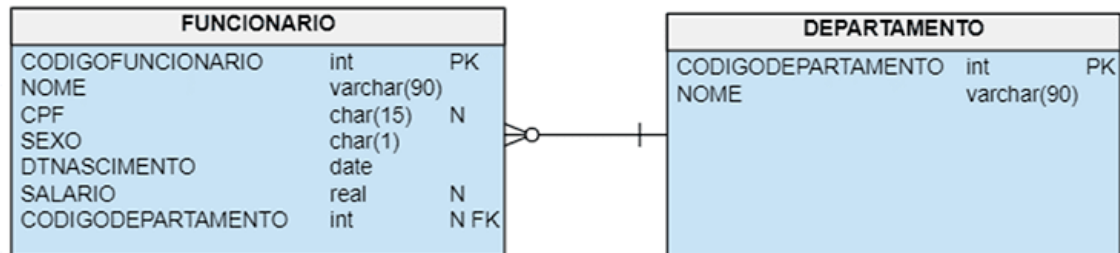




Operar subconsultas aninhadas e correlatas

SUBCONSULTAS

Vamos construir consultas que dependem dos - ou usam - resultados de outras consultas para recuperar informações de interesse. Essa categoria é conhecida por subconsulta, uma consulta sobre o resultado de outra consulta.



```
CREATE TABLE DEPARTAMENTO (  
    CODIGODEPARTAMENTO int NOT NULL,  
    NOME varchar(90) NOT NULL,  
    CONSTRAINT DEPARTAMENTO_pk PRIMARY KEY  
    (CODIGODEPARTAMENTO)  
);
```

```
CREATE TABLE FUNCIONARIO (  
    CODIGOFUNCIONARIO int NOT NULL,  
    NOME varchar(90) NOT NULL,  
    CPF char(15) NULL,  
    SEXO char(1) NOT NULL,  
    DTNASCIMENTO date NOT NULL,  
    SALARIO real NULL,  
    CODIGODEPARTAMENTO int NULL,  
    CONSTRAINT FUNCIONARIO_pk PRIMARY KEY  
    (CODIGOFUNCIONARIO)  
);
```

```
ALTER TABLE FUNCIONARIO ADD CONSTRAINT FUNCIONARIO_DEPARTAMENTO  
FOREIGN KEY (CODIGODEPARTAMENTO)  
REFERENCES DEPARTAMENTO (CODIGODEPARTAMENTO) ;
```

```
INSERT INTO DEPARTAMENTO(CODIGODEPARTAMENTO,NOME) VALUES(1,'Tecnologia  
da Informação');  
INSERT INTO DEPARTAMENTO(CODIGODEPARTAMENTO,NOME)  
VALUES(2,'Contabilidade');  
INSERT INTO DEPARTAMENTO(CODIGODEPARTAMENTO,NOME)  
VALUES(3,'Marketing');
```

```

INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF,
SEXO,DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO) VALUES (1,'ROBERTA
SILVA BRASIL',NULL,'F','20/02/1980',7000,1);
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF,
SEXO,DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO) VALUES (2,'MARIA SILVA
BRASIL',NULL,'F','20/09/1988',9500,2);
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF,
SEXO,DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO) VALUES (3,'GABRIELLA
PEREIRA LIMA',NULL,'F','20/02/1990',6000,1);
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF,
SEXO,DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO) VALUES (4,'MARCOS
PEREIRA BRASIL',NULL,'M','20/02/1999',6000,2);
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF,
SEXO,DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO) VALUES (5,'HEMERSON
SILVA BRASIL',NULL,'M','20/12/1992',4000,NULL);

```

```

SELECT * FROM DEPARTAMENTO;
SELECT * FROM FUNCIONARIO;

```

TABLE FUNCIONARIO; (equivalente a SELECT * FROM FUNCIONARIO;)

Em resumo a subconsulta é usada quando temos que obter um dado que não está disponível em determinada tabela.

Subconsultas aninhadas

Uma subconsulta aninhada ocorre quando é necessário obter dados que dependem do resultado de uma - ou mais - consulta(s) mais interna(s). Para isso, cria-se uma condição na cláusula WHERE de forma a envolver o resultado da subconsulta em algum tipo de teste.

Exemplos:

Consulta 01: Retornar o código e o nome do(s) funcionário(s) que ganha(m) o maior salário.

```

SELECT CODIGOFUNCIONARIO,NOME
FROM FUNCIONARIO
WHERE SALARIO= (
    SELECT MAX(SALARIO)
    FROM FUNCIONARIO);

```

Consulta 02: Retornar o código, o nome e o salário do(s) funcionário(s) que ganha(m) mais que a média salarial dos colaboradores.

```

SELECT CODIGOFUNCIONARIO,NOME,SALARIO
FROM FUNCIONARIO
WHERE SALARIO>(
    SELECT AVG(SALARIO)
    FROM FUNCIONARIO
);

```

Consulta 03: Retornar o código, o nome e o salário do(s) funcionário(s) que ganha(m) menos que a média salarial dos colaboradores do departamento de Tecnologia da Informação.

```
SELECT CODIGOFUNCIONARIO, NOME, SALARIO
FROM FUNCIONARIO
WHERE SALARIO < (

    SELECT AVG(SALARIO)
    FROM FUNCIONARIO
    WHERE CODIGODEPARTAMENTO IN (

        SELECT CODIGODEPARTAMENTO
        FROM DEPARTAMENTO
        WHERE NOME='Tecnologia da Informação'
    ));
```

Há outra maneira de resolver a consulta 03?

Sim, você pode trocar uma subconsulta por uma junção. O código a seguir produz os mesmos resultados:

```
SELECT CODIGOFUNCIONARIO, NOME, SALARIO
FROM FUNCIONARIO
WHERE SALARIO < (

    SELECT AVG(SALARIO)
    FROM FUNCIONARIO F JOIN DEPARTAMENTO D ON
    (F.CODIGODEPARTAMENTO=D.CODIGODEPARTAMENTO)
    WHERE D.NOME = 'Tecnologia da Informação');
```

Consulta 04: Quantos funcionários recebem menos que a funcionária que possui o maior salário entre as colaboradoras de sexo feminino?

Subconsultas correlatas

Uma subconsulta correlata – ou correlacionada – é um tipo especial de consulta aninhada. Uma consulta correlata ocorre quando é necessário obter dados que dependem do resultado de uma - ou mais - consulta(s) mais interna(s). Para isso, cria-se uma condição na cláusula WHERE de forma a envolver o resultado da subconsulta em algum tipo de teste.

Consulta 05: Retornar o código, o nome e o salário do(s) funcionário(s) que ganha(m) mais que a média salarial dos colaboradores do departamento ao qual pertencem.

```
SELECT CODIGOFUNCIONARIO,NOME,SALARIO
FROM FUNCIONARIO F
WHERE SALARIO>(
    SELECT AVG(SALARIO)
    FROM FUNCIONARIO
    WHERE CODIGODEPARTAMENTO=F.CODIGODEPARTAMENTO
```

);

Trata-se de uma consulta com lógica de construção semelhante ao que foi desenvolvido na consulta 02. No entanto, a média salarial é calculada levando em consideração somente os funcionários de cada departamento. Isso ocorre em função da condição WHERE declarada na linha 6.

Há outra maneira de resolver a consulta 05?

Sim, você pode trocar uma subconsulta por uma junção. O código a seguir produz os mesmos resultados:

```
SELECT CODIGOFUNCIONARIO,NOME,SALARIO
FROM FUNCIONARIO F
    JOIN (
        SELECT CODIGODEPARTAMENTO,AVG(SALARIO) AS MEDIA
        FROM FUNCIONARIO
        GROUP BY CODIGODEPARTAMENTO) TESTE
ON F.CODIGODEPARTAMENTO=TESTE.CODIGODEPARTAMENTO
WHERE SALARIO > MEDIA;
```

Consulta 06: suponha que surgiu a necessidade de equiparar os salários dos funcionários que atuam no mesmo departamento. Os funcionários de cada departamento terão salário atualizado em função do maior salário dos seus setores.

Perceba que a listagem está ordenada pela coluna CODIGODEPARTAMENTO. O maior salário de funcionário pertencente ao departamento 1 é R\$ 7.000; em relação ao departamento 2, R\$ 9.500. Note também que há um funcionário sem a informação sobre departamento.

```
UPDATE FUNCIONARIO F
SET SALARIO = (
    SELECT MAX(SALARIO)
    FROM FUNCIONARIO
    WHERE CODIGODEPARTAMENTO=F.CODIGODEPARTAMENTO
)
WHERE F.CODIGODEPARTAMENTO IS NOT NULL;
```

Consulta correlacionada com uso de [NOT] EXISTS

Podemos utilizar o operador EXISTS em uma consulta correlacionada. Tal operador testa a existência de linha(s) retornada(s) por alguma subconsulta.

Consulta 07: exibir o código e o nome do departamento onde há pelo menos um funcionário alocado.

```
SELECT D.CODIGODEPARTAMENTO, D.NOME
FROM DEPARTAMENTO D
WHERE EXISTS (
    SELECT F.CODIGODEPARTAMENTO
    FROM FUNCIONARIO F
    WHERE D.CODIGODEPARTAMENTO=F.CODIGODEPARTAMENTO
);
```

A subconsulta correlacionada (linhas 4 a 6) é executada. Caso haja pelo menos uma linha em seu retorno, a avaliação da cláusula WHERE retorna verdadeiro e as colunas especificadas na linha 1 são exibidas.

Finalmente, se estivéssemos interessados em saber se há departamento sem ocorrência de colaborador alocado, bastaria usar a negação (NOT), conforme a seguir:

```
SELECT D.CODIGODEPARTAMENTO, D.NOME
FROM DEPARTAMENTO D
WHERE NOT EXISTS (
    SELECT F.CODIGODEPARTAMENTO
    FROM FUNCIONARIO F
    WHERE D.CODIGODEPARTAMENTO=F.CODIGODEPARTAMENTO
);
```