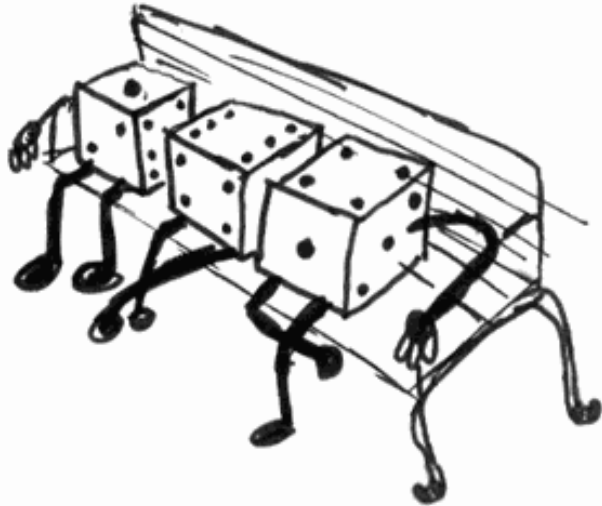
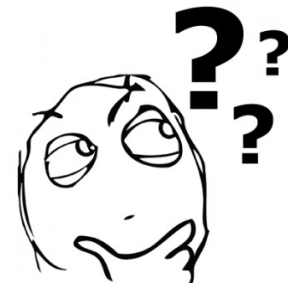


O BANCO DE DADOS



Banco de Dados

Instruções SQL – MySQL Command Line



```
MySQL 5.5 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.5.29 MySQL Community Server (GPL)

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

O que veremos hoje ?

9. Linguagem SQL

Instruções DML no MySQL 5.5 Command Line



```
MySQL 5.5 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.5.29 MySQL Community Server (GPL)

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

INSTRUÇÕES DML - INSERT

- O comando usado para inserir dados é o INSERT. A síntese básica do comando INSERT é a seguinte:

INSERT INTO (campo1, campo2) VALUES (valor1, valor2);

- Cada valor é inserido no campo que corresponde à posição do valor na lista: valor1 é inserido no campo1, valor2 no campo2 e assim por diante.
- Os valores devem ser separados com uma vírgula e se o tipo do campo for texto deve estar entre aspas duplas ou simples.

INSTRUÇÕES DML - INSERT

```
mysql> USE venda;  
Database changed  
mysql> DESCRIBE cliente;
```

| Field | Type | Null | Key | Default | Extra |
|------------|-------------|------|-----|---------|----------------|
| cliente_id | int(8) | NO | PRI | NULL | auto_increment |
| nome | varchar(50) | NO | | NULL | |
| endereco | varchar(80) | NO | | NULL | |
| cidade | varchar(30) | NO | | NULL | |

```
4 rows in set (0.01 sec)
```

```
mysql> INSERT INTO cliente(nome,endereco,cidade)  
-> VALUES('Francisco','Rua: 24 de Maio nº 324','Fortaleza');  
Query OK, 1 row affected (0.05 sec)
```

INSTRUÇÕES DML - SELECT

- A forma mais simples de se fazer um SELECT é recuperando todos os dados de uma tabela. A síntese básica é:

SELECT * FROM <nome da tabela>;

- O * (asterisco) substitui os nomes de todas as colunas, e todas serão selecionadas para o resultado da consulta. A instrução FROM indica de qual tabela estamos buscando os dados.
- Caso não fosse de nosso desejo mostrar todas as colunas no resultado da consulta, bastaria nomear as colunas que deveriam aparecer no lugar do * (asterisco) e separadas por vírgula.

SELECT <coluna1>,<coluna2> FROM <nome da tabela>;

INSTRUÇÕES DML - SELECT

```
mysql> USE venda;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_venda |
+-----+
| cliente          |
| pedido           |
+-----+
2 rows in set (0.00 sec)

mysql> SELECT * FROM cliente;
+-----+-----+-----+-----+
| cliente_id | nome      | endereco                                     | cidade    |
+-----+-----+-----+-----+
|          1 | Francisco | Rua: 24 de Maio nº 324                     | Fortaleza |
|          2 | Pedro     | Av. Augusto dos Anjos nº 2674              | Fortaleza |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

INSTRUÇÕES DML - SELECT

| | |
|-----------------|---|
| FROM | Utilizada para especificar a tabela que se vai selecionar os registros. |
| WHERE | Utilizada para especificar as condições que devem reunir os registros que serão selecionados. |
| GROUP BY | Utilizada para separar os registros selecionados em grupos específicos. |
| HAVING | Utilizada para expressar a condição que deve satisfazer cada grupo. |
| ORDER BY | Utilizada para ordenar os registros selecionados com uma ordem específica. |
| DISTINCT | Utilizada para selecionar dados sem repetição. |

INSTRUÇÕES DML - SELECT

```
mysql> DESCRIBE cliente;
```

| Field | Type | Null | Key | Default | Extra |
|------------|-------------|------|-----|---------|----------------|
| cliente_id | int(8) | NO | PRI | NULL | auto_increment |
| nome | varchar(50) | NO | | NULL | |
| endereco | varchar(80) | NO | | NULL | |
| cidade | varchar(30) | NO | | NULL | |

```
4 rows in set (0.08 sec)
```

```
mysql> SELECT nome,endereco FROM cliente WHERE cliente_id = 2;
```

| nome | endereco |
|-------|-------------------------------|
| Pedro | Av. Augusto dos Anjos nº 2674 |

```
1 row in set (0.05 sec)
```


INSTRUÇÕES DML - SELECT

```
mysql> SELECT nome,preco FROM produto ORDER BY preco ASC;
```

| nome | preco |
|----------|-------|
| Sal | 0.70 |
| Biscoito | 1.20 |
| Farinha | 1.80 |
| Vinagre | 1.95 |
| Leite | 2.35 |
| Cafe | 4.10 |
| Acucar | 5.40 |
| Arroz | 6.50 |

```
8 rows in set (0.00 sec)
```

INSTRUÇÕES DML - UPDATE

O comando para atualizar os dados é UPDATE, ele possui a seguinte sintaxe:

**UPDATE <tabela> SET <campo> = “novo valor” WHERE
<condição > ;**

- tabela: nome da tabela que será modificada
- set: define qual campo será alterado
- campo: campo que terá seu valor alterado
- novo valor: valor que substituirá o antigo dado cadastrado em campo
- where: se não for informado, a tabela inteira será atualizada
- condição: regra que impõe condição para execução do comando

```
mysql> SELECT * FROM cliente;
```

| cliente_id | nome | endereco | cidade |
|------------|-----------|-------------------------------|-----------|
| 1 | Francisco | Rua: 24 de Maio nº 324 | Fortaleza |
| 2 | Pedro | Av. Augusto dos Anjos nº 2674 | Fortaleza |

```
2 rows in set (0.42 sec)
```

```
mysql> UPDATE cliente SET endereco = "Av. C nº 687 4ª Etapa Conjunto Ceará"  
-> WHERE nome = "Francisco";
```

```
Query OK, 1 row affected (0.07 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> SELECT * FROM cliente;
```

| cliente_id | nome | endereco | cidade |
|------------|-----------|--------------------------------------|-----------|
| 1 | Francisco | Av. C nº 687 4ª Etapa Conjunto Ceará | Fortaleza |
| 2 | Pedro | Av. Augusto dos Anjos nº 2674 | Fortaleza |

```
2 rows in set (0.00 sec)
```

INSTRUÇÕES DML - UPDATE

- Também podemos alterar mais de um campo de uma vez. Suponhamos que o cliente Pedro se mudou para outra cidade, precisamos alterar o endereço e a cidade atual, não precisamos criar dois UPDATES basta separá-los por vírgula.

**UPDATE <tabela> SET <campo1> = “valor1”,
<campo2> = “valor2” WHERE <condição> ;**

```
mysql> SELECT * FROM cliente;
```

| cliente_id | nome | endereco | cidade |
|------------|-----------|--------------------------------------|-----------|
| 1 | Francisco | Av. C nº 687 4ª Etapa Conjunto Ceará | Fortaleza |
| 2 | Pedro | Av. Augusto dos Anjos nº 2674 | Fortaleza |

```
2 rows in set (0.00 sec)
```

```
mysql> UPDATE cliente SET endereco = "Rua Joaquim Mota nº 260" ,  
-> cidade = "Caucaia" WHERE nome = "Pedro";
```

```
Query OK, 1 row affected (0.08 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> SELECT * FROM cliente;
```

| cliente_id | nome | endereço | cidade |
|------------|-----------|--------------------------------------|-----------|
| 1 | Francisco | Av. C nº 687 4ª Etapa Conjunto Ceará | Fortaleza |
| 2 | Pedro | Rua Joaquim Mota nº 260 | Caucaia |

```
2 rows in set (0.00 sec)
```

INSTRUÇÕES DML - DELETE

A forma mais simples de se fazer um DELETE é excluindo todos os dados de uma tabela. A síntese básica é:

DELETE FROM < nome da tabela >;

Se não for especificada nenhuma condição então serão excluídos todos os dados da tabela, porém se você quer excluir somente um registro é preciso usar a cláusula WHERE informando qual será a condição para deletar.

DELETE FROM < nome da tabela > WHERE < condição >;

INSTRUÇÕES DML - DELETE

```
mysql> DELETE FROM cliente WHERE cliente_id = 1;  
Query OK, 1 row affected (0.12 sec)
```

```
mysql> SELECT * FROM cliente;
```

| cliente_id | nome | endereco | cidade |
|------------|-------|-------------------------|---------|
| 2 | Pedro | Rua Joaquim Mota nº 260 | Caucaia |

```
1 row in set (0.00 sec)
```


FUNÇÕES DE AGREGAÇÃO

| FUNÇÕES DE AGREGAÇÃO | |
|----------------------|---|
| COUNT | Utilizada para devolver o número de registros da seleção. |
| SUM | Utilizada para devolver a soma de todos os valores de um campo determinado. |
| MAX | Utilizada para devolver o valor mais alto de um campo especificado. |
| MIN | Utilizada para devolver o valor mais baixo de um campo especificado. |
| AVG | Utilizada para calcular a media dos valores de um campo determinado. |

FUNÇÕES DE AGRAGAÇÃO

- Exemplo Contagem (COUNT)
 - `SELECT COUNT(campo) FROM < nome da tabela > ;`
- Exemplo SOMA (SUM)
 - `SELECT SUM(campo) FROM < nome da tabela > ;`
- Exemplo MAXIMO (MAX)
 - `SELECT MAX(campo) FROM < nome da tabela > ;`
- Exemplo MINIMO (MIN)
 - `SELECT MIN(campo) FROM < nome da tabela > ;`
- Exemplo MÉDIA (AVG)
 - `SELECT AVG(campo) FROM < nome da tabela > ;`

| OPERADORES RELACIONAIS | |
|------------------------|---|
| < | Menor que |
| > | Maior que |
| < > | Diferente |
| <= | Menor ou igual que |
| >= | Maior ou igual que |
| = | Igual a |
| BETWEEN | Utilizado para especificar um intervalo de valores. |
| LIKE | Utilizado na comparação de um modelo e para especificar registros de um banco de dados."Like" + extensão % vai significar buscar todos resultados com o mesmo início da extensão. |

COMANDOS LIKE

| COMANDO | DESCRIÇÃO |
|------------|--|
| LIKE 'A%' | Todas as palavras que iniciem com a letra A; |
| LIKE '%A' | Todas que terminem com a letra A; |
| LIKE '%A%' | Todas que tenham a letra A 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; |
| LIKE '_A_' | String de três caracteres cuja segunda letra seja A, independentemente do primeiro ou do último caractere; |
| LIKE '%A_' | Todos que tenham a letra A na penúltima posição e a última seja qualquer outro caractere; |
| LIKE '_A%' | Todos que tenham a letra A na segunda posição e o primeiro caractere seja qualquer um; |

COMANDOS LIKE

```
mysql> SELECT * FROM contatos WHERE nome LIKE 'A%';
```

| id | nome | email | telefone |
|----|-----------|-------------------------|---------------|
| 1 | Alice | alicebarros@hotmail.com | (85) 32945832 |
| 2 | Alexandre | alexandre@hotmail.com | (85) 87948242 |

```
2 rows in set (0.04 sec)
```

```
mysql> SELECT * FROM contatos WHERE nome LIKE '%A';
```

| id | nome | email | telefone |
|----|---------|----------------------|---------------|
| 4 | Carla | carlinha@gmail.com | (85) 88533294 |
| 6 | Erica | ericasouza@yahoo.com | (85) 91432504 |
| 8 | Laura | laura@hotmail.com | (85) 32954232 |
| 10 | Vitoria | vitoria@hotmail.com | (85) 85492504 |

```
4 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM contatos WHERE nome LIKE '%A%';
```

| id | nome | email | telefone |
|----|-----------|-------------------------|---------------|
| 1 | Alice | alicebarros@hotmail.com | (85) 32945832 |
| 2 | Alexandre | alexandre@hotmail.com | (85) 87948242 |
| 4 | Carla | carlinha@gmail.com | (85) 88533294 |
| 5 | Davi | davi@gmail.com | (85) 34895322 |
| 6 | Erica | ericasouza@yahoo.com | (85) 91432504 |
| 8 | Laura | laura@hotmail.com | (85) 32954232 |
| 9 | Marcio | marcio_123@gmail.com | (85) 87426584 |
| 10 | Vitoria | vitoria@hotmail.com | (85) 85492504 |

```
8 rows in set (0.00 sec)
```

COMANDO BETWEEN

- O comando BETWEEN permite fazer a seleção de um intervalo, entre um e outro. A sintaxe da cláusula BETWEEN é a seguinte:

SELECT * FROM alunos WHERE idade BETWEEN 10 AND 20;

- Este comando irá selecionar todas as linhas cuja coluna tiver um valor entre 10 e 20. Os valores podem ser números, texto ou datas. Poderíamos de outra forma obter o mesmo resultado que seria:

SELECT * FROM alunos WHERE idade >= 10 AND idade <= 20;

COMANDO BETWEEN

```
mysql> SELECT * FROM alunos;
```

| id | nome | idade | responsavel | contato |
|----|---------|-------|-------------|----------|
| 1 | Daniel | 15 | Maria | 87945252 |
| 2 | Rafael | 18 | Francisca | 32853952 |
| 3 | Juliana | 23 | Graça | 91583206 |
| 4 | Hugo | 16 | Dores | 87935725 |
| 5 | Emerson | 25 | Eunice | 34895325 |
| 6 | Lucas | 17 | João | 32853925 |
| 7 | Sara | 14 | Maria | 87936306 |
| 8 | Natalia | 28 | Leonardo | 91328536 |

```
8 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM alunos WHERE idade BETWEEN 10 AND 20;
```

| id | nome | idade | responsavel | contato |
|----|--------|-------|-------------|----------|
| 1 | Daniel | 15 | Maria | 87945252 |
| 2 | Rafael | 18 | Francisca | 32853952 |
| 4 | Hugo | 16 | Dores | 87935725 |
| 6 | Lucas | 17 | João | 32853925 |
| 7 | Sara | 14 | Maria | 87936306 |

```
5 rows in set (0.46 sec)
```

```
mysql> SELECT * FROM alunos WHERE idade >= 10 AND idade <= 20;
```

| id | nome | idade | responsavel | contato |
|----|--------|-------|-------------|----------|
| 1 | Daniel | 15 | Maria | 87945252 |
| 2 | Rafael | 18 | Francisca | 32853952 |
| 4 | Hugo | 16 | Dores | 87935725 |
| 6 | Lucas | 17 | João | 32853925 |
| 7 | Sara | 14 | Maria | 87936306 |

```
5 rows in set (0.00 sec)
```


ATIVIDADE

Dada a tabela Alunos a seguir, escreva os comandos SQL que se pede:

| Matricula | Nome | Sexo | Idade |
|-----------|------------------|------|-------|
| 1 | Marcelo Medeiros | M | 35 |
| 2 | Ana Paula Berlim | F | 25 |
| 3 | Lucas Silva | M | 7 |
| 4 | Caroline Silva | F | 19 |
| 5 | Djalma Medeiros | M | 65 |
| 6 | Artur Paes | M | 5 |
| 7 | Eduarda Duda | F | 8 |

ATIVIDADE

Crie a tabela alunos acima.

Insira os dados corretamente.

Liste todos os alunos do sexo masculino

Liste todos os alunos que possuem o sobrenome Medeiros, ordenados por idade.

Liste a média de idade dos alunos.

Mostre a maior idade dos alunos.

Liste a quantidade de alunos com idade menor que 20 anos.

Liste a Matricula e a idade do aluno chamado „Lucas Silva“.

Informe o nome e a idade do aluno mais jovem.

Liste a quantidade de alunos com idade entre 10 e 20 anos que sejam do sexo masculino.

Liste a quantidade de mulheres cadastradas na tabela Alunos.

Informe o sexo e a quantidade de ocorrências agrupadas por sexo.

Liste os nomes dos alunos que começam com a letra „A“ e de sexo masculino.

Altere a idade da aluna Caroline Silva para 20 anos.

Exclua a aluna Djalma Medeiros da tabela.