FIEP SESI SENAI IEL

SENAI

Linguagem Do Banco de Dados

Prof. Sergio Luiz

Conteúdo

FIEP SESI SENAI IEL

SENAI

Prof. Sergio Luiz

01 - Junções- Join

Introdução à junções

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

É comum ao elaborarmos uma consulta termos a necessidade de trazer dados de diferentes tabelas.

Sergio.lsilveira@sesisenaipr.org.br

01 - Junções- Join

Introdução à junções

Para criarmos esta seleção devemos definir os critérios de agrupamento para trazer estes dados.

Estes critérios são chamados de Junções.

Sergio.lsilveira@sesisenaipr.org.br

01 - Junções- Join

Introdução à 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.

Sergio.lsilveira@sesisenaipr.org.br

02 - 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**.

Sergio.lsilveira@sesisenaipr.org.br

02 - Junção de produto cartesiano

Para trazer apenas os campos necessários da consulta:

- No **SELECT** utilizamos em vez do nome do campo simples, o **nome_tabela.nome_campo;**
- Em **FROM** utilizamos os nomes das tabelas que possuem os campos que queremos trazer;
- Em **WHERE** determinamos a dependência das tabelas, lembrando que ***a relação entre as tabelas é efetuada pela chave estrangeira***.
- Desta forma o **WHERE** sempre especifica as chaves estrangeiras que ligam as tabelas.

Sergio.lsilveira@sesisenaipr.org.br

02 - Junção de produto cartesiano

Como exemplo, vamos imaginar que possuímos duas tabelas relacionadas:

Cliente **e** **Profissão**

A tabela **Profissão** contém o atributo:

Cod_Profissao,

Nome_Profissao

Sergio.lsilveira@sesisenaipr.org.br

02 - Junção de produto cartesiano

```
CREATE TABLE Profissao (
Cod_Profissao INT NOT NULL,
Nome_Profissao VARCHAR (60) NOT NULL,
PRIMARY KEY (Cod_Profissao));
```

Sergio.lsilveira@sesisenaipr.org.br

02 - Junção de produto cartesiano

Agora inserimos dados nestas tabelas.
Em Profissão vamos inserir três profissões básicas:

```
INSERT INTO Profissao (
Cod_Profissao,
Nome_Profissao)
VALUES
('0025','Programador'),
('0026','Analista de BD'),
('0027','Suporte'),
('0028','Estagiario');
```

Sergio.lsilveira@sesisenaipr.org.br

02 - Junção de produto cartesiano

A **tabela Cliente** armazena os dados pessoais do Cliente

```
CREATE TABLE Cliente1 (
Cod_Cliente INT NOT NULL,
Nome_Cliente VARCHAR (60) NOT NULL,
Data_Nascimento DATE,
Telefone CHAR (9),
Cod_Profissao INT,
PRIMARY KEY (Cod_Cliente) ,
FOREIGN KEY (Cod_Profissao)
REFERENCES Profissao(Cod_Profissao));
```

Sergio.lsilveira@sesisenaipr.org.br

02 - Junção de produto cartesiano

Possuímos agora três profissões com respectivos códigos.

Agora vamos inserir dados na tabela Cliente:

Sergio.lsilveira@sesisenaipr.org.br

02 - Junção de produto cartesiano

```
INSERT INTO Cliente1 (
Cod_Cliente,
Nome_Cliente,
Data_Nascimento,
Telefone,
Cod_Profissao )
VALUES
('121','João Pereira','1980-09-20','3456-7890',25),
('122','Maria Barros','1972-01-22','3456-7891',26),
('123','José Mendes','1983-04-29','3456-7892',27),
('124','Rogerio Cavalcante','1990-01-12','3456-7894',28);
```

Sergio.lsilveira@sesisenaipr.org.br

02 - Junção de produto cartesiano

A **tabela Pedido** vai armazenar dados referente a pedidos do cliente.

```
CREATE TABLE Pedido (
Num_Pedido INT NOT NULL,
Cod_Cliente INT,
Tot_Pedido DECIMAL(10,2),
PRIMARY KEY (Num_Pedido) ,
FOREIGN KEY (Cod_Cliente)
REFERENCES Cliente1(Cod_Cliente));
```

Sergio.lsilveira@sesisenaipr.org.br

02 - Junção de produto cartesiano

Insira pedidos na **tabela Pedido**

```
INSERT INTO Pedido (
Num_Pedido,
Cod_Cliente,
Tot_Pedido)
VALUES
('203', '121',800.00),
('204', '122',900.00),
('205', '123',1200.00);
```

Sergio.lsilveira@sesisenaipr.org.br

FIEP SESI SENAI IEL

SENAI

VISUALIZANDO DADOS

Junção por Nome de Coluna

Conceito

03 - Visualizando Dados

Para visualizarmos **todos os dados contidos nas duas tabelas** após a inserção dos dados, podemos utilizar:

Sergio.lsilveira@sesisenaipr.org.br

03 - Visualizando Dados

Para visualizarmos **todos os dados contidos nas duas tabelas** após a inserção dos dados, podemos utilizar:

```
SELECT * FROM Cliente1, Profissao;
```

Sergio.lsilveira@sesisenaipr.org.br

03 - Visualizando Dados

Resultado da Consulta

Cod_Cliente	Nome_Cliente	Data_Nascimento	Telefone	Cod_Profissao	Cod_Profissao	Nome_Profissao
121	João Pereira	1980-09-20	3456-7890	25	25	Programador
122	Maria Barros	1972-01-22	3456-7891	26	25	Programador
123	José Mendes	1983-04-29	3456-7892	27	25	Programador
124	Rogerio Cavalcante	1990-01-12	3456-7894	28	25	Programador
121	João Pereira	1980-09-20	3456-7890	25	26	Analista de BD
122	Maria Barros	1972-01-22	3456-7891	26	26	Analista de BD
123	José Mendes	1983-04-29	3456-7892	27	26	Analista de BD
124	Rogerio Cavalcante	1990-01-12	3456-7894	28	26	Analista de BD
121	João Pereira	1980-09-20	3456-7890	25	27	Suporte
122	Maria Barros	1972-01-22	3456-7891	26	27	Suporte
123	José Mendes	1983-04-29	3456-7892	27	27	Suporte
124	Rogerio Cavalcante	1990-01-12	3456-7894	28	27	Suporte
121	João Pereira	1980-09-20	3456-7890	25	28	Estagiario
122	Maria Barros	1972-01-22	3456-7891	26	28	Estagiario
123	José Mendes	1983-04-29	3456-7892	27	28	Estagiario
124	Rogerio Cavalcante	1990-01-12	3456-7894	28	28	Estagiario

Sergio.lsilveira@sesisenaipr.org.br

03 - Visualizando Dados

Porém, se quisermos trazer apenas o **Nome do Cliente** e o seu **Cargo**, podemos fazer uma junção de produto cartesiano.

Sergio.lsilveira@sesisenaipr.org.br

03 - Visualizando Dados

Nesta seleção, trazemos um campo de cada tabela após o **SELECT**, mencionamos as tabelas de quais elas se originam no **FROM**, e no **WHERE** especificamos a ligação entre as tabelas.

Sergio.lsilveira@sesisenaipr.org.br

03 - Visualizando Dados

Note que **Cod_Profissao** é a **chave estrangeira** na **tabela Cliente**, que referencia diretamente a ***chave primária*** da **tabela Profissao**.

Sergio.lsilveira@sesisenaipr.org.br

03 - Visualizando Dados

```
SELECT Nome_Cliente,Nome_Profissao
FROM Cliente1, Profissao WHERE
Profissao.Cod_Profissao=
Cliente1.Cod_Profissao;
```

Nome_Cliente	Nome_Profissao
João Pereira	Programador
Maria Barros	Analista de BD
José Mendes	Suporte
Rogerio Cavalcante	Estagiario

Sergio.lsilveira@sesisenaipr.org.br

FIEP SESI SENAI IEL

SENAI

VISUALIZANDO DADOS

Junção Interna (Inner Join)

Conceito

04 - Junção Interna (Inner Join)

Uma **Junção 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.

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

Sergio.lsilveira@sesisenaipr.org.br

04 - Junção Interna (Inner Join)

Podemos especificar **duas formas diferentes** de expressar esta junção: **a explícita utiliza a palavra JOIN**, enquanto **a implícita utiliza ','** para separar as tabelas a combinar na cláusula **FROM** do **SELECT**.

Sergio.lsilveira@sesisenaipr.org.br

04 - Junção Interna (Inner Join)

Então sempre é gerado o produto cruzado do qual são selecionadas as combinações que cumpram a cláusula **WHERE**.

Sergio.lsilveira@sesisenaipr.org.br

04 - Junção Interna (Inner Join)

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.**

Sergio.lsilveira@sesisenaipr.org.br

04 - Junção Interna (Inner Join)

Como exemplo, a consulta seguinte traz todos os registros da **tabela Cliente** e encontra todas as combinações com a **tabela Profissao**.

A cláusula **JOIN** compara os valores da ***coluna Profissao*** de **Cliente** com a ***coluna Codigo*** da **Profissao**.

Sergio.lsilveira@sesisenaipr.org.br

04 - Junção Interna (Inner Join)

Quando não existe os dados não atendem as condições especificadas, eles não são retornados.

Sergio.lsilveira@sesisenaipr.org.br

FIEP SESI SENAI IEL

SENAI

PRATICANDO

Prof. Sergio Luiz

04 - Junção Interna (Inner Join)

```
SELECT
Cliente1.Nome_Cliente,Pedido.Num_Pedi
do FROM Cliente1 INNER JOIN Pedido ON
Cliente1.Cod_cliente =
Pedido.Cod_Cliente;
```

Nome_Cliente	Num_Pedido
João Pereira	203
Maria Barros	204
José Mendes	205

Sergio.lsilveira@sesisenaipr.org.br

VISUALIZANDO DADOS

Junção Externa (Outer Join)

Conceito

05 - Junção Externa (Outer Join)

Uma **Junção Externa** é uma seleção que não requer que os registros de uma tabela possuam registros equivalentes em outra.

O registro é mantido na pseudo-tabela se não existe outro registro que lhe corresponda.

Sergio.lsilveira@sesisenaipr.org.br

05 - Junção Externa (Outer Join)

Este tipo de junção se subdivide ***dependendo da tabela do qual admitiremos os registros que não possuem correspondência***:

- a tabela esquerda;
- a direita
- ou ambas.

Sergio.lsilveira@sesisenaipr.org.br

VISUALIZANDO DADOS

Junção Externa (Left Outer Join)

Conceito

06 - Junção Ext. (Left Outer Join)

O resultado desta seleção **sempre contém todos os registros da tabela esquerda** (***isto é, a primeira tabela mencionada na consulta***), mesmo quando não exista registros correspondentes na tabela direita.

Sergio.lsilveira@sesisenaipr.org.br

06 - Junção Ext. (Left Outer Join)

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

Sergio.lsilveira@sesisenaipr.org.br

06 - Junção Ext. (Left Outer Join)

Se por exemplo inserimos na **tabela Cliente** um **Cliente que não possua valor em seu campo Profissao**, ou **possua um valor que não tem correspondente no Codigo na tabela Profissão**, e efetuarmos a seleção com **LEFT OUTER JOIN** a seleção será efetuada trazendo todos os dados da **tabela Cliente**, e os correspondentes na **tabela Profissao**, e *quando não houver estes correspondentes,* ***trará o valor NULL***.

Sergio.lsilveira@sesisenaipr.org.br

FIEP SESI SENAI IEL

SENAI

PRATICANDO

Junção Externa (Left Outer Join)

Exemplo

06 - Junção Ext. (Left Outer Join)

```
SELECT DISTINCT * FROM Cliente1
LEFT OUTER JOIN Profissao ON
Cliente1.Cod_Profissao =
Profissao.Cod_Profissao;
```

Cod_Cliente	Nome_Cliente	Data_Nascimento	Telefone	Cod_Profissao
121	João Pereira	1980-09-20	3456-7890	25
122	Maria Barros	1972-01-22	3456-7891	26
123	José Mendes	1983-04-29	3456-7892	27
124	Rogerio Cavalcante	1990-01-12	3456-7894	28

Cod_Profissao	Nome_Profissao
25	Programador
26	Analista de BD
27	Suporte
28	Estagiario

Sergio.lsilveira@sesisenaipr.org.br

FIEP SESI SENAI IEL

SENAI

VISUALIZANDO DADOS

Junção Ext. (Right Outer Join)

Conceito

07 - Junção Ext. (Right Outer Join)

Esta operação 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.

Sergio.lsilveira@sesisenaipr.org.br

07 - Junção Ext. (Right Outer Join)

Nestes casos, o valor **NULL** é retornado quando não há correspondência.

Sergio.lsilveira@sesisenaipr.org.br

07 - Junção Ext. (Right Outer Join)

Como exemplo, imaginemos que possuímos **diversas Profissões com respectivos códigos que não possuem correspondentes na tabela Clientes**.

Esta consulta traz todas estas Profissões mesmo que não haja esta correspondência:

Sergio.lsilveira@sesisenaipr.org.br

07 - Junção Ext. (Right Outer Join)

```
SELECT * FROM Cliente1
RIGHT OUTER JOIN Profissao ON
Cliente1.Cod_Profissao =
Profissao.Cod_Profissao;
```

Cod_Cliente	Nome_Cliente	Data_Nascimento	Telefone	Cod_Profissao	Cod_Profissao	Nome_Profissao
121	João Pereira	1980-09-20	3456-7890	25	25	Programador
122	Maria Barros	1972-01-22	3456-7891	26	26	Analista de BD
123	José Mendes	1983-04-29	3456-7892	27	27	Suporte
124	Rogerio Cavalcante	1990-01-12	3456-7894	28	28	Estagiario
NULL	NULL	NULL	NULL	NULL	29	Web Designer

Sergio.lsilveira@sesisenaipr.org.br

FIEP SESI SENAI IEL

SENAI

VISUALIZANDO DADOS

Junção Ext.
(Full Outer Join)

Conceito

08 - Junção Ext. (Full Outer Join)

Esta **operação apresenta todos os dados das tabelas à esquerda e à direita**, mesmo que não possuam correspondência em outra tabela.

Sergio.lsilveira@sesisenaipr.org.br

08 - Junção Ext. (Full Outer Join)

A tabela combinada possuirá assim todos os registros de ambas as tabelas e apresentará valores nulos para os registros sem correspondência.

Sergio.lsilveira@sesisenaipr.org.br

08 - Junção Ext. (Full Outer Join)

EXEMPLO 01

DIGITE E DIGA O QUE ACONTECEU!

```
SELECT * FROM Cliente1
FULL OUTER JOIN Profissao ON
Cliente1.Cod_Profissao =
Profissao.Cod_Profissao;
```

NÃO FUNCIONOU NO MySQL

Sergio.lsilveira@sesisenaipr.org.br

08 - Junção Ext. (Full Outer Join)

EXEMPLO 02

DIGITE O CÓDIGO DO PROXIMO SLIDE
E DIGA O QUE ACONTECEU!

Sergio.lsilveira@sesisenaipr.org.br

08 - Junção Ext. (Full Outer Join)

```
SELECT * FROM Cliente1
LEFT JOIN Profissao ON
Cliente1.Cod_Profissao =
Profissao.Cod_Profissao
UNION
SELECT * FROM Cliente1
RIGHT JOIN Profissao ON
Cliente1.Cod_Profissao =
Profissao.Cod_Profissao WHERE
Cliente1.Cod_Profissao IS NULL;
```

Sergio.lsilveira@sesisenaipr.org.br

08 - Junção Ext. (Full Outer Join)

Cod_Cliente	Nome_Cliente	Data_Nascimento	Telefone	Cod_Profissao	Cod_Profissao	Nome_Profissao
121	João Pereira	1980-09-20	3456-7890	25	25	Programador
122	Maria Barros	1972-01-22	3456-7891	26	26	Analista de BD
123	José Mendes	1983-04-29	3456-7892	27	27	Suporte
124	Rogerio Cavalcante	1990-01-12	3456-7894	28	28	Estagiario
NULL	NULL	NULL	NULL	NULL	29	Web Designer

Sergio.lsilveira@sesisenaipr.org.br

08 - Junção Ext. (Full Outer Join)

```
SELECT * FROM Cliente1
LEFT JOIN Profissao ON
Cliente1.Cod_Profissao =
Profissao.Cod_Profissao
UNION
SELECT * FROM Cliente1
RIGHT JOIN Profissao ON
Cliente1.Cod_Profissao =
Profissao.Cod_Profissao WHERE
Cliente1.Cod_Profissao IS NOT NULL;
```

Sergio.lsilveira@sesisenaipr.org.br

08 - Junção Ext. (Full Outer Join)

Cod_Cliente	Nome_Cliente	Data_Nascimento	Telefone	Cod_Profissao	Cod_Profissao	Nome_Profissao
121	João Pereira	1980-09-20	3456-7890	25	25	Programador
122	Maria Barros	1972-01-22	3456-7891	26	26	Analista de BD
123	José Mendes	1983-04-29	3456-7892	27	27	Suporte
124	Rogerio Cavalcante	1990-01-12	3456-7894	28	28	Estagiario

Sergio.lsilveira@sesisenaipr.org.br

09 - REFERENCIAS

MySQL Workbench – Junções - Prof.ª. Marlene da Silva Maximiano de Oliveira & Prof.ª. Alessandra Aparecida da Silva

Sergio.lsilveira@sesisenaipr.org.br

FIM

Sergio.lsilveira@sesisenaipr.org.br