



REVISÃO PARA PROVA – 1º BIMESTRE

BANCO DE DADOS II – PROFESSORA: ANDRÉIA MACIEL DA SILVA

PESQUISA NOS DADOS USANDO SELECT

- O comando SELECT é o mais básicos dos comandos SQL. Ele retorna determinadas colunas de determinadas tabelas (FROM), atendendo certas condições (WHERE). Pode-se combinar essas cláusulas para criar comandos de consulta intuitivos em SQL.
- `SELECT * FROM Nome_tabela WHERE condição;`

SELECT

Vários campos

```
graph TD; A[Vários campos] --> B[SELECT NOME, IDADE, DATANASCIMENTO FROM TESTE]; B --> C[No exemplo agora estamos trazendo um conjunto de dados da tabela TESTE, quando especificamos outros campos, separamos por vírgula.];
```

```
SELECT NOME, IDADE, DATANASCIMENTO FROM TESTE
```

No exemplo agora estamos trazendo um conjunto de dados da tabela TESTE, quando especificamos outros campos, separamos por vírgula.



CLÁUSULA DISTINCT

Não mostra eventuais valores repetidos de colunas.

Sintaxe: `SELECT DISTINCT coluna FROM tabela;`

OPERADORES DE COMPARAÇÃO

Operador	Descrição	Exemplo
A=B	A é igual a B	preco=200
A>B	A é maior que B	Preco>200
A>=B	A é maior ou igual a B	Preco>=200
A<B	A é menor que B	Preco<200
A<=B	A é menor ou igual a B	Preco<=200
A<>B	A é diferente de B	Preco<>200

OPERADORES LÓGICOS

Operador	Descrição	Exemplo
AND	A AND B	codigo>=200 AND preco=100
OR	A OR B	codigo>=200 OR preco=100
NOT	A NOT B	NOT preco=100

LIKE

Quando não se sabe exatamente o que se está procurando, utiliza-se caracteres curingas % com o comando LIKE (semelhante a), e isso vai pesquisar por sequências de caracteres que correspondem aos padrões especificados.

```
SELECT * FROM produtos WHERE nome_produto LIKE "%ão";
```

Este comando procura padrões utilizando um caractere curinga.

WHERE nome LIKE 'a%'	Encontra todos os valores que iniciam com "a"
WHERE nome LIKE '%a'	Encontra todos os valores que terminam "a"
WHERE nome LIKE '%or%'	Localiza quaisquer valores que tenham "ou" em qualquer posição

Código	Nome do produto	Preço
101	Melão	8
104	Limão	2
108	Mamão	10

SELECT – ORDER BY



ORDER BY organiza os resultados de acordo com uma ou mais colunas da tabela, podendo definir a ordem do resultados como crescente ou decrescente.



```
SELECT * FROM aluno ORDER BY nome;
```



DESC ou ASC

BETWEEN

- É usado para obter um intervalo de valores.
- `SELECT * FROM produtos WHERE preco BETWEEN 150 and 200;`

IN

- É empregado para percorrer uma lista de valores e buscar dados dentro dessa lista.
- `SELECT * FROM produtos WHERE codigo IN (10, 20, 30);`

FUNÇÕES DE AGRUPAMENTO - MAX, MIN, AVG, SUM E COUNT

MIN = Valor Mínimo de um conjunto de valores

MAX = Valor Máximo de um conjunto de valores

AVG = Média Aritmética de um conjunto de valores

SUM = Total (Soma) de um conjunto de valores

COUNT = Contar quantidade total de itens

GROUP BY

- É usado para formar grupos e categorizar os resultados por meio desses grupos. Este comando sempre estará associado a uma função agregada uma vez que seu objetivo é o de aplicar uma função agregada, a um grupo de registros.
- Exemplo:
- `SELECT distrito, AVG(preco) FROM produtos GROUP BY distrito;`

HAVING

- Vimos que o WHERE é o comando que permite definir condições para uma consulta. No entanto, é muito comum precisarmos aplicar uma condição aos grupos formados pelo GROUP BY em vez de aplicar a condição a todos os registros da tabela. Para aplicar a condição aos grupos, usamos o comando HAVING.
- Exemplo:
- `SELECT distrito, AVG(preco) FROM produtos GROUP BY distrito HAVING AVG(preco) >= 200;`

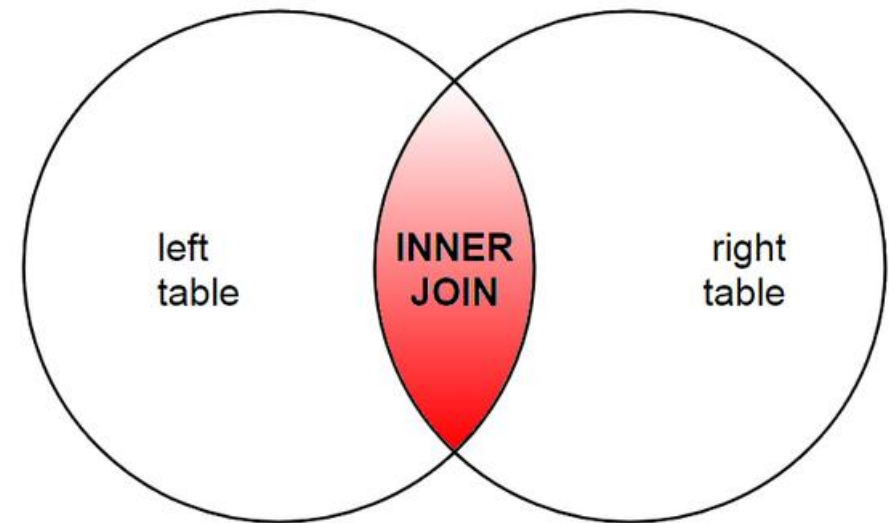
TRABALHANDO COM JOINS

- Ela faz a ligação entre as diferentes entidades pelos respectivos atributos desejados. Geralmente, interliga as chaves **primárias e estrangeiras**.

INNER JOIN

- Retorna apenas os valores iguais na interligação de campos distintos.

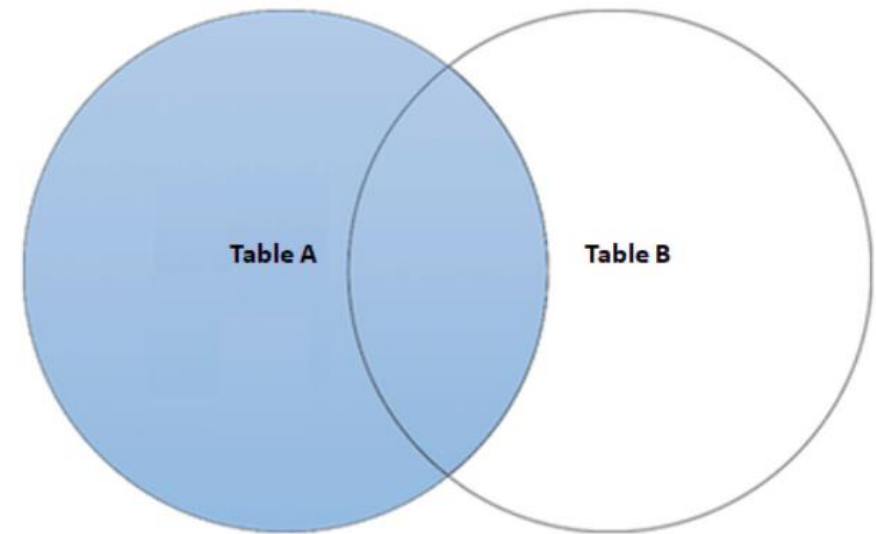
```
SELECT column_name(s)  
FROM table1  
INNER JOIN table2  
ON table1.column_name = table2.column_name;
```



LEFT JOIN

- Retorna todos os registros da tabela da esquerda (tabela1) e os registros correspondentes (se houver) da tabela da direita (tabela2).

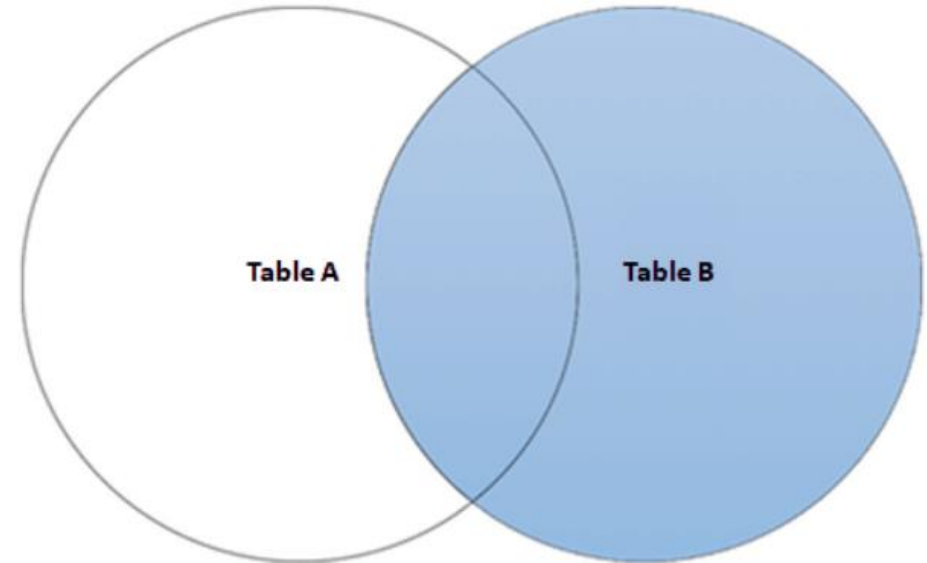
```
SELECT column_name(s)  
FROM table1  
LEFT JOIN table2  
ON table1.column_name = table2.column_name;
```



RIGHT JOIN

- Retorna todos os registros da tabela direita (tabela2) e os registros correspondentes (se houver) da tabela esquerda (tabela1).

```
SELECT column_nome(s)
FROM table1
RIGHT JOIN table2
ON table1.column_nome = table2.column_nome;
```



JUNTANDO TRÊS OU MAIS TABELAS

Exemplo:

```
SELECT aluno.nomeAluno, curso.nome_curso, cidade.cidade FROM aluno  
INNER JOIN curso ON aluno.idCurso = curso.id  
INNER JOIN cidade ON aluno.idCidade = cidade.id;
```



EXEMPLOS



```
select * from telefone  
right join clientes on  
clientes.idCliente=telefono.idCliente;
```

```
select * from telefone  
inner join clientes on  
clientes.idCliente=telefono.idCliente;
```

```
select * from endereco  
left join clientes on clientes.idCliente=endereco.idCliente  
left join telefone on clientes.idCliente=telefono.idCliente;
```



ATIVIDADE AVALIATIVA

