

Universidade Presbiteriana Mackenzie



Banco de Dados – Aula 18 Linguagem SQL SELECT com consultas aninhadas

Profa. Elisângela Botelho Gracias

Faculdade de Computação e Informática

Consultas Aninhadas

- Uma **consulta aninhada** é aquela que tem outra consulta embutida dentro dela, sendo que a consulta embutida é chamada de **subconsulta**
- Uma **subconsulta** aparece tipicamente dentro da cláusula WHERE de uma consulta, mas pode aparecer também na cláusula SELECT, FROM ou HAVING
- Uma **subconsulta** pode conter outras **subconsultas** aninhadas

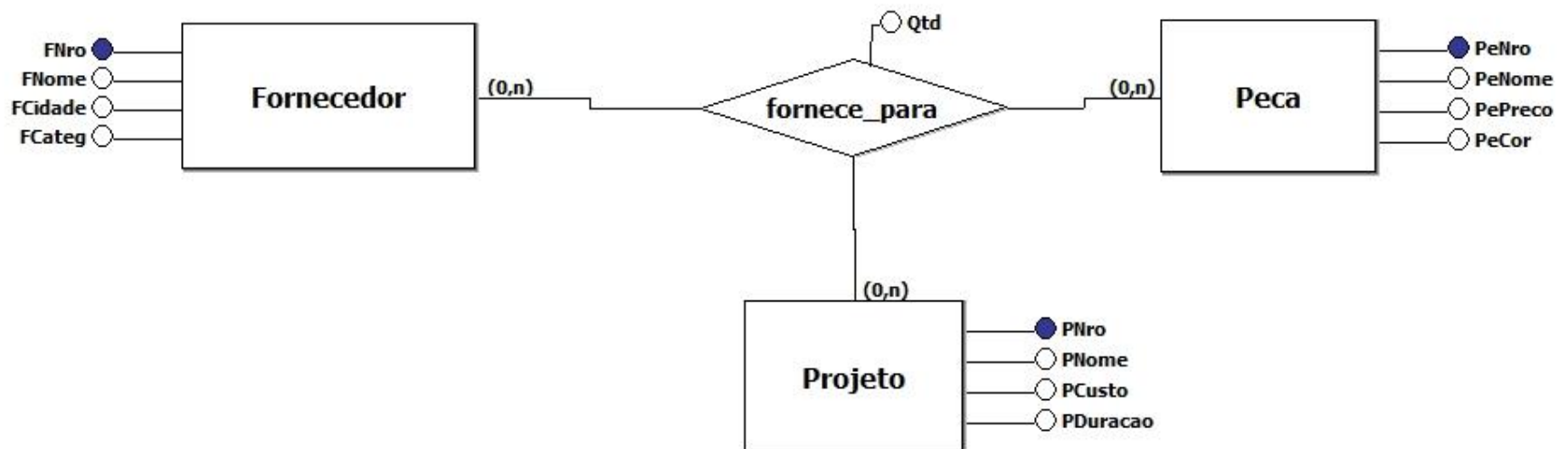
Consultas Aninhadas

- Uma **subconsulta** sempre irá retornar um valor ou um conjunto de valores para uma consulta aninhada
- Então, o que está sendo **retornado por uma subconsulta** tem que ser **compatível** com o que se está **comparando**
- Tente observar isso nos exemplos que serão mostrados, a seguir

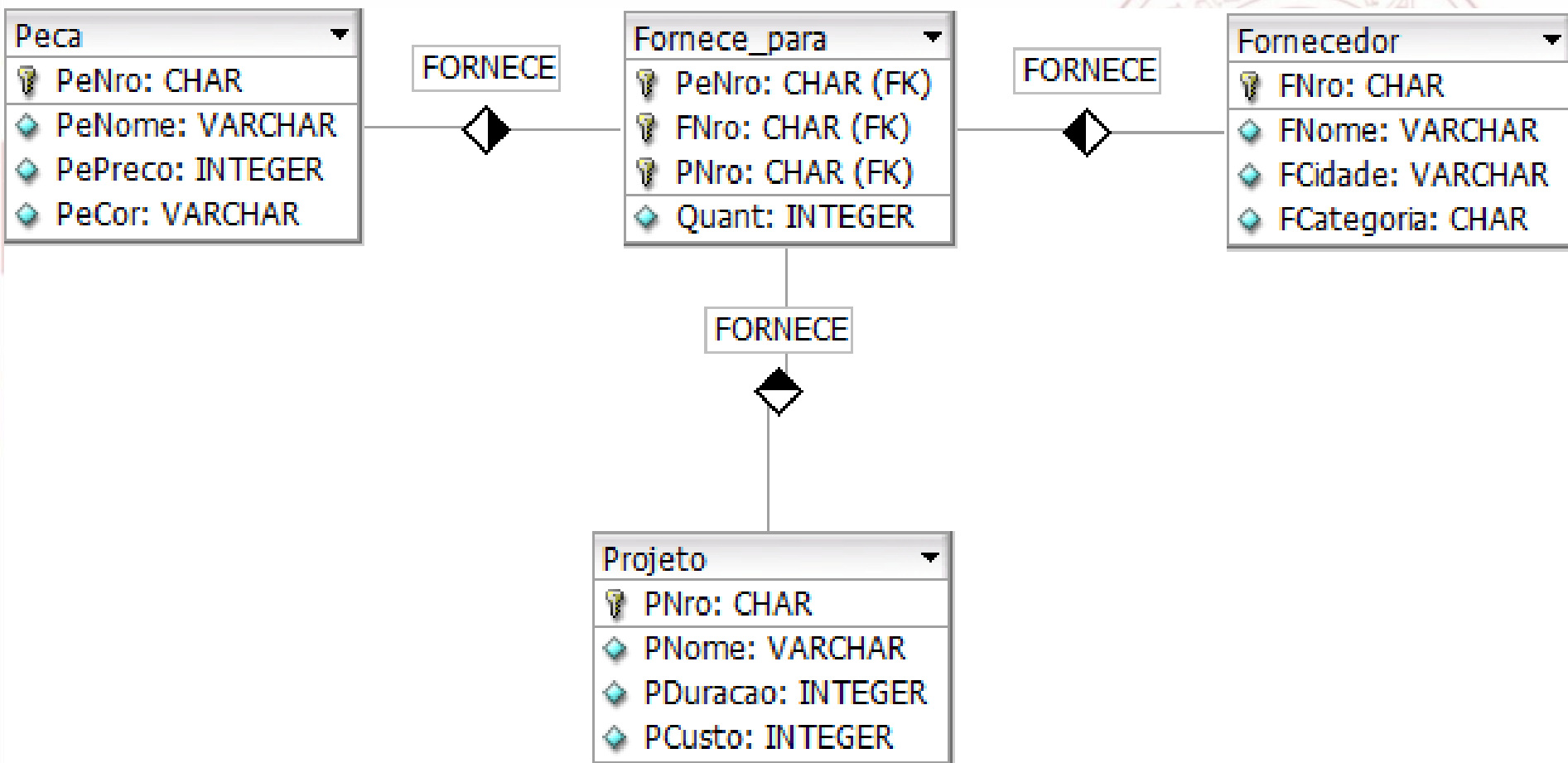


Banco de Dados Exemplo

Modelo Entidade-Relacionamento



Modelo Relacional



Exemplo de Banco de Dados

Considere o seguinte modelo relacional (chaves primárias estão sublinhadas:

PECA = {PeNro, PeNome, PePreco, PeCor}

FORNECEDOR = {FNro, FNome, FCidade, FCateg}

PROJETO = {PNro, PNome, PDuracao, PCusto}

FORNECE_PARA = {PeNro, FNro, PNro, Quant}

- PeNro é chave estrangeira que referencia a tabela Peca
- FNro é chave estrangeira que referencia a tabela Fornecedor
- PNro é chave estrangeira que referencia a tabela Projeto

Peca

PeNro	PeNome	PePreço	PeCor
PE1	Cinto	22	Azul
PE2	Volante	18	Vermelho
PE3	Lanterna	14	Preto
PE4	Limpador	09	Amarelo
PE5	Painel	43	Vermelho

Fornecedor

FNro	FNome	FCidade	FCateg
F1	Plastec	Campinas	B
F2	CM	São Paulo	D
F3	Kirurgic	Campinas	A
F4	Piloto	Piracicaba	A
F5	Equipament	São Carlos	C

Projeto

PNro	PNome	PDuração	PCusto
P1	Detroit	5	43000
P2	Pegasus	3	37000
P3	Alfa	2	26700
P4	Sea	3	21200
P5	Paraíso	1	17000

Fornece_para

PeNro	FNro	PNro	Quant
PE1	F5	P4	5
PE2	F2	P2	1
PE3	F3	P4	2
PE4	F4	P5	3
PE5	F1	P1	1
PE2	F2	P3	1
PE4	F3	P5	2

--Script de criação do Banco de Dados:

```
DROP TABLE Fornece_Para CASCADE CONSTRAINT;  
DROP TABLE Projeto CASCADE CONSTRAINT;  
DROP TABLE Fornecedor CASCADE CONSTRAINT;  
DROP TABLE Peca CASCADE CONSTRAINT;
```

```
CREATE TABLE Peca (  
  PeNro CHAR(4),  
  PeNome VARCHAR(30) NOT NULL,  
  PePreco INTEGER NOT NULL,  
  PeCor VARCHAR(20) NOT NULL,  
  PRIMARY KEY(PeNro));
```

```
CREATE TABLE Fornecedor (  
  FNro CHAR(4),  
  FNome VARCHAR(30) NOT NULL,  
  FCidade VARCHAR(30) NOT NULL,  
  FCategoria CHAR(1) NOT NULL,  
  PRIMARY KEY(FNro));
```

```
CREATE TABLE Projeto (  
  PNro CHAR(4),  
  PNome VARCHAR(30) NOT NULL,  
  PDuracao INTEGER NOT NULL,  
  PCusto INTEGER NOT NULL,  
  PRIMARY KEY(PNro));
```

```
CREATE TABLE Fornece_para (  
  PeNro CHAR(4),  
  FNro CHAR(4),  
  PNro CHAR(4),  
  Quant INTEGER,  
  PRIMARY KEY(PeNro,FNro,PNro),  
  FOREIGN KEY(PeNro) REFERENCES Peca(PeNro),  
  FOREIGN KEY(FNro) REFERENCES Fornecedor(FNro),  
  FOREIGN KEY(PNro) REFERENCES Projeto(PNro));
```

```
INSERT INTO Peca VALUES ('PE1', 'Cinto', 22, 'Azul');
INSERT INTO Peca VALUES ('PE2', 'Volante', 18, 'Vermelho');
INSERT INTO Peca VALUES ('PE3', 'Lanterna', 14, 'Preto');
INSERT INTO Peca VALUES ('PE4', 'Limpador', 9, 'Amarelo');
INSERT INTO Peca VALUES ('PE5', 'Painel', 43, 'Vermelho');

INSERT INTO Fornecedor VALUES ('F1', 'Plastec', 'Campinas', 'B');
INSERT INTO Fornecedor VALUES ('F2', 'CM', 'Sao Paulo', 'D');
INSERT INTO Fornecedor VALUES ('F3', 'Kirurgic', 'Campinas', 'A');
INSERT INTO Fornecedor VALUES ('F4', 'Piloto', 'Piracicaba', 'A');
INSERT INTO Fornecedor VALUES ('F5', 'Equipament', 'Sao Carlos', 'C');

INSERT INTO Projeto VALUES ('P1', 'Detroit', 5, 43000);
INSERT INTO Projeto VALUES ('P2', 'Pegasus', 3, 37000);
INSERT INTO Projeto VALUES ('P3', 'Alfa', 2, 26700);
INSERT INTO Projeto VALUES ('P4', 'Sea', 3, 21200);
INSERT INTO Projeto VALUES ('P5', 'Paraiso', 1, 17000);

INSERT INTO Fornece_para VALUES ('PE1', 'F5', 'P4', 5);
INSERT INTO Fornece_para VALUES ('PE2', 'F2', 'P2', 1);
INSERT INTO Fornece_para VALUES ('PE3', 'F3', 'P4', 2);
INSERT INTO Fornece_para VALUES ('PE4', 'F4', 'P5', 3);
INSERT INTO Fornece_para VALUES ('PE5', 'F1', 'P1', 1);
INSERT INTO Fornece_para VALUES ('PE2', 'F2', 'P3', 1);
INSERT INTO Fornece_para VALUES ('PE4', 'F3', 'P5', 2);
COMMIT;
```

Consultas Aninhadas

- Exemplo1 (subconsulta): Obtenha o nome das peças cujo preço é superior ao preço médio das peças.

SELECT PeNome

FROM Peca

WHERE PePreco > (**SELECT** **AVG**(PePreco)
FROM Peca);

PeNome
Cinto
Painel

Consultas Aninhadas

- Exemplo2 (subconsulta): Obtenha o nome da peça de menor preço.

SELECT PeNome

FROM Peca

WHERE PePreco = (**SELECT MIN**(PePreco)
FROM Peca);

PeNome
Limpador

Consultas Aninhadas

- Exemplo3 (subconsulta): Obtenha o nome das peças utilizadas no projeto P5.

```
SELECT PeNome  
FROM Peca  
WHERE PeNro IN (SELECT PeNro  
                FROM Fornece_para  
                WHERE (PNro = 'P5')  
                );
```

PeNome
Limpador

Consultas Aninhadas

- Observe que o Exemplo3 também poderia ser feito utilizando o INNER JOIN

SELECT DISTINCT PE.PeNome

FROM Peca PE **INNER JOIN** Fornece_para FP

ON (FP.PeNro = PE.PeNro)

WHERE (FP.PNro = 'P5');

Consultas Aninhadas

- Exemplo4 (subconsulta): Obtenha, sem repetição e em ordem crescente, o nome dos fornecedores que forneceram peças para algum projeto.

```
SELECT DISTINCT FNome  
FROM Fornecedor  
WHERE FNro IN (SELECT FNro  
                FROM Fornece_para)  
ORDER BY FNome ASC;
```

FNome
CM
Equipament
Kirurgic
Piloto
Plastec

Consultas Aninhadas

- Observe que o Exemplo4 também poderia ser feito utilizando o INNER JOIN

```
SELECT DISTINCT F.FNome  
FROM Fornece_para FP INNER JOIN Fornecedor F  
ON (FP.FNro = F.FNro)  
ORDER BY F.FNome ASC;
```

Consultas Aninhadas

Exemplo5 (subconsulta): Obtenha os nomes das peças utilizadas nos projetos com duração maior que 3 meses.

SELECT PeNome

FROM Peca

WHERE PeNro **IN** (**SELECT** PeNro

FROM Fornece_para

WHERE PNro **IN** (**SELECT** PNro

FROM Projeto

WHERE (PDuracao >3)
)

);

PeNome
Painel

Consultas Aninhadas

- Observe que o Exemplo5 também poderia ser feito utilizando o INNER JOIN

SELECT PE.PeNome

FROM Projeto P **INNER JOIN** Fornece_para FP

ON (P.PNro = FP.PNro)

INNER JOIN Peca PE

ON (PE.PeNro = FP.PeNro)

WHERE (P.PDuracao >3);

Consultas Aninhadas

- Exemplo6 (subconsulta): Obtenha os nomes das peças fornecidas por algum fornecedor de Piracicaba.

SELECT PeNome

FROM Peca

WHERE PeNro **IN** (**SELECT** PeNro

FROM Fornece_para

WHERE FNro **IN** (**SELECT** FNro

FROM Fornecedor

WHERE (FCidade = 'Piracicaba')

)

);

PeNome
Limpador

Consultas Aninhadas

- Observe que o Exemplo6 também poderia ser feito utilizando o INNER JOIN

```
SELECT PE.PeNome  
FROM Fornecedor F INNER JOIN Fornece_para FP  
ON (F.FNro = FP.FNro)  
INNER JOIN Peca PE  
ON (PE.PeNro = FP.PeNro)  
WHERE (F.FCidade = 'Piracicaba');
```


Consultas Aninhadas

- Exemplo7 (subconsulta): Obtenha os nomes das peças que não são fornecidas por fornecedores da categoria A.

```
SELECT PeNome
FROM Peca
WHERE PeNro IN (SELECT PeNro
                  FROM Fornece_para
                  WHERE FNro NOT IN (SELECT FNro
                                   FROM Fornecedor
                                   WHERE (FCategoria = 'A')
                                   )
                  );
```

PeNome
Cinto
Volante
Painel

Consultas Aninhadas

- Observe que o Exemplo7 também poderia ser feito utilizando o INNER JOIN

```
SELECT DISTINCT PE.PeNome  
FROM Fornecedor F INNER JOIN Fornece_para FP  
ON (F.FNro = FP.FNro)  
INNER JOIN Peca PE  
ON (PE.PeNro = FP.PeNro)  
WHERE (F.FCategoria <> 'A');
```

Consultas Aninhadas

- Exemplo8 (subconsulta): Obtenha o nome de cada projeto e o total de peças utilizadas em cada um, mas desde que esse total seja maior que o total de peças utilizadas no projeto 'Paraíso'.

```
SELECT P.PNome, SUM(FP.Quant) AS Total  
FROM Fornece_para FP INNER JOIN Projeto P  
ON (FP.PNro = P.PNro)  
GROUP BY P.PNome  
HAVING SUM(FP.Quant) > (SELECT SUM(FP.Quant)
```

```
FROM Fornece_para FP INNER JOIN Projeto P  
ON (FP.PNro = P.PNro)  
WHERE (P.PNome = 'Paraíso')  
);
```

PNome	Total
Sea	7

UPDATE e DELETE com subconsulta

- Uma subconsulta pode ser utilizada nos comandos UPDATE e DELETE e isso será necessário quando:
 - a atualização dos dados for feita em uma determinada tabela, mas os critérios para essa atualização envolvem dados de outras tabelas
 - a eliminação de dados for feita em uma determinada tabela, mas os critérios para essa eliminação envolvem dados de outras tabelas

UPDATE com subconsulta

- Exemplo9 (UPDATE com subconsulta): Atualize para 10 a quantidade da peça 'Volante' fornecida pelo fornecedor F2 para o projeto P2.

UPDATE Fornece_para

SET Quant = 10

WHERE (FNro = 'F2') **AND** (PNro = 'P2')

AND PeNro **IN** (**SELECT** PeNro

FROM Peca

WHERE (PeNome = 'Volante')

);

UPDATE com subconsulta

- Exemplo10 (UPDATE com subconsulta): Atualize a peça 'PE4' para que a sua cor seja igual a cor da peça 'PE3'.

UPDATE Peca

SET PeCor = (**SELECT** PeCor

FROM Peca

WHERE (PeNro = 'PE3')

)

WHERE (PeNro = 'PE4');

DELETE com subconsulta

- Exemplo11 (DELETE com subconsulta): Elimine da tabela Fornece_Para todas as informações referentes ao fornecedor de nome 'CM'.

DELETE

FROM Fornece_para

WHERE FNro IN (**SELECT** FNro

FROM Fornecedor

WHERE (FNome = 'CM')

);

UNION

- A operação **UNION** da linguagem SQL é a mesma operação de união da teoria de conjuntos e da Álgebra Relacional (como já foi visto em BD-I)
- A união de $R \cup S$ inclui todas as tuplas que estão em R ou em S , ou tanto em R quanto em S . Esta operação elimina as tuplas duplicadas
- Para utilizar o **UNION**, é necessário que as relações envolvidas tenham o **mesmo tipo de tupla**, ou seja, isso é a **compatibilidade de união**

UNION

- Exemplo (**UNION**): Obtenha o código das peças que possuem preço superior a 15 OU então o código daquelas peças que foram utilizadas em algum projeto em uma quantidade superior a 4.

```
SELECT PeNro  
FROM Peca  
WHERE (PePreco > 15)  
UNION  
SELECT PeNro  
FROM Fornece_para  
WHERE (Quant > 4);
```

PeNro
PE1
PE2
PE5

Obrigado

Profa. Elisângela Botelho Gracias
elisangela.botelho@mackenzie.br

