

FIAP GRADUAÇÃO

# ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

## ESTRATÉGIA E IMPLEMENTAÇÃO DE ESTRUTURAS DE DADOS

Profª. Rita de Cássia Rodrigues  
rita@fiap.com.br

AULA 24 – COMANDOS DQL/DRS (SQL)

(DRS → LINGUAGEM PARA RECUPERAÇÃO DE DADOS)

(DQL → LINGUAGEM PARA CONSULTA DE DADOS)

JUNÇÃO DE TABELAS – JOIN

PESQUISA A MÚLTIPLAS TABELAS

- ✓ Objetivo
- ✓ Conceitos referentes a linguagem de consulta/recuperação de dados
- ✓ Revisão dos Conceitos
- ✓ Exercícios

- ❑ Aplicar os conceitos da linguagem SQL durante a implementação do banco de dados

---

# **Conteúdo Programático referente a esta aula**

FIAP

- ☐ Linguagem para consulta/recuperação de dados
  - ☐ DRS/DQL (SELECT)
    - ☐ Junções (consulta a duas ou mais tabelas)
  - ☐ Exercícios

# Linguagem SQL

## SQL - Structured Query Language

### (Linguagem Estruturada de Consulta)

FIAP



É uma **linguagem de definição, manipulação e controle de banco de dados.**

DRS – Linguagem para recuperação de dados ou

DQL – Linguagem para consulta de dados

Utilizada pesquisar e apresentar dados já armazenados no banco de dados.

**Exemplo: SELECT .**

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP

### Sintaxe Básica

```
SELECT tabela1.coluna, tabela2.coluna  
FROM tabela1
```

```
[CROSS JOIN tabela2] |
```

```
[NATURAL JOIN tabela2] |
```

```
[JOIN tabela2 USING (nome_coluna)] |
```

```
[JOIN tabela2
```

```
ON (tabela1.nome_coluna = tabela2.nome_coluna)] |
```

```
[LEFT|RIGHT|FULL OUTER JOIN tabela2
```

```
ON (tabela1.nome_coluna = tabela2.nome_coluna)] ;
```



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



Utilizaremos o exercício **IMPLANTAÇÃO DE PROJETOS** na manipulação dos dados utilizando SQL.

Scripts necessários para os exemplos desta aula:

- ✓ [SCRIPT\\_AULA24\\_DDL\\_IMPLANTACAO\\_PROJETO.sql](#)
- ✓ [SCRIPT\\_AULA24\\_DML\\_IMPLANTACAO\\_PROJETO.sql](#)
- ✓ [SCRIPT\\_AULA24\\_EXEMPLO\\_RECURSAO.sql](#)

**Nota:** Os arquivos se encontram no portal da FIAP.

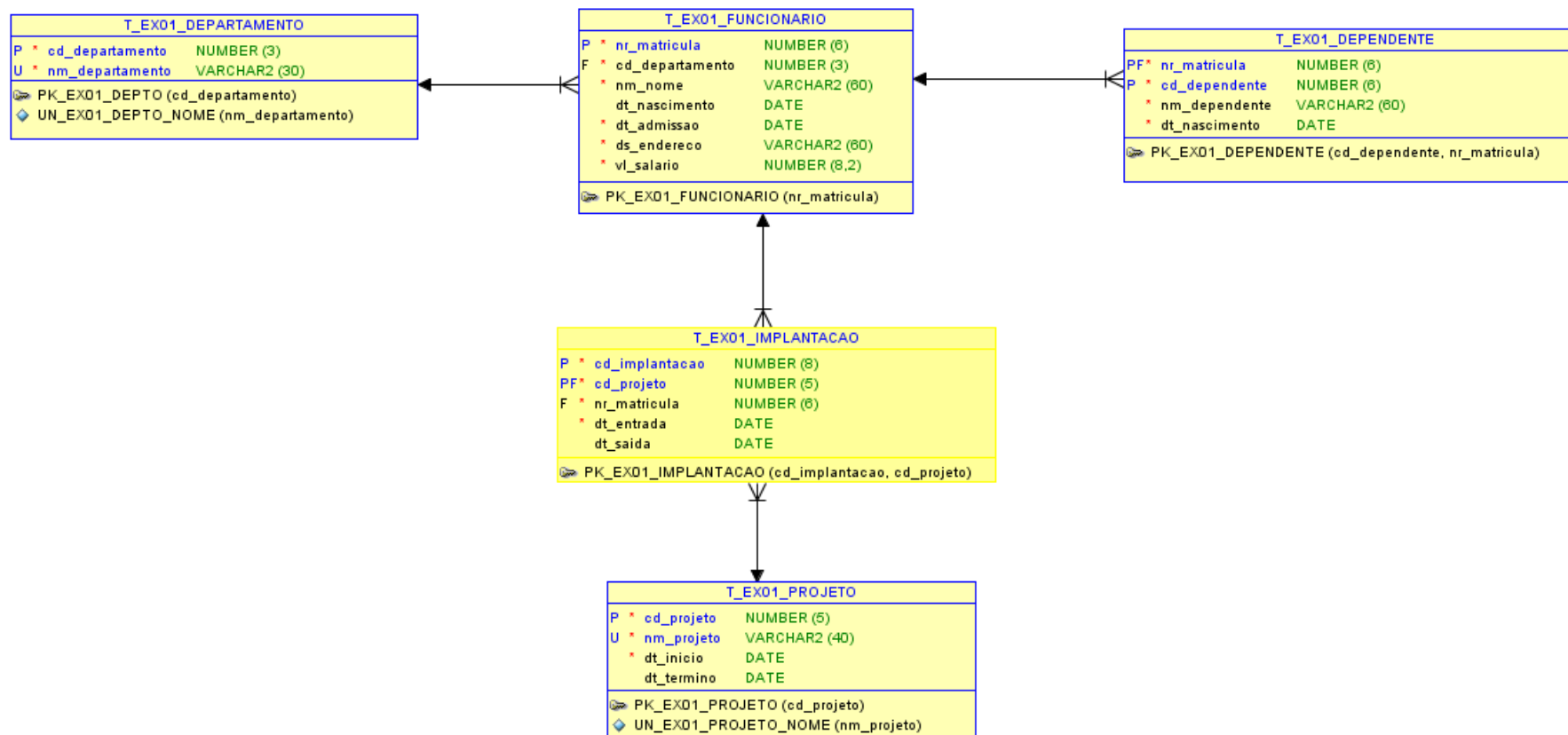




# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### Modelo Relacional ou Físico – IMPLANTAÇÃO DE PROJETOS



---

# **Linguagem SQL**

## **Consulta dos Dados utilizando a linguagem SQL**

FIAP



## **EXEMPLOS: SELECT UTILIZANDO JUNÇÕES**



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



### Pesquisa em Múltiplas Tabelas

Se faz necessário quando precisamos recuperar informações que estão distribuídas em uma ou várias tabelas, a esta necessidade damos o nome de JUNÇÕES (JOIN).

A operação JUNÇÃO (JOIN) é usada para combinar registros relacionados, dentro de um único registro. Esta operação nos permite processar os relacionamentos entre as relações.



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



### Condições de Join

Há dois tipos baseadas no operador utilizado:

**EQUIJOINS:** utilizam o operador de igualdade ( = ). São consideradas como consequência de um modelo relacional, ou seja, espelham a relação dos dados representados em um modelo E-R.

**NO-EQUIJOINS:** Utilizam um operador que não é o de igualdade, como <, > , BETWEEN etc. São consideradas como junções atípicas, onde não há a presença de chave primária e nem mesmo chaves estrangeiras. Os *relacionamentos são feitos através de colunas normalmente não chaves primárias e nem únicas.*

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



### Tipos de Join

Há três tipos diferentes de Joins:

Joins INTERNAS : retornam uma linha somente quando as colunas da join contêm valores que satisfazem a condição.

Joins EXTERNAS: retornam uma linha mesmo quando uma das colunas na condição de join contém um valor nulo.

AutoJoins: retornam linhas unidas na mesma tabela.

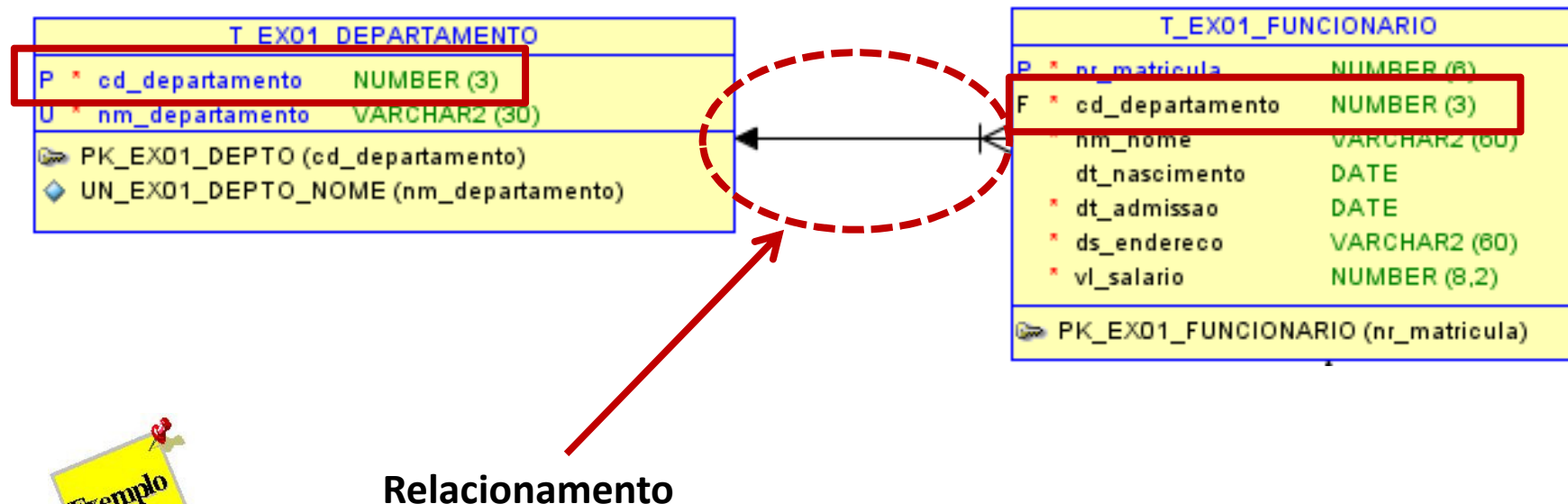


# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### Pesquisa em Múltiplas Tabelas

Para ilustrar, imagine que precisássemos recuperar o nome do departamento em que o funcionário está trabalhando,



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP

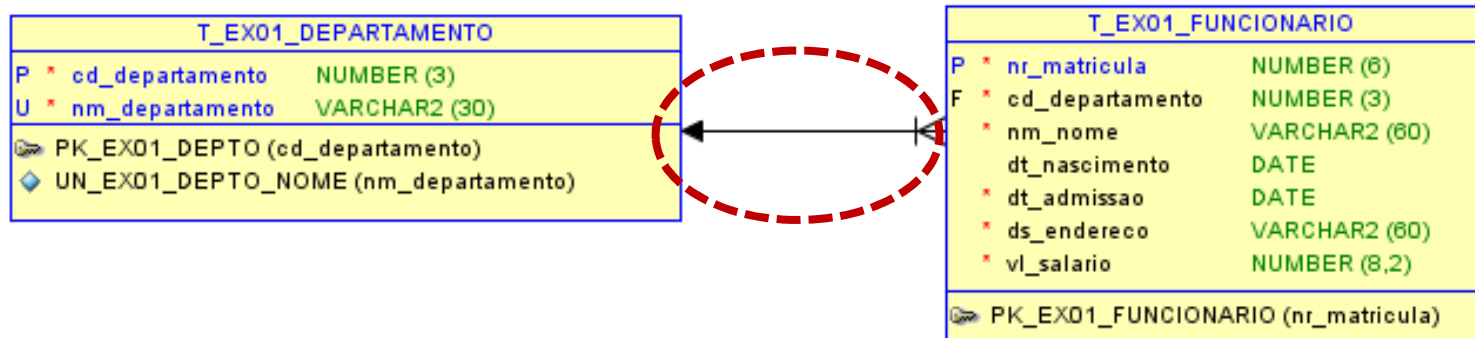


### Pesquisa em Múltiplas Tabelas

Observe que o **nome do departamento** está armazenado na tabela **“DEPARTAMENTO”** que representa o cadastro dos departamentos.

Perceba que existe uma associação entre estas duas tabelas, observamos a **chave estrangeira “CD\_DEPARTAMENTO”** na tabela **“FUNCIONARIO”**.

Para que possamos recuperar o nome do departamento para cada registro da tabela **“FUNCIONARIO”** precisaremos **combinar (JUNÇÃO)** estas duas tabelas.



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

Tabela: FUNCIONARIO

Tabela: DEPARAMENTO

R	NR_MATRICULA	R	CD_DEPARTAMENTO	R	NM_NOME	R	DT_NASCIMENTO
1	1234	1		1	ANA MARIA	10/02/1986	00:00:00
2	1235	2		2	ROSA MARIA	15/08/1970	00:00:00
3	3245	1		1	ANTONIA CAMARGO	18/07/1986	00:00:00
4	2233	3		3	JOÃO DA SILVA	(null)	
5	7866	3		3	JOAQUIM XAVIER	21/04/1987	00:00:00
6	9876	4		4	JOSÉ MARIA	08/09/1998	00:00:00
7	9899	4		4	JOSÉ LISBOA	07/10/1996	00:00:00

R	CD_DEPARTAMENTO	R	NM_DEPARTAMENTO
1	4	4	COMERCIAL
2	1	1	FINANCEIRO
3	2	2	MARKETING
4	3	3	RECURSOS HUMANOS
5	5	5	TECNOLOGIA DA INFORMAÇÃO

### Resultado da Junção:

R	MATRICULA	R	COD. DEPTO	R	DEPARTAMENTO	R	FUNCIONARIO
1	1234	1	FINANCEIRO				ANA MARIA
2	1235	2	MARKETING				ROSA MARIA
3	3245	1	FINANCEIRO				ANTONIA CAMARGO
4	2233	3	RECURSOS HUMANOS				JOÃO DA SILVA
5	7866	3	RECURSOS HUMANOS				JOAQUIM XAVIER
6	9876	4	COMERCIAL				JOSÉ MARIA
7	9899	4	COMERCIAL				JOSÉ LISBOA





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



### CROSS JOIN (Junção Cruzada)

Também conhecida por Produto Cartesiano.

Ocorrerá um produto cartesiano sempre que:

- ☐ Não houver uma condição para a união (ausência da cláusula WHERE);
- ☐ Condição de união entre as tabelas inválida (cláusula WHERE incorreta);
- ☐ Todas as linhas da primeira tabela estiverem unidas a todas as linhas da segunda tabela.

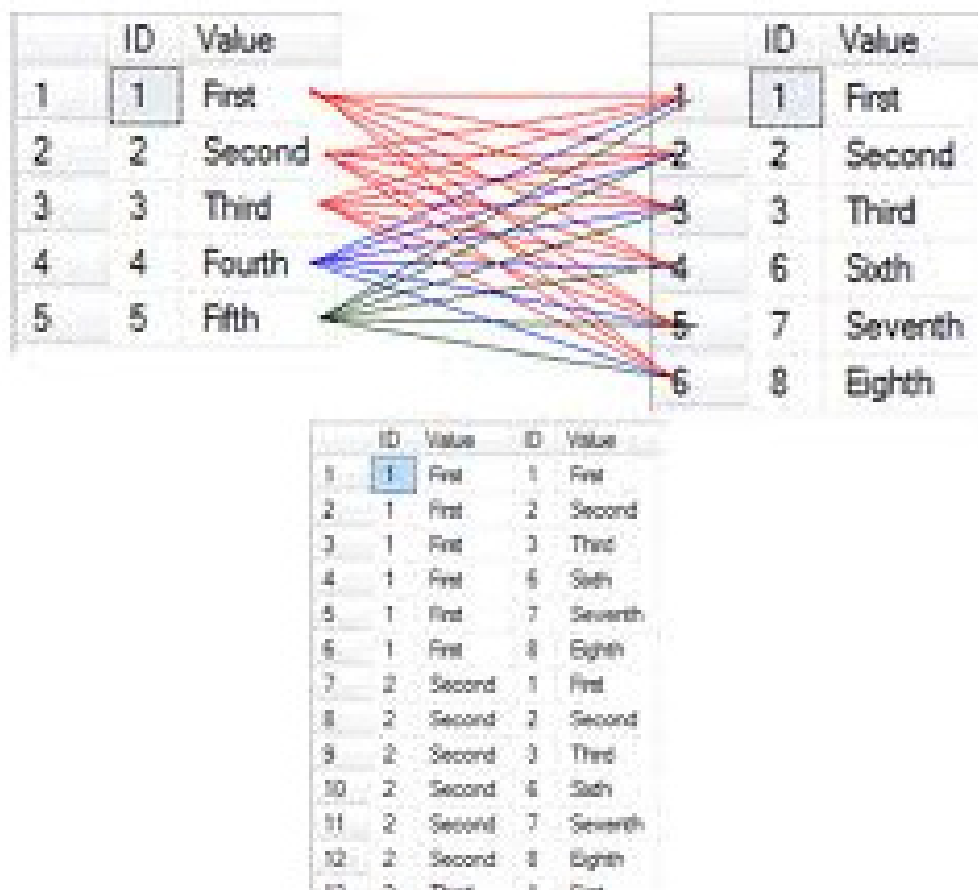
Para evitar um produto cartesiano, utilize uma condição válida para junção na cláusula WHERE.

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### CROSS JOIN (Junção Cruzada)

Exemplo:



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



### CROSS JOIN (Junção Cruzada)

Faça uma junção cruzada entre as tabelas: “DEPARTAMENTO” e “FUNCIONARIO”.

Consulte as tabelas “DEPARTAMENTO” e “FUNCIONARIO” separadamente. Observe a tabela “DEPARTAMENTO” possui 5 linhas e a tabela “FUNCIONARIO”, possui 7 linhas:

Utilize o comando SELECT abaixo:

-- EXEMPLO - CROSS JOIN (PRODUTO CARTESIANO)

SELECT \*

FROM T\_EX01\_DEPARTAMENTO ,  
T\_EX01\_FUNCIONARIO;

-- UTILIZANDO O PADRÃO SQL/99

SELECT \*

FROM T\_EX01\_DEPARTAMENTO CROSS JOIN T\_EX01\_FUNCIONARIO;





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### CROSS JOIN (Junção Cruzada)

Executando o exemplo anterior teremos o seguinte resultado:

Todas as Linhas Extraídas: 35 em 0,011 segundos

	CD_DEPARTAMENTO	NM_DEPARTAMENTO	NR_MATRICULA	CD_DEPARTAMENTO_1	NM_NOME	DT_NASCIMENTO
1	4	COMERCIAL	1234	1	ANA MARIA	10/02/1986 00:00:00
2	4	COMERCIAL	1235	2	ROSA MARIA	15/08/1970 00:00:00
3	4	COMERCIAL	3245	1	ANTONIA CAMARGO	18/07/1986 00:00:00
4	4	COMERCIAL	2233	3	JOÃO DA SILVA	(null)
5	4	COMERCIAL	7866	3	JOAQUIM XAVIER	21/04/1987 00:00:00
6	4	COMERCIAL	9876	4	JOSÉ MARIA	08/09/1998 00:00:00
7	4	COMERCIAL	9899	4	JOSÉ LISBOA	07/10/1996 00:00:00
8	1	FINANCEIRO	1234	1	ANA MARIA	10/02/1986 00:00:00
9	1	FINANCEIRO	1235	2	ROSA MARIA	15/08/1970 00:00:00
10	1	FINANCEIRO	3245	1	ANTONIA CAMARGO	18/07/1986 00:00:00

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL



### CROSS JOIN (Junção Cruzada)

O que ocorreu no exemplo foi que as linhas da primeira tabela foram combinadas com as linhas da segunda tabela, demonstrando um resultado na maior parte das vezes indesejado.

Veja que a tabela “DEPARTAMENTO”, possuía 5 linhas e a tabela “FUNCIONARIO” possuía 7 linhas, o resultado foram  $(5 \times 7 = 35)$  35 linhas exibidas como resultado da nossa busca.

Percebemos então a necessidade de colocar a condição após a cláusula WHERE associando a chave primária e estrangeira das tabelas. Utilize o modelo de dados para facilitar a criação de sua consulta SQL.

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



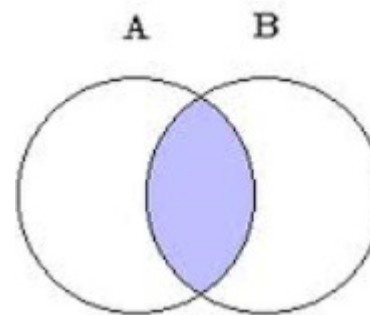
### NATURAL JOIN

A junção do tipo **NATURAL JOIN** segue a **padronização ANSI** e implica por sua vez em criar a **junção entre duas tabelas**. A junção será feita através de **colunas com o mesmo nome nas duas tabelas**. Por isso devemos ter cuidado caso as colunas tenham nomes diferentes em uma tabela e em outra.

Esse tipo de junção surgiu a partir da padronização ANSI 99.

Caso as colunas tiverem o mesmo nome, mas tipos de dados diferentes será retornado um erro.

É um tipo de junção interna.





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

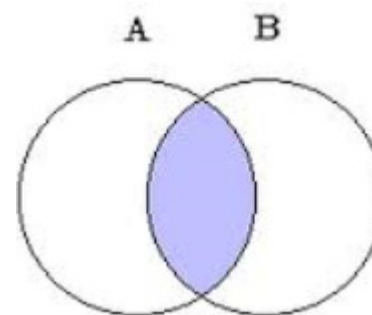
### NATURAL JOIN

Crie um NATURAL JOIN entre as tabelas “DEPARTAMENTO” e “FUNCIONARIO”.

Utilize o comando SELECT abaixo:

-- EXEMPLO - NATURAL JOIN

```
SELECT F.NR_MATRICULA      "MATRICULA" ,  
       CD_DEPARTAMENTO      "COD. DEPTO" ,  
       D.NM_DEPARTAMENTO    "DEPARTAMENTO",  
       F.NM_NOME            "FUNCIONARIO"  
  
FROM T_EX01_DEPARTAMENTO D NATURAL JOIN  
     T_EX01_FUNCIONARIO F;
```



**Nota:** Não aplique o qualificador na coluna “CD\_DEPARTAMENTO”, acarretará um erro. Quando utilizamos NATURAL JOIN não podemos utilizar qualificadores para a colunas chave primária/estrangeira.

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL



### NATURAL JOIN

Executando o exemplo anterior teremos o seguinte resultado:

Todas as Linhas Extraídas: 7 em 0,002 segundos				
	 MATRICULA	 COD. DEPTO	 DEPARTAMENTO	 FUNCIONARIO
1	1234		1 FINANCEIRO	ANA MARIA
2	1235		2 MARKETING	ROSA MARIA
3	3245		1 FINANCEIRO	ANTONIA CAMARGO
4	2233		3 RECURSOS HUMANOS	JOÃO DA SILVA
5	7866		3 RECURSOS HUMANOS	JOAQUIM XAVIER
6	9876		4 COMERCIAL	JOSÉ MARIA
7	9899		4 COMERCIAL	JOSÉ LISBOA



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

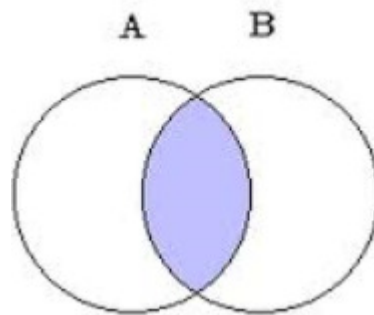
### NATURAL JOIN UTILIZANDO A CLÁUSULA USING



Caso as colunas possuam o mesmo nome, mas tipos de dados diferentes, podemos modificar a cláusula NATURAL JOIN com a **cláusula USING** para especificar as colunas que devem ser usadas em uma junção.

**Não utilize apelido ou nome de tabelas nas colunas** onde a referência for feita (chave primária/chave estrangeira).

As cláusulas **NATURAL JOIN** e **USING** são **excludentes**, ou seja, não podem ser utilizadas juntas.





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

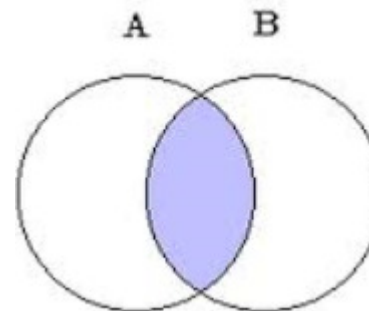
### NATURAL JOIN UTILIZANDO A CLÁUSULA USING

Crie um NATURAL JOIN entre as tabelas “DEPARTAMENTO” e “FUNCIONARIO”, utilizando a cláusula USING.

Utilize o comando SELECT abaixo:

-- EXEMPLO - USING

```
SELECT F.NR_MATRICULA      "MATRICULA" ,  
       CD_DEPARTAMENTO      "COD. DEPTO" ,  
       D.NM_DEPARTAMENTO    "DEPARTAMENTO",  
       F.NM_NOME            "FUNCIONARIO"  
FROM T_EX01_DEPARTAMENTO D JOIN  
     T_EX01_FUNCIONARIO F  
     USING ( CD_DEPARTAMENTO );
```



**Nota:** Não aplique o qualificador na coluna “CD\_DEPARTAMENTO”, acarretará um erro. Quando utilizamos USING não podemos utilizar qualificadores para as colunas chave primária/estrangeira.

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP

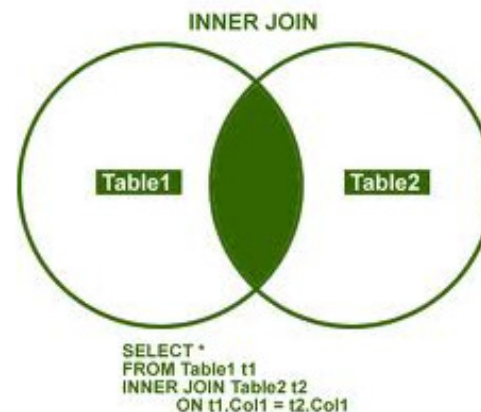


### INNER JOIN

Nesse tipo de junção, caso sejam unidas duas tabelas serão **exibidos todos os dados existentes nas duas tabelas envolvidas na consulta.**

Denomina-se **união regular** as uniões que têm a cláusula **WHERE** indicando a **chave primária à estrangeira das tabelas** afetadas pelo comando SELECT.

É um tipo de junção interna.



(C) <http://blog.SQLAuthority.com>



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### INNER JOIN

Crie um INNER JOIN entre as tabelas “DEPARTAMENTO” e “FUNCIONARIO”.

Utilize o comando SELECT abaixo:

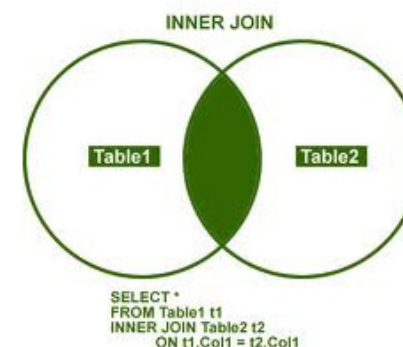
-- EXEMPLO COM INNER JOIN - PADRÃO SQL/99

```
SELECT F.NR_MATRICULA      "MATRICULA" ,
       F.CD_DEPARTAMENTO    "COD. DEPTO" ,
       D.NM_DEPARTAMENTO    "DEPARTAMENTO",
       F.NM_NOME            "FUNCIONARIO"

FROM T_EX01_DEPARTAMENTO D INNER JOIN T_EX01_FUNCIONARIO F

ON ( D.CD_DEPARTAMENTO = F.CD_DEPARTAMENTO )

ORDER BY D.NM_DEPARTAMENTO;
```



(C) <http://blog.SQLAuthority.com>



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### INNER JOIN

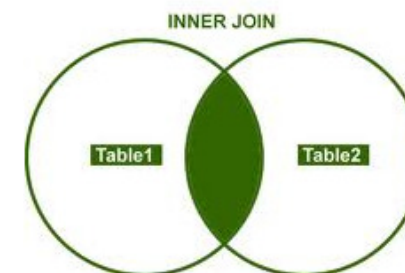
Podemos escrever alternativamente o comando, conforme exemplo:

-- EXEMPLO COM INNER JOIN - PADRÃO ORACLE (ANTES SQL/99)

```
SELECT F.NR_MATRICULA      "MATRICULA" ,
       F.CD_DEPARTAMENTO    "COD. DEPTO" ,
       D.NM_DEPARTAMENTO    "DEPARTAMENTO",
       F.NM_NOME            "FUNCIONARIO"
FROM   T_EX01_DEPARTAMENTO D ,
       T_EX01_FUNCIONARIO  F
WHERE  D.CD_DEPARTAMENTO = F.CD_DEPARTAMENTO
ORDER BY D.NM_DEPARTAMENTO;
```

CHAVE PRIMÁRIA

CHAVE ESTRANGEIRA



```
SELECT *
FROM Table1 t1
INNER JOIN Table2 t2
ON t1.Col1 = t2.Col1
```

(C) <http://blog.SQLAuthority.com>



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### INNER JOIN UTILIZANDO A CLÁUSULA USING

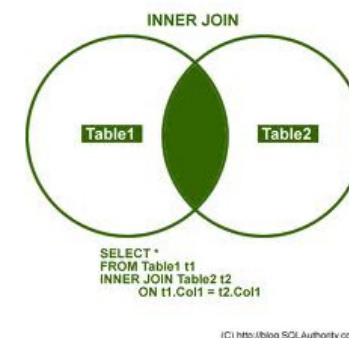
Se usarmos o mesmo nome para a chave primária e chave estrangeira, poderemos escrever o comando conforme exemplo abaixo:

#### -- EXEMPLO COM INNER JOIN – PADRÃO SQL/99 – CLÁUSULA USING

```
SELECT F.NR_MATRICULA      "MATRICULA" ,
       CD_DEPARTAMENTO      "COD. DEPTO" ,
       D.NM_DEPARTAMENTO    "DEPARTAMENTO",
       F.NM_NOME            "FUNCIONARIO"
```

```
FROM T_EX01_DEPARTAMENTO D INNER JOIN T_EX01_FUNCIONARIO F
     USING ( CD_DEPARTAMENTO )
```

```
ORDER BY D.NM_DEPARTAMENTO;
```



**Nota:** Não aplique o qualificador na coluna “CD\_DEPARTAMENTO”, acarretará um erro. Quando utilizamos USING não podemos utilizar qualificadores para a colunas chave primária/estrangeira.

---

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### UNIÃO DE DUAS OU MAIS TABELAS



Exemplificando iremos recuperar os dados de um funcionário mostrando os projetos em que ele está atuando.

Para facilitar observe o modelo de dados, para identificar as chaves primária e estrangeira.





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### INNER JOIN (UNIÃO DE DUAS OU MAIS TABELAS)

Recuperar os dados de um funcionário mostrando os projetos em que ele está atuando.

Utilize o comando SELECT abaixo:

-- EXEMPLO COM INNER JOIN - PADRÃO SQL/99

```
SELECT F.NR_MATRICULA      "MATRICULA" ,
       F.NM_NOME           "FUNCIONARIO" ,
       P.NM_PROJETO        "PROJETO" ,
       I.DT_ENTRADA        "ENTRADA" ,
       I.DT_SAIDA          "SAIDA"

FROM T_EX01_PROJETO P INNER JOIN T_EX01_IMPLANTACAO I
     ON ( P.CD_PROJETO = I.CD_PROJETO )

INNER JOIN T_EX01_FUNCIONARIO F
     ON ( F.NR_MATRICULA = I.NR_MATRICULA )

ORDER BY F.NM_NOME ;
```





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### INNER JOIN UTILIZANDO A CLÁUSULA USING

### (UNIÃO DE DUAS OU MAIS TABELAS)

-- EXEMPLO COM INNER JOIN - PADRÃO SQL/99 - CLÁUSULA USING

```
SELECT NR_MATRICULA      "MATRICULA" ,
       F.NM_NOME          "FUNCIONARIO" ,
       P.NM_PROJETO       "PROJETO" ,
       I.DT_ENTRADA       "ENTRADA" ,
       I.DT_SAIDA         "SAIDA"

FROM T_EX01_PROJETO P INNER JOIN T_EX01_IMPLANTACAO I
     USING ( CD_PROJETO )
     INNER JOIN T_EX01_FUNCIONARIO F
     USING ( NR_MATRICULA )

ORDER BY F.NM_NOME ;
```

**Nota:** Não aplique o qualificador na coluna “NR\_MATRICULA”, acarretará um erro. Quando utilizamos USING não podemos utilizar qualificadores para a colunas chave primária/estrangeira.

# Linguagem SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### INNER JOIN (UNIÃO DE DUAS OU MAIS TABELAS)



Podemos escrever alternativamente o comando, conforme exemplo:

-- EXEMPLO COM INNER JOIN - PADRÃO ORACLE (ANTES SQL/99)

```
SELECT F.NR_MATRICULA      "MATRICULA" ,
       F.NM_NOME           "FUNCIONARIO" ,
       P.NM_PROJETO        "PROJETO" ,
       I.DT_ENTRADA        "ENTRADA" ,
       I.DT_SAIDA          "SAIDA"

FROM T_EX01_PROJETO P, T_EX01_FUNCIONARIO F, T_EX01_IMPLANTACAO I

WHERE P.CD_PROJETO = I.CD_PROJETO AND

      F.NR_MATRICULA = I.NR_MATRICULA

ORDER BY F.NM_NOME ;
```



# Linguagem SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### DICA:



No padrão SQL a ordem em que colocamos as tabelas na cláusula FROM determina quais tabelas serão pesquisadas primeiro. Logo se colocarmos as tabelas menores primeiro a busca ficará mais rápida.

Portanto deixe as tabelas maiores, sempre que possível, para o final da cláusula.



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### OUTER JOIN

Define-se união externa como aquela que inclui linhas no resultado da busca mesmo que não haja relação entre as duas tabelas que estão sendo unidas.

Onde não há informação o banco de dados insere NULL.

Existem 3 formas de realizar a união externa:

- ☐ Left Outer Join
- ☐ Right Outer Join
- ☐ Full Outer Join

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP

### LEFT JOIN



Todas as linhas da tabela a esquerda serão recuperadas, independentemente da existência de ocorrências relacionadas na tabela da direita.

Preserva as linhas sem correspondência da primeira tabela (esquerda), juntando-as com uma linha nula na forma da segunda tabela (direita).





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### LEFT JOIN

Recuperar os projetos que possuem ou não registros de implantações.

Utilize o comando SELECT abaixo:

-- EXEMPLO LEFT JOIN - PADRÃO SQL/99

```
SELECT    P.CD_PROJETO          "CODIGO" ,
          P.NM_PROJETO          "PROJETO" ,
          P.DT_INICIO           "DATA INICIO" ,
          I.NR_MATRICULA        "MATRÍCULA FUNCIONÁRIO" ,
          I.DT_ENTRADA          "ENTRADA"

FROM T_EX01_PROJETO P LEFT OUTER JOIN T_EX01_IMPLANTACAO I
      ON ( P.CD_PROJETO = I.CD_PROJETO );
```



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### LEFT JOIN UTILIZANDO A CLÁUSULA USING

-- EXEMPLO LEFT JOIN - PADRÃO SQL/99 - CLÁUSULA USING

```
SELECT    CD_PROJETO          "CODIGO" ,
          P.NM_PROJETO        "PROJETO" ,
          P.DT_INICIO         "DATA INICIO" ,
          I.NR_MATRICULA       "MATRÍCULA FUNCIONÁRIO" ,
          I.DT_ENTRADA         "ENTRADA"

FROM T_EX01_PROJETO P LEFT OUTER JOIN T_EX01_IMPLANTACAO I

      USING ( CD_PROJETO );
```

**Nota:** Não aplique o qualificador na coluna “CD\_PROJETO”, acarretará um erro. Quando utilizamos USING não podemos utilizar qualificadores para a colunas chave primária/estrangeira.



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### LEFT JOIN UTILIZANDO O OPERADOR (+)

-- EXEMPLO LEFT JOIN - PADRÃO ORACLE - OPERADOR ( + )

```
SELECT    P.CD_PROJETO          "CODIGO" ,
          P.NM_PROJETO          "PROJETO" ,
          P.DT_INICIO           "DATA INICIO" ,
          I.NR_MATRICULA        "MATRÍCULA FUNCIONÁRIO" ,
          I.DT_ENTRADA          "ENTRADA"

FROM T_EX01_PROJETO P , T_EX01_IMPLANTACAO I

WHERE P.CD_PROJETO = I.CD_PROJETO (+) ;
```

**Nota:** O operador (+) deverá ser inserido ao lado da coluna que poderá ter o valor nulo, neste exemplo é o código do projeto na tabela "IMPLANTACAO".

Este operador só poderá ser colocado em um dos lados. Caso coloque nos dois lados será gerado um erro.





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### LEFT JOIN UTILIZANDO O OPERADOR (+)

-- EXEMPLO LEFT JOIN - PADRÃO ORACLE - OPERADOR ( + )

-- EXEMPLO DE PROJETOS SEM IMPLANTAÇÕES

```
SELECT    P.CD_PROJETO          "CODIGO" ,
          P.NM_PROJETO          "PROJETO" ,
          P.DT_INICIO           "DATA INICIO" ,
          I.NR_MATRICULA        "MATRÍCULA FUNCIONÁRIO" ,
          I.DT_ENTRADA          "ENTRADA"

FROM      T_EX01_PROJETO P , T_EX01_IMPLANTACAO I

WHERE     P.CD_PROJETO  = I.CD_PROJETO (+)

AND       I.CD_PROJETO IS NULL;
```



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

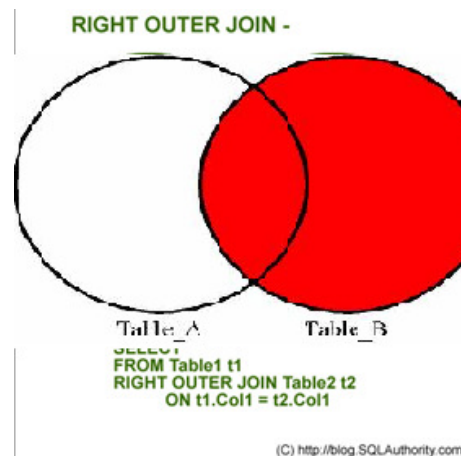
FIAP



### RIGHT JOIN

Todas as linhas da tabela a direita serão recuperadas, independentemente da existência de ocorrências relacionadas na tabela da esquerda.

Preserva as linhas sem correspondência da segunda tabela (direita), juntando-as com uma linha nula na forma da primeira tabela (esquerda).





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### RIGHT JOIN

Recuperar os dados de um funcionário mostrando os projetos em que ele está ou não atuando.

Utilize o comando SELECT abaixo:

-- EXEMPLO RIGHT JOIN - PADRÃO SQL/99

```
SELECT    F.NR_MATRICULA      "MATRICULA FUNCIONÁRIO" ,
          I.CD_PROJETO        "CÓDIGO PROJETO" ,
          I.DT_ENTRADA        "DATA ENTRADA",
          I.DT_SAIDA          "DATA SAÍDA"

FROM T_EX01_IMPLANTACAO I RIGHT OUTER JOIN T_EX01_FUNCIONARIO F
      ON ( F.NR_MATRICULA = I.NR_MATRICULA );
```

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### RIGHT JOIN UTILIZANDO A CLÁUSULA USING

FIAP



-- EXEMPLO RIGHT JOIN - PADRÃO SQL/99 - CLÁUSULA USING

```
SELECT    NR_MATRICULA      "MATRICULA FUNCIONÁRIO" ,  
          I.CD_PROJETO      "CÓDIGO PROJETO" ,  
          I.DT_ENTRADA      "DATA ENTRADA",  
          I.DT_SAIDA        "DATA SAÍDA"  
  
FROM T_EX01_IMPLANTACAO I RIGHT OUTER JOIN T_EX01_FUNCIONARIO F  
      USING ( NR_MATRICULA );
```

**Nota:** Não aplique o qualificador na coluna “NR\_MATRICULA”, acarretará um erro. Quando utilizamos USING não podemos utilizar qualificadores para a colunas chave primária/estrangeira.



# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### RIGHT JOIN UTILIZANDO O OPERADOR (+)

FIAP



-- EXEMPLO RIGHT JOIN - PADRÃO ORACLE - OPERADOR ( + )

```
SELECT    F.NR_MATRICULA      "MATRICULA FUNCIONÁRIO" ,
          I.CD_PROJETO        "CÓDIGO PROJETO" ,
          I.DT_ENTRADA        "DATA ENTRADA",
          I.DT_SAIDA          "DATA SAÍDA"

FROM T_EX01_IMPLANTACAO I , T_EX01_FUNCIONARIO F

WHERE I.NR_MATRICULA (+) = F.NR_MATRICULA;
```

**Nota:** O operador (+) deverá ser inserido ao lado da coluna que poderá ter o valor nulo, neste exemplo é a matrícula do funcionário na tabela “IMPLANTACAO”.

Este operador só poderá ser colocado em um dos lados. Caso coloque nos dois lados será gerado um erro.

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### RIGHT JOIN UTILIZANDO O OPERADOR (+)

FIAP



-- EXEMPLO RIGHT JOIN - PADRÃO ORACLE - OPERADOR ( + )

-- MOSTRANDO APENAS OS FUNCIONÁRIOS SEM IMPLANTAÇÕES

```
SELECT    F.NR_MATRICULA      "MATRICULA FUNCIONÁRIO" ,
          I.CD_PROJETO        "CÓDIGO PROJETO" ,
          I.DT_ENTRADA        "DATA ENTRADA",
          I.DT_SAIDA          "DATA SAÍDA"

FROM      T_EX01_IMPLANTACAO I , T_EX01_FUNCIONARIO F

WHERE     I.NR_MATRICULA (+) = F.NR_MATRICULA AND

          I.NR_MATRICULA IS NULL;
```

# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

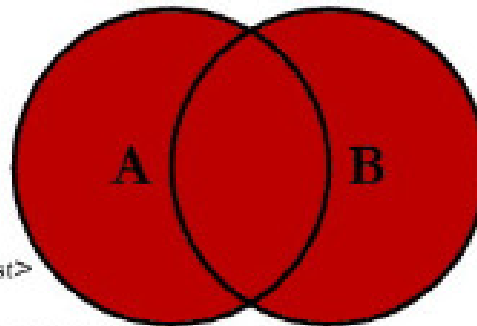
### FULL OUTER JOIN

FIAP



Todas as linhas da tabela a direita e da esquerda serão recuperadas, independentemente da existência de ocorrências relacionadas na tabela da esquerda ou da direita.

```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### FULL OUTER JOIN

Utilize o comando SELECT abaixo:

-- EXEMPLO FULL JOIN - PADRÃO SQL/99

```
SELECT    P.CD_PROJETO          "CODIGO" ,
          P.NM_PROJETO          "PROJETO" ,
          P.DT_INICIO           "DATA INICIO" ,
          I.NR_MATRICULA        "MATRÍCULA FUNCIONÁRIO" ,
          I.DT_ENTRADA           "ENTRADA"

FROM T_EX01_PROJETO P FULL OUTER JOIN T_EX01_IMPLANTACAO I
      ON ( P.CD_PROJETO = I.CD_PROJETO ) ;
```





# Linguagem SQL

## Consulta dos Dados utilizando a linguagem SQL

### FULL JOIN UTILIZANDO A CLÁUSULA USING

-- EXEMPLO FULL JOIN - PADRÃO SQL/99 - CLÁUSULA USING

```
SELECT    CD_PROJETO          "CODIGO" ,
          P.NM_PROJETO        "PROJETO" ,
          P.DT_INICIO         "DATA INICIO" ,
          I.NR_MATRICULA       "MATRÍCULA FUNCIONÁRIO" ,
          I.DT_ENTRADA         "ENTRADA"

FROM T_EX01_PROJETO P FULL OUTER JOIN T_EX01_IMPLANTACAO I
      USING ( CD_PROJETO );
```

**Nota:** Não aplique o qualificador na coluna “CD\_PROJETO”, acarretará um erro. Quando utilizamos USING não podemos utilizar qualificadores para a colunas chave primária/estrangeira.



# Linguagem SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### União de Tabelas sem colunas em comum (non-equijoin)



Existem situações onde mesmo não havendo um relacionamento entre as tabelas, há o relacionamento de uma coluna com o intervalo de outras colunas em outras tabelas.



#### EXEMPLOS:

- Os empregados estão associados a uma faixa salarial. Caso criemos as tabelas relacionadas quando houver mudança de faixa teremos de ir na tabela de empregados para alterar a faixa salarial (FK) do empregado e vice versa, quando alterar o salário do empregado precisamos buscar a nova faixa e associá-la novamente ao empregado.



# Linguagem SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### União de Tabelas sem colunas em comum (non-equijoin)



#### EXEMPLO:

Exemplos como faixa de imposto de renda, alíquotas de impostos em geral, enquadramentos baseados em valores e ou quantidades, são exemplos típicos de consultas non equi-join e nesse caso as tabelas não seriam relacionadas através do modelo e sim através da aplicação.



# Linguagem SQL

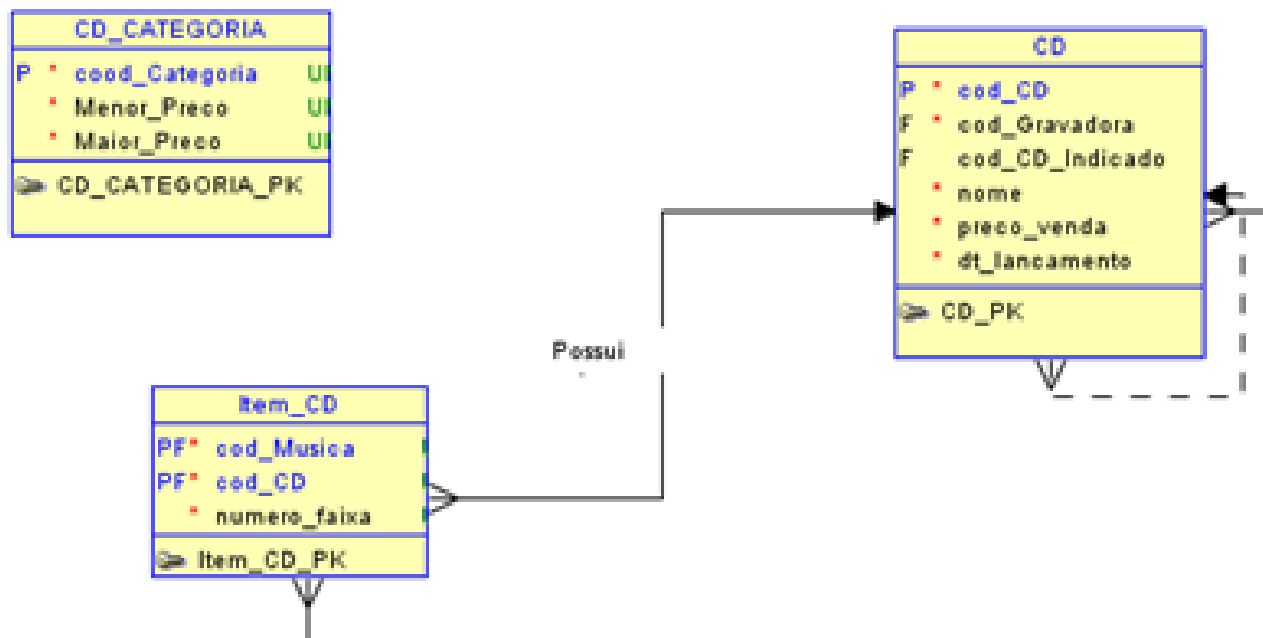
FIAP



## Consulta dos Dados utilizando a linguagem SQL

### União de Tabelas sem colunas em comum (non-equijoin)

Visualizando a modelagem observamos que a tabela categoria possui a faixa de preços dos CD's.



# Linguagem SQL

FIAP



## Consulta dos Dados utilizando a linguagem SQL

### União de Tabelas sem colunas em comum (non-equijoin)

Conteúdo das tabelas para visualização:

Tabela: CATEGORIA

CODIGO_CATEGORIA	MENOR_PRECO	MAIOR_PRECO
2	25	45
1	18.5	32
3	38	52

3 rows selected

Tabela: CD

CODIGO_CD	NOME_CD	PRECO_VENDA
1	PERFIL	23.5
3	AURORA	36.5
2	TODOS	40

3 rows selected

### Resultado da consulta entre as tabelas CD e CATEGORIA

NOME_CD	PRECO_VENDA	CODIGO_CATEGORIA
PERFIL	23.5	1
AURORA	36.5	2
TODOS	40	2
TODOS	40	3

4 rows selected



# Linguagem SQL

FIAP



## Consulta dos Dados utilizando a linguagem SQL

### União de Tabelas sem colunas em comum (non-equijoin)

Comando SQL:

```
SELECT A.NOME_CD ,  
       A.PRECO_VENDA,  
       B.CODIGO_CATEGORIA  
FROM   CD A, CATEGORIA B  
WHERE  A.PRECO_VENDA BETWEEN B.MENOR_PRECO AND  
                                     B.MAIOR_PRECO;
```

**Nota:** Veja que a condição está utilizando a coluna preço de venda da tabela “CD”, comparando com a faixa de preços da tabela “CATEGORIA”.



# Linguagem SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### SELF JOIN ou AUTOJUNÇÃO



As junções do tipo SELF JOIN são utilizadas em casos onde temos no modelo de dados, a figura do **auto-relacionamento** (relacionamento recursivo). Nesse caso precisamos acessar a mesma tabela duas vezes uma para recuperar os registros a outra para buscar os dados relacionados a mesma (auto-relacionamento).

Nesse caso se apenas indicarmos a tabela duas vezes sem usar um apelido (alias) resultará em erro.

Portanto é necessário o uso de apelido nas tabelas e nomes de colunas sempre que tratarmos de SELF JOIN.



# Linguagem SQL

FIAP



## Consulta dos Dados utilizando a linguagem SQL

### SELF JOIN ou AUTOJUNÇÃO

EXEMPLO:

Exibir o código e o nome dos gerentes dos funcionários de uma empresa.

Modelagem de Dados:





# Linguagem SQL

FIAP



## Consulta dos Dados utilizando a linguagem SQL

### SELF JOIN ou AUTOJUNÇÃO

Conteúdo da tabela para visualização:

Tabela: FUNCIONARIO

Todas as Linhas Extraídas: 5 em 0,002 segundos							
	MATRICULA	COD_DEPTO	COD_CARGO	GERENTE	NOME	CPF	RG
1	1	1	1	3	RITA	12345678901	11222333X
2	2	1	1	3	JOAO	12342233445	11333444X
3	3	1	1	(null)	ROSA	12234543441	11444555X
4	4	1	1	(null)	MARIA	1555566651	116667778
5	5	1	1	3	ANA	1343322901	119764333

### Resultado da consulta RECURSIVA

	MATRICULA	FUNCIONÁRIO	COD. GERENTE	GERENTE
1	5	ANA	3	ROSA
2	2	JOAO	3	ROSA
3	1	RITA	3	ROSA



# Linguagem SQL

FIAP



## Consulta dos Dados utilizando a linguagem SQL

### SELF JOIN ou AUTOJUNÇÃO

Comando SQL :

-- EXEMPLO AUTOJUNÇÃO (SELF JOIN) - PADRÃO ORACLE

```
SELECT FUNC.Matricula ,  
        FUNC.NOME "FUNCIONÁRIO",  
        FUNC.GERENTE "COD. GERENTE",  
        GER.NOME "GERENTE"  
FROM    T_AULA16_2_FUNCIONARIO FUNC ,  
        T_AULA16_2_FUNCIONARIO GER  
WHERE   GER.MATRICULA = FUNC.GERENTE;
```

# Linguagem SQL

FIAP



## Consulta dos Dados utilizando a linguagem SQL

### SELF JOIN ou AUTOJUNÇÃO

Comando SQL :

-- EXEMPLO AUTOJUNÇÃO (SELF JOIN) – PADRÃO SQL/99

```
SELECT FUNC.Matricula ,  
       FUNC.NOME "FUNCIONÁRIO",  
       FUNC.GERENTE "COD. GERENTE",  
       GER.NOME "GERENTE"  
FROM T_AULA16_2_FUNCIONARIO FUNC INNER JOIN  
     T_AULA16_2_FUNCIONARIO GER  
ON ( GER.MATRICULA = FUNC.GERENTE );
```



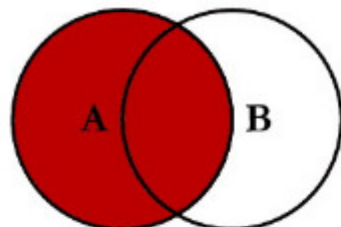
# Linguagem SQL

FIAP

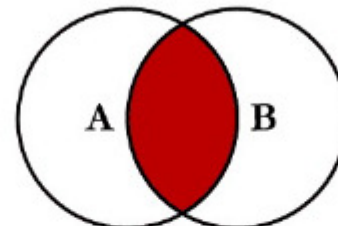
## Consulta dos Dados utilizando a linguagem SQL



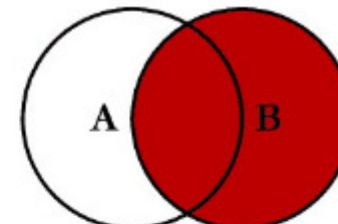
### SQL JOINS



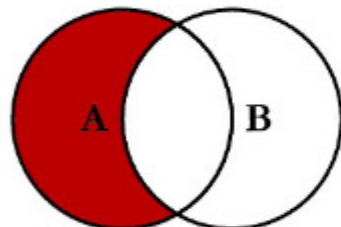
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



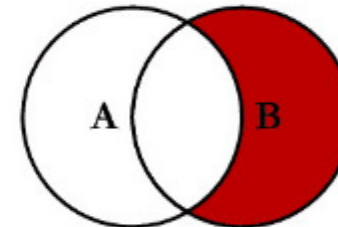
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



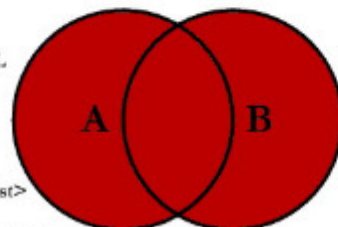
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



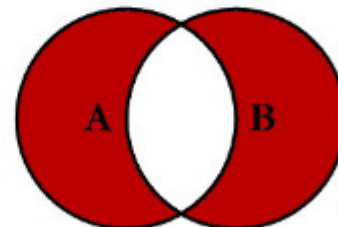
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```



---

## **| Próxima aula estudaremos**

FIAP

- ☐ Revisão de conceitos através de exercícios

## I REFERÊNCIAS



- MACHADO, Felipe Nery R. Banco de Dados - Projeto e Implementação. Érica, 2004.
- Páginas: 330, 331.
- ELMASRI, R.; NAVATHE, S.B. Sistemas de Banco de Dados: Fundamentos e Aplicações. Pearson, 2005. Páginas: 153, 154.
- PRICE, JASON, ORACLE DATABASE 11 g – SQL Domine SQL e PL-SQL no banco de Dados Oracle, Bookman, 2008. Capítulos: 2.
- **Outros:**
- Manual Oficial Oracle – Introdução ao Oracle 9i (SQL) - **Oracle Corporation, 2000, 2001.**

**Copyright © 2016 Profa. Rita de Cássia Rodrigues**

**Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).**