

Banco de Dados

Introdução ao MySQL

Wolney Henrique Queiroz Freitas
wolney.freitas@professores.unifavip.edu.br

MySQL

- O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL como interface;
- É atualmente um dos sistemas de gerenciamento de bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo.

MySQL: Características

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, C#, Visual Basic, Python, Perl, PHP, ASP e Ruby);
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de novos hardware;
- Facilidade no manuseio;
- É um Software Livre com base na GPL (entretanto, se o programa que acessar o Mysql não for GPL, uma licença comercial deverá ser adquirida);
- Contempla a utilização de vários Storage Engines como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid;
- Suporta controle transacional;
- Suporta Triggers;
- Suporta Cursors (Non-Scrollable e Non-Updatable);
- Suporta Stored Procedures e Functions;
- Replicação facilmente configurável;
- Interfaces gráficas (MySQL Toolkit) de fácil utilização cedidos pela MySQL Inc.

MySQL: Links para instalação

Opção recomendada e a que será utilizada nas aulas em laboratório, SGBD Mysql + ferramenta gráfica para sua utilização:

- MySQL
 - <https://dev.mysql.com/downloads/installer/>

MySQL: Links para instalação

- Se não quiser instalar o MySQL + MySQL Workbench, você pode optar por instalar o pacote Xampp, que vem com o MySQL e a ferramenta gráfica PhpMyAdmin:
 - https://www.apachefriends.org/pt_br/index.html

Laboratório de Banco de Dados

SQL: Operações de manipulação das estruturas do BD

Wolney Henrique Queiroz Freitas
wolney.freitas@professores.unifavip.edu.br

Roteiro de Aula

- SQL
 - Criar, atualizar e deletar banco de dados;
 - Criar, atualizar e deletar tabelas.

Linguagem SQL

- Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional;
- A linguagem é um grande padrão de banco de dados. Isto decorre da sua simplicidade e facilidade de uso.

Linguagem SQL

- A linguagem SQL é dividida em subconjuntos de acordo com as operações que queremos efetuar sobre um banco de dados, tais como:
 - DDL - Linguagem de Definição de Dados
 - Permite ao utilizador definir tabelas novas e elementos associados;
 - DML - Linguagem de Manipulação de Dados
 - Inclusões, alterações e exclusões de dados;
 - DQL - Linguagem de Consulta de Dados
 - Consulta a dados.

Criar Banco de Dados

- Comando SQL para criar um banco de dados:

```
create database loja_db;
```

Criar Banco de Dados

- Na criação de um banco de dados podemos configurar propriedades e regras (Constraints) específicas para nosso banco de dados;
- Exemplo
 - Trabalhar com Charset uft-8, mais apropriado para línguas latinas.

Criar Banco de Dados

- Comando SQL para criar um banco de dados definindo o charset e o collate no MySQL:

```
create database loja_db  
default character set utf8  
default collate UTF8_GENERAL_CI;
```

Criar Banco de Dados

- Diferença entre CHARSET e COLLATE:
- Charset
 - O Charset define o conjunto de caracteres que você irá utilizar para armazenar dados em forma de texto, onde cada caractere recebe um número;
- Collate
 - Collate é o termo utilizado para definir o conjunto de regras que o servidor irá utilizar para ordenação e comparação entre textos, ou seja, como será o funcionamento dos operadores =, >, <, order by, etc.

Atualizar Banco de Dados

- Comando SQL para atualizar propriedades de um banco de dados no MySQL:

```
alter database loja_db  
default character set utf8  
default collate UTF8_GENERAL_CI;
```

Deletar Banco de Dados

- Comando SQL para deletar um banco de dados:

```
drop database loja_db;
```

Criar Tabelas

- Comando `create table`

- Sintaxe:

```
create table nome_da_tabela (  
    nome_da_coluna1 tipo_da_coluna1(tamanho_da_coluna_1)  
    restrições da coluna 1,  
    nome_da_coluna2 tipo_da_coluna2(tamanho_da_coluna_2)  
    restrições da coluna 2,  
    ...  
    declarar chave primária e chaves estrangeiras da tabela  
)
```


Criar Tabelas

- Ao se criar uma estrutura de uma tabela é necessário que o usuário forneça, para cada coluna, as seguintes informações:
 - Tipo de Dado
 - Ex: varchar, float, date, int, etc;
 - Tamanho;
 - Regras / Restrições (Constraints)
 - Ex: unique, not null, autoincrement, etc.

Criar Tabelas: Tipo de Dado

- Tipos Primitivos do MySQL:
- Numérico
 - Inteiro (TinyInt, SmallInt, Int, MediumInt, BigInt);
 - Real (Decimal, Float, Double, Real);
 - Lógico (Bit, Boolean);
- Data/Tempo
 - Date, DateTime, TimeStamp, Time, Year;
- Literal
 - Caractere (Char, VarChar);
 - Texto (TinyText, Text, MediumText, LongText);
 - Binário (TinyBlob, Blob, MediumBlob, LongBlob);
 - Coleção (Enum, Set);
- Espacial
 - Geometry, Point, Polygon, MultiPolygon.

Criar Tabelas: Constraints

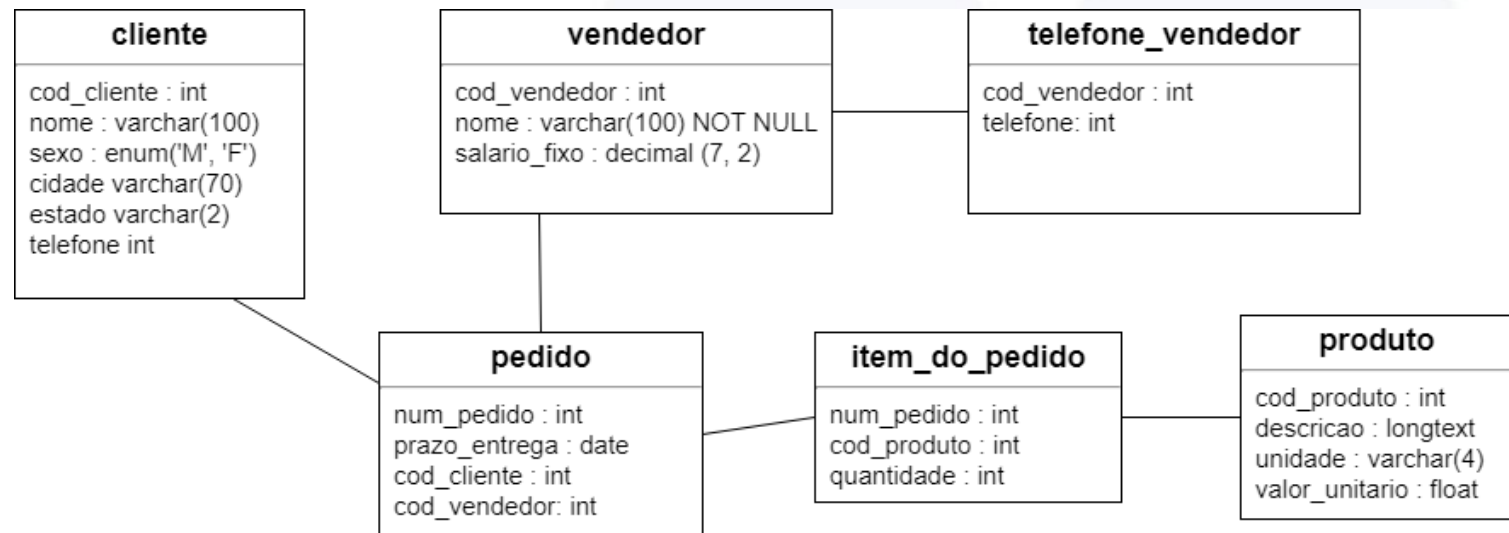
- As Restrições são regras aplicadas nas colunas de uma tabela;
- São usadas para limitar os tipos de dados que são inseridos;
- Podem ser especificadas no momento de criação da tabela (CREATE) ou após a tabela ter sido criada (ALTER).

Criar Tabelas: Constraints

- As principais constraints MySQL são as seguintes:
- NOT NULL
 - A constraint NOT NULL impõe a uma coluna a NÃO aceitar valores NULL;
- UNIQUE
 - Identifica de forma única cada registro em uma tabela de um banco de dados;
- PRIMARY KEY
 - A restrição PRIMARY KEY (Chave Primária) identifica de forma única cada registro em uma tabela de banco de dados;
- FOREIGN KEY
 - Uma FOREIGN KEY (Chave Estrangeira) em uma tabela é um campo que aponta para uma chave primária em outra tabela. Desta forma, é usada para criar os relacionamentos entre as tabelas no banco de dados;
- DEFAULT
 - A restrição DEFAULT é usada para inserir um valor padrão especificado em uma coluna;
- Autoincrement
 - O valor de uma coluna numérica em uma tabela é preenchido automaticamente para cada registro de forma incremental.

Criar Tabelas: Exemplo

- Estrutura do banco de dados da loja:



Criar Tabelas: Exemplo

- Tabela Cliente

```
create table cliente (  
  cod_cliente int,  
  nome varchar(100),  
  sexo enum('M', 'F'),  
  cidade varchar(70),  
  estado varchar(2),  
  telefone int,  
  
  constraint pk_cliente primary key(cod_cliente)  
);
```

Criar Tabelas: Exemplo

- Tabela Vendedor

```
create table vendedor (  
  cod_vendedor int,  
  nome varchar(200) NOT NULL,  
  salario_fixo decimal(7, 2),  
  
  constraint pk_vendedor primary key(cod_vendedor)  
);
```

Criar Tabelas: Exemplo

- Tabela Telefone Vendedor

```
create table telefone_vendedor(  
  cod_vendedor int,  
  telefone int,  
  
  constraint pk_telefone_vendedor primary  
  key(cod_vendedor, telefone),  
  constraint fk_vendedor_telefone_vendedor foreign  
  key(cod_vendedor) references vendedor(cod_vendedor)  
);
```


Criar Tabelas: Exemplo

- Tabela Produto

```
create table produto (  
  cod_produto int,  
  descricao longtext,  
  unidade varchar(4),  
  valor_unitario float,  
  
  constraint pk_produto primary key(cod_produto)  
);
```

Criar Tabelas: Exemplo

- Tabela Pedido

```
create table pedido (  
    num_pedido int,  
    prazo_entrega date,  
    cod_cliente int,  
    cod_vendedor int,  
  
    constraint pk_pedido primary key(num_pedido),  
    constraint fk_cliente_pedido foreign key(cod_cliente) references  
cliente(cod_cliente),  
    constraint fk_vendedor_pedido foreign key(cod_vendedor) references  
vendedor(cod_vendedor)  
);
```

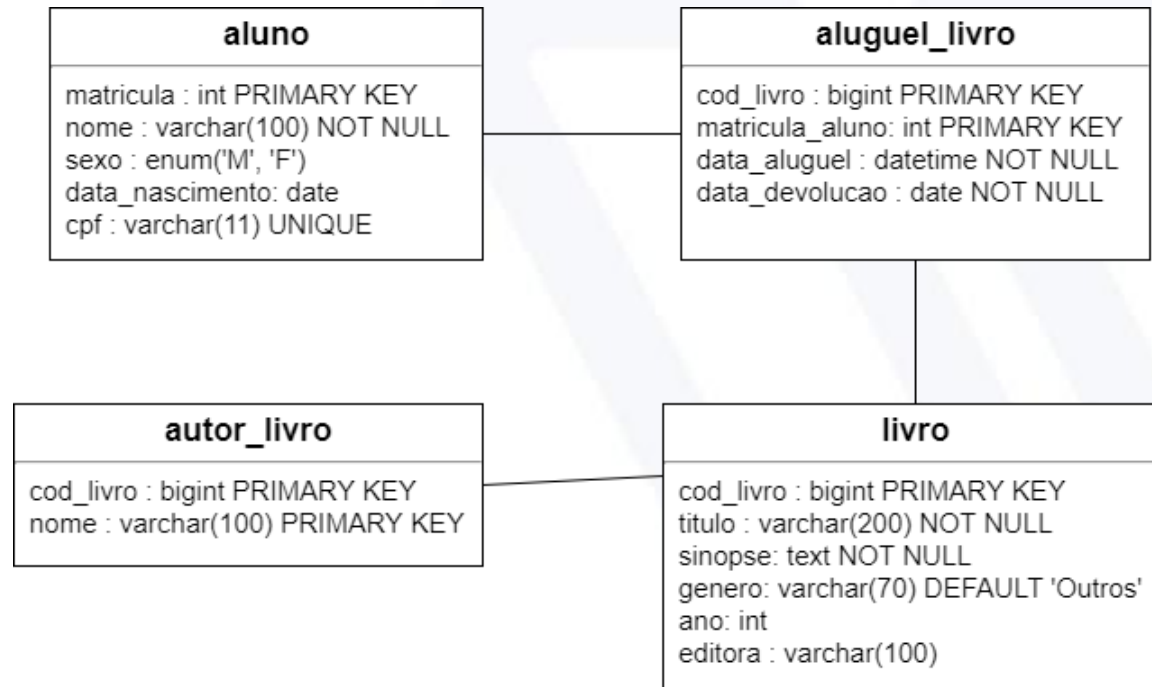
Criar Tabelas: Exemplo

- Tabela Item do pedido

```
create table item_do_pedido(  
    num_pedido int,  
    cod_produto int,  
    quantidade int,  
  
    constraint pk_item_do_produto primary key(num_pedido,  
cod_produto),  
    constraint fk_pedido_item_do_produto foreign key(num_pedido)  
references pedido(num_pedido)  
);
```

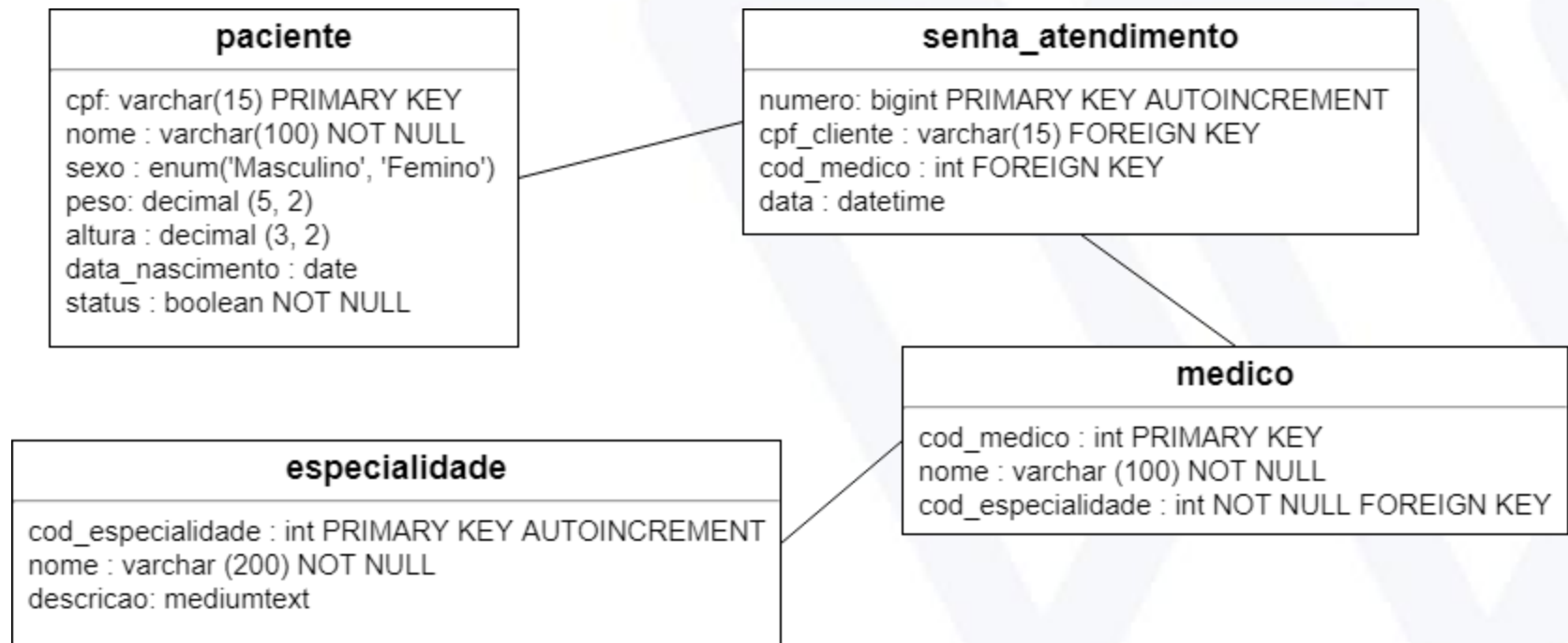
Vamos Praticar...

- Crie os comandos SQL para criar as tabelas do banco de dados de alugueis da biblioteca de uma faculdade:



Vamos Praticar...

- Crie os comandos SQL para criar as tabelas do banco de dados de atendimentos de um consultório médico:



Deletar Tabelas

- Comando **drop table**

```
drop table item_do_pedido;
```

Alterar Tabelas

- Comando **alter table**

alter table <tabela> <ADD | MODIFY | DROP>
(coluna)

Alterar Tabelas

- Adicionar coluna data_nascimento na tabela cliente:
`alter table cliente add (data_nascimento date);`
- Deletar coluna data_nascimento na tabela cliente:
`alter table cliente drop data_nascimento;`
- Para modificar o tamanho da coluna unidade na tabela produto
`alter table produto MODIFY unidade varchar(10);`

Alterar Tabelas: Renomear Tabela

```
alter table cliente  
rename to cliente2;
```

```
alter table cliente2  
rename to cliente;
```

Alterar Tabelas: Renomear Coluna

`alter table produto`

`change unidade unidade_medida varchar(10);`



Faci facid FACIMP FBV fmc Faculdade
Martha Falcão ISL UNIFAVIP UNI  METROCAMP RUY BARBOSA | AREA1 UniFBV UniFanor