

Mysql

“Quanto mais informações você
busca, mais você evolui.”

Vitor Ramos

SQL

Structured Query Language - Linguagem de Consulta Estruturada

- Desenvolvida pela IBM, nos anos 70;
- Inicialmente chamada SEQUEL
- Parte do SystemR (protótipo de BD relacional)

Concorrência

- Relational Software Inc - lançou o Oracle (baseado em SQL)

SQL

- surgem vários SGBD baseados em SQL
- Problemas - falta de padrões
- American National Standard Institute (ANSI) e International Standards Organizations (ISO) definem padrão SQL.

SQL1 -1986, com modific. em 1989

SQL2 -1992

SQL3 –1999

e depois disso tivemos as atualizações

Categorias

- **DDL – Data Definition Language**
 - Composta pelos comandos Create, Alter e Drop
 - Responsável por dar forma ao banco de dados
- **DML – Data Manipulation Language**
 - Comandos Select, Insert, Delete e Update
 - Responsável por manipular os dados acrescentando, modificando, apagando e fazendo consultas
- **DCL – Data Control Language**
 - Subgrupo da DML, composta pelos comandos Grant e Revoke
 - Responsável por controlar acesso dos usuários aos dados.

MySQL

- Características
 - Um dos bancos de dados relacionais mais rápidos do mercado
 - Tem linguagem simples
 - Suportado por Sistemas com filosofia UNIX, embora outros SO também forneçam suporte, como Windows por exemplo
 - Código fonte aberto
 - Alta estabilidade
 - Baixo custo

Conceitos

- Banco de dados
 - Conjunto de tabelas
- Tabela
 - Conjunto de registros(Ex. Conjunto de Alunos)
- Registro
 - Conjunto de campos que representa uma entidade (Ex. Dados de um Aluno)
- Campo
 - Menor unidade de informação a ser armazenada

Conceitos

- Tabela Alunos

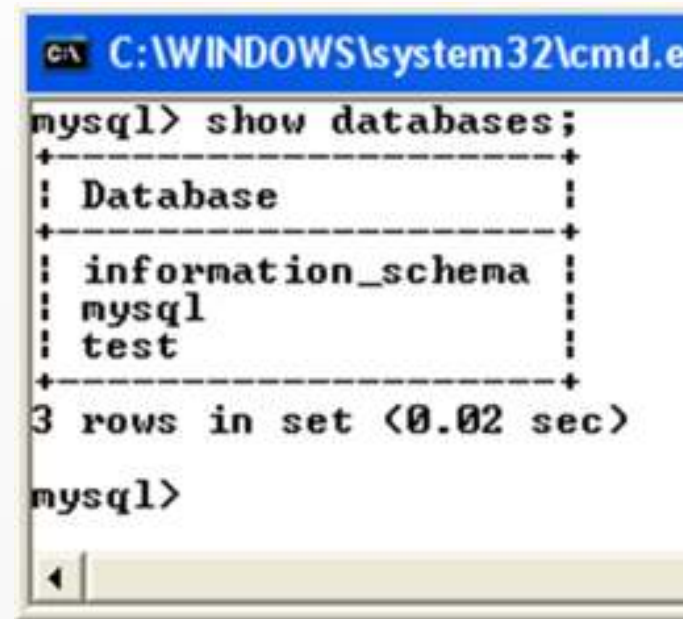
Campos

RA	nome	email
12345	Guilherme	Gui@uol.com.br
12346	Elaine	Brito@cotil.unicamp.br
12347	Roberto	Beto@vivax.com.br
12348	José	Ze@gmail.com.br

Registros

Bancos de dados

- O MySQL já vem com dois bancos de dados criados.
- Para visualizá-los basta executar o comando `show databases;`



```
C:\WINDOWS\system32\cmd.e
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| test      |
+-----+
3 rows in set (0.02 sec)

mysql>
```


Criação de Banco de Dados

- Quando se cria um banco de dados com o MySQL em ambiente Windows, é criada apenas uma pasta vazia, dentro da qual serão armazenados os arquivos gerados utilizando os comandos para criar tabelas.
- Como o MySQL trata o banco de dados como um esquema, temos duas opções de sintaxe para criar um banco de dados:

Mysql> CREATE DATABASE Exemplo;

ou

Mysql> CREATE SCHEMA Exemplo;

Visualizar Bancos de Dados

- `Mysql> SHOW DATABASES; [Enter]`
 - `+-----+`
 - `| Databases`
 - `+-----+`
 - `| information_schema`
 - `| mysql`
 - `| test`
 - `| Exemplo`
 - `+-----+`
- } criados na instalação do MySQL

Nomenclatura - banco de dados:

- Máximo 64 caracteres;
- Permitido: letras, números, traços, underlines;
- Proibido: barras e pontos;
- Evitar: acentos e cedilhas.

Excluindo banco de dados

- Ao apagar um banco de dados, todas as tabelas e dados também serão excluídos.

Mysql> **DROP DATABASE** Exemplo;

Ou

Mysql> **DROP SCHEMA** Exemplo;

- Com DROP também podemos usar a clausula IF EXISTS:

Mysql> **DROP DATABASE IF EXISTS** Exemplo;

Selecionando um banco de dados

- Podemos ter vários bancos de dados, mas só podemos manipular um por vez.
- Mysql> USE Exemplo;

Dentro do Banco de dados

- Bancos de dados: pasta vazia
- Precisamos criar tabelas dentro dele
- Para criar Tabelas precisamos definir sua estrutura = seus campos
- Exemplo: Tabela: Aluno
Campos: RA, nome, email
- Também precisamos definir tipos dos Campos

Tipos de Dados

- No MySQL os tipos de dados são divididos em três grupos:
 - Tipos Numéricos
 - Tipos de Data
 - Tipo de Cadeia

Tipos Numéricos

- Exatos:
 - NUMERIC
 - DECIMAL
 - INTEGER
 - SMALLINT, entre outros
- Aproximados:
 - FLOAT
 - REAL ou DOUBLE
 - DECIMAL ou DEC.

Tipos Numéricos

Tipo	Bytes	De	Até
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INT ou Integer	4	-2147483648	2147483647
BIGINT	8	-9223372036854775808	9223372036854775807
Bit ou Bool	1	Inteiro que pode ser 0 ou 1	

Tipos Numéricos

- NUMERIC e DECIMAL
 - implementados como o mesmo tipo.
 - preserva a exatidão (Ex. dados monetários).
- Exemplo:

Precisão: número de dígitos
que serão armazenados

Escala: número de dígitos
que serão armazenados
após o ponto decimal

– salario DECIMAL(5,2)

Tipos Numéricos

- **FLOAT** - números aproximados
 - Números pequenos
 - permite uma especificação opcional da precisão
 - quando não especifica precisão utiliza quatro bytes
- **REAL e DOUBLE**
 - não aceitam especificações de precisão.
 - implementados como valores de ponto flutuante de 8 bytes de dupla precisão

Tipos de Data e Hora

- DATE
 - para apenas do valor da data, sem a parte da hora.
 - formato 'YYYY-MM-DD'.
 - faixa de '1001-01-01' até '9999-12-31'.
- DATETIME
 - para valores que contém data e a hora.
 - formato 'YYYY-MM-DD HH:MM:SS'.
 - faixa de '1001-01-01 00:00:00' até '9999-12-31 23:59:59'.

Tipos de Data e Hora

- TIME
 - formato 'HH:MM:SS'
 - faixa '-838:59:59' até '838:59:59'.
 - Hora é grande pois pode ser usado para intervalos de tempo entre dois eventos. Por exemplo e não só para hora do dia que seria até 24
- YEAR
 - formato com 2 ou 4 algarismos.
 - faixa de 1901 até 2155.

Tipos de Cadeia

- CHAR(N)
 - caracteres alfanuméricos
 - tamanho é fixo e instaurado ao ser criado.
 - pode ter de 0 a 255 caracteres.
 - Exemplo: endereço CHAR(30);
 - Observe que não há acentos nem cedilhas no nome do campo, pois muitos servidores não acentuam, e sua tabela teria difícil acesso.

Tipos de Cadeia

- **VARCHAR(N)**
 - aloca apenas o espaço necessário para gravação
 - CHAR tamanho definido fixo (mesmo que não usado, aquele espaço em disco é alocado)
 - trocamos espaço por velocidade, pois este campo é 50% mais lento que o anterior.
 - Exemplo: endereço VARCHAR(30);
 - Define um campo chamado endereço que pode conter até 30 letras. Se você preencher apenas duas, o campo não ocupará todos os 30 bytes, mas apenas 2.

Tipos de Cadeia

- TEXT/BLOB
 - para guardar grandes quantidades de caracteres.
 - pode conter de 0 a 65535 bytes,
 - TEXT não é sensível a letras maiúsculas e minúscula quando uma comparação é realizada, e o BLOB sim.

Tipos de Cadeia

- MediumTEXT/MediumBLOB
 - Máximo de 16.777.215 caracteres.
- LongTEXT/LongBLOB
 - Máximo de 4.294.967.295 caracteres.

Tipos de Cadeia

- SET

- permite que o usuário faça uma escolha dado determinado número de opções.
- cada campo pode conter até, 64 opções.
- Exemplo:
 - curso SET("informatica", "geomatica") NOT NULL;
 - Neste exemplo este campo pode conter apenas os seguintes itens: " "
 - "informatica"
 - "geomatica"
 - "informatica,geomatica"

Tipos de Cadeia

- ENUM
 - semelhante ao SET, com a diferença que apenas um valor pode ser escolhido.
 - Exemplo:
 - sexo ENUM("masculino", "feminino") NOT NULL;
 - Neste exemplo este campo pode conter os seguintes valores: “ “
 - “masculino“
 - “feminino“

Tabelas

- SQL oferece três instruções para definição do esquema da base de dados:
- **Create Table**
 - define a estrutura da tabela
 - cria a tabela vazia
- **Drop Table**
 - Elimina a tabela da base de dados
- **Alter Table**
 - Permite modificar a definição da tabela

Criação de Tabela

```
mysql> CREATE TABLE teste(  
    > codigo INT,  
    > nome CHAR(15),  
    > email CHAR(25));
```

Nome da tabela

Tipos dos campos

Nome dos campos

Restrições

- Uma maneira de limitar os dados que podem ser inseridos em uma tabela é a definição de tipo.
- Podemos desejar definir outras restrições como:
 - Dados de um coluna em relação a outras colunas ou linhas.
 - Valores aceitáveis.

Restrições

- Não Nulo:
 - Define que uma coluna não pode conter valor nulo.
 - A cláusula NOT NULL especifica que uma coluna não admite valor vazio.
 - Exemplo: nome CHAR(15) NOT NULL;
 - o campo nome não pode ser vazio, é de preenchimento obrigatório.

Restrições

- Unicidade:

- Define que os dados contidos na coluna (ou grupo de colunas) sejam únicos (não se repitam) em relação a todas as outras linhas da tabela.

- A cláusula UNIQUE especifica que os dados não se repetem.

Restrições

```
CREATE TABLE alunos (  
    ra      char(5) UNIQUE,  
    nome char(50),  
    nasc  date );
```

Ou

```
CREATE TABLE alunos (  
    ra      char(5),  
    nome char(50),  
    nasc  date,  
    UNIQUE (ra) );
```

Restrições

Se uma restrição de unicidade faz referencia a um grupo de colunas, elas são separadas por vírgula:

```
CREATE TABLE exemplo (  
  a    char(5),  
  b    char(50),  
  c    date,  
  UNIQUE (a,c)    );
```

Restrições

- Chave primária:
- A chave primária indica que a coluna, ou grupo de colunas, pode ser utilizado como identificador único para as linhas da tabela
- É a junção de restrição de unicidade com a restrição de não nulo. Apenas a unicidade não garante identificador único pois não exclui os valores nulos.
- Uma tabela pode ter no máximo uma chave primária, mas pode ter várias restrições de unicidade e de não nulo.

Restrições

```
CREATE TABLE alunos (  
    ra      char(5) UNIQUE NOT NULL,  
    nome char(50),  
    nasc  date    );
```

ou

```
CREATE TABLE alunos (  
    ra char(5) PRIMARY KEY,  
    nome char(50),  
    nasc date);
```

A chave primária pode ser composta de vários atributos:

```
CREATE TABLE exemplo (  
    a      char(5),  
    b      char(50),  
    c      date,  
    PRIMARY KEY (a,c));
```

Restrições

- **Auto incremento:**
 - Este recurso, faz com que conforme novos registros são criados, automaticamente estes obtém valores que correspondem ao valor deste mesmo campo no registro anterior, somado a 1.
 - Exemplo: código `INT AUTO_INCREMENT`;
 - Soma um a cada registro automaticamente neste campo. Começando de 1, com inserção subsequente.

Praticando...

- Antes de criar uma tabela, precisamos selecionar o banco de dados dentro do qual ela será criada, para isso utilize o comando `USE nome do banco;`
- Criar tabela:

```
CREATE TABLE teste( codigo INTEGER  
    AUTO_INCREMENT NOT NULL,  
    nome CHAR(15) NOT NULL,  
    email CHAR(30),  
    telefone CHAR(8),  
    PRIMARY KEY(codigo));
```

Comandos relativos as tabelas:

- Mostrar tabelas - Lista todas as tabelas existentes no banco de dados atual.

```
mysql>show tables;
```

- Mostrar colunas - Mostra as colunas da tabela.

```
mysql>show columns from teste;
```

- Mostrar estrutura - Mostra a estrutura da tabela.

```
mysql>describe teste;
```

Alteração de Tabelas

- Quando notamos que as necessidades da aplicação mudaram ou que foi cometido um erro, podemos modificar a estrutura das tabelas já criadas.
- Podemos incluir ou excluir colunas, restrições, modificar nome de coluna ou da própria tabela.
- Tudo isso pode ser feito através do comando `ALTER TABLE`

Modificação da estrutura de uma tabela

- ADD <campo> <tipo>
 - Insere novo campo
- DROP <campo>
 - Remove determinado campo
- MODIFY<campo><tipo>
 - Modifica o tipo de determinado campo

Alter Table - ADD

- Inserir na tabela teste o campo nascimento que conterá a data de nascimento dos cadastrados.

```
mysql>alter table teste add nascimento date;
```

A nova coluna não pode possuir a restrição de não-nulo, porque a coluna inicialmente deve conter valores nulos. Porém, a restrição de não-nulo pode ser adicionada posteriormente.

Observe as alterações com o describe teste;

Alter Table - ADD

- Inserir na tabela teste o campo endereço após o campo nome.

```
mysql>alter table teste add endereco char(50)  
after nome;
```

Observe as alterações com o describe teste;

Obs. Para inserir antes de todos os outros campos use
first

Alter Table - MODIFY

- O campo email foi criado com limite de 30 caracteres. Observe isso com o comando describe teste;
- Trocar para 40 caracteres .
- `mysql>alter table teste modify email CHAR(40);`
- Execute o describe teste; novamente para observar a alteração.

Alter Table - CHANGE

- Trocar no nome da coluna email por e_mail .
- `mysql>alter table teste change email e_mail char(30);`
- Execute o `describe teste;` para observar a alteração.

Alter Table - Drop

- Abaixo vemos como excluir o campo codigo da tabela Teste:

```
mysql>alter table teste drop codigo;
```

Observe as alterações com o describe teste;

Alter Table – ADD Primary Key

- Abaixo vemos como definir o campo nome como chave para a tabela Teste:

```
mysql>alter table teste add primary key  
      (nome,nascimento);
```

Observe as alterações com o describe teste;

Alter Table - Drop Primary Key

- Exclui a chave primária, mas não a coluna

```
mysql>alter table teste drop primary key;
```

Observe as alterações com o describe teste;

Renomeando a tabela

- Para alterar o nome da tabela A para B

```
mysql> ALTER TABLE A RENAME TO B;
```

Drop Table

- Exclui a tabela. Todos os dados e definições da tabela são removidos, assim tenha cuidado com este comando!

Observação: não vamos executar este comando pois vamos continuar usando a tabela teste em aula apenas observe o comando.

```
mysql>drop table teste;
```

Praticando...

- Faça as alterações necessárias na tabela para que ela tenha as características iniciais;
- Tabela teste
 - código inteiro, auto-incremento, chave primária
 - nome CHAR(15)
 - email CHAR(30)
 - telefone CHAR(8)