

MySQL – RELACIONAMENTOS INNER JOIN

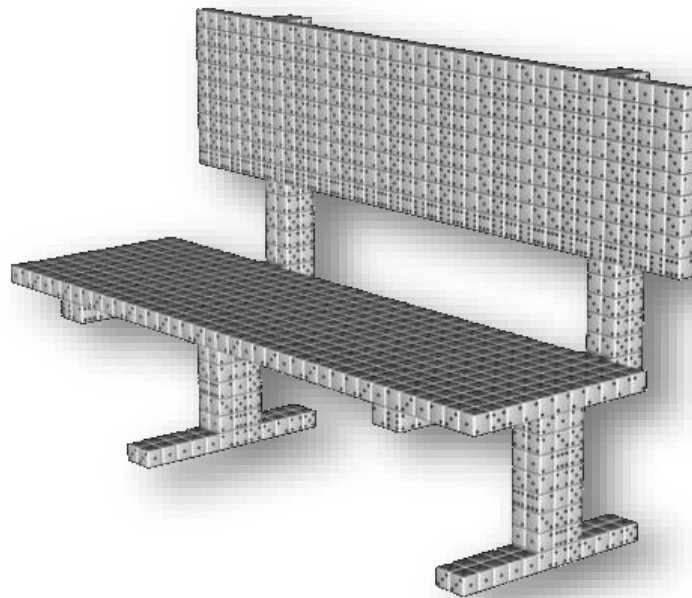
FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HERCULANO DE BIASI

herculano.debiasi@unoesc.edu.br

TÓPICOS

- Restrição de integridade referencial
- Operações de junção
- Junções internas (INNER JOINS)



RESTRIÇÃO DE INTEGRIDADE REFERENCIAL

❑ Atributos

- ❑ PRIMARY KEY (PK) : Define o campo como chave primária
- ❑ BINARY (B) : Armazena caracteres no formato binário, sendo *case sensitive*
- ❑ UNSIGNED: Aceita apenas valores positivos em campos numéricos
- ❑ ZEROFILL (ZF) : Preenche campos numéricos com 0s à esquerda até o tamanho máximo do campo que foi informado na sua criação (atributo depreciado no MySQL 8)
- ❑ AUTO_INCREMENT (AI) : O MySQL automaticamente incrementa o valor do campo
 - ❑ É possível definir o valor inicial do autoincremento ou então modificar o próximo valor
 - ❑ É possível também definir qual será o passo de autoincremento (por padrão é 1)
- ❑ GENERATED (G) : Coluna 'gerada' a partir de fórmulas ou derivadas de outras colunas, podendo ser virtual (VIRTUAL, que é o padrão) ou armazenada (STORED)

RESTRIÇÃO DE INTEGRIDADE REFERENCIAL

❑ Restrições (*constraints*)

- ❑ NOT NULL (NN) : Não aceita valores nulos no campo
- ❑ UNIQUE (UQ) : Valor do campo deverá ser único, ou seja, dentro da tabela não serão aceitos dois registros diferentes com o mesmo valor para o campo
- ❑ DEFAULT (DF) : Permite definir um valor padrão para a coluna
 - ❑ Este valor será inserido automaticamente nos registros se nenhum valor for especificado para a coluna em questão
- ❑ CHECK (CK) : Assegura que os valores de uma coluna satisfazem certas condições
- ❑ PRIMARY KEY (PK) : A restrição de chave primária é utilizada para identificar cada registro de forma única
- ❑ FOREIGN KEY (FK) : Também chamada de restrição de integridade referencial, especifica que o valor da coluna (ou grupo de colunas) deve corresponder a algum valor existente em uma linha de outra tabela
- ❑ INDEX (IN) : Usado para criar e recuperar dados de uma forma muito eficiente

RESTRIÇÃO DE INTEGRIDADE REFERENCIAL

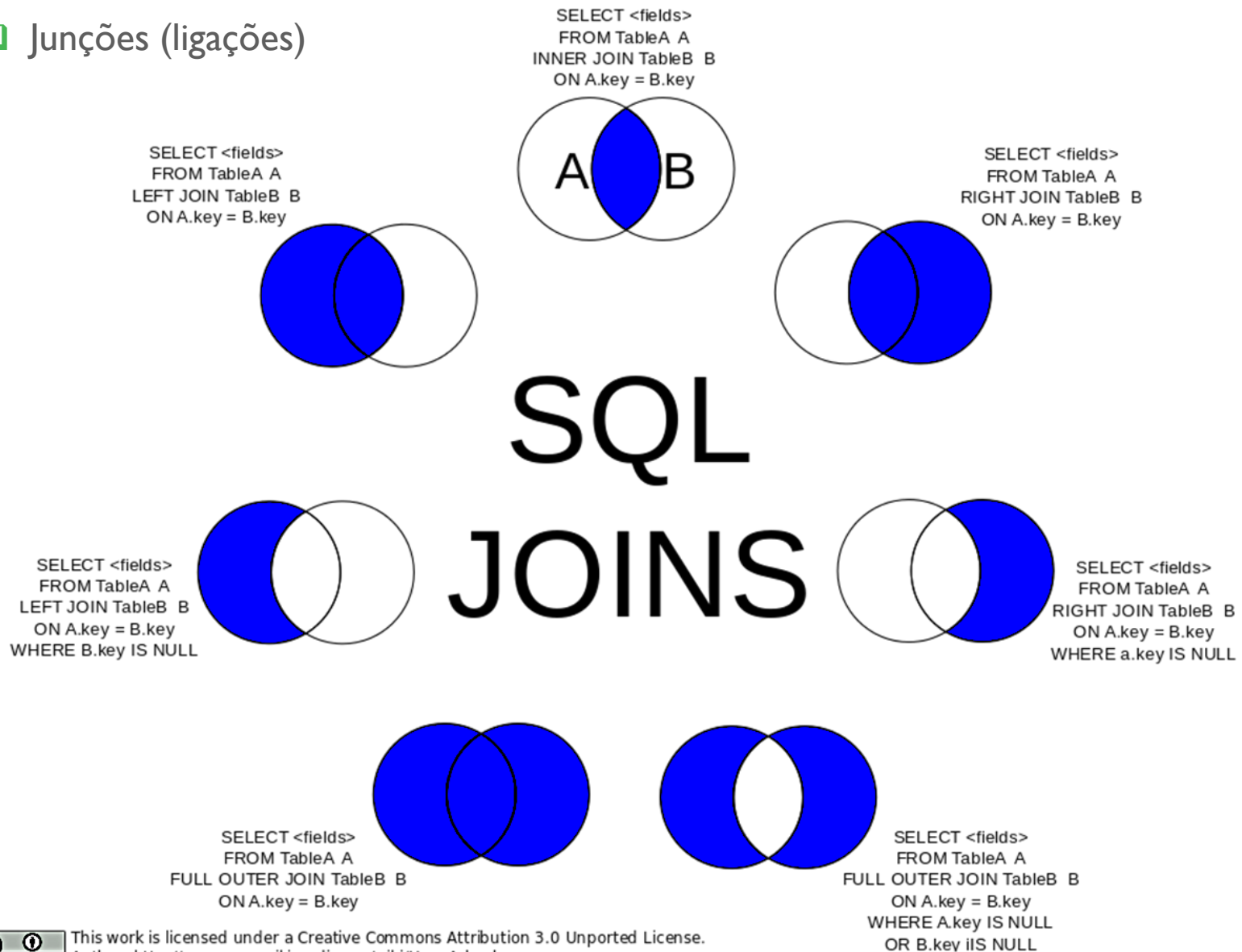
- Restrição de integridade referencial: Em um banco de dados relacional, quando um registro de uma tabela depende de outro em uma outra tabela, é necessário criar regras ou restrições para que o registro “pai” não possa ser excluído se ele tiver “filhos” (dependentes)
- Este tipo de controle é chamado de integridade referencial, pois tem como objetivo manter a integridade (coerência) das tabelas
- As restrições de integridade referencial são implementadas através de chave primária na tabela pai e chave estrangeira na tabela filha
- Os 2 principais tipos de integridade referencial são
 - CASCADE: Quando removemos um registro da tabela usando a chave estrangeira, os registros relacionados ao que foi removido serão automaticamente eliminados
 - RESTRICT: Não permite a remoção de registros que possuam relacionamentos em outras tabelas – trata-se de uma opção mais segura

OPERAÇÕES DE JUNÇÃO

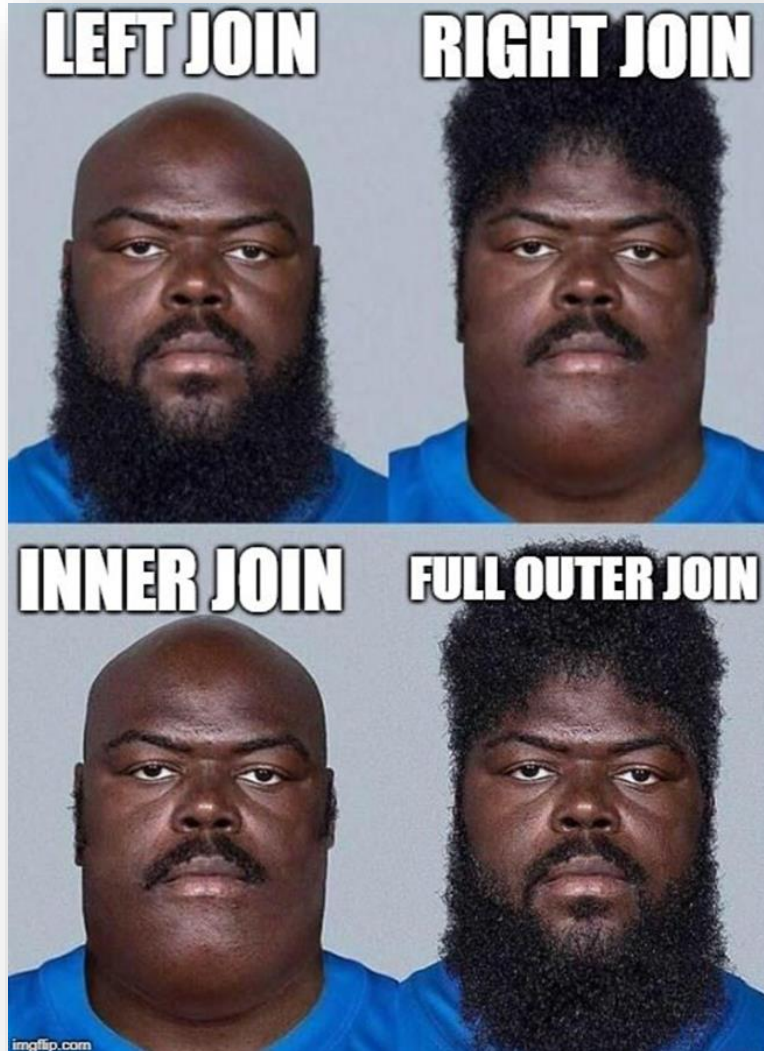
- ❑ Junção interna (INNER JOIN)
- ❑ Junções externas (OUTER JOIN)
 - ❑ Externa à esquerda (LEFT OUTER JOIN)
 - ❑ Externa à direita (RIGHT OUTER JOIN)
 - ❑ Externa total, ou completa (FULL OUTER JOIN)

OPERAÇÕES DE JUNÇÃO

■ Junções (ligações)



OPERAÇÕES DE JUNÇÃO



JUNÇÕES INTERNAS (INNER JOINS)

- ❑ Uma junção, ou ligação interna é caracterizada por uma seleção que retorna apenas os dados que atendem às condições de junção, isto é, quais linhas de uma tabela se relacionam com as linhas de outras tabelas
- ❑ Há duas formas diferentes de expressar esta junção
 - ❑ Explícita utilizando a palavra `JOIN`
 - ❑ Implícita utilizando ',' para separar as tabelas a combinar na cláusula `FROM` do `SELECT`, gerando assim o produto cruzado do qual combinações desejadas devem ser selecionadas através da cláusula `WHERE`

JUNÇÕES INTERNAS (INNER JOINS)

- ❑ No padrão SQL:2003, a operação de junção pode ser expressa diretamente na cláusula `FROM`, utilizando as palavras-chaves `[INNER] JOIN` em vez de ser expressa nas cláusulas `FROM` e `WHERE`

```
SELECT NumOfer, NumCurso, NomeProf, SobrenomeProf
FROM Oferecimento, Professor
WHERE TrimestreOfer = 'OUTONO' AND AnoOfer = 2005
    AND ClassificacaoProf = 'ASSISTENTE' AND NumCurso LIKE 'SI%'
    AND Professor.CPFProf = Oferecimento.CPFProf
```

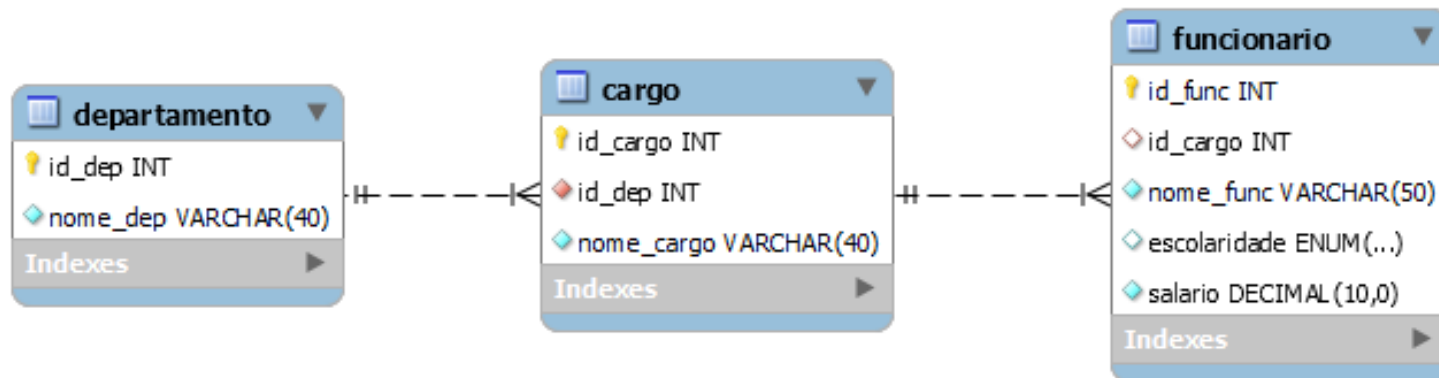
```
SELECT NumOfer, NumCurso, NomeProf, SobrenomeProf
FROM Oferecimento INNER JOIN Professor
    ON Professor.CPFProf = Oferecimento.CPFProf
WHERE TrimestreOfer = 'OUTONO' AND AnoOfer = 2005
    AND ClassificacaoProf = 'ASSISTENTE' AND NumCurso LIKE 'SI%'
```

JUNÇÕES INTERNAS (INNER JOINS)

- ❑ É necessário ter algum cuidado quando se combinam colunas com valores nulos (NULL), já que o valor nulo não se combina com outro valor, ou outro valor nulo, exceto quando se agregam predicados como `IS NULL` ou `IS NOT NULL`
- ❑ Tuplas nas quais os atributos de junção possuem o valor `NULL`, ou para as quais a condição de junção é falsa, não aparecem no resultado da operação `JOIN`
- ❑ Se nenhuma combinação de tuplas satisfaz a condição de junção, o resultado da operação `JOIN` é uma relação vazia (sem tuplas)

JUNÇÕES INTERNAS (INNER JOINS)

Exemplo com 3 tabelas



id_dep	nome_dep
1	Tecnologia da Informação
2	Marketing
3	Financeiro

id_cargo	id_dep	nome_cargo
1	1	Analista de Sistemas
2	1	Programador
3	2	Analista de Marketing
4	2	Designer
5	3	Analista de Negócios
6	3	Contador

id_func	id_cargo	nome_func	escolaridade	salario
1	2	André	Especialização	1000
2	1	Fabiana	Ensino fundamental	2000
3	3	Luis	Graduação	700
4	5	Alfredo	Especialização	2300
5	4	Patricia	Doutorado	2600
6	6	Ricardo	Graduação	1000
7	5	Thais	Graduação	1100
8	6	Pedro	Ensino médio	900
9	4	Flavia	Mestrado	2300
10	2	Fernanda	Ensino médio	800

JUNÇÕES INTERNAS (INNER JOINS)

Exemplo com 3 tabelas

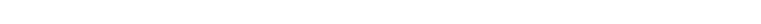
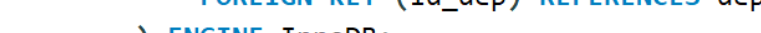
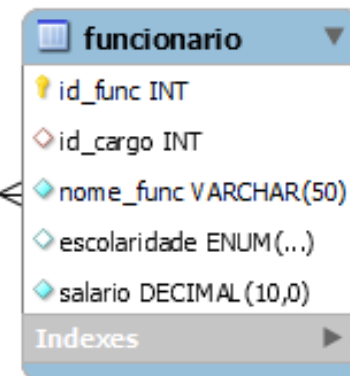
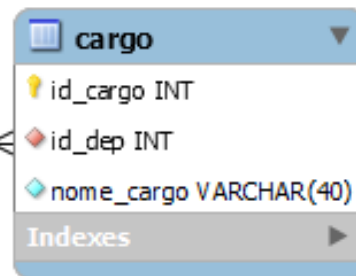
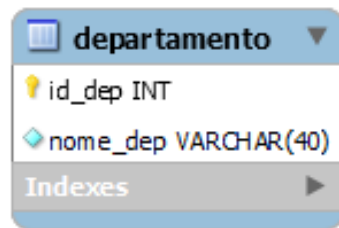
id_func	nome_dep	nome_cargo	nome_func	escolaridade	salario
1	Tecnologia da Informação	Programador	André	Especialização	1000
2	Tecnologia da Informação	Analista de Sistemas	Fabiana	Ensino fundamental	2000
3	Marketing	Analista de Marketing	Luis	Graduação	700
4	Financeiro	Analista de Negócios	Alfredo	Especialização	2300
5	Marketing	Designer	Patricia	Doutorado	2600
6	Financeiro	Contador	Ricardo	Graduação	1000
7	Financeiro	Analista de Negócios	Thais	Graduação	1100
8	Financeiro	Contador	Pedro	Ensino médio	900
9	Marketing	Designer	Flavia	Mestrado	2300
10	Tecnologia da Informação	Programador	Fernanda	Ensino médio	800



JUNÇÕES INTERNAS (INNER JOINS)

Exemplo com 3 tabelas

```
CREATE TABLE IF NOT EXISTS funcionario (  
  id_func INT AUTO_INCREMENT PRIMARY KEY,  
  id_cargo INT,  
  nome_func VARCHAR(50) NOT NULL,  
  escolaridade ENUM('Ensino fundamental', 'Ensino médio',  
                    'Graduação', 'Especialização', 'Mestrado',  
                    'Doutorado'),  
  salario DECIMAL NOT NULL,  
  FOREIGN KEY(id_cargo) REFERENCES cargo(id_cargo)  
) ENGINE=InnoDB;
```



```
CREATE TABLE IF NOT EXISTS departamento (  
  id_dep INT AUTO_INCREMENT PRIMARY KEY,  
  nome_dep VARCHAR(40) NOT NULL  
) ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS cargo (  
  id_cargo INT AUTO_INCREMENT PRIMARY KEY,  
  id_dep INT NOT NULL,  
  nome_cargo VARCHAR(40) NOT NULL,  
  FOREIGN KEY (id_dep) REFERENCES departamento(id_dep)  
) ENGINE=InnoDB;
```

JUNÇÕES INTERNAS (INNER JOINS)

Exemplo de CROSS JOIN

id_dep	nome_dep
1	Tecnologia da Informação
2	Marketing
3	Financeiro

Departamentos

id_cargo	id_dep	nome_cargo
1	1	Analista de Sistemas
2	1	Programador
3	2	Analista de Marketing
4	2	Designer
5	3	Analista de Negócios
6	3	Contador

Cargos

id_func	id_cargo	nome_func	escolaridade	salario
1	2	André	Especialização	1000
2	1	Fabiana	Ensino fundamental	2000
3	3	Luis	Graduação	700
4	5	Alfredo	Especialização	2300
5	4	Patricia	Doutorado	2600
6	6	Ricardo	Graduação	1000
7	5	Thais	Graduação	1100
8	6	Pedro	Ensino médio	900
9	4	Flavia	Mestrado	2300
10	2	Fernanda	Ensino médio	800

Funcionários

	id_dep	nome_dep	id_cargo	id_dep	nome_cargo
▶	1	Tecnologia da Informação	1	1	Analista de Sistemas
	1	Tecnologia da Informação	2	1	Programador
	1	Tecnologia da Informação	3	2	Analista de Marketing
	1	Tecnologia da Informação	4	2	Designer
	1	Tecnologia da Informação	5	3	Analista de Negócios
	1	Tecnologia da Informação	6	3	Contador
	2	Marketing	1	1	Analista de Sistemas
	2	Marketing	2	1	Programador
	2	Marketing	3	2	Analista de Marketing
	2	Marketing	4	2	Designer
	2	Marketing	5	3	Analista de Negócios
	2	Marketing	6	3	Contador
	3	Financeiro	1	1	Analista de Sistemas
	3	Financeiro	2	1	Programador
	3	Financeiro	3	2	Analista de Marketing
	3	Financeiro	4	2	Designer
	3	Financeiro	5	3	Analista de Negócios
	3	Financeiro	6	3	Contador

```
SELECT * FROM departamento, cargo
ORDER BY departamento.id_dep;
```

JUNÇÕES INTERNAS (INNER JOINS)

Exemplos de consultas

id_dep	nome_dep
1	Tecnologia da Informação
2	Marketing
3	Financeiro

Departamentos

id_cargo	id_dep	nome_cargo
1	1	Analista de Sistemas
2	1	Programador
3	2	Analista de Marketing
4	2	Designer
5	3	Analista de Negócios
6	3	Contador

Cargos

id_func	id_cargo	nome_func	escolaridade	salario
1	2	André	Especialização	1000
2	1	Fabiana	Ensino fundamental	2000
3	3	Luis	Graduação	700
4	5	Alfredo	Especialização	2300
5	4	Patricia	Doutorado	2600
6	6	Ricardo	Graduação	1000
7	5	Thais	Graduação	1100
8	6	Pedro	Ensino médio	900
9	4	Flavia	Mestrado	2300
10	2	Fernanda	Ensino médio	800

Funcionários

```
SELECT nome_cargo, nome_dep
FROM cargo c, departamento d
WHERE c.id_dep=d.id_dep;
```

```
SELECT nome_cargo, nome_dep FROM cargo c
JOIN departamento d ON c.id_dep=d.id_dep;
```

```
SELECT nome_cargo, nome_dep FROM cargo
JOIN departamento USING(id_dep);
```

```
SELECT nome_cargo, nome_dep FROM cargo
NATURAL JOIN departamento;
```

	nome_cargo	nome_dep
►	Analista de Sistemas	Tecnologia da Informação
	Programador	Tecnologia da Informação
	Analista de Marketing	Marketing
	Designer	Marketing
	Analista de Negócios	Financeiro
	Contador	Financeiro

JUNÇÕES INTERNAS (INNER JOINS)

Exemplos de consultas

id_dep	nome_dep
1	Tecnologia da Informação
2	Marketing
3	Financeiro

Departamentos

id_cargo	id_dep	nome_cargo
1	1	Analista de Sistemas
2	1	Programador
3	2	Analista de Marketing
4	2	Designer
5	3	Analista de Negócios
6	3	Contador

Cargos

id_func	id_cargo	nome_func	escolaridade	salario
1	2	André	Especialização	1000
2	1	Fabiana	Ensino fundamental	2000
3	3	Luis	Graduação	700
4	5	Alfredo	Especialização	2300
5	4	Patricia	Doutorado	2600
6	6	Ricardo	Graduação	1000
7	5	Thais	Graduação	1100
8	6	Pedro	Ensino médio	900
9	4	Flavia	Mestrado	2300
10	2	Fernanda	Ensino médio	800

Funcionários

```
SELECT nome_func, f.id_cargo, nome_cargo
FROM funcionario f, cargo c
WHERE f.id_cargo=c.id_cargo
ORDER BY id_func;
```

```
SELECT nome_func, f.id_cargo, nome_cargo
FROM funcionario f
JOIN cargo c ON f.id_cargo=c.id_cargo
ORDER BY id_func;
```

```
SELECT nome_func, id_cargo, nome_cargo
FROM funcionario
JOIN cargo USING(id_cargo)
ORDER BY id_func;
```

```
SELECT nome_func, id_cargo, nome_cargo
FROM funcionario
NATURAL JOIN cargo
ORDER BY id_func;
```

	nome_func	id_cargo	nome_cargo
▶	André	2	Programador
	Fabiana	1	Analista de Sistemas
	Luis	3	Analista de Marketing
	Alfredo	5	Analista de Negócios
	Patricia	4	Designer
	Ricardo	6	Contador
	Thais	5	Analista de Negócios
	Pedro	6	Contador
	Flavia	4	Designer
	Fernanda	2	Programador

JUNÇÕES INTERNAS (INNER JOINS)

Exemplos de consultas

id_dep	nome_dep
1	Tecnologia da Informação
2	Marketing
3	Financeiro

Departamentos

id_cargo	id_dep	nome_cargo
1	1	Analista de Sistemas
2	1	Programador
3	2	Analista de Marketing
4	2	Designer
5	3	Analista de Negócios
6	3	Contador

Cargos

id_func	id_cargo	nome_func	escolaridade	salario
1	2	André	Especialização	1000
2	1	Fabiana	Ensino fundamental	2000
3	3	Luis	Graduação	700
4	5	Alfredo	Especialização	2300
5	4	Patricia	Doutorado	2600
6	6	Ricardo	Graduação	1000
7	5	Thais	Graduação	1100
8	6	Pedro	Ensino médio	900
9	4	Flavia	Mestrado	2300
10	2	Fernanda	Ensino médio	800

Funcionários

```
SELECT nome_func, nome_cargo, nome_dep
FROM funcionario f, cargo c, departamento d
WHERE f.id_cargo=c.id_cargo AND c.id_dep=d.id_dep
ORDER BY id_func;
```

```
SELECT nome_func, nome_cargo, nome_dep
FROM funcionario f
JOIN cargo c ON f.id_cargo=c.id_cargo
JOIN departamento d ON c.id_dep=d.id_dep
ORDER BY id_func;
```

```
SELECT nome_func, nome_cargo, nome_dep
FROM funcionario
JOIN cargo USING(id_cargo)
JOIN departamento USING(id_dep)
ORDER BY id_func;
```

```
SELECT nome_func, nome_cargo, nome_dep
FROM funcionario
NATURAL JOIN cargo
NATURAL JOIN departamento
ORDER BY id_func;
```

	nome_func	nome_cargo	nome_dep
▶	André	Programador	Tecnologia da Informação
	Fabiana	Analista de Sistemas	Tecnologia da Informação
	Luis	Analista de Marketing	Marketing
	Alfredo	Analista de Negócios	Financeiro
	Patricia	Designer	Marketing
	Ricardo	Contador	Financeiro
	Thais	Analista de Negócios	Financeiro
	Pedro	Contador	Financeiro
	Flavia	Designer	Marketing
	Fernanda	Programador	Tecnologia da Informação