Universidade Presbiteriana Mackenzie

Banco de Dados – Aula 16

Linguagem SQL -

SELECT com várias tabelas

INNER JOIN

Profa. Elisângela Botelho Gracias

Faculdade de Computação e Informática

Introdução

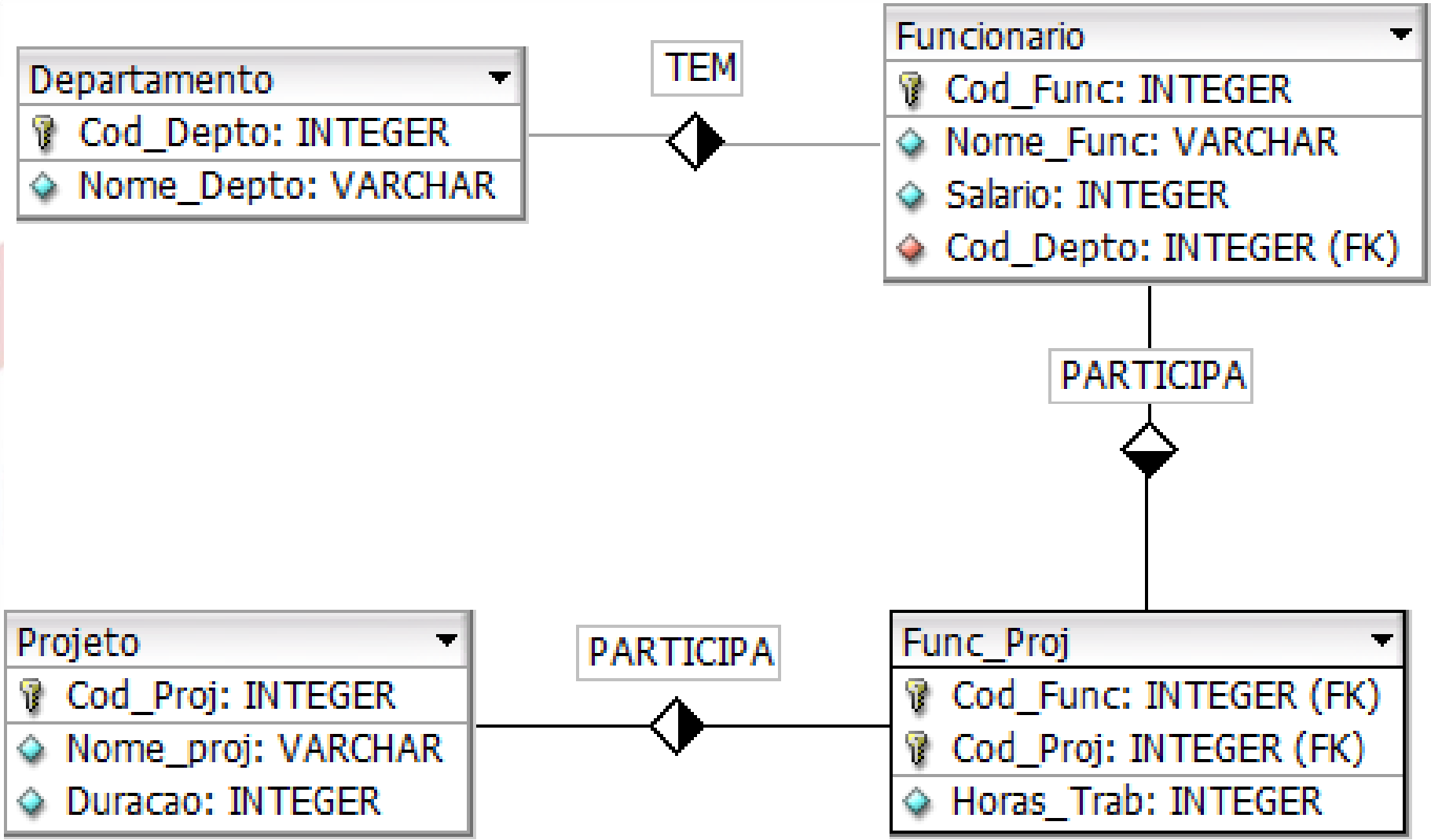
Considere o seguinte Banco de Dados para esta aula (chave primária está sublinhada)

- Departamento = {Cod_Depto, Nome_Depto}
- Funcionario = {Cod_Func, Nome_Func, Salario, Cod_Depto}

  - *Cod_depto é chave estrangeira que referencia o atributo Cod_depto da tabela Departamento*
- Projeto = {Cod_Proj, Nome_Proj, Duracao}
- Func_Proj = {Cod_Func, Cod_Proj, Horas_Trab}

  - *Cod_Func é chave estrangeira que referencia o atributo Cod_Func da tabela Funcionario*

  - *Cod_Proj é chave estrangeira que referencia o atributo Cod_Proj da tabela Projeto*


Departamento
Cod_Depto: INTEGER
Nome_Depto: VARCHAR
TEM
Funcionario
Cod_Func: INTEGER
Nome_Func: VARCHAR
Salario: INTEGER
Cod_Depto: INTEGER (FK)
PARTICIPA
Projeto
Cod_Proj: INTEGER
Nome_proj: VARCHAR
Duracao: INTEGER
PARTICIPA
Func_Proj
Cod_Func: INTEGER (FK)
Cod_Proj: INTEGER (FK)
Horas_Trab: INTEGER

Departamento

Cod_Depto	Nome_Depto
1	Marketing
2	Vendas
3	Dados
4	Pesquisa

Funcionário

Cod_Func	Nome_Func	Salario	Cod_Depto
101	Joao da Silva	2000	2
102	Mario Souza	1500	1
103	Sergio Santos	2400	2
104	Maria Castro	1200	1
105	Marcio Santana	1400	4

Projeto

Cod_Proj	Nome_Proj	Duracao
1001	Sistema A	2
1002	Sistema B	6
1003	Sistema X	4

Func_Proj

Cod_Func	Cod_Proj	Horas_Trab
101	1001	24
101	1002	160
102	1001	56
102	1003	45
103	1001	86
103	1003	64
104	1001	46

```
-- Script de Criação do BD Projeto

DROP TABLE Func_Proj CASCADE CONSTRAINT;
DROP TABLE Projeto CASCADE CONSTRAINT;
DROP TABLE Funcionario CASCADE CONSTRAINT;
DROP TABLE Departamento CASCADE CONSTRAINT;

CREATE TABLE Departamento
(Cod_Depto INTEGER,
 Nome_Depto VARCHAR(20) NOT NULL,
 PRIMARY KEY(Cod_Depto));

CREATE TABLE Funcionario
(Cod_Func INTEGER,
 Nome_Func VARCHAR(20) NOT NULL,
 Salario INTEGER NOT NULL,
 Cod_Depto INTEGER NOT NULL,
 PRIMARY KEY(Cod_Func),
 FOREIGN KEY (Cod_Depto) REFERENCES Departamento (Cod_Depto));

CREATE TABLE Projeto
(Cod_Proj INTEGER,
 Nome_Proj VARCHAR(20) NOT NULL,
 Duracao INTEGER NOT NULL,
 PRIMARY KEY(Cod_Proj));

CREATE TABLE Func_Proj
(Cod_Func INTEGER,
 Cod_Proj INTEGER,
 Horas_Trab INTEGER,
 PRIMARY KEY(Cod_Func, Cod_Proj),
 FOREIGN KEY (Cod_Func) REFERENCES Funcionario(Cod_Func),
 FOREIGN KEY (Cod_Proj) REFERENCES Projeto(Cod_Proj));
```

```
INSERT INTO Departamento (Cod_Depto, Nome_Depto) VALUES (1, 'Marketing');
INSERT INTO Departamento (Cod_Depto, Nome_Depto) VALUES (2, 'Vendas');
INSERT INTO Departamento (Cod_Depto, Nome_Depto) VALUES (3, 'Dados');
INSERT INTO Departamento (Cod_Depto, Nome_Depto) VALUES (4, 'Pesquisa');

INSERT INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto) VALUES (101, 'Joao da Silva Santos', 2000, 2);
INSERT INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto) VALUES (102, 'Mario Souza', 1500, 1);
INSERT INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto) VALUES (103, 'Sergio Silva Santos', 2400, 2);
INSERT INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto) VALUES (104, 'Maria Castro', 1200, 1);
INSERT INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto) VALUES (105, 'Marcio Silva Santana', 1400, 4);

INSERT INTO Projeto (Cod_Proj, Nome_Proj, Duracao) VALUES (1001, 'SistemaA', 2);
INSERT INTO Projeto (Cod_Proj, Nome_Proj, Duracao) VALUES (1002, 'SistemaB', 6);
INSERT INTO Projeto (Cod_Proj, Nome_Proj, Duracao) VALUES (1003, 'SistemaX', 4);

INSERT INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab) VALUES (101, 1001, 24);
INSERT INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab) VALUES (101, 1002, 160);
INSERT INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab) VALUES (102, 1001, 56);
INSERT INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab) VALUES (102, 1003, 45);
INSERT INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab) VALUES (103, 1001, 86);
INSERT INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab) VALUES (103, 1003, 64);
INSERT INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab) VALUES (104, 1001, 46);
COMMIT;
```

Introdução

- A junção de várias tabelas pode ser feita na cláusula WHERE, colocando-se os nomes de todas as tabelas envolvidas na cláusula FROM

Introdução

- Exemplo (junção na cláusula WHERE): Obtenha o nome de cada funcionário e o nome do departamento que cada um pertence, em ordem crescente do nome do departamento.

```
SELECT F.Nome_Func, D.Nome_Depto
FROM Funcionario F, Departamento D
WHERE (F.Cod_depto = D.Cod_depto)
ORDER BY D.Nome_Depto ASC;
```

Nome_Func	Nome_Depto
Mario Souza	Marketing
Maria Castro	Marketing
Marcio Silva Santana	Pesquisa
Joao da Silva Santos	Vendas
Sergio Silva Santos	Vendas

INNER JOIN

INNER JOIN

- No padrão SQL:2003, a operação de junção de várias tabelas pode ser expressa diretamente na cláusula FROM, em vez de ser expressa nas cláusulas FROM e WHERE

INNER JOIN

- Para executar uma operação de junção na cláusula FROM, utilize as palavras-chave INNER JOIN e as condições de junção são indicadas pela palavra-chave ON, dentro da cláusula FROM (observe que a condição de junção não aparece mais na cláusula WHERE).

INNER JOIN

- A **junção COMPLETA** das tabelas Funcionario e Departamento é obtida através do seguinte comando:

```
SELECT *
FROM Funcionario F INNER JOIN  Departamento D
ON (F.Cod_depto = D.Cod_depto);
```

INNER JOIN

- A junção COMPLETA das tabelas Funcionario e Departamento gera o seguinte resultado:

Cod_Func	Nome_Func	Salario	Cod_Depto	Cod_Depto	Nome_Depto
101	Joao da Silva Santos	2000	2	2	Vendas
102	Mario Souza	1500	1	1	Marketing
103	Sergio Silva Santos	2400	2	2	Vendas
104	Maria Castro	1200	1	1	Marketing
105	Marcio Silva Santana	1400	4	4	Pesquisa

INNER JOIN

- Exemplo (INNER JOIN): Obtenha o nome de cada funcionário e o nome do departamento que cada um pertence, em ordem crescente do nome de departamento.

```
SELECT F.Nome_Func, D.Nome_Depto
FROM Funcionario F INNER JOIN  Departamento D
ON (F.Cod_depto = D.Cod_depto)
ORDER BY D.Nome_Depto ASC;
```

Nome_Func	Nome_Depto
Mario Souza	Marketing
Maria Castro	Marketing
Marcio Silva Santana	Pesquisa
Joao da Silva Santos	Vendas
Sergio Silva Santos	Vendas

INNER JOIN

- nome de cada
  ento que cada um
  nome de departamento.

epto

Departamento D

Observe que o departamento 'Dados' não apareceu no resultado, pois ele não tem funcionário. O INNER JOIN traz somente as linhas combinadas, de acordo com a condição de junção.

Nome_Func	Nome_Depto
Mario Souza	Marketing
Maria Castro	Marketing
Marcio Silva Santana	Pesquisa
Joao da Silva Santos	Vendas
Sergio Silva Santos	Vendas

INNER JOIN

- Exemplo (INNER JOIN): Obtenha o nome de cada departamento (que tenha funcionário) e, para cada departamento, o número de funcionários, em ordem crescente do nome de departamento.

```
SELECT D.Nome_Depto, COUNT(F.Cod_Depto) AS Total
FROM Departamento D INNER JOIN Funcionario F
ON (D.Cod_depto = F.Cod_depto)
GROUP BY D.Nome_Depto
ORDER BY D.Nome_Depto ASC;
```

Nome_Depto	Total
Marketing	2
Pesquisa	1
Vendas	2

INNER JOIN

- Exemplo (INNER JOIN): Obtenha o nome de cada departamento (que tenha funcionário) e, para cada departamento, o número de funcionários, em ordem crescente do nome de departamento.

```
SELECT D.Nome_Depto, COUNT(F.Cod_Depto) AS Total
FROM Departamento D INNER JOIN Funcionario F
ON (D.Cod_depto = F.Cod_depto)
GROUP BY D.Nome_Depto
ORDER BY D.Nome_Depto ASC;
```

Nome_Depto	Total
Marketing	2
Pesquisa	1
Vendas	2

JOIN

o nome de cada
ionário) e, para cada
ncionários, em ordem
nento.

NT(F.Cod_Depto) AS Total
R JOIN Funcionario F
pto,

ORDER BY D.Nome_Depto ASC;

Observe que o departamento 'Dados' não apareceu no resultado, pois ele não tem funcionário. O INNER JOIN traz somente as linhas combinadas, de acordo com a condição de junção.

Nome_Depto	Total
Marketing	2
Pesquisa	1
Vendas	2

INNER JOIN com mais de 2 tabelas

- Exemplo (INNER JOIN): Obtenha o nome de cada funcionário e o nome dos projetos que cada um trabalhou

```
SELECT F.Nome_Func, P.Nome_Proj

FROM Funcionario F  INNER JOIN Func_Proj FP

ON (F.Cod_Func = FP.Cod_Func)

INNER JOIN Projeto P ON (FP.Cod_Proj = P.Cod_Proj);
```

INNER JOIN

- Observe o resultado, a seguir, da consulta anterior
- Neste caso, o funcionário de nome **'Marcio Silva Santana' não apareceu no resultado**, pois ele não participou de nenhum projeto, portanto **não teve linhas combinadas**

Nome_Func	Nome_Proj
Joao da Silva Santos	SistemaA
Joao da Silva Santos	SistemaB
Mario Souza	SistemaA
Mario Souza	SistemaX
Sergio Silva Santos	SistemaA
Sergio Silva Santos	SistemaX
Maria Castro	SistemaA

Obrigado

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