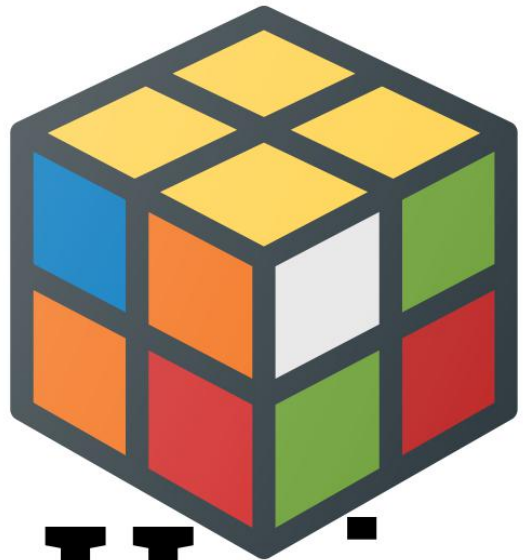


Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Junção de Tabelas

Junção de Tabelas

Contextualizando...

Junção de Tabelas

Contextualizando...

Em um banco de dados podemos ter duas ou mais tabelas relacionadas.

É bastante comum que ao elaborarmos uma consulta termos a necessidade de trazer dados de diferentes tabelas (Nós vimos uma forma de fazer isso na aula 03 deste módulo).

Para criarmos esta seleção de dados devemos definir os critérios de agrupamento para trazer estes dados. Estes critérios são chamados de **Junções**.

Uma junção de tabelas cria uma pseudo-tabela derivada de duas ou mais tabelas de acordo com as regras especificadas, e que são parecidas com as regras da **Teoria dos Conjuntos**.

OBS: Não se preocupe com os códigos SQL executados nesta aula. Iremos executá-los no banco de dados nas próximas seções quando tivermos os servidores banco de dados devidamente instalados e configurados.

OBS: As diferentes junções muitas vezes são formas diferentes de se fazer a mesma coisa.

Junção de Tabelas

Base de Dados utilizada...

Junção de Tabelas

Base de Dados utilizada...

```sql
CREATE DATABASE juncao;

USE juncao;

CREATE TABLE profissoes(
    id INT NOT NULL AUTO_INCREMENT,
    cargo VARCHAR(60) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE clientes(
    id INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(60) NOT NULL,
    data_nascimento DATE NOT NULL,
    telefone VARCHAR(10) NOT NULL,
    id_profissao int NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_profissao) REFERENCES profissoes(id)
);

```

Junção de Tabelas

Base de Dados utilizada...

```sql
CREATE DATABASE juncao;

USE juncao;

CREATE TABLE profissoes(
    id INT NOT NULL AUTO_INCREMENT,
    cargo VARCHAR(60) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE clientes(
    id INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(60) NOT NULL,
    data_nascimento DATE NOT NULL,
    telefone VARCHAR(10) NOT NULL,
    id_profissao int NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_profissao) REFERENCES profissoes(id)
);

```

Comando SQL para criar o banco de dados. (DDL);

Junção de Tabelas

Base de Dados utilizada...

```sql
CREATE DATABASE juncao;

USE juncao;

CREATE TABLE profissoes(
    id INT NOT NULL AUTO_INCREMENT,
    cargo VARCHAR(60) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE clientes(
    id INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(60) NOT NULL,
    data_nascimento DATE NOT NULL,
    telefone VARCHAR(10) NOT NULL,
    id_profissao int NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_profissao) REFERENCES profissoes(id)
);

```

Comando SQL para utilizar um banco de dados (DML);

Junção de Tabelas

Base de Dados utilizada...

```sql
CREATE DATABASE juncao;

USE juncao;

CREATE TABLE profissoes(
    id INT NOT NULL AUTO_INCREMENT,
    cargo VARCHAR(60) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE clientes(
    id INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(60) NOT NULL,
    data_nascimento DATE NOT NULL,
    telefone VARCHAR(10) NOT NULL,
    id_profissao int NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_profissao) REFERENCES profissoes(id)
);
```

Comando SQL para criar tabela (DDL);

Aqui estamos criando uma tabela chamada 'profissoes' com dois campos: id: inteiro, não nulo e auto incremento; e cargo: caracteres variados com tamanho máximo 60 caracteres, não nulo.

Definimos como chave primária desta tabela o campo 'id'.

Junção de Tabelas

Base de Dados utilizada...

```sql
CREATE DATABASE juncao;

USE juncao;

CREATE TABLE profissoes(
    id INT NOT NULL AUTO_INCREMENT,
    cargo VARCHAR(60) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE clientes(
    id INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(60) NOT NULL,
    data_nascimento DATE NOT NULL,
    telefone VARCHAR(10) NOT NULL,
    id_profissao int NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_profissao) REFERENCES profissoes(id)
);
```

Comando SQL para criar tabela (DDL);

Aqui estamos criando uma tabela chamada 'clientes' com cinco campos:
id: inteiro, não nulo e auto incremento;
nome: caracteres variados com tamanho máximo 60 caracteres, não nulo.
data_nascimento: data, não nulo;
telefone: caracteres variados com tamanho máximo 10 caracteres, não nulo;
id_profissao: inteiro, não nulo;

Definimos como chave primária desta tabela o campo 'id'.

Definimos que esta tabela tem uma chave estrangeira, relacionando o campo 'id_profissao' com a tabela 'profissoes'.

Junção de Tabelas

Base de Dados utilizada...

```sql
INSERT INTO profissoes (cargo) VALUES ('Programador');
INSERT INTO profissoes (cargo) VALUES ('Analista de Sistemas');
INSERT INTO profissoes (cargo) VALUES ('Suporte');
INSERT INTO profissoes (cargo) VALUES ('Gerente');

INSERT INTO clientes (nome, data_nascimento, telefone, id_profissao) VALUES ('João Pereira', '1981-06-15', '1234-5688', 1);
INSERT INTO clientes (nome, data_nascimento, telefone, id_profissao) VALUES ('Ricardo da Silva', '1973-10-10', '2234-5669', 2);
INSERT INTO clientes (nome, data_nascimento, telefone, id_profissao) VALUES ('Felipe Oliveira', '1987-08-01', '4234-5640', 3);
INSERT INTO clientes (nome, data_nascimento, telefone, id_profissao) VALUES ('Mário Pirez', '1991-02-05', '5234-5621', 1);
```

Utilizando comandos SQL (DML) para inserir dados nas tabelas.

OBS: Note o formato da data de nascimento; yyyy-mm-dd

Junção de Tabelas

Junção de produto cartesiano...

Uma junção de produto cartesiano é uma junção entre duas tabelas que origina uma terceira tabela constituída por todos os elementos da primeira combinadas com todos os elementos da segunda.

Junção de Tabelas

Junção de produto cartesiano...Exemplo

Aprendemos em outras aulas que se quisermos selecionar todos os dados da tabela profissoes podemos fazer:

*SELECT * FROM profissoes;*

Junção de Tabelas

Junção de produto cartesiano...Exemplo

Aprendemos em outras aulas que se quisermos selecionar todos os dados da tabela profissoes podemos fazer:

*SELECT * FROM profissoes;*

id	cargo
1	Programador
2	Analista de Sistemas
3	Suporte
4	Gerente

Junção de Tabelas

Junção de produto cartesiano...Exemplo

Desta forma, se quisermos selecionar todos os dados da tabela clientes podemos fazer:

*SELECT * FROM clientes;*

Junção de Tabelas

Junção de produto cartesiano...Exemplo

Desta forma, se quisermos selecionar todos os dados da tabela clientes podemos fazer:

*SELECT * FROM clientes;*

id	nome	data_nascimento	telefone	id_profissao
1	João Pereira	1981-06-15	1234-5688	1
2	Ricardo da Silva	1973-10-10	2234-5669	2
3	Felipe Oliveira	1987-08-01	4234-5640	3
4	Mário Pirez	1991-02-05	5234-5621	1

Junção de Tabelas

Junção de produto cartesiano...Exemplo

Ainda se quisermos selecionar todos os dados da tabela profissoes e clientes podemos fazer:

*SELECT * FROM profissoes, clientes;*

Junção de Tabelas

Junção de produto cartesiano...Exemplo

Ainda se quisermos selecionar todos os dados da tabela profissoes e clientes podemos fazer:

*SELECT * FROM profissoes, clientes;*

id	cargo	id	nome	data_nascimento	telefone	id_profissao
1	Programador	1	João Pereira	1981-06-15	1234-5688	1
2	Analista de Sistemas	1	João Pereira	1981-06-15	1234-5688	1
3	Suporte	1	João Pereira	1981-06-15	1234-5688	1
4	Gerente	1	João Pereira	1981-06-15	1234-5688	1
1	Programador	2	Ricardo da Silva	1973-10-10	2234-5669	2
2	Analista de Sistemas	2	Ricardo da Silva	1973-10-10	2234-5669	2
3	Suporte	2	Ricardo da Silva	1973-10-10	2234-5669	2
4	Gerente	2	Ricardo da Silva	1973-10-10	2234-5669	2
1	Programador	3	Felipe Oliveira	1987-08-01	4234-5640	3
2	Analista de Sistemas	3	Felipe Oliveira	1987-08-01	4234-5640	3
3	Suporte	3	Felipe Oliveira	1987-08-01	4234-5640	3
4	Gerente	3	Felipe Oliveira	1987-08-01	4234-5640	3
1	Programador	4	Mário Pirez	1991-02-05	5234-5621	1
2	Analista de Sistemas	4	Mário Pirez	1991-02-05	5234-5621	1
3	Suporte	4	Mário Pirez	1991-02-05	5234-5621	1
4	Gerente	4	Mário Pirez	1991-02-05	5234-5621	1

Junção de Tabelas

Junção de produto cartesiano...Exemplo

Ainda se quisermos selecionar todos os dados da tabela profissoes e clientes podemos fazer:

*SELECT * FROM profissoes, clientes;*

id	cargo	id	nome	data_nascimento	telefone	id_profissao
1	Programador	1	João Pereira	1981-06-15	1234-5688	1
2	Analista de Sistemas	1	João Pereira	1981-06-15	1234-5688	1
3	Suporte	1	João Pereira	1981-06-15	1234-5688	1
4	Gerente	1	João Pereira	1981-06-15	1234-5688	1
1	Programador	2	Ricardo da Silva	1973-10-10	2234-5669	2
2	Analista de Sistemas	2	Ricardo da Silva	1973-10-10	2234-5669	2
3	Suporte	2	Ricardo da Silva	1973-10-10	2234-5669	2
4	Gerente	2	Ricardo da Silva	1973-10-10	2234-5669	2
1	Programador	3	Felipe Oliveira	1987-08-01	4234-5640	3
2	Analista de Sistemas	3	Felipe Oliveira	1987-08-01	4234-5640	3
3	Suporte	3	Felipe Oliveira	1987-08-01	4234-5640	3
4	Gerente	3	Felipe Oliveira	1987-08-01	4234-5640	3
1	Programador	4	Mário Pirez	1991-02-05	5234-5621	1
2	Analista de Sistemas	4	Mário Pirez	1991-02-05	5234-5621	1
3	Suporte	4	Mário Pirez	1991-02-05	5234-5621	1
4	Gerente	4	Mário Pirez	1991-02-05	5234-5621	1

Note que temos um problema com o resultado dos dados selecionados.

Note que para cada cliente foi repetido os nomes para cada profissão.

Já sabemos como resolver isso com o que aprendemos na aula passada...

Junção de Tabelas

Junção de produto cartesiano...Exemplo

Ainda se quisermos selecionar todos os dados da tabela profissoes e clientes podemos fazer:

```
SELECT c.id, c.nome, c.data_nascimento, c.telefone, p.cargo
        FROM clientes AS c, profissoes AS p
        WHERE c.id_profissao = p.id;
```

Junção de Tabelas

Junção de produto cartesiano...Exemplo

Ainda se quisermos selecionar todos os dados da tabela profissoes e clientes podemos fazer:

```
SELECT c.id, c.nome, c.data_nascimento, c.telefone, p.cargo
        FROM clientes AS c, profissoes AS p
        WHERE c.id_profissao = p.id;
```

id	nome	data_nascimento	telefone	cargo
1	João Pereira	1981-06-15	1234-5688	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	Suporte
4	Mário Pirez	1991-02-05	5234-5621	Programador

Veja que agora os dados retornados pela consulta fazem sentido. Cada cliente tem somente 1 cargo.

Descobrimos desta forma que aquela consulta que fizemos na aula passada com múltiplas tabelas, assim como esta, se trata da Junção de Produto Cartesiano.

Junção de Tabelas

Junção Interna (Inner Join)...

Junção de Tabelas

Junção Interna (Inner Join)...

Uma junção interna é caracterizada por uma consulta 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.

Para isso utilizamos a cláusula **ON**, que é semelhante à cláusula **WHERE**.

Junção de Tabelas

Junção Interna (Inner Join)...Exemplo

```
SELECT c.id, c.nome, c.data_nascimento, c.telefone, p.cargo
FROM clientes AS c INNER JOIN profissoes AS p
ON  c.id_profissao = p.id;
```

No exemplo acima teremos o mesmo resultado da junção por produto cartesiano.

Junção de Tabelas

Junção Interna (Inner Join)...Exemplo

```
SELECT c.id, c.nome, c.data_nascimento, c.telefone, p.cargo
FROM clientes AS c INNER JOIN profissoes AS p
ON  c.id_profissao = p.id;
```

No exemplo acima teremos o mesmo resultado da junção por produto cartesiano.

id	nome	data_nascimento	telefone	cargo
1	João Pereira	1981-06-15	1234-5688	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	Suporte
4	Mário Pirez	1991-02-05	5234-5621	Programador

Junção de Tabelas

Junção Externa (Outer Join)...

Junção de Tabelas

Junção Externa (Outer Join)...

Uma junção externa é uma consulta que não requer que os registros de uma tabela possuam registros equivalentes em outra.

Este tipo de junção se subdivide dependendo da tabela do qual admitiremos os registros que não possuem correspondência: a tabela da esquerda, a direita ou ambas.

Junção de Tabelas

Junção Externa (Outer Join)...Left Outer Join

O resultado desta consulta sempre contém todos os registros da tabela esquerda (ou seja, a primeira tabela mencionada na consulta), mesmo quando não exista registros correspondentes na tabela direira.

Desta forma, esta consulta retorna todos os valores da tabela esquerda com os valores da tabela direita correspondente, ou quando não há correspondência retorna um valor NULL.

Junção de Tabelas

Junção Externa (Outer Join)...Left Outer Join..Exemplo

*SELECT * FROM clientes*
LEFT OUTER JOIN profissoes
ON clientes.id_profissao = profissoes.id;

No exemplo acima temos, todos os dados das tabelas clientes e profissoes com os dados da tabela profissoes correspondendo aos dados da tabela clientes (tabela esquerda).

id	nome	data_nascimento	telefone	id_profissao	id	cargo
1	João Pereira	1981-06-15	1234-5688	1	1	Programador
4	Mário Pirez	1991-02-05	5234-5621	1	1	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	2	2	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	3	3	Suporte

Junção de Tabelas

Junção Externa (Outer Join)...Right Outer Join

Esta consulta é inversa à anterior e retorna sempre todos os registros da tabela à direita (a segunda tabela mencionada na consulta), mesmo se não existir registro correspondente na tabela à esquerda. Nestes casos, o valor NULL é retornado quando não há correspondência.

Junção de Tabelas

Junção Externa (Outer Join)...Right Outer Join..Exemplo

*SELECT * FROM clientes*
RIGHT OUTER JOIN profissoes
ON clientes.id_profissao = profissoes.id;

No exemplo acima, trazemos novamente os dados das duas tabelas mas desta vez os dados da tabela da direita (profissoes) foi apresentada de acordo com a tabela à esquerda.

Junção de Tabelas

Junção Externa (Outer Join)...Right Outer Join..Exemplo

*SELECT * FROM clientes*
RIGHT OUTER JOIN profissoes
ON clientes.id_profissao = profissoes.id;

No exemplo acima, trazemos novamente os dados das duas tabelas mas desta vez os dados da tabela da direita (profissoes) foi apresentada de acordo com a tabela à esquerda.

id	nome	data_nascimento	telefone	id_profissao	id	cargo
1	João Pereira	1981-06-15	1234-5688	1	1	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	2	2	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	3	3	Suporte
4	Mário Pirez	1991-02-05	5234-5621	1	1	Programador
NULL	NULL	NULL	NULL	NULL	4	Gerente

Junção de Tabelas

Junção Externa (Outer Join)...Right Outer Join..Exemplo

*SELECT * FROM clientes*
RIGHT OUTER JOIN profissoes
ON clientes.id_profissao = profissoes.id;

No exemplo acima, trazemos novamente os dados das duas tabelas mas desta vez os dados da tabela da direita (profissoes) foi apresentada de acordo com a tabela à esquerda.

id	nome	data_nascimento	telefone	id_profissao	id	cargo
1	João Pereira	1981-06-15	1234-5688	1	1	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	2	2	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	3	3	Suporte
4	Mário Pirez	1991-02-05	5234-5621	1	1	Programador
NULL	NULL	NULL	NULL	NULL	4	Gerente

Junção de Tabelas

Junção Externa (Outer Join)...Full Outer Join

Junção de Tabelas

Junção Externa (Outer Join)...Full Outer Join

Esta consulta apresenta todos os dados das tabelas à esquerda e à direita, mesmo que não possuam correspondência em outra tabela. A tabela combinada possuirá assim todos os registros de ambas as tabelas e apresentará os valores nulos para os registros sem correspondência.

Junção de Tabelas

Junção Externa (Outer Join)...Full Outer Join..Exemplo

*SELECT * FROM clientes*
FULL OUTER JOIN profissoes
ON clientes.id_profissao = profissoes.id;

Esta consulta traz então os dados de ambas tabelas de acordo com suas correspondências e caso não tenha preenche o valor com NULL.

* Esta junção não funciona no MySQL mas pode ser simulada utilizando um LEFT JOIN e um RIGHT JOIN.

Junção de Tabelas

Junção Externa (Outer Join)...Full Outer Join..Exemplo

*SELECT * FROM clientes*
FULL OUTER JOIN profissoes
ON clientes.id_profissao = profissoes.id;

Esta consulta traz então os dados de ambas tabelas de acordo com suas correspondências e caso não tenha preenche o valor com NULL.

* Esta junção não funciona no MySQL mas pode ser simulada utilizando um LEFT JOIN e um RIGHT JOIN.

id	nome	data_nascimento	telefone	id_profissao	id	cargo
1	João Pereira	1981-06-15	1234-5688	1	1	Programador
4	Mário Pirez	1991-02-05	5234-5621	1	1	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	2	2	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	3	3	Suporte
NULL	NULL	NULL	NULL	NULL	4	Gerente

Junção de Tabelas

Junção Externa (Outer Join)...Full Outer Join..Exemplo

```
SELECT * FROM clientes
LEFT OUTER JOIN profissoes
ON clientes.id_profissao = profissoes.id
UNION
SELECT * FROM clientes
RIGHT OUTER JOIN profissoes
ON clientes.id_profissao = profissoes.id;
```

* Versão MySQL.

id	nome	data_nascimento	telefone	id_profissao	id	cargo
1	João Pereira	1981-06-15	1234-5688	1	1	Programador
4	Mário Pirez	1991-02-05	5234-5621	1	1	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	2	2	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	3	3	Suporte
NULL	NULL	NULL	NULL	NULL	4	Gerente

Junção de Tabelas

Junção Cruzada (Cross Join)...

Junção de Tabelas

Junção Cruzada (Cross Join)...

Esta consulta é usada quando queremos juntar duas ou mais tabelas por cruzamento. Ou seja, para cada linha de uma tabela queremos todos os dados da outra tabela ou vice-versa.

Junção de Tabelas

Junção Cruzada (Cross Join)...Exemplo

```
SELECT c.id, c.nome, c.data_nascimento, c.telefone, p.cargo
FROM clientes AS c
CROSS JOIN profissoes AS p;
```

Neste exemplo para cada cliente colocamos um linha com cada profissão.

Junção de Tabelas

Junção Cruzada (Cross Join)...Exemplo

```
SELECT c.id, c.nome, c.data_nascimento, c.telefone, p.cargo
FROM clientes AS c
CROSS JOIN profissoes AS p;
```

Neste exemplo para cada cliente colocamos um linha com cada profissão.

id	nome	data_nascimento	telefone	cargo
1	João Pereira	1981-06-15	1234-5688	Programador
1	João Pereira	1981-06-15	1234-5688	Analista de Sistemas
1	João Pereira	1981-06-15	1234-5688	Suporte
1	João Pereira	1981-06-15	1234-5688	Gerente
2	Ricardo da Silva	1973-10-10	2234-5669	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	Analista de Sistemas
2	Ricardo da Silva	1973-10-10	2234-5669	Suporte
2	Ricardo da Silva	1973-10-10	2234-5669	Gerente
3	Felipe Oliveira	1987-08-01	4234-5640	Programador
3	Felipe Oliveira	1987-08-01	4234-5640	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	Suporte
3	Felipe Oliveira	1987-08-01	4234-5640	Gerente
4	Mário Pirez	1991-02-05	5234-5621	Programador
4	Mário Pirez	1991-02-05	5234-5621	Analista de Sistemas
4	Mário Pirez	1991-02-05	5234-5621	Suporte
4	Mário Pirez	1991-02-05	5234-5621	Gerente

Junção de Tabelas

Auto Junção (Self Join)...

Junção de Tabelas

Auto Junção (Self Join)...

Esta consulta é uma auto junção de uma tabela a si mesma.

Junção de Tabelas

Auto Junção (Self Join)...Exemplo

Esta consulta é uma auto junção de uma tabela a si mesma.

```sql
CREATE TABLE consumidor(
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(50) NOT NULL,
    contato VARCHAR(50) NOT NULL,
    endereco VARCHAR(100) NOT NULL,
    cidade VARCHAR(100) NOT NULL,
    cep VARCHAR(20) NOT NULL,
    pais VARCHAR(50) NOT NULL
);


INSERT INTO consumidor (nome, contato, endereco, cidade, cep, pais) VALUES ('Alfredo Nunes', 'Maria Nunes', 'Rua da paz, 47', 'São Paulo', '123.456-87', 'Brasil');
INSERT INTO consumidor (nome, contato, endereco, cidade, cep, pais) VALUES ('Ana Trujillo', 'Guilherme Souza', 'Rua Dourada, 452', 'Goiânia', '232.984-23', 'Brasil');
INSERT INTO consumidor (nome, contato, endereco, cidade, cep, pais) VALUES ('Leandro Veloz', 'Pedro Siqueira', 'Rua Vazia, 72', 'São Paulo', '936.738-23', 'Brasil');
```

Criamos uma nova tabela e **populamos** a mesma para servir como exemplo.

Junção de Tabelas

Auto Junção (Self Join)...Exemplo

```
SELECT a.nome AS Consumidor1, b.nome AS Consumidor2, a.cidade
FROM consumidor AS a
INNER JOIN consumidor AS b
ON a.id <> b.id
AND a.cidade = b.cidade;
```

No exemplo acima estamos realizando um Self Join onde os ids sejam diferentes mas a cidade seja igual.

Junção de Tabelas

Auto Junção (Self Join)...Exemplo

```
SELECT a.nome AS Consumidor1, b.nome AS Consumidor2, a.cidade
FROM consumidor AS a
INNER JOIN consumidor AS b
ON a.id <> b.id
AND a.cidade = b.cidade;
```

No exemplo acima estamos realizando um Self Join onde os ids sejam diferentes mas a cidade seja igual.

Consumidor1	Consumidor2	cidade
Alfredo Nunes	Leandro Veloz	São Paulo
Leandro Veloz	Alfredo Nunes	São Paulo

Junção de Tabelas

Junção Baseada em Comparador (Equi-Join)...

Junção de Tabelas

Junção Baseada em Comparador (Equi-Join)...

Uma junção **Equi-Join** é um tipo específico de junção baseada em comparador, que usa apenas comparações de igualdade na junção.

Junção de Tabelas

Junção Baseada em Comparador (Equi-Join)...Exemplo

*SELECT **
FROM clientes JOIN profissoes
ON clientes.id_profissao = profissoes.id;

Neste exemplo estamos realizando um **Equi-Join** utilizando como comparador os campos de relacionamento.

Junção de Tabelas

Junção Baseada em Comparador (Equi-Join)...Exemplo

*SELECT **
FROM clientes JOIN profissoes
ON clientes.id_profissao = profissoes.id;

Neste exemplo estamos realizando um **Equi-Join** utilizando como comparador os campos de relacionamento.

id	nome	data_nascimento	telefone	id_profissao	id	cargo
1	João Pereira	1981-06-15	1234-5688	1	1	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	2	2	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	3	3	Suporte
4	Mário Pirez	1991-02-05	5234-5621	1	1	Programador

Junção de Tabelas

Junção Natutal (Natural Join)...

Junção de Tabelas

Junção Natutal (Natural Join)...

Uma junção **Natural-Join** é um caso especial de Equi-Join. O resultado desta junção é o conjunto de todas as combinações que são iguais em seus nomes de atributos comuns.

Junção de Tabelas

Junção Natutal (Natural Join)...Exemplo

```
SELECT *
FROM clientes NATURAL JOIN profissoes;
```

Neste exemplo a junção natural acontece com os campos comuns do relacionamento.

Junção de Tabelas

Junção Natutal (Natural Join)...Exemplo

```
SELECT *
FROM clientes NATURAL JOIN profissoes;
```

Neste exemplo a junção natural acontece com os campos comuns* do relacionamento.

id	nome	data_nascimento	telefone	id_profissao	cargo
1	João Pereira	1981-06-15	1234-5688	1	Programador
2	Ricardo da Silva	1973-10-10	2234-5669	2	Analista de Sistemas
3	Felipe Oliveira	1987-08-01	4234-5640	3	Suporte
4	Mário Pirez	1991-02-05	5234-5621	1	Gerente

* O único campo comum entre as duas tabelas é o campo 'id'. Desta forma, veja que o resultado está errado, já que João Pereira e Mário Pirez tem a mesma profissão...

Esta junção só 'funciona' bem se os campos chaves (onde acontece os relacionamentos) tiverem o mesmo nome em ambas as tabelas.

Junção de Tabelas

Resumo...

Junção de produto cartesiano é uma junção entre duas tabelas que origina uma terceira tabela constituída por todos os elementos da primeira tabela combinada com todos os elementos da segunda.

Junção Interna (Inner Join) todas as linhas de uma tabela se relacionam com todas as linhas de outras tabelas se elas tiverem ao menos 1 campo em comum.

Junção Externa Esquerda (Left Outer Join) traz todos os registros da tabela esqueda mesmo quando não exista registros correspondentes na tabela direita.

Junção Externa Direita (Right Outer Join) traz todos os registros da tabela direita mesmo quando não exista registros correspondentes na tabela esquerda.

Junção Externa Completa (Full Outer Join) apresenta todos os dados das tabelas à esquerda e à direita, mesmo que não possuam correspondência em outra tabela.

Junção de Tabelas

Resumo...

Junção Cruzada (Cross Join) é uma junção entre todos os campos de ambas as tabelas.

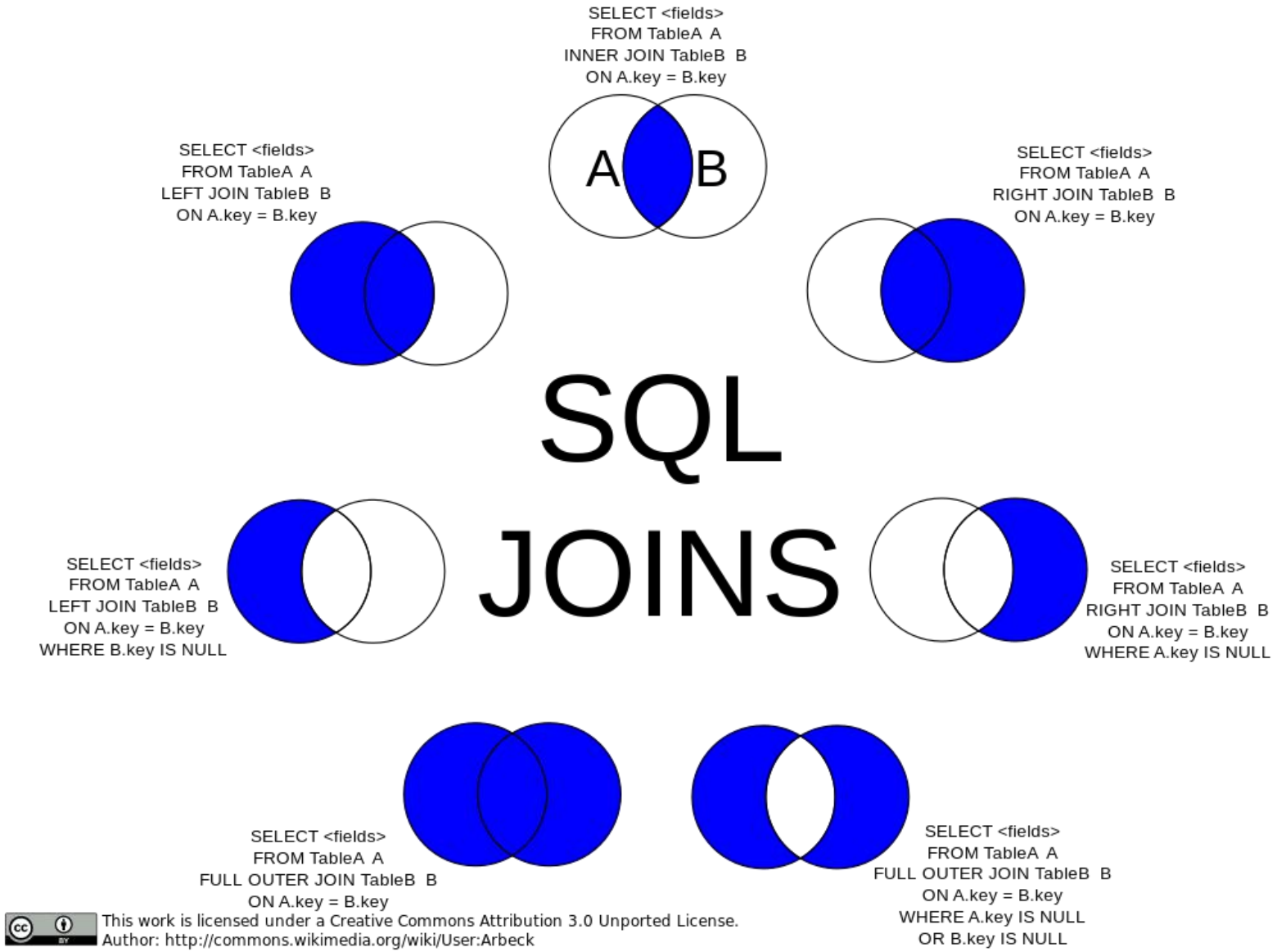
Auto Junção (Self Join) realiza uma auto junção da própria tabela.

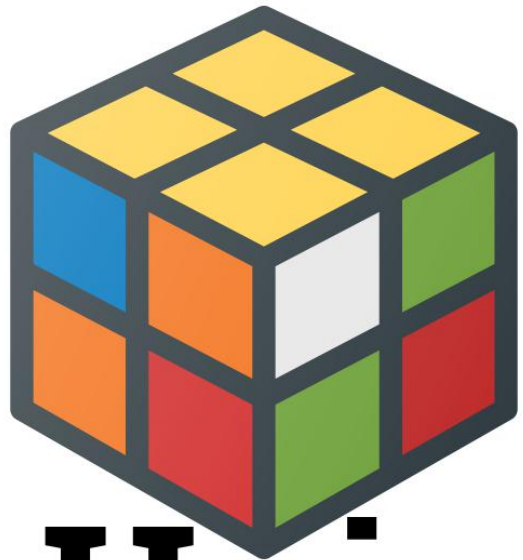
Junção Baseada em Comparador (Equi-Join) traz todos os registros das tabelas utilizando operador de comparação.

Junção Natural (Natural Join) traz todos os registros das tabelas de acordo com os nomes de atributos em comum.

Junção de Tabelas

Resumo...





Geek University

Evolua seu lado geek!

www.geekuniversity.com.br