

Banco de Dados - MYSQL

Profª Débora Cabral Nazário

MYSQL - Características

- Velocidade proporcionada pela sua implementação leve
- Distribuição gratuita
- Facilidade de integração com servidor Web
- E linguagens de programação de desenvolvimento de sites dinâmicos, especialmente a linguagem PHP

MYSQL - Características

- Capacidade de lidar com um número ilimitado de usuários
- Capacidade de manipular mais de 50 milhões de registros
- Execução rápida de comandos
- Sistema de segurança simples e funcional
- Possui API's para C, C++, Java, Perl, PHP, Python, TCL.

MYSQL - Características

- Foi desenvolvido para várias plataformas incluindo ambientes Unix, OS/2 e Windows.
- Permite operações e funções nas cláusulas select e where
- Suporte as funções SQL (group by, order by)
- Além de funções de grupo como:
Count(), avg(), sum(), std(), max(), min()
- Permite a seleção de diferentes tabelas de diferentes bases de dados em uma mesma query

MYSQL - Características

- Possui algoritmos de criptografia de senhas, fornecendo assim segurança aos dados gravados nas tabelas.
- Permite conexões via TCP/IP
- Acesso via terminal

MYSQL - Características

- Comandos são sucedidos por ponto e vírgula ';'
- Banco de dados são estruturas complexas de dados
- Gravados em tabelas (colunas - tipo)
- Em forma de registros

Tipos de Campos no Mysql

- Suporta uma grande extensão de tipos:
 - numéricos, caracteres, caracteres de tamanho variável, tipos enumerados
- Char(tamanho) - Caracter de tamanho fixo - 1 a 255
 - caracteres alfanuméricos, como endereços e nomes
- Varchar(tamanho) - aloca apenas o tamanho necessário para a gravação, mas é mais lento
- Int - Valor inteiro
- * Float - Valor de ponto flutuante

Tipos de Campos no Mysql

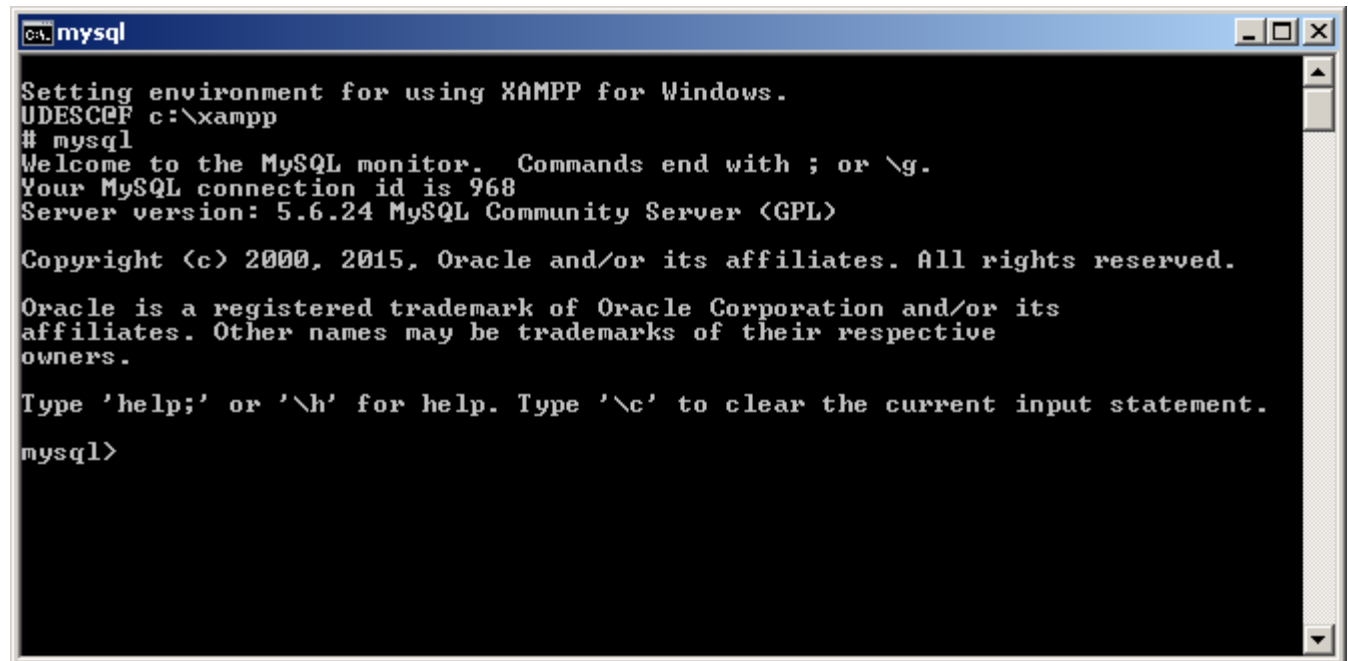
- Date - Campo de data padrão
 - padrão 'AAAA-MM-DD'
- Time - Campo de tempo padrão
- Text/Blob - grande quantidade de informação - o a 65535 bytes
 - Text - não é sensível a maiúsculas e minúsculas
 - Blob - case sensitive
- SET - permite o usuário escolher um determinado número de opções - 64 opções
- ENUM - semelhante ao SET, mas pode escolher apenas uma opção

MYSQL

- Registros
 - conjunto de campos relacionados
 - ex. nome char(15), email char(25), telefone int;
- Tabelas
 - conjunto de registros forma uma tabela

MYSQL

- Para conectar com o servidor mySQL
 - c:\xampp\mysql\bin\mysql
 - XAMPP Control Panel – Shell – mysql
 - mysql>



```
mysql
Setting environment for using XAMPP for Windows.
UDESCOF c:\xampp
# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 968
Server version: 5.6.24 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, 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>
```

MYSQL

- Trocar o banco de dados
 - Use <nome_bd>;
 - Database changed
- Para desconectar do servidor mySQL
 - QUIT; ou EXIT;

MYSQL

- Ajuda do mySQL
 - Help
- Mostrar todas as bases de dados já criadas
 - Show databases;
- Mostrar todas as variáveis disponíveis do mySQL
 - Show variables;

MYSQL

- Mostrar várias informações estatísticas da seção mySQL aberta
 - Show status;
- Criar um banco de dados
- Create database <nome_bd>;
 - c:\xampp\mysql\bin>mysql -u root
 - mysql -h localhost -u root -p (ubuntu)
 - mysql> create database odaw;
- Mostrar todas as tabelas do banco de dados em uso
 - Show tables;

Linguagem SQL

- Operações básicas:
 - Incluir, Apagar, Alterar, Pesquisar
- Criar uma tabela
 - Create table nome_tabela (campo tipo,campo tipo...)
 - create table teste (codigo int, nome char(15), email char(25), telefone int);
- Tabelas devem ter nomes diferentes
- Coluna pode ter até 64 letras
- Coluna pode começar com número
- Mas não pode ter somente números

Linguagem SQL

- Mostrar a definição de uma tabela
 - Describe <nome_tabela>;
- Chave Primária
 - único para cada registro
 - nome char(15) primary key
 - ex. com chave primária:
 - create table teste (codigo int, nome char(15) primary key, email char(25), telefone int);

Linguagem SQL

- Auto incremento
 - soma um a cada registro automaticamente
 - código `int auto_increment`
 - `create table teste (codigo int auto_increment primary key, nome char(15), email char(25), telefone int);`
 - Mostrar colunas - estrutura da tabela
 - `show columns from teste`
 - `describe teste`

Linguagem SQL

cmd XAMPP for Windows - mysql -u root

mysql> use odaw;

Database changed

mysql> show tables;

```
+-----+
| Tables_in_odaw |
+-----+
| teste          |
+-----+
```

1 row in set (0.00 sec)

mysql> show columns from teste;

Field	Type	Null	Key	Default	Extra
codigo	int(11)	NO	PRI	NULL	auto_increment
nome	char(15)	YES		NULL	
email	char(25)	YES		NULL	
telefone	int(11)	YES		NULL	

4 rows in set (0.03 sec)

mysql> _

Linguagem SQL

- Inserindo registros
 - insert into nome_tabela values (valor1, valor2)
 - insert into teste values
(NULL, 'Debora', 'debora.nazario@udesc.br',
123456);
 - INSERT INTO teste (nome, email) VALUES
(`Maria`, 'maria.souza@udesc.br');

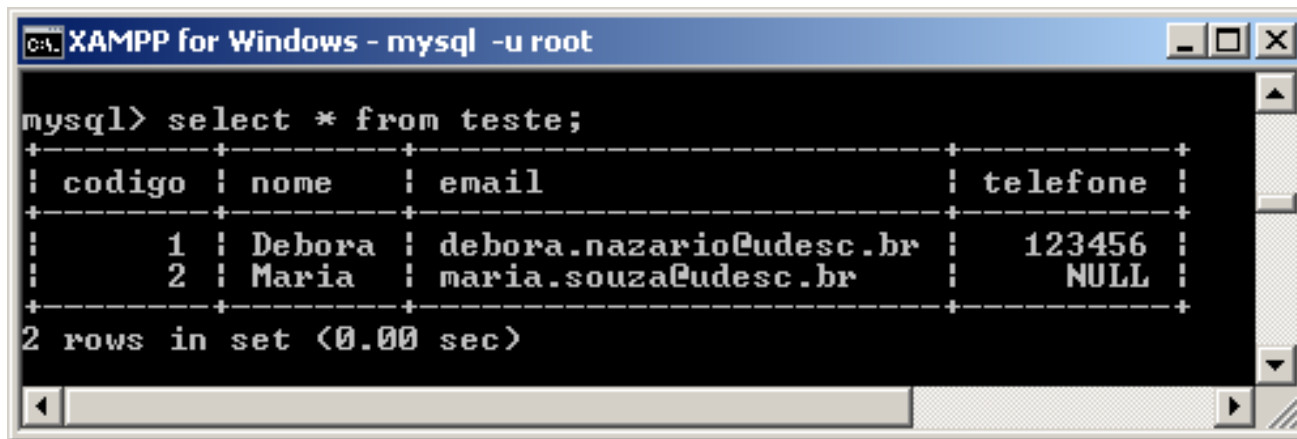
Linguagem SQL

■ Insert sintaxe

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
    [INTO] tbl_name [(col_name,...)]
    VALUES ((expression | DEFAULT),...),(...),...
    [ ON DUPLICATE KEY UPDATE col_name=expression, ... ]
or INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
    [INTO] tbl_name [(col_name,...)]
    SELECT ...
or INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
    [INTO] tbl_name
    SET col_name=(expression | DEFAULT), ...
    [ ON DUPLICATE KEY UPDATE col_name=expression, ... ]
```

Linguagem SQL

- Pesquisando registros
- `SELECT * FROM teste;`
- lista todos os registros da tabela teste
- `SELECT * FROM teste WHERE (nome = "Debora");`



The screenshot shows a terminal window titled "XAMPP for Windows - mysql -u root". The prompt is "mysql>". The user has entered the command "select * from teste;". The output is a table with four columns: "codigo", "nome", "email", and "telefone". There are two rows of data. The first row has values 1, Debora, debora.nazario@udesc.br, and 123456. The second row has values 2, Maria, maria.souza@udesc.br, and NULL. Below the table, it says "2 rows in set (0.00 sec)".

```
mysql> select * from teste;
+-----+-----+-----+-----+
| codigo | nome  | email                               | telefone |
+-----+-----+-----+-----+
| 1      | Debora | debora.nazario@udesc.br           | 123456   |
| 2      | Maria  | maria.souza@udesc.br              | NULL     |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Linguagem SQL

■ Select Sintaxe

```
SELECT [STRAIGHT_JOIN]
      [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
      [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
      [HIGH_PRIORITY]
      [DISTINCT | DISTINCTROW | ALL]
      select_expression,...
      [INTO {OUTFILE | DUMPFILE} 'file_name' export_options]
```

Linguagem SQL

■ Select Sintaxe

```
[FROM table_references  
  [WHERE where_definition]  
  [GROUP BY {unsigned_integer | col_name | formula} [ASC | DESC], ...  
  [WITH ROLLUP]]  
[HAVING where_definition]  
[ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC], ...]  
[LIMIT [offset,] row_count | row_count OFFSET offset]  
[PROCEDURE procedure_name(argument_list)]  
[FOR UPDATE | LOCK IN SHARE MODE]]
```

Linguagem SQL

- Apagando registros
- `DELETE FROM teste WHERE (codigo=1);`
- `TRUNCATE TABLE nome_tabela`
- apaga todos os registros da tabela

Linguagem SQL

■ Delete sintaxe

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM table_name  
    [WHERE where_definition]  
    [ORDER BY ...]  
    [LIMIT row_count]
```

or:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] table_name[.*] [,  
    table_name[.*] ...]  
FROM table-references  
[WHERE where_definition]
```


Linguagem SQL

- Alterando registros
- UPDATE teste SET nome = 'Debora C Nazario' WHERE nome = 'Debora';

Linguagem SQL

- Update Sintaxe

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name  
  SET col_name1=expr1 [, col_name2=expr2 ...]  
  [WHERE where_definition]  
  [ORDER BY ...]  
  [LIMIT row_count]
```

Linguagem SQL

- Operadores Lógicos
- AND
 - `SELECT * FROM teste WHERE (nome = 'Debora' AND email='debora.nazario@udesc.br')`
- OR
 - `SELECT * FROM teste WHERE (nome = 'Debora' OR email='debora.nazario@udesc.br')`
- NOT (!)
 - `SELECT * FROM teste WHERE (nome != 'Debora')`

Linguagem SQL

- ORDER BY

- ordena as informações por determinado(s) campos
- `SELECT * FROM teste ORDER BY nome;`

- LIKE

- pesquisa valores "aproximados"
- `SELECT * FROM teste WHERE nome LIKE 'DE%'`
- selecionará todos os nomes que começam com "DE"
- ex. DEBORA, DENISE

Linguagem SQL

■ LIKE

- `SELECT * FROM teste WHERE nome LIKE '%RA'`
- selecionará todos os nomes que terminam com "RA"
- ex. DEBORA, MARA

- `SELECT * FROM teste WHERE nome LIKE '%NO%'`
- selecionará todos os nomes que tenham "NO" em qualquer posição
- ex. FABIANO, MANOEL, NORMA

Linguagem SQL

- Apagar uma tabela
- `DROP TABLE nome_tabela`
- `DROP TABLE teste`
- Alterar a estrutura de uma tabela
- `ALTER TABLE nome_tabela ...`
- `ALTER [IGNORE] TABLE tbl_name
alter_specification [, alter_specification] ...`

Linguagem SQL

alter_specification:

```
ADD [COLUMN] create_definition [FIRST | AFTER column_name ]  
| ADD [COLUMN] (create_definition, create_definition,...)  
| ADD INDEX [index_name] [index_type] (index_col_name,...)  
| ADD [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...)  
| ADD [CONSTRAINT [symbol]] UNIQUE [index_name] [index_type]  
  (index_col_name,...)  
| ADD FULLTEXT [index_name] (index_col_name,...)  
| ADD [CONSTRAINT [symbol]] FOREIGN KEY [index_name] (index_col_name,...)  
  [reference_definition]  
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}  
| CHANGE [COLUMN] old_col_name create_definition  
  [FIRST | AFTER column_name]
```

Linguagem SQL

- | MODIFY [COLUMN] create_definition [FIRST | AFTER column_name]
- | DROP [COLUMN] col_name
- | DROP PRIMARY KEY
- | DROP INDEX index_name
- | DISABLE KEYS
- | ENABLE KEYS
- | RENAME [TO] new_tbl_name
- | ORDER BY col
- | CHARACTER SET character_set_name [COLLATE collation_name]
- | table_options

Linguagem SQL

- Exemplo:
- `CREATE TABLE t1 (a INTEGER, b CHAR(10));`
 - cria a tabela t1, com campos a e b
- `ALTER TABLE t1 RENAME t2;`
 - a tabela t1 foi renomeada para t2
- `ALTER TABLE t2 MODIFY a TINYINT NOT NULL, CHANGE b c CHAR(20);`
 - altera a tabela t2: a muda de INTEGER para TINYINT NOT NULL
 - e b muda para c e passa de CHAR(10) para CHAR(20)

Linguagem SQL

- `ALTER TABLE t2 ADD d TIMESTAMP;`
 - adiciona o campo d
- `ALTER TABLE t2 ADD INDEX (d), ADD PRIMARY KEY (a);`
 - adiciona um índice d, e chave primária a
- `ALTER TABLE t2 DROP COLUMN c;`
 - apaga a coluna c

Linguagem SQL

- Apagar o base de dados
 - `drop database nome_base`
- Bibliografia:
 - www.mysql.com

Exercício

- Criar uma base de dados
- Criar pelo menos uma tabela com pelo menos 3 campos
- Executar comandos para:
 - Inserir dados
 - Alterar dados
 - Visualizar dados
 - Apagar dados
- Apagar tabela
- Apagar base de dados