

Desenvolvimento para a Web



Módulo 2

Roteiro

- ▣ Banco de Dados
- ▣ Node.JS
- ▣ HTML - CSS - JavaScript

Recursos

❑ Github

- <https://github.com/>
- https://rogerdudler.github.io/git-guide/index.pt_BR.html

❑ GitHub – Repositório do Curso

- <https://github.com/luisalvarenga-pde/>

❑ Instalação do MySQL

- <https://dicasdeprogramacao.com.br/como-instalar-o-mysql-no-windows/>

Banco de Dados



Imersão no MySQL

Objetivos

- ❑ Entender o que é um Banco de Dados
- ❑ Instalar o MySQL
- ❑ DDL - Comandos de criação de objetos de banco de dados (tabelas e visões)
- ❑ DML - Comandos de manipulação de dados
- ❑ MER - Modelo de Entidades e Relacionamentos

Introdução ao MySQL

- ❑ MySQL é um Sistema de Gerenciamento de Banco de Dados Relacional (SGBD) que implementa a Linguagem de Consulta Estruturada – SQL (Structured Query Language)
- ❑ MySQL é compatível com o padrão SQL ISO/ANSI
- ❑ A Oracle fornece uma versão comercial do MySQL

Recursos

- Endereço de download do MySQL

<http://dev.mysql.com/downloads/>

- Endereço dos scripts para criação do banco de dados e carga nas tabelas

https://github.com/luisalvarenga-pde/banco-de-dados/blob/master/exemplo/banco/banco_bd.sql

https://github.com/luisalvarenga-pde/banco-de-dados/blob/master/exemplo/banco/banco_dados.sql

Acesso ao MySQL

▣ Digitar no prompt de comando (cmd)

>mysql -u [usuario] -h [endereço-do-servidor] -p

>Enter password:[senha]

Comandos

❑ mysql>

- Usuário entra com um comando
- Mysql envia o comando para o servidor executa-lo
- Mysql apresenta o resultado
- Mysql aguarda entrada de novo comando

mysql>

- ❑ Um comando pode ocupar várias linhas
- ❑ Um comando, normalmente, consiste de declarações SQL seguidas de um ponto-e-vírgula

Prompt de Comandos

prompt	significado
mysql>	Pronto para receber um novo comando.
->	Aguardando a próxima linha em um comando com múltiplas linhas.
'>	Aguardando a próxima linha, esperando pela complementação do texto que iniciou com uma aspas simples (``'``).
">	Aguardando a próxima linha, esperando pela complementação do texto que iniciou com uma aspas duplas (""").
`>	Aguardando a próxima linha, esperando pela complementação do texto que iniciou com um acento grave (`` ` ``).
/*>	Aguardando a próxima linha, esperando pela complementação do texto que iniciou com um comentário (/*).

Comandos

- ▣ Ajuda \h
- ▣ Sair \q
- ▣ Cancelar\Interromper um comando \c

Banco de Dados e Tabelas

- ❑ Listar os bancos de dados no SGBD
 - >show databases;
- ❑ Acessar um banco de dados
 - >Use [nome-do-banco-de-dados]
- ❑ Mostrar o banco de dados em uso atualmente
 - > select database();
- ❑ Mostrar as tabelas do banco de dados em uso
 - >show tables;
- ❑ Mostrar a estrutura de uma tabela
 - > describe [nome-da-tabela];

Banco - Exemplo

agencia (nome-agencia, cidade-agencia, ativos)

cliente (nome-cliente, endereço-cliente, cidade-cliente)

conta (numero-conta, nome-agencia, saldo)

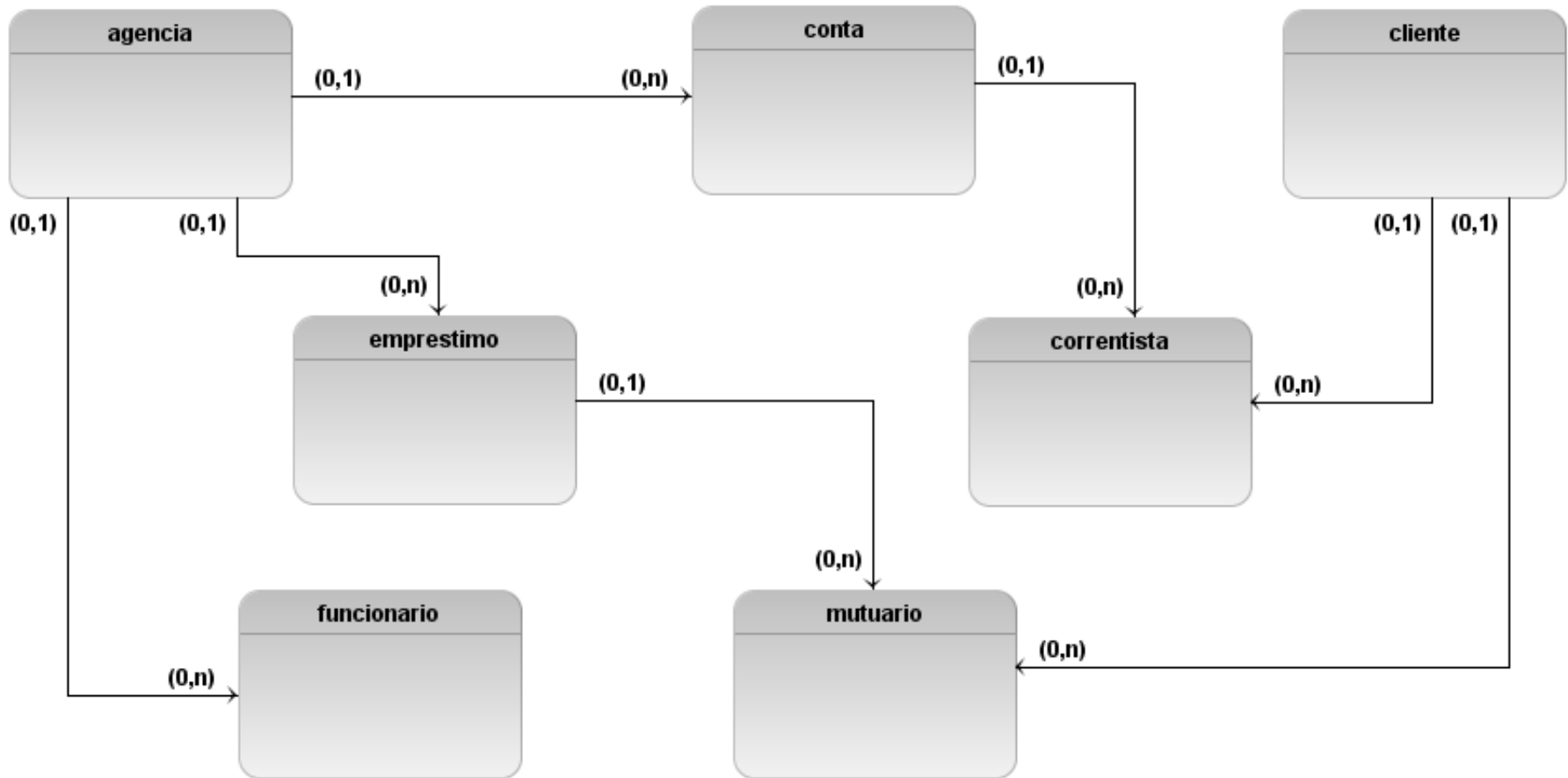
*emprestimo (numero-emprestimo, nome-agencia,
montante)*

correntista (nome-cliente, numero-conta)

mutuario (nome-cliente, numero-emprestimo)

funcionario (nome-funcionario, nome-agencia, salario)

Banco - Exemplo de MER



CREATE DATABASE

- ❑ Um banco de dados no Mysql é criado com o comando **CREATE DATABASE**:
 - **create database** [nome-do-banco-de-dados]
- ❑ Exemplo
 - **create database** *meubancodedados*

Script SQL para criação das tabelas

- ❑ O script SQL para criar o banco de dados 'banco' pode ser encontrado em:

https://github.com/luisalvarenga-pde/banco-de-dados/blob/master/exemplo/banco/banco_bd.sql

https://github.com/luisalvarenga-pde/banco-de-dados/blob/master/exemplo/banco/banco_dados.sql

Query

- ❑ Encontrar todos os empréstimos realizados na agência “Rio de Janeiro” com valor superior a R\$ 1.100,00.

```
select numero_emprestimo  
  from emprestimo  
 where nome_agencia = 'Rio de Janeiro'  
       and montante > 1100;
```

- ❑ Encontrar os empréstimos com montante entre R\$ 1.000,00 e R\$ 1.500,00 (ou seja, \geq R\$ 1.000,00 e \leq R\$ 1.500,00)

```
select numero_emprestimo  
  from emprestimo  
 where monante between 1000 and 1500;
```

Query

- ❑ Encontre todas as agencias que possuem valor de ativo superior ao da agência de 'Brasilia'.

```
select distinct T.nome_agencia
  from agencia as T
 join agencia as S
    on T.ativos > S.ativos
 where S.branch_city = 'Brasilia';
```

- ❑ Encontrar o nome do cliente, o número e montante do empréstimo tomado pelos clientes em qualquer agencia.

```
select T.nome_cliente, T.numero_emprestimo, S.montante
  from mutuario as T
 join emprestimo as S
    on T.numero_emprestimo > S.numero_emprestimo;
```

Operações com conjuntos

- ▣ Encontre todos os clientes que tenham um empréstimo, uma conta ou ambos:

```
(select nome_cliente from correntista)
union
(select nome_cliente from mutuario);
```

- ▣ Encontre todos os clientes que possuem uma conta mas não um empréstimo. (O MySQL não dispõe da declaração **minus**)

```
select nome_cliente
  from correntista
 where nome_cliente not in
       (select nome_cliente from mutuario);
```

Funções de Agregação

- ▣ Encontre a quantidade de clientes de cada agência.

```
select nome_agencia, count (distinct nome_cliente)
from correntista crt
join conta cnt
on crt.numero_conta = cnt.numero_conta
group by nome_agencia;
```

- ▣ Encontre todas as agências com saldo médio superior a R\$ 500,00.

```
select nome_agencia, avg (saldo)
from conta
group by nome_agencia
having avg (saldo) > 500;
```

Subquery

- ▣ Encontre todos os cliente que possuem uma conta e um emprestimo no banco.

```
select distinct nome_cliente  
  from mutuario  
 where nome_cliente in  
      (select nome_cliente from correntista);
```

- ▣ Encontre todos os clientes que possuem um emprestimo no banco, mas não uma conta.

```
select distinct nome_cliente  
  from mutuario  
 where nome_cliente not in  
      (select nome_cliente from correntista);
```

Subquery

- ▣ Encontre todas as agencias com ativos superiores aos da agencia de 'Niterói'.

```
select nome_agencia
from agencia
where ativos > all
    (select ativos
     from agencia
     where nome_cidade = 'Niterói');
```

Visões

- ❑ Criar uma View contendo as agência e seus clientes.

```
create view vw_clientes as  
  (select nome_agencia, nome_cliente  
    from correntista crt  
    join account cnt  
      on crt.numero_conta = cnt.numero_conta)  
union  
  (select nome_agencia, nome_cliente  
    from mutuario mtr  
    join emprestimo ept  
      on mtr.numero_emprestimo = ept.numero_emprestimo);
```

Junções

- ❑ **Join operations** - tome duas relações e retorne uma outra relação como resultado.
- ❑ São utilizadas frequentemente como Subqueries na clausula **from**
- ❑ **Join condition** – define os campos que correspondem-se nas relações e os campos que estão presentes no resultado da junção.
- ❑ **Join type** – define como os campos que não correspondem à junção são tratados.

Join types

inner join

left outer join

right outer join

full outer join

Junções - Exemplos

emprestimo

numero_emprestimo	nome_agencia	montante
L-170	Niterói	3.000,00
L-230	Rio de Janeiro	4.000,00
L-260	São Paulo	1.700,00

mutuario

nome_cliente	numero_emprestimo
João	L-170
Samuel	L-230
Heitor	L-155

■ Note:

- Faltam os dados do empréstimo do mutuário Heitor (empréstimo L-155)
- Faltam os dados do mutuário para o empréstimo L-260.

Junções - Exemplos

emprestimo

numero_emprestimo	nome_agencia	montante
L-170	Niterói	3.000,00
L-230	Rio de Janeiro	4.000,00
L-260	São Paulo	1.700,00

mutuario

nome_cliente	numero_emprestimo
João	L-170
Samuel	L-230
Heitor	L-155

```
select *  
  from emprestimo emp  
 inner join mutuario mut  
    on emp.numero_emprestimo = mut.numero_mutuario;
```

numero_emprestimo	nome_agencia	montante	nome_cliente	numero_emprestimo
L-170	Niterói	3.000,00	João	L-170
L-230	Rio de Janeiro	4.000,00	Samuel	L-230

Junções - Exemplos

emprestimo

numero_emprestimo	nome_agencia	montante
L-170	Niterói	3.000,00
L-230	Rio de Janeiro	4.000,00
L-260	São Paulo	1.700,00

mutuario

nome_cliente	numero_emprestimo
João	L-170
Samuel	L-230
Heitor	L-155

```
select *  
  from emprestimo emp  
 left join mutuario mut  
       on emp.numero_emprestimo = mut.numero_mutuario;
```

numero_emprestimo	nome_agencia	montante	nome_cliente	numero_emprestimo
L-170	Niterói	3.000,00	João	L-170
L-230	Rio de Janeiro	4.000,00	Samuel	L-230
L-260	São Paulo	1.700,00	NULL	NULL

Modificando o Banco de Dados

- Aumentar em 7% todas as contas com saldo superior a R\$ 800,00, as demais em 8%.

```
update conta
  set saldo = saldo * 1.07
where saldo > 800;
```

```
update conta
  set saldo = saldo * 1.08
where saldo <= 800;
```

Modificando o Banco de Dados

- Aumentar em 6% todas as contas com saldo superior a R\$ 700,00, as demais em 5%.

```
update conta
  set saldo =
    case
      when saldo <= 700 then saldo *1.05
      else saldo * 1.06
    end;
```

Modificando o Banco de Dados

- ❑ Excluir os registros das contas que possuem saldo inferior à média de saldo das contas do banco.

```
delete from conta  
where saldo < (select avg (saldo) from conta);
```

- ❑ Adicionar um novo registro na tabela *conta*

```
insert into conta  
values ('A-9732', 'Rio de Janeiro', 1200);
```