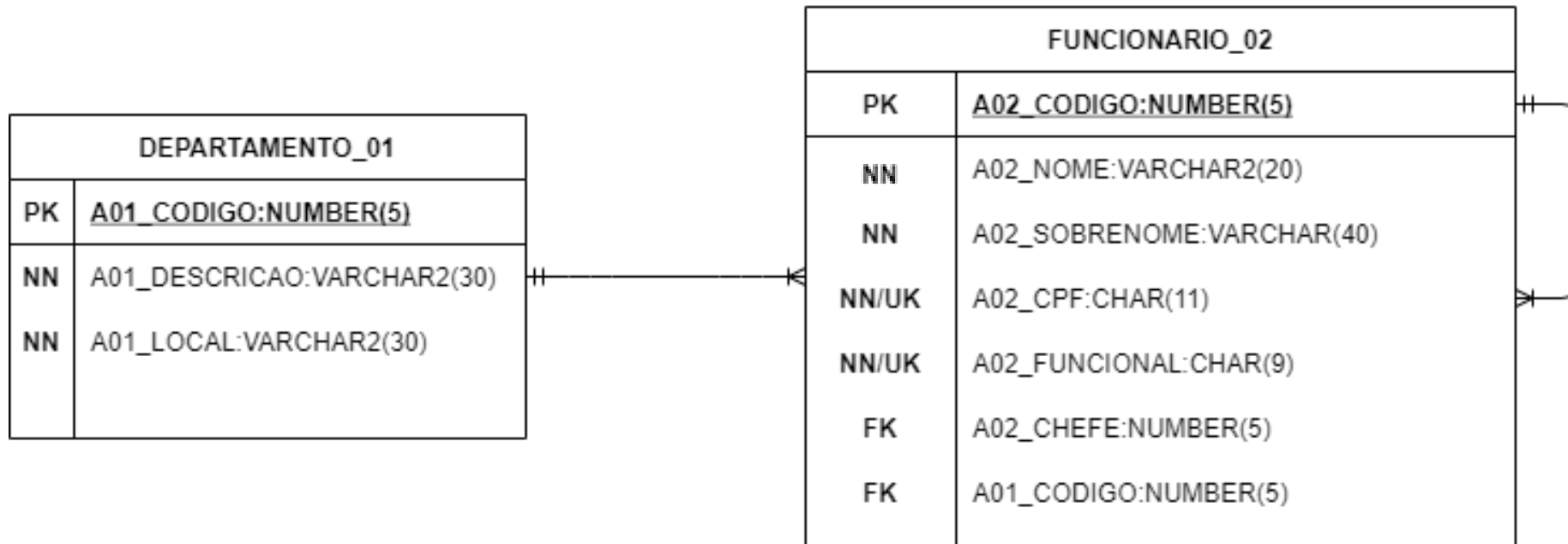


Prática de Banco de Dados

Carlos Baltazar & Henry Ungaro
Laboratório de Banco de Dados
UNIP

Uma empresa está informatizando seu registro de funcionários. A princípio, a empresa possui informações sobre cada funcionário como: nome; sobrenome; CPF; identidade funcional; responsável; departamento e local. Assim, implemente o banco de dados descrito na modelagem a seguir:

MER Físico



```
CREATE TABLE DEPARTAMENTO_01  
(  
    A01_CODIGO      NUMBER(5)    PRIMARY KEY,  
    A01_DESCRICAO   VARCHAR2(30) NOT NULL,  
    A01_LOCAL       VARCHAR2(30) NOT NULL  
);  
  
DESC DEPARTAMENTO_01;
```

```
CREATE TABLE FUNCIONARIO_02
(
    A02_CODIGO          NUMBER(5)          PRIMARY KEY,
    A02_NOME            VARCHAR2(20)       NOT NULL,
    A02_SOBRENOME       VARCHAR2(40)       NOT NULL,
    A02_CPF             CHAR(11)           UNIQUE NOT NULL,
    A02_FUNCIONAL       CHAR(9)            UNIQUE NOT NULL,
    A02_CHEFE           NUMBER(5),
    A01_CODIGO          NUMBER(5)          NOT NULL,
    FOREIGN KEY (A02_CHEFE) REFERENCES FUNCIONARIO_02 (A02_CODIGO),
    FOREIGN KEY (A01_CODIGO) REFERENCES DEPARTAMENTO_01 (A01_CODIGO)
);

DESC FUNCIONARIO_02;
```

Insira as tuplas descritas no projeto inicial do banco de dados:

Nome da tabela:	DEPARTAMENTO_01		
Nome do campo:	A01_CODIGO	A01_DESCRICAO	A01_LOCAL
Tipo de chave:	PK	-	-
Único/nulo:	UK,NN	NN	NN
Tipo de dado:	NUMBER(5)	VARCHAR2(30)	VARCHAR2(30)
Dados simples:	1	Presidência	23º andar
	2	Diretoria	10º andar
	3	R.H.	3º andar
	4	T.I.	Sub-solo

```
INSERT INTO DEPARTAMENTO_01 (A01_CODIGO, A01_DESCRICAO, A01_LOCAL)  
VALUES (1,'Presidência','23º andar');
```

```
INSERT INTO DEPARTAMENTO_01 (A01_CODIGO, A01_DESCRICAO, A01_LOCAL)  
VALUES (2,'Diretoria','10º andar');
```

```
INSERT INTO DEPARTAMENTO_01 (A01_CODIGO, A01_DESCRICAO, A01_LOCAL)  
VALUES (3,'R.H.','3º andar');
```

```
INSERT INTO DEPARTAMENTO_01 (A01_CODIGO, A01_DESCRICAO, A01_LOCAL)  
VALUES (4,'T.I.','subsolo');
```

```
SELECT * FROM DEPARTAMENTO_01;
```

Insira as tuplas descritas no projeto inicial do banco de dados:

Nome da tabela:	FUNCIONARIO_02						
Nome de campos:	A02_CODIGO	A02_NOME	A02_SOBRENOME	A02_CPF	A02_FUNCIONAL	A02_CHEFE	A01_CODIGO
Tipo de chave:	PK	-	-	-	-	FK	FK
Único/nulo:	NN	NN	NN	NN/UK	NN/UK	-	NN
Tipo de dado:	NUMBER(5)	Varchar2(20)	VARCHAR2(40)	CHAR(11)	Char(9)	Number(5)	Number(5)
Dados simples:	1	João	da Silva	12345678910	A23456789	-	1
	2	Maria	da Silva	23456789011	A34567890	1	2
	3	Renan	Souza	34567891212	B45678910	2	4
	4	Jose	dos Santos	45678912313	C56789123	2	3


```
INSERT INTO FUNCIONARIO_02 (A02_CODIGO, A02_NOME, A02_SOBRENOME, A02_CPF,  
A02_FUNCIONAL, A01_CODIGO) VALUES (1,'João','da Silva','12345678910','A23456789',1);  
  
INSERT INTO FUNCIONARIO_02 (A02_CODIGO, A02_NOME, A02_SOBRENOME, A02_CPF,  
A02_FUNCIONAL, A02_CHEFE, A01_CODIGO) VALUES (2,'Maria','da Silva','23456789011','A34567890',1,2);  
  
INSERT INTO FUNCIONARIO_02 (A02_CODIGO, A02_NOME, A02_SOBRENOME, A02_CPF,  
A02_FUNCIONAL, A02_CHEFE, A01_CODIGO) VALUES (3,'Renan','Souza','34567891212','B45678910',2,3);  
  
INSERT INTO FUNCIONARIO_02 (A02_CODIGO, A02_NOME, A02_SOBRENOME, A02_CPF,  
A02_FUNCIONAL, A02_CHEFE, A01_CODIGO) VALUES (4,'José','dos Santos','45678912313','C56789123',2,4);  
  
SELECT * FROM FUNCIONARIO_02;
```

Consultas SQL com condição (filtro) e operadores lógicos

```
SELECT <campos> FROM <tabela>  
WHERE condição1  
      AND condição2
```

```
SELECT  
  DISTINCT <campos da seleção>  
FROM <tabela>  
WHERE <condição da pesquisa>
```

```
SELECT <campos> FROM <tabela>  
WHERE condição1  
      OR condição2
```

Nas consultas SQL temos **Operadores Lógicos** que permitem refinar ainda mais os filtros aplicados.

AND - Retorna TRUE se ambas as condições forem verdadeiras

OR - Retorna TRUE se uma das condições for verdadeira

NOT - Retorna TRUE se a condição seguinte for falsa

Além disso temos também **Operadores de Comparação** para estipular condições que comparam uma expressão a outro valor ou expressão.

Operadores:

= Igual a

<> Diferente de

> Maior que

>= Maior ou igual a

< Menor que

<= Menor ou igual a

Selecionando a descrição e o local de todos os departamentos:

```
SELECT A01_DESCRICAO, A01_LOCAL  
FROM DEPARTAMENTO_01;
```

```
SELECT A01_DESCRICAO AS DEPARTAMENTO, A01_LOCAL AS ANDAR  
FROM DEPARTAMENTO_01;
```

Obs.: `SELECT * FROM DEPARTAMENTO_01` (seleciona todos os campos da tabela)

Selecionando somente a descrição e o local da Presidência:

```
SELECT A01_DESCRICAO, A01_LOCAL  
FROM DEPARTAMENTO_01  
WHERE A01_CODIGO = 1;
```

```
SELECT A01_DESCRICAO AS DEPARTAMENTO, A01_LOCAL AS ANDAR  
FROM DEPARTAMENTO_01  
WHERE A01_CODIGO = 1;
```

Obs.: Verifique também código > 1 , código < 1, código <> 1

Selecionando somente a descrição e o local da Presidência ou da TI:

```
SELECT A01_DESCRICAO, A01_LOCAL  
FROM DEPARTAMENTO_01  
WHERE A01_CODIGO = 1 OR A01_CODIGO = 4;
```

```
SELECT A01_DESCRICAO AS DEPARTAMENTO, A01_LOCAL AS ANDAR  
FROM DEPARTAMENTO_01  
WHERE A01_CODIGO = 1 OR A01_CODIGO = 4;
```

Obs.: uso do operador lógico (or)

Selecionando somente o nome e o sobrenome de todos os funcionários.

```
SELECT A02_NOME, A02_SOBRENOME FROM FUNCIONARIO_02;
```

```
SELECT A02_NOME AS NOME, A02_SOBRENOME AS SOBRENOME  
FROM FUNCIONARIO_02;
```

Obs.: `SELECT * FROM FUNCIONARIO_02` (seleciona todos os campos da tabela)

Selecionando somente o nome e o CPF de todos os funcionários.

```
SELECT A02_NOME, A02_CPF FROM FUNCIONARIO_02;
```

```
SELECT A02_NOME AS NOME, A02_CPF AS CPF  
FROM FUNCIONARIO_02;
```

Selecionando o nome, CPF e identidade funcional do funcionário Renan:

```
SELECT A02_NOME AS NOME, A02_CPF AS CPF, A02_FUNCIONAL AS  
FUNCIONAL  
FROM FUNCIONARIO_02  
WHERE A02_NOME = 'Renan';
```

Selecionando o nome, sobrenome e identidade funcional do funcionário com CPF '12345678910':

```
SELECT A02_NOME AS NOME, A02_SOBRENOME AS SOBRENOME,  
A02_FUNCIONAL AS FUNCIONAL  
FROM FUNCIONARIO_02  
WHERE A02_CPF = '12345678910';
```

Selecionando o nome, sobrenome e identidade funcional dos funcionários que trabalham no RH

```
SELECT A02_NOME AS NOME, A02_SOBRENOME AS SOBRENOME,  
A02_FUNCIONAL AS FUNCIONAL  
FROM FUNCIONARIO_02  
WHERE A01_CODIGO = 3;
```

Selecionando o nome, sobrenome e identidade funcional dos funcionários que trabalham no RH ou na área de TI

```
SELECT A02_NOME AS NOME, A02_SOBRENOME AS SOBRENOME,  
A02_FUNCIONAL AS FUNCIONAL  
FROM FUNCIONARIO_02  
WHERE A01_CODIGO = 3 OR A01_CODIGO = 4;
```

Obs.: uso de operador lógico (or)

Selecionando o nome, sobrenome e identidade funcional dos funcionários que trabalham no RH ou na área de TI

```
SELECT A02_NOME AS NOME, A02_SOBRENOME AS SOBRENOME,  
A02_FUNCIONAL AS FUNCIONAL  
FROM FUNCIONARIO_02  
WHERE A01_CODIGO IN (3 , 4);
```

Obs.: uso de operador IN – verifique também com a opção NOT IN (3,4)

Selecionando o nome, sobrenome e CPF dos funcionários que possuem sobrenome 'da Silva'

```
SELECT A02_NOME AS NOME, A02_SOBRENOME AS SOBRENOME,  
A02_CPF AS CPF  
FROM FUNCIONARIO_02  
WHERE A02_SOBRENOME = 'da Silva';
```

Selecionando o nome, sobrenome e CPF dos funcionários que possuem funcional A23456789 ou A34567890

```
SELECT A02_NOME AS NOME, A02_SOBRENOME AS SOBRENOME,  
A02_CPF AS CPF  
FROM FUNCIONARIO_02  
WHERE A02_FUNCIONAL = 'A23456789' OR A02_FUNCIONAL = 'A34567890';
```

Obs.: uso de operador lógico (or)

Comando de seleção agregando dados de duas tabelas diferentes:

```
SELECT APELIDO1.ATRIBUTO, APELIDO2.ATRIBUTO  
FROM TABELA1 APELIDO1, TABELA2 APELIDO2  
WHERE CONDICA0;
```

```
SELECT F.A02_NOME, F.A02_CPF, F.A02_FUNCIONAL, D.A01_DESCRICAO  
FROM DEPARTAMENTO_01 D, FUNCIONARIO_02 F  
WHERE F.A01_CODIGO = D.A01_CODIGO;
```

Obs.: tabela com apelido ou alias
agregando dados de duas tabelas diferentes

Comando de alteração de uma tupla (**UPDATE**)

```
UPDATE TABELA SET ATRIBUTO1 = VALOR1, ATRIBUTO2 = VALOR2  
WHERE CONDICAO;
```

Alterando o local do departamento de T.I. do subsolo para o 4º andar:

```
UPDATE DEPARTAMENTO_01 SET A01_LOCAL = '4º andar'  
WHERE A01_DESCRICAO = 'T.I.';
```

Comando de eliminação de uma tupla (**DELETE**):

```
DELETE FROM TABELA WHERE CONDICAO;
```

Excluindo todos os funcionários do departamento de T.I.

```
DELETE FROM FUNCIONARIO_02  
WHERE A01_CODIGO = 4;
```

Agora vamos fazer mais exercícios conforme exemplo abaixo:

Uma escola está informatizando seu registro de alunos e possui algumas informações sobre cada aluno como: nome; cidade; estado e email.

Assim, implemente o banco de dados descrito a seguir:

Criando a tabela de alunos

```
CREATE TABLE ALUNO_01
```

```
(  
  A01_CODIGO      NUMBER(3) PRIMARY KEY,  
  A01_NOME        varchar2(30) NOT NULL,  
  A01_CIDADE      varchar2(20) NOT NULL,  
  A01_ESTADO      char(2)      NOT NULL,  
  A01_EMAIL       varchar2(40)  
);
```

- INSERT INTO ALUNO_01 (A01_CODIGO, A01_NOME, A01_CIDADE, A01_ESTADO, A01_EMAIL) VALUES (1, 'JOSE VIEIRA DA SILVA', 'Sao Paulo', 'SP', 'jose@hotmail.com')
- INSERT INTO ALUNO_01 (A01_CODIGO, A01_NOME, A01_CIDADE, A01_ESTADO, A01_EMAIL) VALUES (2, 'MARISA DIAS DE OLIVEIRA', 'Rio de Janeiro', 'RJ', 'marisa@gmail.com')
- INSERT INTO ALUNO_01 (A01_CODIGO, A01_NOME, A01_CIDADE, A01_ESTADO, A01_EMAIL) VALUES (3, 'TALITA BASTOS DE SA', 'Santos', 'SP', 'tali@outlook.com.br')
- INSERT INTO ALUNO_01 (A01_CODIGO, A01_NOME, A01_CIDADE, A01_ESTADO, A01_EMAIL) VALUES (4, 'ALEXANDRA MENDES DE OLIVEIRA', 'Campinas', 'SP', 'ale22@gmail.com')

Insira mais dez registros na tabela ALUNO e realize as consultas SQL abaixo:

- ✓ Liste todos os registros da cidade de São Paulo
- ✓ Liste todos os registros com código do aluno maior que 5
- ✓ Liste todos os registros com nome igual a 'Alexandra' ou 'Mariana'

- ✓ Liste todos os registros do estado de São Paulo
- ✓ Liste todos os registros dos alunos dos estados de SP, RJ e MG
- ✓ Liste todos os registros com código do aluno maior que 8 e menor que 12
- ✓ Liste todos os registros do estado de São Paulo e que tenha código maior que 6

Modificações nos dados (update)

- ✓ Modifique o nome do aluno José Vieira para 'José Moreira Dantas da Silva'
- ✓ Modifique o estado da aluna Alexandra Mendes para 'MG'
- ✓ Modifique a cidade da aluna Talita Bastos para 'Araraquara'
- ✓ Modifique o email do aluno José Moreira para 'josemor@gmail.com'

Eliminação de dados (delete)

- ✓ Elimine os alunos da cidade de Campinas
- ✓ Elimine os alunos do estado de MG
- ✓ Elimine os alunos com código 3 e 4
- ✓ Elimie os alunos dos estados de RJ e SC

Agora crie mais tabelas e treine os comandos SQL com filtros e operadores

(exemplos: pagamentos, produtos, clientes, fornecedores, cursos, veículos)

Selecione o nome, CPF, identidade funcional, departamento e o nome do chefe para cada funcionário da empresa:

```
SELECT F.A02_NOME AS NOME, F.A02_CPF AS CPF, F.A02_FUNCIONAL,  
D.A01_DESCRICAO AS DEPARTAMENTO, (SELECT F2.A02_NOME FROM  
FUNCIONARIO_02 F2 WHERE F2.A02_CODIGO = F.A02_CHEFE) AS CHEFE  
FROM FUNCIONARIO_02 F, DEPARTAMENTO_01 D  
WHERE F.A01_CODIGO = D.A01_CODIGO;
```