```
\c bd;
```

3. No banco de dados **bd**, criar uma tabela chamada **DEPARTAMENTO** com os seguintes atributos:

- **IDDEPTO**, integer, obrigatório.
- **NOMEDEPTO**, texto (30 caracteres), opcional.

```
CREATE TABLE DEPARTAMENTO (  
    IDDEPTO INTEGER NOT NULL ,  
    NOMEDEPTO VARCHAR (30) );
```

4. Executar o comando **PostgreSQL** para descrever a estrutura da tabela **DEPARTAMENTO**:

```
\d DEPARTAMENTO;
```

5. Executar o comando **SQL** para **alterar** a estrutura da tabela **DEPARTAMENTO** e incluir uma regra de integridade de chave primária (**CONSTRAINT**). Empregar o campo **IDDEPTO** como chave primária da tabela. Nomear essa **CONSTRAINT** pelo nome **PK_DEPARTAMENTO**:

```
ALTER TABLE DEPARTAMENTO ADD CONSTRAINT PK_DEPARTAMENTO PRIMARY  
KEY(IDDEPTO);
```

6. Executar o comando **PostgreSQL** para exibir a estrutura da tabela **DEPARTAMENTO**:

```
\d DEPARTAMENTO;
```

7. Inserir na tabela **DEPARTAMENTO**, as seguintes tuplas:

(10, 'Compras')
(20, 'Engenharia')
(40, 'Vendas')
(55, 'Financeiro')

INSERT INTO DEPARTAMENTO VALUES

**(10, 'Compras'),
(20, 'Engenharia'),
(40, 'Vendas'),
(55, 'Financeiro');**

8. Executar o comando **SQL** para listar as tuplas da tabela **DEPARTAMENTO**:

SELECT * FROM DEPARTAMENTO;

9. No banco de dados **bd**, criar uma tabela chamada **FUNCIONARIO** com os seguintes atributos:

- **IDFUNC**, integer, obrigatório;
- **NOMEFUNC**, texto (40 caracteres), obrigatório;
- **CPF**, texto (14 caracteres), obrigatório, chave única;
- **IDDEPTO** integer.

CREATE TABLE FUNCIONARIO (

**IDFUNC INTEGER NOT NULL ,
NOMEFUNC VARCHAR (40) NOT NULL,
CPF CHAR(14) NOT NULL UNIQUE,
IDDEPTO INTEGER,**

10. Executar o comando **SQL** para **alterar** a estrutura da tabela **FUNCIONARIO** e incluir uma regra de integridade de chave primária (**CONSTRAINT**). Empregar o campo **IDFUNC** como chave primária da tabela. Nomear essa **CONSTRAINT** pelo nome **PK_FUNCIONARIO**:

```
ALTER TABLE FUNCIONARIO ADD CONSTRAINT PK_FUNCIONARIO PRIMARY KEY(IDFUNC);
```

11. Executar o comando **SQL** para alterar a estrutura da tabela **FUNCIONARIO** e incluir uma regra de integridade de chave estrangeira (**CONSTRAINT**). Empregar o campo **IDDEPTO** como chave estrangeira referenciada da tabela **DEPARTAMENTO**. Nomear essa **CONSTRAINT** pelo nome **FK_FUNCIONARIO**. Especificar que um departamento não deve ser alterado nem deletado caso haja algum funcionário alocado no correspondente departamento;

```
ALTER TABLE FUNCIONARIO ADD CONSTRAINT FK_FUNCIONARIO FOREIGN KEY(IDDEPTO) REFERENCES DEPARTAMENTO(IDDEPTO) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

12. Executar o comando **PostgreSQL** para exibir a estrutura da tabela **FUNCIONARIO**:

```
\d FUNCIONARIO;
```

13. Inserir as seguintes tuplas na tabela **FUNCIONARIO**:

```
(3456, 'Paulo de Souza Alves', '345.987.123-98',10 )
(9872, 'Jose da Silva', '987.243.098-01',10 )
(1890, 'Pedro Rangel de Souza', '112.872.340-81',NULL)
(4680, 'Angela Silva Medeiros', '567.982.045-27',40 )
```

```
INSERT INTO FUNCIONARIO VALUES
```

```
(3456, 'Paulo de Souza Alves', '345.987.123-98',10 ),
(9872, 'Jose da Silva', '987.243.098-01',10 ),
(1890, 'Pedro Rangel de Souza', '112.872.340-81',NULL),
(4680, 'Angela Silva Medeiros', '567.982.045-27',40 );
```

14. Executar o comando **SQL** para listar as tuplas da tabela **FUNCIONARIO**:

```
SELECT * FROM FUNCIONARIO;
```

15. Executar o comando **SQL** para deletar o Departamento **10**:

```
DELETE FROM DEPARTAMENTO WHERE idDepto = 10;
```

16. Caso tenha havido algum erro, anotar a mensagem e justificar o erro encontrado:

ERROR: update or delete on table "departamento" violates foreign key constraint

DETAIL: key (iddepto) = (10) is still referenced from table "funcionario".

17. Executar o comando **SQL** para alterar a regra de integridade de chave estrangeira **FK_FUNCIONARIO**. Especificar que ao se deletar um departamento, os funcionários alocados devem ter código de departamento nulo;

ALTER TABLE FUNCIONARIO DROP CONSTRAINT FK_FUNCIONARIO;

ALTER TABLE FUNCIONARIO ADD CONSTRAINT FK_FUNCIONARIO FOREIGN KEY(IDDEPTO) REFERENCES DEPARTAMENTO(IDDEPTO) ON DELETE SET NULL ON UPDATE SET NULL;

18. Executar o comando **SQL** para deletar o Departamento **10**:

DELETE FROM DEPARTAMENTO WHERE idDepto = 10;

19. Executar o comando **SQL** para visualizar os registros das tabelas **DEPARTAMENTO** e **FUNCIONARIO**;

SELECT * FROM FUNCIONARIO;

SELECT * FROM DEPARTAMENTO;

20. Deletar os dados das tabelas **DEPARTAMENTO** e **FUNCIONARIO**.

DELETE FROM FUNCIONARIO;

DELETE FROM DEPARTAMENTO;

21. Reinsérer os dados das tabelas **DEPARTAMENTO** e **FUNCIONARIO**.

INSERT INTO DEPARTAMENTO VALUES

**(10, 'Compras'),
(20, 'Engenharia'),
(40, 'Vendas'),
(55, 'Financeiro');**

INSERT INTO FUNCIONARIO VALUES

**(3456, 'Paulo de Souza Alves', '345.987.123-98',10),
(9872, 'Jose da Silva', '987.243.098-01',10),
(1890, 'Pedro Rangel de Souza', '112.872.340-81',NULL),
(4680, 'Angela Silva Medeiros', '567.982.045-27',40);**

22. Executar o comando **SQL** alterar a regra de integridade de chave estrangeira **FK_FUNCIONARIO**. Especificar que ao se deletar um departamento no qual haja funcionários alocados, todos os funcionários também serão deletados juntamente com o departamento.

ALTER TABLE FUNCIONARIO DROP CONSTRAINT FK_FUNCIONARIO;

**ALTER TABLE FUNCIONARIO ADD CONSTRAINT FK_FUNCIONARIO FOREIGN
KEY(IDDEPTO) REFERENCES DEPARTAMENTO(IDDEPTO) ON DELETE CASCADE;**

23. Executar o comando **SQL** para deletar o Departamento **10**:

DELETE FROM DEPARTAMENTO WHERE idDepto = 10;

24. Executar o comando **SQL** para visualizar os registros das tabelas **DEPARTAMENTO** e **FUNCIONARIO**;

**SELECT * FROM FUNCIONARIO;
SELECT * FROM DEPARTAMENTO;**