



Instituto de Tecnologia - ITec
Ciência da Computação

Banco de Dados

Prof. Eder Pazinatto

Prof. Eder Costella

Linguagem SQL – DQL



❑ LINGUAGEM DE CONSULTA DE DADOS (DQL):

- ❑ O comando DQL - instrução utilizada para a obtenção dos registros/linhas das tabelas do bancos de dados.**
- ❑ Os comandos para consulta de dados:**
 - ❑ SELECT**





SELECT – Estrutura de uma consulta SQL

SELECT <lista de atributos>

- Lista os atributos/colunas cujos valores são recuperados pela consulta. O asterisco * pode ser usado para listar todos as colunas.

* campos/colunas de mais de uma tabela pode fazer parte da seleção. Se possuírem o mesmo nome é possível renomear

FROM <lista de tabelas>

- Lista de nomes de tabelas necessárias ao processamento da consulta

WHERE <condição/filtro>

- uma expressão condicional que identifica as tuplas(linhas), conforme um critério, a serem recuperadas pela consulta



* Dica – uso do SELECT

SELECT < o que você quer como resultado? >

FROM < onde estão dos dados? >

WHERE < como você quer? >





SELECT – Exemplo de consulta SQL

SELECT nome, idade, email

FROM aluno

Seleciona o nome, idade e email de todos alunos da tabela aluno

SELECT nome, idade, email

FROM aluno

WHERE idade >= 18

Seleciona o nome, idade e email de todos alunos com idade
Maior ou Igual a 18



Operadores permitidos dentro da cláusula WHERE

Operador	Descrição
=	igual
<>	diferente
>	maior que
<	menor que
>=	maior ou igual
<=	menor ou igual
BETWEEN	entre
LIKE	busca por um padrão
IN	aceita apenas um valor do conjunto especificado
AND	operador lógico E
OR	operador lógico OU





SELECT - uso do WHERE

- ❑ Nas consultas podemos também selecionar um conjunto de registros em uma ou mais tabelas que atenda a uma determinada condição definida.
- ❑ Exemplo:
 - ❑ ORDER BY <nome-campo> {ASC | DESC} - quando utilizada apresenta o resultado da consulta ordenado de forma crescente ou decrescente pelos campos definidos.



SELECT

- ❑ ALL (comando padrão/default): retorna todos os registros da seleção mesmo com duplicidade

❑ SELECT ALL valor
❑ FROM compra

<u>Valor</u>
87.89
9.99
12.50
9.99
9.99

- ❑ DISTINCT: retorna somente registros sem duplicidade

❑ SELECT DISTINCT valor
❑ FROM compra
❑ ORDER BY valor

<u>Valor</u>
9.99
12.50
87.89





□ SQL – Operadores Especiais

- Há alguns operadores que são utilizados para determinar melhor as linhas que queremos filtrar.

□ Exemplos:

- IS NULL: usado para identificar se o conteúdo de uma coluna foi ou não inicializado. Exemplo:

```
SELECT      nome  
FROM        aluno  
WHERE       email IS NULL
```

- IS NOT NULL: somente aqueles campos que possuírem conteúdos atribuídos serão mostrados. Exemplo:

```
SELECT      nome  
FROM        aluno  
WHERE       email IS NOT NULL
```

□ SQL – Operadores Especiais

- BETWEEN: usado para determinar um intervalo de busca; muito usado em datas e simplifica o uso do operador AND.
Exemplo:

```
SELECT      nome  
FROM        aluno  
WHERE       idade BETWEEN 15 AND 30
```

- IN: permite comparar o valor de uma coluna com um conjunto de valores – mais usado em subquery. Exemplo:

```
SELECT      *  
FROM        aluno  
WHERE       idade IN (21,26,30) -- ou idade = 21 or idade = 26...
```



❑ SQL – Operadores Especiais

- ❑ EXTRACT: A função **extract** extrai parte de uma data ou intervalo

- ❑ Exemplos:

```
SELECT extract(day from dataEvento)
FROM evento
```

*/*filtra todos os dias com eventos, independente do mês e ano*/*

```
SELECT *
FROM evento
WHERE extract(year from dataEvento) = 2025
```

*/*filtra somente os evento que foram realizados em 2025*/*



❑ SQL – Operadores Especiais

- ❑ Operação em Texto - - Funções UPPER e LOWER:

Converte o campo (texto) para que seja exibido todo em caracteres maiúsculos ou minúsculos respectivamente.

Exemplos:

```
SELECT matricula, UPPER(nome) FROM Aluno
```

Saída => campo nome em maiúsculo

```
SELECT matricula
```

```
FROM Aluno
```

```
WHERE LOWER(nome) = 'ana'
```

Saída => lista apenas as matrícula com nome ana tudo em minúsculo





- A função **SUBSTR** tem como objetivo extrair parte do conteúdo do campo selecionado, de acordo com os parâmetros informados na função, independente do tipo do campo (numérico, texto ou data).

Exemplo

SELECT matricula, SUBSTR(nome,1,4) FROM Aluno
saída -> exibe do campo nome, apenas o conteúdo a partir do primeiro, os próximos 4 caracteres.

A função **REPLACE** tem como objetivo possibilitar a substituição de caracteres específicos por outros.

Exemplo

SELECT matricula, REPLACE(nome,'A','*') FROM Aluno
saída -> substituindo o caractere 'A' por '' do campo nome.*



□ SQL – Operadores Especiais

- LIKE: usado para comparar cadeia de caracteres. Utiliza dois caracteres reservados '%' (substitui um número arbitrário de caracteres) e '-' (substitui um único caractere).

□ Exemplo:

```
SELECT      nome
  FROM       aluno
 WHERE      email LIKE '%@upf%'
```

Tabela – uso do Like



(Operador Like - fonte: <http://www.devmedia.com.br>)

Expressão	Resultado
LIKE 'Juca%'	Qualquer string que iniciem com Juca.
LIKE '%Silva'	Qualquer string que terminem com Silva.
LIKE '%Santos%'	Qualquer string que tenha Santos em qualquer posição.
LIKE 'A_'	String de dois caracteres que tenham a primeira letra A e o segundo caractere seja qualquer outro.
LIKE '_A'	String de dois caracteres cujo primeiro caractere seja qualquer um e a última letra seja a letra A.
LIKE '_A_'	String de três caracteres cuja segunda letra seja A, independentemente do primeiro ou do último caractere.
LIKE '%A_'	Qualquer string que tenha a letra A na penúltima posição e a última seja qualquer outro caractere.
LIKE '_A%'	Qualquer string que tenha a letra A na segunda posição e o primeiro caractere seja qualquer outro caractere.
LIKE '___'	Qualquer string com exatamente três caracteres.
LIKE '___%'	Qualquer string com pelo menos três caracteres.
LIKE '%%"'	Qualquer string que tenha o caractere ' em qualquer posição.

