

SQL - Comando SELECT

David Lima <david.lima@ifal.edu.br>

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Seleção de colunas específicas em uma tabela

SELECT <nome(s) da(s) colunas> **FROM** <tabela>;

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Seleção de todas as colunas de uma tabela

SELECT * FROM <tabela>;

SQL – Extração de Dados

- Comando Select (Exemplos)
 - Seleção de todas as colunas de uma tabela
- Listar todo o conteúdo de vendedor.
 - **SELECT * FROM** vendedor;

SQL – Extração de Dados

- Comando Select (exemplos)
 - Seleção de colunas específicas em uma tabela
- Listar todos os produtos com as respectivas descrições, nomes e valores unitários.
 - **SELECT** descricao, nome, valor_unitario **FROM** produto;
- Listar da tabela CLIENTE, o nome do cliente, cidade, cep e estado
 - **SELECT** nome, cidade, cep, estado **FROM** cliente;

SQL – Extração de Dados

- Comando Select (contextualização)
 - Alterando o Heading (cabeçalho) da coluna
- Por default, o heading (nome da coluna criado no BD) apresentado na saída do SELECT é o nome da coluna na tabela.
- O SQL permite que se apresente a saída de um SELECT com cabeçalhos de colunas ao nosso gosto.

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Alterando o Heading (cabeçalho) da coluna

SELECT <coluna> **AS** <novo nome da coluna> **FROM** <tabela>;

SQL – Extração de Dados

- Comando Select (Exemplos)
 - Alterando o Heading (cabeçalho) da coluna
- Listar da tabela CLIENTE, a cidade e o cep do cliente como cidade_cliente e cep_cliente.
 - **SELECT** cidade **AS** cidade_cliente, cep **AS** cep_cliente **FROM** cliente;

SQL – Extração de Dados

- Comando Select (Exemplos)
 - Manipulando dados numéricos: Operadores Aritméticos
- Listar da tabela VENDEDOR, o nome do vendedor, o salário fixo e o triplo do mesmo.
 - **SELECT** nome, salario_fixo, (salario_fixo * 3) **FROM** vendedor;
 - **SELECT** nome, salario_fixo, (salario_fixo * 3) **AS** salario_triplo **FROM** vendedor;
- Listar da tabela VENDEDOR, o nome do vendedor, o salário fixo e o dobro de seu salário acrescido de 350.
 - **SELECT** nome, salario_fixo, ((salario_fixo * 2) + (350)) **AS** dobro_salario_mais **FROM** vendedor;

SQL – Extração de Dados

- Comando Select (contextualização)
 - Selecionando apenas algumas Linhas da Tabela
- A cláusula **WHERE** em um comando **SELECT** especifica quais linhas queremos obter, baseada em <condições de seleção>

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Selecionando apenas algumas Linhas da Tabela

SELECT <nome(s) da(s) colunas> **FROM** <tabela> **WHERE** <condições de seleção>

Operadores

a) De Comparação

b) Lógicos

c) BETWEEN e NOT BETWEEN

d) LIKE e NOT LIKE

e) IN e NOT IN

f) IS NULL e IS NOT NULL

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

Operadores de Comparação

- Listar o número do pedido, o código do produto e a quantidade de itens do pedido com a quantidade igual a 30.
 - **SELECT** FK_num_pedido, FK_cod_produto, quantidade **FROM** itempedido **WHERE** quantidade = 30;
- Quais os clientes que moram em Maceió?
 - **SELECT** nome **FROM** cliente **WHERE** cidade = 'Maceió';

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

Operadores de Comparação

- Quais os clientes que não moram em Maceió?
 - **SELECT** nome **FROM** cliente **WHERE** cidade <> 'Maceió';
- Quais os produtos com valor unitário menor ou igual a R\$ 2?
 - **SELECT** descricao **FROM** produto **WHERE** valor_unitario <= 2;

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

Operadores Lógicos

- Listar os produtos que tenham valor unitário menor ou igual a R\$ 10 e maior que R\$ 1 da tabela PRODUTO.
 - **SELECT** descricao **FROM** produto **WHERE** valor_unitario <= 10 **AND** valor_unitario > 1;
- Liste os vendedores e seus respectivos salários, que pertencem a faixa de comissão 'A' ou que ganham um salário fixo acima de R\$ 500.
 - **SELECT** nome **FROM** vendedor **WHERE** faixa_comissao = 'A' **OR** salario_fixo > 500;

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

Operadores Lógicos

- Mostrar todos os pedidos que não tenham prazo de entrega igual a 15 dias.
 - **SELECT** num_pedido **FROM** pedido **WHERE NOT** (prazo_entrega = 15);
- Mostrar todos os pedidos que não tenham prazo de entrega igual a 15 dias (sem utilizar o NOT).
 - **SELECT** num_pedido **FROM** pedido **WHERE** (prazo_entrega <> 15);

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Selecionando apenas algumas Linhas da Tabela

BETWEEN e NOT BETWEEN

WHERE <nome da coluna> **BETWEEN** <valor1> **AND** <valor2>;

WHERE <nome da coluna> **NOT BETWEEN** <valor1> **AND** <valor2>;

SQL – Extração de Dados

- Comando Select (contextualização)
 - Selecionando apenas algumas Linhas da Tabela

BETWEEN e NOT BETWEEN

- Este operador propicia a pesquisa por uma determinada coluna e selecionando as linhas cujo valor da coluna esteja dentro de uma faixa determinada de valores, sem a necessidade dos operadores \geq , \leq e AND.
- Tanto o <valor1> quanto o <valor2> têm de ser do mesmo tipo de dado da coluna.

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

BETWEEN e NOT BETWEEN

- Listar o código e a descrição dos produtos que tenham o valor unitário na faixa de R\$ 0,32 e R\$ 2,00.
 - **SELECT** cod_produto, descricao **FROM** produto **WHERE** valor_unitario **BETWEEN** 0.32 AND 2;
- Listar o código e a descrição dos produtos que não tenham o valor unitário na faixa de R\$ 1,00 e R\$ 2,00.
 - **SELECT** cod_produto, descricao **FROM** produto **WHERE** valor_unitario **NOT BETWEEN** 1 AND 2;

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Selecionando apenas algumas Linhas da Tabela

LIKE e NOT LIKE

WHERE <nome da coluna> **LIKE** <valor>

WHERE <nome da coluna> **NOT LIKE** <valor>

SQL – Extração de Dados

- Comando Select (contextualização)
 - Selecionando apenas algumas Linhas da Tabela

LIKE e NOT LIKE

- Os operadores LIKE e NOT LIKE só trabalham sobre colunas que sejam do tipo **VARCHAR**.
- Eles têm praticamente o mesmo funcionamento que os operadores = e <>, porém o poder desses operadores está na utilização do símbolo (%) que pode fazer o papel de “coringa”.
- % - substitui uma palavra

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

LIKE e NOT LIKE

- LIKE 'LÁPIS %' pode enxergar os seguintes registros:
 - o 'LÁPIS PRETO'
 - o 'LÁPIS BORRACHA'
 - o 'LÁPIS CERA'
- Ou seja, todos os registros que contenham 'LÁPIS' seguido de qualquer palavra ou conjunto de caracteres.

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

LIKE e NOT LIKE

- LIKE ‘%ÃO’ pode enxergar os seguintes registros:
 - o ‘JOÃO’
 - o ‘SÃO JOÃO’
 - o ‘ALÇAPÃO’
 - o ‘CÃO’
- Ou seja, pode enxergar qualquer nome que termine com “ÃO”.

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

LIKE e NOT LIKE

- Listar todos os produtos que tenham o seu nome começando por “p”.
 - **SELECT** descricao **FROM** produto **WHERE** descricao **LIKE** ‘p%’;
- Listar todos os produtos que tenham o seu nome terminando com “o”.
 - **SELECT** descricao **FROM** produto **WHERE** descricao **LIKE** ‘%o’;

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

LIKE e NOT LIKE

- Listar os vendedores que não começam por “Jo”.
 - **SELECT** nome **FROM** vendedor **WHERE** nome **NOT LIKE** 'Jo%';

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Selecionando apenas algumas Linhas da Tabela

IN e NOT IN

WHERE <nome da coluna> **IN** <valor>

WHERE <nome da coluna> **NOT IN** <valor>

- Esses operadores pesquisam registros que estão ou não contidos no conjunto de valores fornecidos.
- Estes operadores minimizam o uso dos operadores =, <>, AND e OR.

SQL – Extração de Dados

- Comando Select (exemplos)
 - Selecionando apenas algumas Linhas da Tabela

IN e NOT IN

- Listar os vendedores que são da faixa de comissão A e B.
 - **SELECT** nome **FROM** vendedor **WHERE** faixa_comissao **IN** ('A', 'B');
- Listar os vendedores que não são da faixa de comissão B e C.
 - **SELECT** nome **FROM** vendedor **WHERE** faixa_comissao **NOT IN** ('B', 'C');

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Selecionando apenas algumas Linhas da Tabela

IS NULL e IS NOT NULL

WHERE <nome da coluna> **IS NULL**

WHERE <nome da coluna> **IS NOT NULL**

- Mostrar os nomes dos clientes que não tenham estado.
 - **SELECT** nome **FROM** cliente **WHERE** estado **IS NULL**;

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Ordenando os Dados Seleccionados

SELECT <nome da(s) coluna(s)> **FROM** <tabela> **WHERE** <condição> **ORDER**
BY <nome da coluna> <**ASC** | **DESC**>;

SQL – Extração de Dados

- Comando Select (contextualização)
 - Ordenando os Dados Seleccionados
- As palavras **ASC** e **DESC** significam, respectivamente, ascendente e descendente.
- A forma ascendente de ordenação é assumida como padrão.

SQL – Extração de Dados

- Comando Select (exemplos)
 - Ordenando os Dados Seleccionados
- Mostrar em ordem alfabética a lista de vendedores e seus respectivos salários fixos.
 - **SELECT** nome, salario_fixo **FROM** vendedor **ORDER BY** nome;
- Listar os nomes, cidades e estados de todos os clientes, ordenados por cidade de forma descendente.
 - **SELECT** nome, cidade, estado **FROM** cliente **ORDER BY** cidade **DESC**;

SQL – Extração de Dados

- Comando Select (exemplos)
 - Ordenando os Dados Seleccionados
- Mostrar a descrição e o nome de todos os produtos que tenham o valor unitário maior que 5, em ordem de valor unitário ascendente.
 - **SELECT** nome, descricao **FROM** produto **WHERE** valor_unitario > 5 **ORDER BY** valor_unitario **ASC**;

SQL – Extração de Dados

- Comando Select (contextualização)
 - Realizando cálculos com Informação Seleccionada
- Com a linguagem SQL pode-se criar um campo que não pertence à tabela original, e seja fruto de cálculo sobre alguns campos da tabela.
 - Mostrar o novo salário fixo dos vendedores, de faixa de comissão 'C', calculado com base no reajuste de 75% acrescido de R\$ 350.00 de bonificação. Ordenar pelo nome do vendedor.

```
SELECT nome, salario_fixo, ((salario_fixo * 1.75) + (350)) AS novo_salario FROM  
vendedor WHERE faixa_comissao = 'C' ORDER BY nome;
```


SQL – Extração de Dados

- Comando Select (exemplos)
 - Utilizando funções de Agregação sobre conjuntos

Buscando Máximos e Mínimos (MAX, MIN)

- Mostrar o menor e o maior salário fixo da tabela VENDEDOR.

```
SELECT MAX(salario_fixo) AS maior, MIN(salario_fixo) AS menor FROM  
vendedor;
```

SQL – Extração de Dados

- Comando Select (exemplos)
 - Utilizando funções de Agregação sobre conjuntos

Calculando médias (AVG)

- Qual a média dos salários fixos dos vendedores?

```
SELECT AVG(salario_fixo) FROM vendedor;
```

SQL – Extração de Dados

- Comando Select (exemplos)
 - Utilizando funções de Agregação sobre conjuntos

Totalizando os valores das colunas (SUM)

- Mostrar a quantidade total pedida para o produto 'Produto 1' de código 1 na tabela itempedido.

```
SELECT SUM(quantidade) AS soma FROM itempedido WHERE cod_produto = 1;
```

SQL – Extração de Dados

- Comando Select (exemplos)
 - Utilizando funções de Agregação sobre conjuntos

Contando os registros (COUNT)

- Quantos vendedores ganham acima de R\$ 300.00 de salário fixo?

SELECT COUNT(*) FROM vendedor WHERE salario_fixo > 300;

- Quantos clientes moram em Maceió?

SELECT COUNT(*) FROM cliente WHERE cidade = 'Maceió';

SQL – Extração de Dados

- Comando Select (exemplos)
 - Utilizando funções de Agregação sobre conjuntos

Utilizando a cláusula (DISTINCT)

- Esta cláusula elimina repetições de valores em relação a uma coluna.
- Listar as cidades distintas, na tabela CLIENTE, ordenadas ascendentemente?

SELECT DISTINCT cidade **FROM** cliente **ORDER BY** cidade;

- Quantas cidades distintas há no cadastro dos clientes?

SELECT COUNT(DISTINCT cidade) **FROM** cliente;

SQL – Extração de Dados

- Comando Select (contextualização)
 - Agrupando Informações Seleccionadas
- Utilizando as cláusula (GROUP BY e HAVING)
- A função de agregação, por si própria, produz um número simples para uma tabela.
- A cláusula GROUP BY organiza esse sumário de dados em grupos, produzindo informação sumarizada para os grupos definidos na tabela objeto de seleção.
- A cláusula HAVING realiza as restrições das linhas resultantes da mesma forma que a cláusula WHERE o faz em um SELECT.

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Agrupando Informações Seleccionadas

Utilizando as cláusula (GROUP BY e HAVING)

SELECT <nome da(s) coluna(s)> **FROM** <tabela> **WHERE** <condição(ões)>
GROUP BY <nome da(s) coluna(s)>; **HAVING** <condição(ões)>;

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Agrupando Informações Seleccionadas

Utilizando as cláusulas (GROUP BY e HAVING)

- Listar o número de produtos que cada pedido contém?

SELECT num_pedido, **COUNT(*) AS** numero_produtos **FROM** itempedido
GROUP BY num_pedido;

HUM_PEDIDO	COD_PRODUTO	QUANTIDADE
11	1	2
11	87	10
11	45	30
12	1	20
12	30	10



HUM_PEDIDO	HUMERO_PRODUTOS
11	3
12	2

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Agrupando Informações Seleccionadas

Utilizando as cláusulas (GROUP BY e HAVING)

- Listar a quantidade total de produtos que cada pedido contém?

```
SELECT num_pedido, SUM(quantidade) FROM itempedido GROUP BY  
num_pedido;
```

HUM_PEDIDO	COD_PRODUTO	QUANTIDADE
11	1	2
11	87	10
11	45	30
12	1	20
12	30	10



HUM_PEDIDO	SUM(QUANTIDADE)
11	42
12	30

SQL – Extração de Dados

- Comando Select (sintaxe)
 - Agrupando Informações Seleccionadas

Utilizando as cláusulas (GROUP BY e HAVING)

- Listar os pedidos que têm mais do que dois produtos?

SELECT num_pedido, **COUNT(*)** **FROM** itempedido **GROUP BY** num_pedido
HAVING COUNT(*) > 2;

HUM_PEDIDO	COD_PRODUTO	QUANTIDADE
11	1	2
11	87	10
11	45	30
12	1	20
12	30	10



HUM_PEDIDO	COUNT(*)
11	3