

3075 - Fundamentos de banco de dados

2204 - Banco de dados

Curso de Sistemas de Informação

Prof. Daniella Vieira

Unidade II

CONCEITOS: MODELAGEM DE DADOS

- Modelos de banco de dados.
- O modelo relacional: domínios, tuplas, atributos, relações, entre outros.
- Diagrama de entidade-relacionamento.
- Mapeamento do modelo conceitual para o modelo relacional.

Referência Bibliográfica

NAVATHE, Elmasri. **Sistema de Banco de Dados**. 6ª Edição. São Paulo: Person Addison Wesley, 2011.



Modelagem de dados

- O Modelo Relacional (MR) representa os dados em um banco de dados como uma coleção de relações (tabelas).
- Cada linha é denominada registro; o nome de uma coluna é chamado de atributo; a tabela é chamada de relação.

EMPREGADO

Matricula	Nome	Sexo	Salário
111	PEDRO	M	R\$ 1.000,00
222	MARIA	F	R\$ 2.000,00
333	JOAO	M	R\$ 120,00
444	ANA	F	R\$ 120,00
321	CARLOS	M	R\$ 150,00
123	CLAUDIA	F	R\$ 359,00
001	MARCOS	M	R\$ 120,00

Domínio

- Um domínio D é uma coleção de valores atômicos (que não podem ser divididos). Um tipo elementar.
- Um domínio está associado a colunas de tabelas.
- Exemplo:
 - Matrícula:** conjunto de valores de três dígitos, numéricos, positivos e inteiros.
 - Nome:** conjunto de caracteres alfabéticos;
 - Sexo:** um caracter alfabético;
 - Salário:** conjunto de valores numéricos monetários, entre 120,00 e 2000,00.

Esquema

- Um esquema de relação é usado para descrever uma relação.
- Um esquema de relação R , denotado por $R(A_1, A_2, \dots, A_n)$, é um conjunto de atributos $R = \{A_1, A_2, \dots, A_n\}$.
- O grau de uma relação corresponde ao número de atributos n de seu esquema de relação.
- No exemplo, a relação EMPREGADO, pode ser representada segundo o seguinte esquema de relação:

EMPREGADO (Matrícula, Nome, Sexo, Salário)

* possui 4 atributos (grau = 4)

Esquema

A relação, ou instância da relação, é modificada com o tempo, para refletir as alterações do mundo real.

O esquema de uma relação é mais estático do que a instância da relação (a alteração do esquema da relação ocorre, por exemplo, quando um novo atributo é adicionado).

Chave de uma relação

- Atributo chave de uma relação
 - Todos os registros de uma relação devem ser distintos. Assim, dois registros não devem ter a mesma combinação de valores para todos os seus atributos.
 - O subconjunto mínimo de atributos de uma relação onde não existem dois registros com a mesma combinação de valores para tais atributos é dito chave da relação. (É comum ocorrer várias chaves candidatas para chave da relação).
 - O valor chave pode ser usado para identificar unicamente um registro em uma relação - chave primária da relação (*primary key* – PK)

Chave de uma relação

- Atributo chave de uma relação
 - A restrição de integridade de entidade indica que nenhum valor de chave primária pode assumir valor **nulo**.
 - A restrição de integridade referencial é uma restrição que é especificada entre duas relações, sendo utilizada para manter a consistência associada ao registro de duas relações (um registro em uma relação está relacionada com um registro em outra relação).
 - A integridade referencial entre relações é implementada através de chave estrangeira (*foreign key* - FK).

Chave de uma relação

- Atributo chave de uma relação

Considere dois esquemas de relação R1 e R2, onde um conjunto de atributos em R1 é denominado chave estrangeira (FK) se satisfaz as seguintes condições:

EMPREGADO (Matricula, Nome, Sexo, Salário, *NumDepto*)

DEPTO (NumDepto, NomeDepto)

Observe o atributo **NumDepto** no esquema de relação **EMPREGADO**. Este atributo é uma chave estrangeira (FK), que referencia a chave primária (PK) no esquema de relação **DEPTO**.

Álgebra relacional

A álgebra relacional consiste em um conjunto de operações utilizadas para manipular relações.

Uma consulta em um banco de dados que segue o modelo relacional é realizada através da aplicação de operações da álgebra relacional.

O resultado de cada operação da álgebra relacional é uma nova relação, que pode ser manipulada por outras operações da álgebra relacional. Assim, as operações da álgebra relacional são realizadas sobre relações inteiras.

A operação de seleção é utilizada para selecionar um subconjunto dos registros de uma relação (linhas de uma tabela), a partir de uma condição de seleção.

σ <condição de seleção> (<nome da relação>)

Álgebra relacional

A operação de seleção é unária; isto é, é aplicada em uma única relação.

O número de registros da relação resultante é sempre menor ou igual ao número de registros da relação utilizada na operação.

$\sigma_{\text{<condição de seleção>}} (\text{<nome da relação>})$

$$\sigma_{\text{sexo=M}} (\text{EMPREGADO})$$

Matrícula	Nome	Sexo	Salário
111	PEDRO	M	R\$ 1.000,00
333	JOAO	M	R\$ 120,00
321	CARLOS	M	R\$ 150,00
001	MARCOS	M	R\$ 120,00

Álgebra relacional

A relação resultante possui os mesmos atributos da relação utilizada na operação.

Na <condição de seleção>, pode-se utilizar os operadores de comparação { = , < , > , ≠ , ≤ , ≥ }. Outros operadores são também utilizados: NOT, AND e OR.

$\sigma_{(sexo=M)AND(salario>120)}(EMPREGADO)$

Matricula	Nome	Sexo	Salario
111	PEDRO	M	R\$ 1.000,00
321	CARLOS	M	R\$ 150,00

Produto Cartesiano

A operação de produto cartesiano é uma operação binária que combina os registros de ambas as relações envolvidas.

Considere duas relações $R (A_1, A_2, \dots, A_n)$ e $S (B_1, B_2, \dots, B_m)$

O resultado do produto cartesiano consiste em uma relação Q com $n + m$ atributos: $Q (A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$.

A relação resultante Q possui um registro para cada possível combinação de registros de R e S .

EMPREGADO (Matricula, Nome, Sexo, Salário, *NumDepto*)

DEPTO (NumDepto, NomeDepto)

RESULT = (Matricula, Nome, Sexo, Salário, ***NumDepto***, NumDepto, NomeDepto)

Operação Binária

Operação Binária: as relações que participam de tais operações devem ser união-compatíveis, significando que as relações $R (A_1, A_2, \dots, A_n)$ e $S (B_1, B_2, \dots, B_n)$:

- devem ter a mesma quantidade de atributos; e
- $D(A_i) = D(B_i)$, para $1 \leq i \leq n$.

Operação Binária

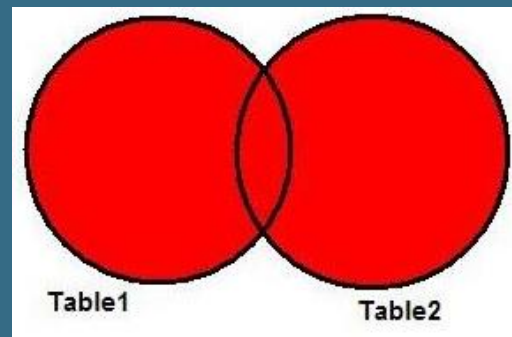
Operação de UNIÃO

A operação de união, denotada por $R \cup S$, resulta em uma relação que inclui todos os registros das relações R e S, onde os registros duplicados serão eliminadas.

TREM (ID_Trem, Chegada, Partina, Linha)

ESTACAO (ID_Estacao, Local)

RESULTADO (ID_Trem, Chegada, Partina, Linha, ID_Estacao, Local)



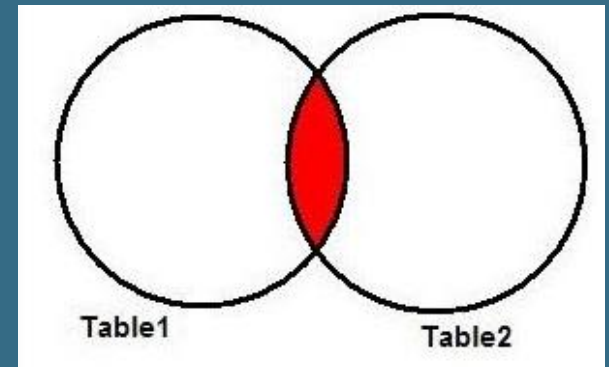
Operação Binária

Operação de interseção

A operação de interseção, denotada por $R \cap S$, resulta em uma relação que inclui todos os registros presentes em ambas as relações.

FUNCIONARIO (ID_Funcionario, Nome, Fone)

CLIENTE (ID_Cliente, Nome, Fone)



Operação Binária

Operação de diferença

- A operação de diferença, denotada por $R - S$, resulta em uma relação que inclui todos os registros de R que não estão presentes em S .

FUNCIONARIO (ID Funcionario, Nome, Fone)

ATESTADO (ID Atestado, Data, Motivo)



Junções

Junções

Operações de junção tomam duas relações e têm como resultado outra relação.

No SQL descreve-se o tipo de junção e condição da junção.

As condições de junção definem quais tuplas das duas relações apresentam correspondência e quais atributos são apresentados no resultado da junção.

Junção (Join)

A operação de junção (join) é utilizada para combinar registros relacionados de duas relações. Os dados são combinados de modo que podem ser comparados ou contrastados.

Resulta nas combinações de registros que satisfazem a condição, enquanto que o produto cartesiano resulta em todas as possíveis combinações de registros.

Matric	Nome	Sexo	Salário	Dep	CodDep	NomeDep
111	PEDRO	M	R\$ 1.000,00	VE	VE	VENDA
222	MARIA	F	R\$ 2.000,00	EN	EN	ENGENHARIA
123	CLAUDIA	F	R\$ 359,00	VE	VE	VENDA
001	MARCOS	M	R\$ 120,00	EN	EN	ENGENHARIA

Junção (Equi Join)

Um equijoin consiste em um join (junção) que utiliza o operador de igualdade (=) na condição de junção.

Matric	Nome	Sexo	Salário	Dep	CodDep	NomeDep
111	PEDRO	M	R\$ 1.000,00	VE	VE	VENDA
222	MARIA	F	R\$ 2.000,00	EN	EN	ENGENHARIA
123	CLAUDIA	F	R\$ 359,00	VE	VE	VENDA
001	MARCOS	M	R\$ 120,00	EN	EN	ENGENHARIA

Junção (Natural Join)

Um natural join consiste em um equijoin sem repetição de colunas envolvidas na condição de junção.

Matric	Nome	Sexo	Salário	Dep	NomeDep
111	PEDRO	M	R\$ 1.000,00	VE	VENDAS
222	MARIA	F	R\$ 2.000,00	EN	ENGENHARIA
123	CLAUDIA	F	R\$ 359,00	VE	VENDAS
001	MARCOS	M	R\$ 120,00	EN	ENGENHARIA

Junção (Outer Join)

Um outer join mantém no resultado da operação as linhas que não satisfazem a condição de um natural join.

Matric	Nome	Sexo	Salário	Depto	NomeDependente
111	PEDRO	M	R\$ 1.000,00	VE	PEDRO FILHO
222	MARIA	F	R\$ 2.000,00	EN	< null >
123	CLAUDIA	F	R\$ 359,00	VE	< null >
001	MARCOS	M	R\$ 120,00	EN	MARCOS JUNIOR

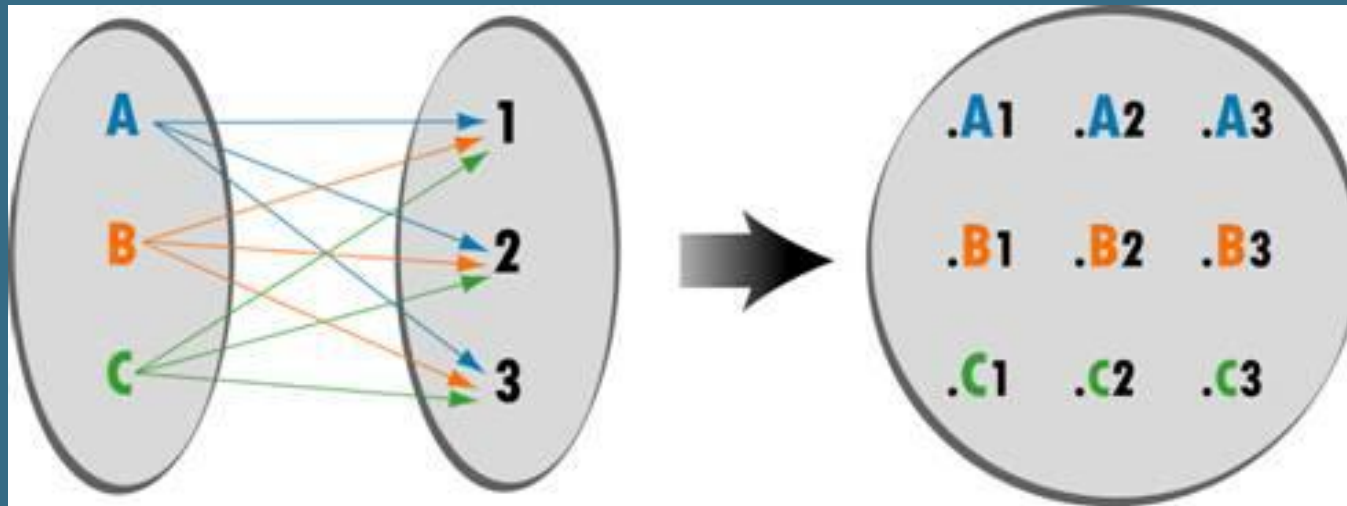
Junção de produto cartesiano

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.

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.

Junção de produto cartesiano



Clark	10
Miller	10
Smith	20
Turner	30

PRODUTO

10	Accounting
20	Research
30	Sales

=

Clark	10	10	Accounting
Clark	10	20	Research
Clark	10	30	Sales
Miller	10	10	Accounting
Miller	10	20	Research
Miller	10	30	Sales
Smith	20	10	Accounting
Smith	20	20	Research
Smith	20	30	Sales
Turner	30	10	Accounting
Turner	30	20	Research
Turner	30	30	Sales

Junção de produto cartesiano

-- Exercício

```
CREATE TABLE Profissao (  
    Prof_Codigo INT NOT NULL AUTO_INCREMENT,  
    Cargo VARCHAR (60) NOT NULL,  
    PRIMARY KEY (Prof_Codigo) );
```

```
CREATE TABLE Cliente(  
    Cli_Codigo INT NOT NULL AUTO_INCREMENT,  
    Nome VARCHAR (60) NOT NULL,  
    Data_Nascimento DATE,  
    Telefone CHAR (8),  
    FK_Profissao INT,  
    PRIMARY KEY (Cli_Codigo) ),  
    FOREIGN KEY (FK_Profissao) REFERENCES  
        Profissao(Prof_Codigo);
```

Junção de produto cartesiano

-- Em Profissao vamos inserir :

```
INSERT INTO Profissao VALUES (1, 'Programador')
```

```
INSERT INTO Profissao VALUES (2, 'Analista de BD')
```

```
INSERT INTO Profissao VALUES (3, 'Suporte')
```

-- Em Cliente vamos inserir :

```
INSERT INTO Cliente VALUES (1, 'João Pereira',  
19820606, '12345678', 1)
```

```
INSERT INTO Cliente VALUES (2, 'José Manuel',  
19750801, '21358271', 2)
```

```
INSERT INTO Cliente VALUES (3, 'Maria Mercedes',  
19851001, '85412587', 3)
```

Junção de produto cartesiano

-- Vamos executar

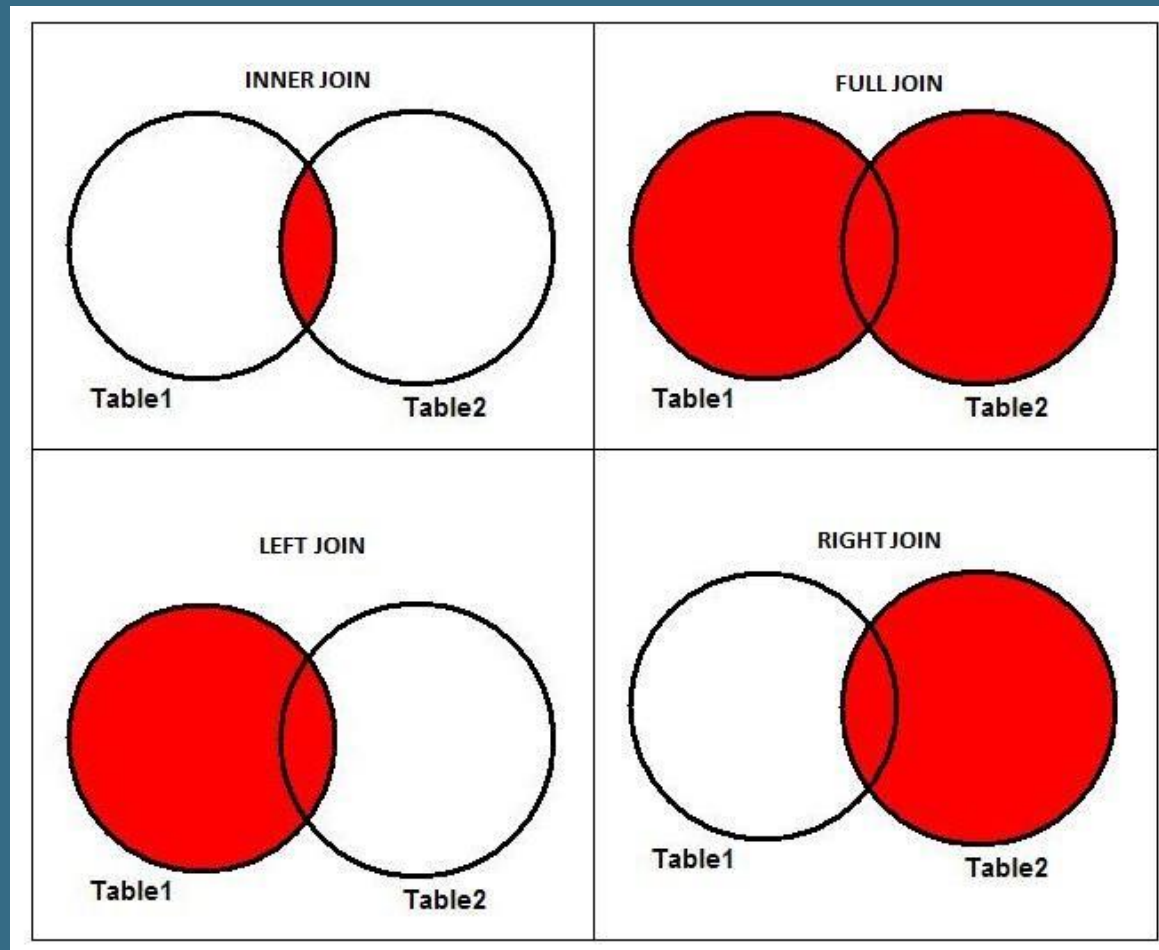
```
SELECT * FROM Cliente, Profissao;
```

```
SELECT Cliente.Nome, Profissao.Cargo  
FROM Cliente, Profissao  
WHERE Cliente.FK_Profissao=Profissao.Prof_Codigo;
```

RETORNA 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. Note que Cliente.FK_Profissao é a chave estrangeira da tabela Cliente, que referencia diretamente a chave primária da tabela Profissao.

Junção Interna e Externa

Junções



Junções

O uso de uma condição de junção é obrigatória para junções externas, mas opcional para junções internas (se for omitido, o resultado será um produto cartesiano).

Cláusula de Junção	Condições de junção
Inner join => join	Natural
Let outer join => left join	On <predicado>
Right outer join => right join	Using (A1, A2, ..., An)
Full outer join => full join	

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.

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.

Junção Interna (Inner Join)

Clark	10	PRODUTO	10	Accounting	=	Clark	10	10	Accounting
Miller	10		20	Research		Clark	10	20	Research
Smith	20		30	Sales		Clark	10	30	Sales
Turner	30					Miller	10	10	Accounting
						Miller	10	20	Research
						Miller	10	30	Sales
						Smith	20	10	Accounting
						Smith	20	20	Research
						Smith	20	30	Sales
						Turner	30	10	Accounting
						Turner	30	20	Research
						Turner	30	30	Sales

```
SELECT tabela1.atributo1, tabela2.atributo2
FROM tabela1
  [INNER] JOIN tabela2
  ON tabela1.atributo1 = tabela2.atributo1;
```

Clark	10	JOIN	10	Accounting	=	Clark	10	10	Accounting
Miller	10		20	Research		Miller	10	10	Accounting
Smith	20		30	Sales		Smith	20	20	Research
Turner	30					Turner	30	30	Sales

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.

```
SELECT
    Cliente.nome, Pedido.cod_cliente, Pedido.num_pedido
FROM Cliente INNER JOIN Pedido
ON Cliente.Cod_cliente = Pedido.Cod_cliente
```

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

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.

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.

Junção Externa (Outer Join)

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.

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.

Junção Externa (Outer Join)

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 `Profissao`, 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`.

```
SELECT *  
FROM Cliente LEFT JOIN Profissao  
ON Cliente.Profissao=Profissao.Codigo;
```

Junção Externa (Outer Join)

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

Junção Externa (Outer Join)

Right Outer Join

Como exemplo, imagine que há diversas profissões com respectivos códigos que não possuem correspondentes na tabela `Cliente`. Esta consulta traz todas estas profissões mesmo que não haja esta correspondência:

```
SELECT *  
FROM Cliente RIGHT JOIN Profissao  
ON Cliente.Profissao = Profissao.Codigo;
```


Junção Externa (Outer Join)

Full Outer Join

Esta operação 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 valores nulos para os registros sem correspondência.

```
SELECT *  
FROM Cliente FULL OUTER JOIN Profissao  
ON Cliente.Profissao=Profissao.Codigo;#
```

Resumo

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

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

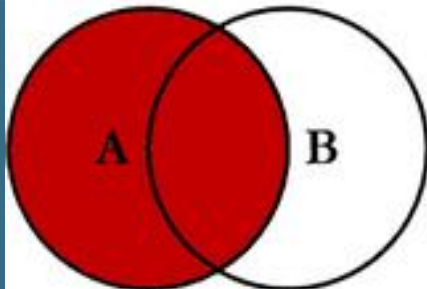
Resumo

Left Outer Join todos os registros da tabela esquerda mesmo quando não exista registros correspondentes na tabela direita.

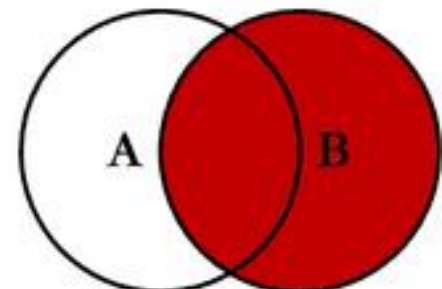
Right Outer Join todos os registros da tabela direita mesmo quando não exista registros correspondentes na tabela esquerda.

Full Outer Join todos os dados das tabelas à esquerda e à direita, mesmo que não possuam correspondência em outra tabela.

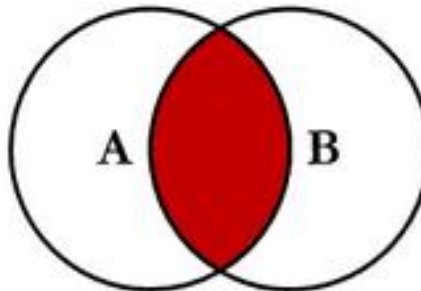
SQL JOINS



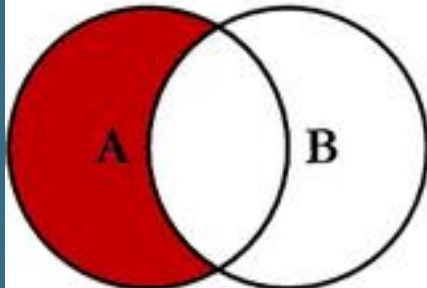
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



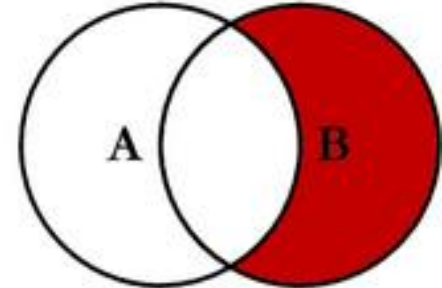
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



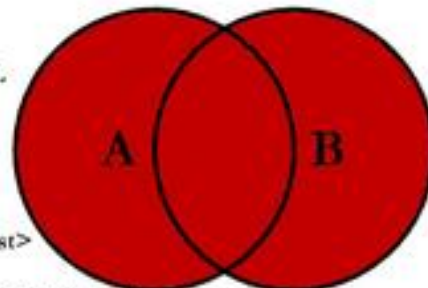
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



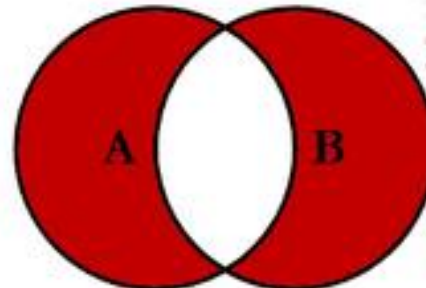
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
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
```

Atividade

