

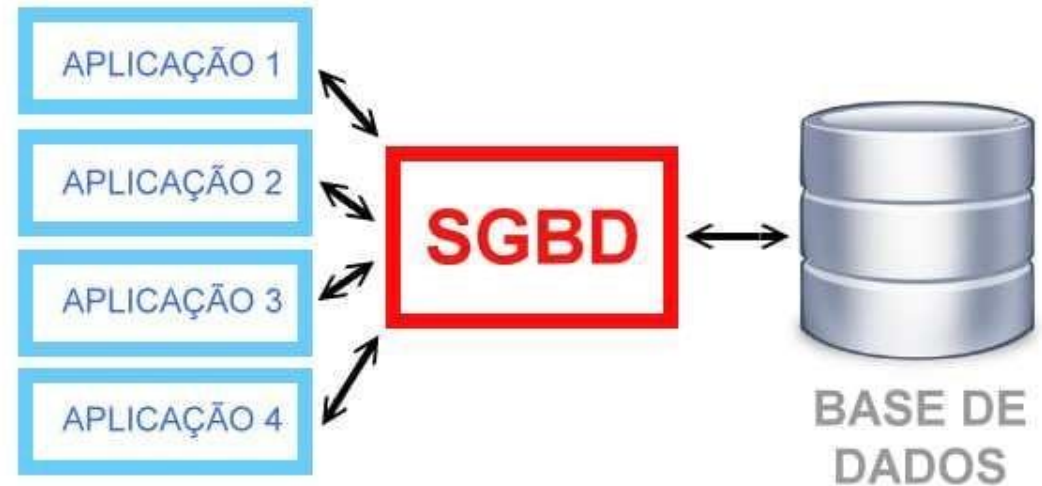
# **INTRODUÇÃO AO USO DO MYSQL EM PROGRAMAS JAVA**

**PROF. ME. HÉLIO ESPERIDIÃO**



# SGBD

- Os SGBD são os elementos mais comuns para persistência de dados utilizados em aplicações comerciais, pois propiciam formas padronizadas para inserção, alteração, remoção e busca de dados. Portanto, é necessário verificar como as interfaces gráficas, quando acionadas pelo usuário, fazem o uso dos SGDBs para gravar seus dados



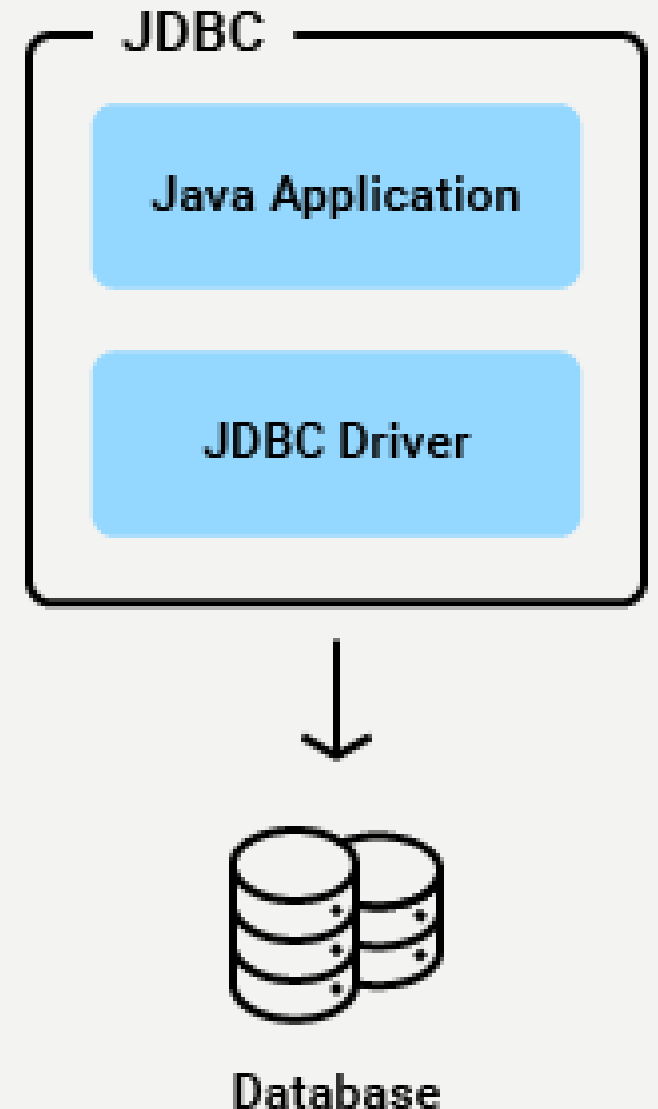
# EXISTEM DIVERSOS SGDBS

- Como existem diversos SGDBs, seria necessário utilizar bibliotecas específicas para cada sistema, o que causaria uma dependência ao tipo de persistência. Para utilizar os SGDBs em Java, especialmente em interfaces gráficas em Java Swing, é indicado utilizar o Java Database Connectivity (JDBC). O JDBC consiste em um conjunto de classes que são incorporadas ao Java Development Kit (JDK)



# JDBC

- Para possibilitar o acesso a diversos SGDBs de forma padronizada sem a necessidade de se utilizar formas específicas para cada sistema de banco de dados (FURGERI, 2015). O JDBC é compatível com diversos sistemas de banco de dados, tais como:
- MySQL
- Oracle
- PostgreSQL



# [HTTPS://DEV.MYSQL.COM/DOWNLOADS/CONNECTOR/J/5.1.HTML](https://dev.mysql.com/downloads/connector/j/5.1.html)

**Generally Available (GA) Releases**

## Connector/J 5.1.48


Select Operating System:  
Platform Independent

[Looking for the latest GA version?](#)

**Recommended Windows Download:**

**MySQL Installer**  
for Windows

All MySQL Products. For All Windows Platforms.  
In One Package.



Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

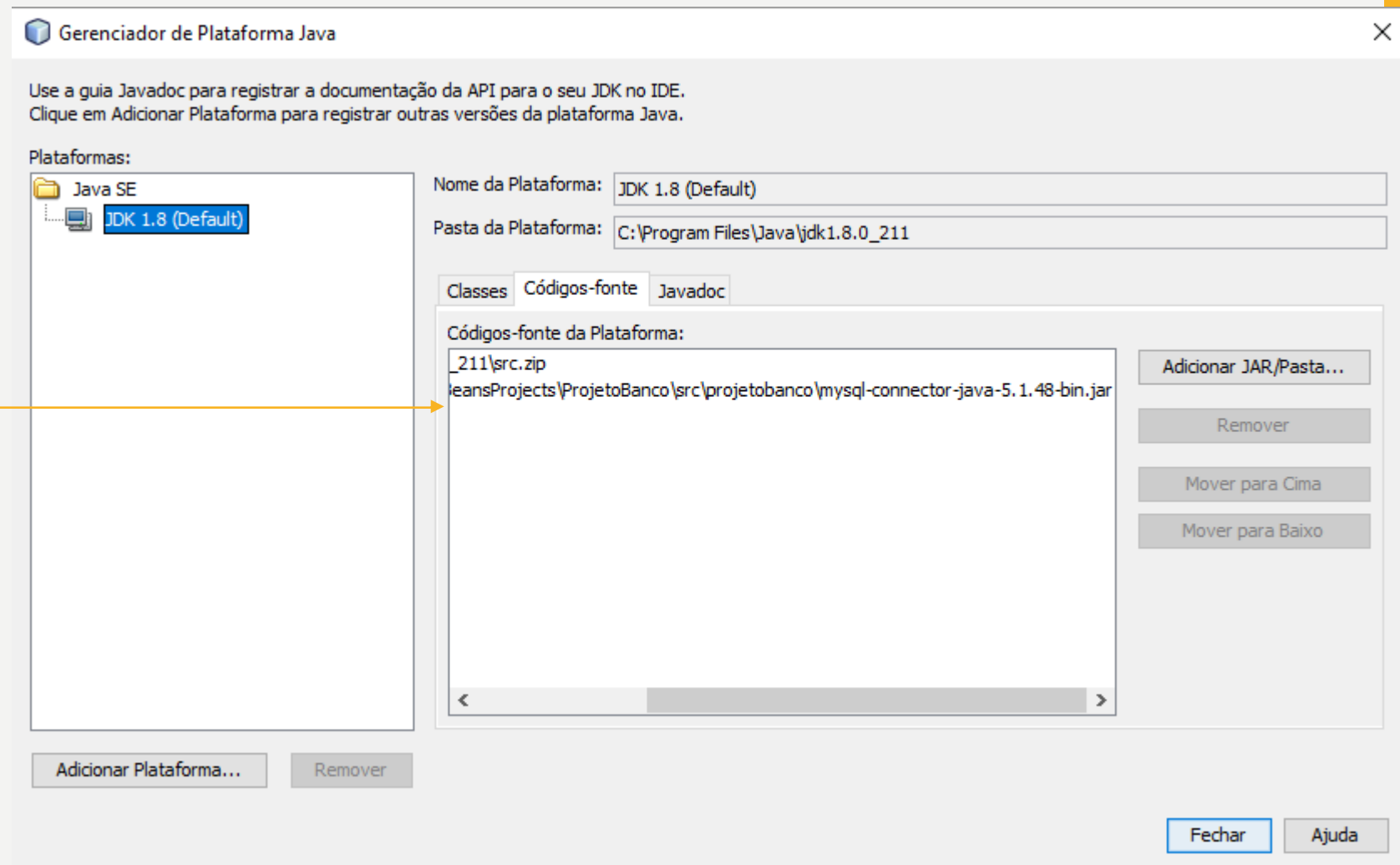
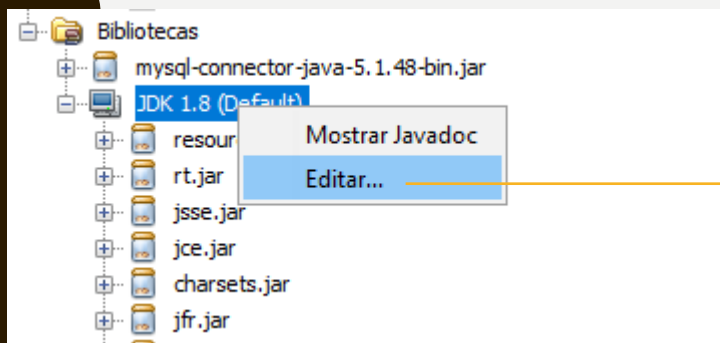
**Windows (x86, 32 & 64-bit), MySQL Installer MSI** [Go to Download Page >](#)

<b>Platform Independent (Architecture Independent), Compressed TAR Archive</b> (mysql-connector-java-5.1.48.tar.gz)	5.1.48	4.2M	<a href="#">Download</a>
<b>Platform Independent (Architecture Independent), ZIP Archive</b> (mysql-connector-java-5.1.48.zip)	5.1.48	4.6M	<a href="#">Download</a>

MD5: 9e6eee4e6df8d3474622bed952513fe5 | Signature

MD5: 5da24facd99964f296ecde32abcd2384 | Signature

# CONFIGURAÇÃO NA FERRAMENTA



# JDBC CONTROLADOR DO DRIVER

O JDBC utiliza as classes que controlam o driver que será utilizado para se conectar no banco de dados indicado.

Para executar o primeiro passo, que consiste em estabelecer a conexão, é necessário garantir que o JDBC conheça o SGBD (HORSTMANN, 2016).

# JAVA.SQL E JAVAX.SQL.

**java.sql.DriverManager:**  
criar a conexão com SGBD.

**java.sql.Connection:**  
preparar a conexão  
com o SGBD e fornecer  
acesso às consultas.

**java.sql.Statement:**  
executar as consultas e  
comandos no SGBD.

**java.sql.ResultSet:**  
recuperar os dados que  
foram buscados, por  
exemplo, um comando de  
select.

**javax.sql.DataSource:**  
agrupar conexões com o  
SGBD.



# STRING DE CONEXÃO

- Para se conectar a um banco de dados, esteja ele implementado em qualquer SGBD, é necessário criar uma string de conexão, ou URL JDBC (Uniform Resource Locator JDBC).
- Essa string informará o “caminho” do banco e apresenta a seguinte sintaxe:

**`jdbc:<driver>:<detalhes da conexão>`**

# EXEMPLOS

Banco de dados	URL JDBC
MySQL	<code>jdbc:mysql://localhost:3306/nomeBancoDeDados</code>
SQL Server	<code>jdbc:sqlserver://localhost;databaseName=nomeBancoDeDados</code>
Oracle	<code>jdbc:oracle:thin@myserver:1521:nomeBancoDeDados</code>

# PARÂMETROS

Os parâmetros da URL alteram conforme o SGBD usado, bem como com as configurações dos sistemas.

Exemplo: “jdbc:mysql://localhost:3306/nomebd”.

Pode ser necessário inserir mais informações, como por exemplo habilitar explicitamente o certificado SSL e o Time Zone, ficando com a seguinte sintaxe:

“jdbc:mysql://localhost:3306/nomebd?useSSL=false&serverTimezone=UTC”.

# UTILIZAR O JDBC – 5 ETAPAS

Estabelecer a conexão.


Criar um objeto da classe statement vinda da conexão para possibilitar a execução das consultas.

Executar as consultas.

Processar os resultados, sendo os dados enviados ou recebidos.

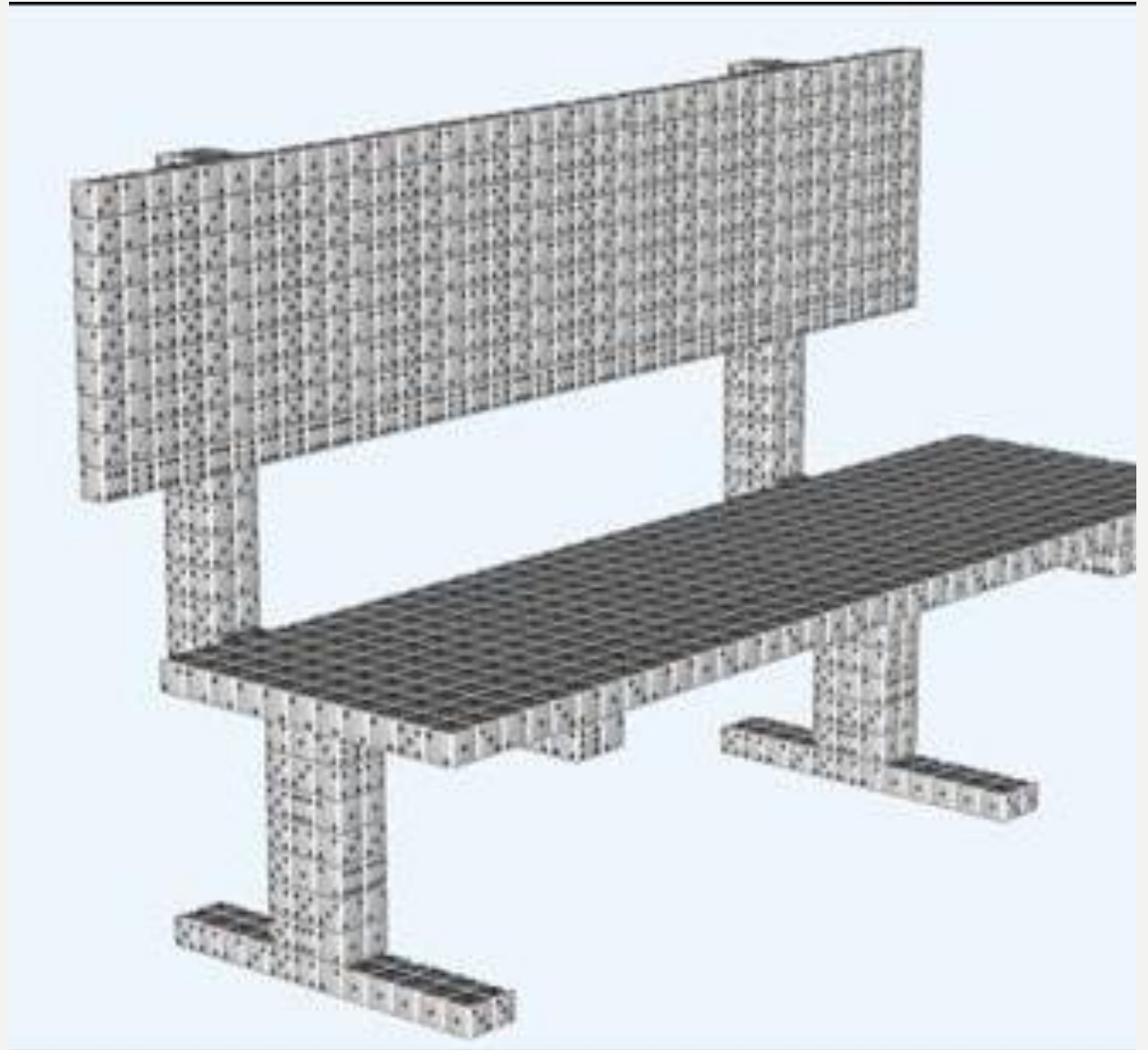
Fechar a conexão

# CRIAR UM BANCO

- Antes de iniciar qualquer ação para utilizar o SGBD, é necessário criar um banco de dados;
  - no caso do MySQL é possível criá-lo utilizando a linha de comando ou empregar alguma ferramenta como o MySQL WorkBench.
- 

# BANCO DE DADOS

- Podemos entender por banco de dados qualquer sistema que reúna e mantenha organizada uma série de informações relacionadas a um determinado assunto.



- Muitas vezes, a implantação da Informática em organizações ocorre de forma evolutiva e gradual. Inicialmente, apenas determinadas funções são automatizadas, à medida que o uso da Informática vai se estabelecendo, novas funções vão sendo vinculadas.

## **COMPARTILHAMENTO DE DADOS**

# O QUE DESCREVE UM MÉDICO?





# ENTIDADE MÉDICO

## MÉDICO

CRM_Médico	Nome_Médico	Especialidade_Médico
21114	Luís Paulo Carvalho	Pediatria
21113	Pedro Estevão Poct	Ginecologia
51024	Maurício Abreu	Neurologia
76004	Simone Almeida	Cardiologia

# O QUE DESCREVE UM PACIENTE?



# ENTIDADE PACIENTE

## PACIENTE

Nome_Paciente	Endereço	Sexo	Idade
Júlio Adamastor	R. Silva Sá, 23/11	Masc	33
Carmem Milhor	R. Dias Melhores 334/122	Fem	18
Sandra Chu Li	Av. Arapanés 4487/1915	Fem	44
Álvaro Medeiros Sá	R. Botica do Ouvidor 44/fundos	Masc	29
Paulo Alengui	Trav. das Camélias 661	Masc	38

# O QUE DESCREVE UMA CONSULTA MÉDICA?

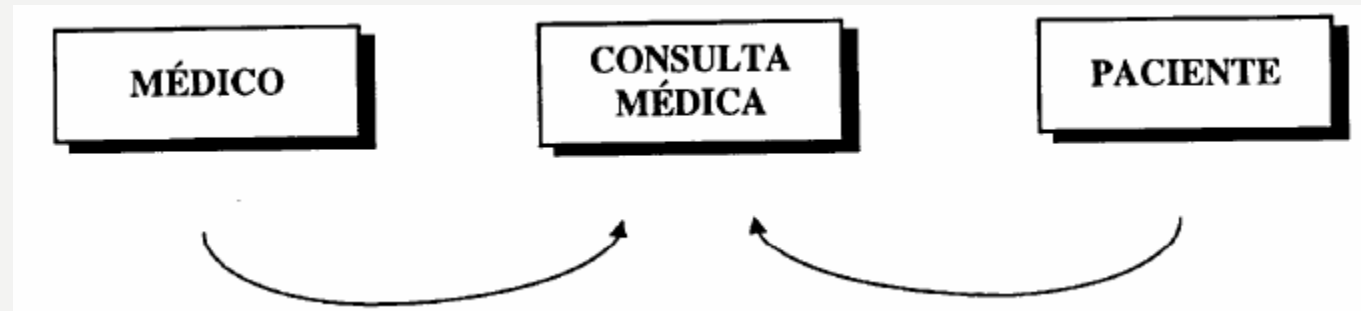


# ENTIDADE CONSULTA MÉDICA

## Consulta Médica

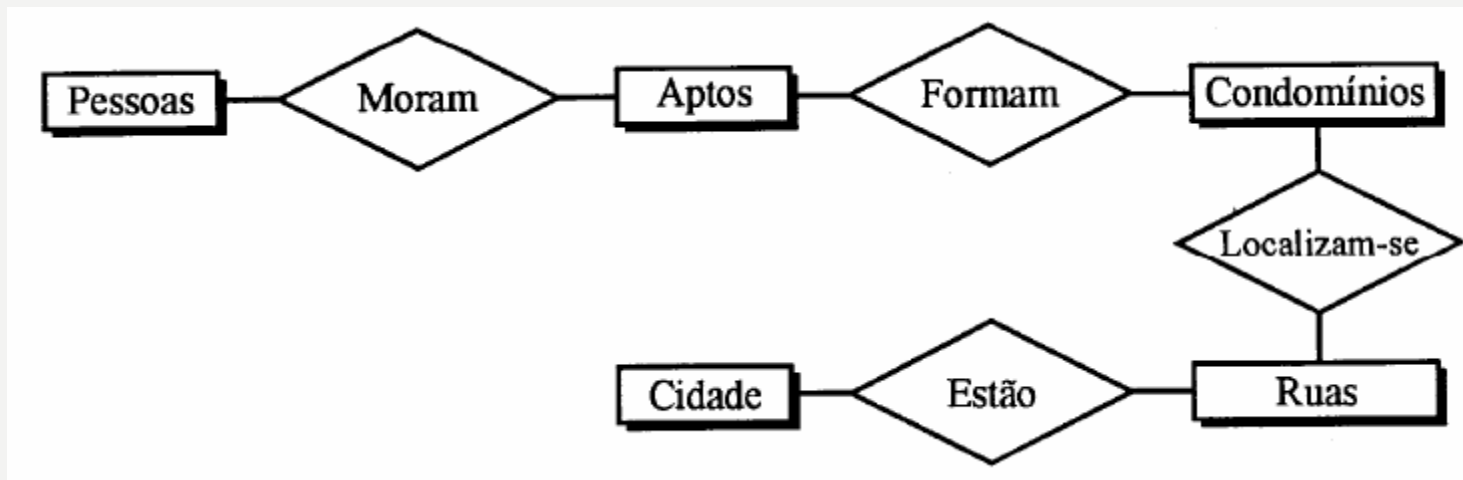
Data_da_Consulta	CRM_do_Médico	Identificação_Paciente
22/04/92	21113	João Pedro Lima
22/04/92	21113	Clara Mathias
21/03/91	14442	Luís Alberto Conde
31/03/92	55555	Maria Luiza Andrade

# ENTIDADE-RELACIONAMENTO

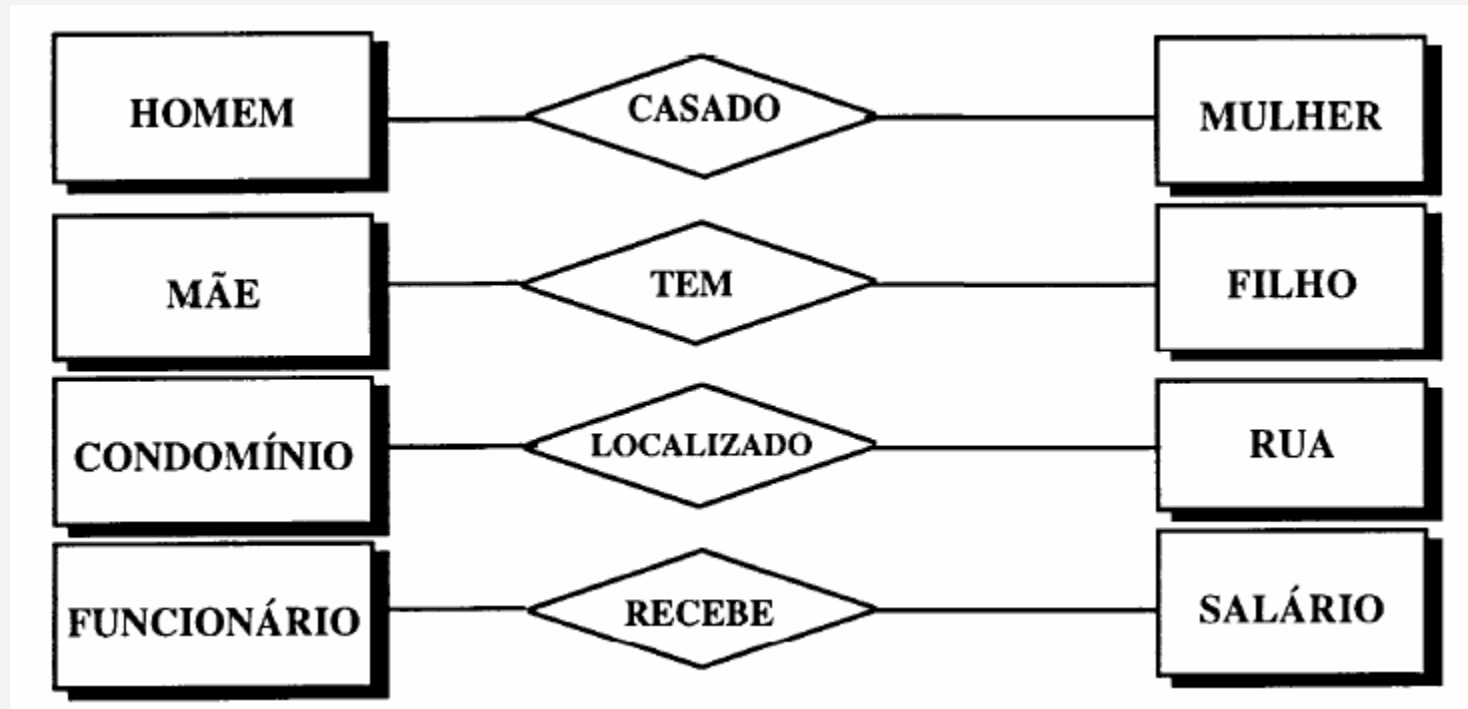


# NOMEANDO RELACIONAMENTOS

- Procure nomear relacionamentos utilizando verbos



# MAIS RELACIONAMENTOS





# RELACIONAMENTO ENTRE ENTIDADES

- Um conjunto de objetos classificados como pessoas (relacionamento PESSOA)
- Um conjunto de objetos classificados como departamentos (relacionamento DEPARTAMENTO)
- Um conjuntos de **associações**, que ligam um departamento a uma pessoa. (relacionamento LOTAÇÃO).

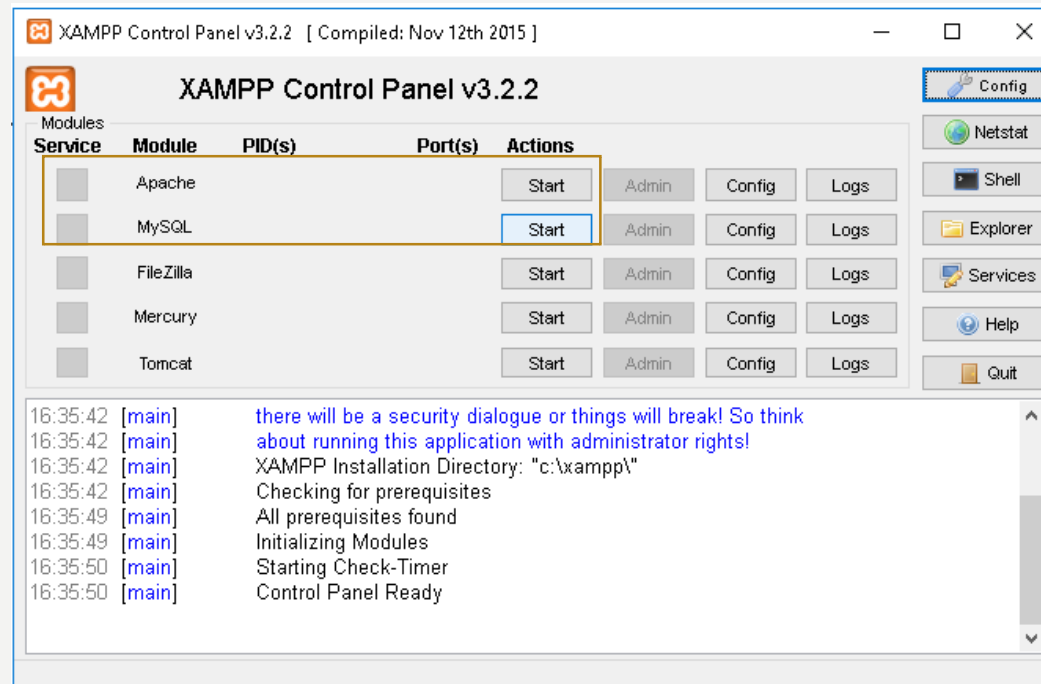


# INICIANDO O XAMPP E SGBD MYSQL

- Xampp é um pacote de programas que incluem

- Mysql
- Apache
- Php
- Phpmyadmin

<https://www.apachefriends.org/download.html>



# HEIDISQL

- É um programa que permite acessar o SGDB Mysql.
- Pode ser adquirido gratuitamente na internet.
- Não é necessário instalar.
  - Possui versão portátil.
- <http://www.heidisql.com/>



# O MYSQL

O MySQL é um SGBD, que utiliza a linguagem SQL como interface.

É atualmente um dos  
SGDBs mais  
populares do mundo

## Usuários

- NASA
- Banco Bradesco
- Nokia
- Sony
- U.S. Federal Reserve Bank,
- Cisco Systems
- Google e outros.



# O MYSQL

ATUALMENTE O MYSQL É  
PROPRIEDADE DA EMPRESA ORACLE

# TIPOS DE DADOS NUMÉRICOS

TinyInt	-128 até 127
Bool	0 ou 1
SmallInt	-32768 até 32767
MediumInt	-8.388.608 até 8.388.607
INTEGER	-2147483648 até 2147483647
BIGINT	-9223372036854775808 até 9223372036854775807
Float	3.402823466E+38 até -1.175494351E-38
Double	-1.7976931348623157E+308 até -2.2250738585072014E-308

# TIPOS DE DADOS NUMÉRICOS

TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT	4 bytes
INTEGER	4 bytes
BIGINT	8 bytes
FLOAT	4 bytes
DOUBLE	8 bytes

## TIPOS ALFANUMÉRICOS

Tipo	Tamanho
Char(n)	n
VarChar(n)	n+1
TINYTEXT	255
Text	65535
MediumText	16.777.215
LongText	4.294.967.295



# CHAR AND VARCHAR TYPES

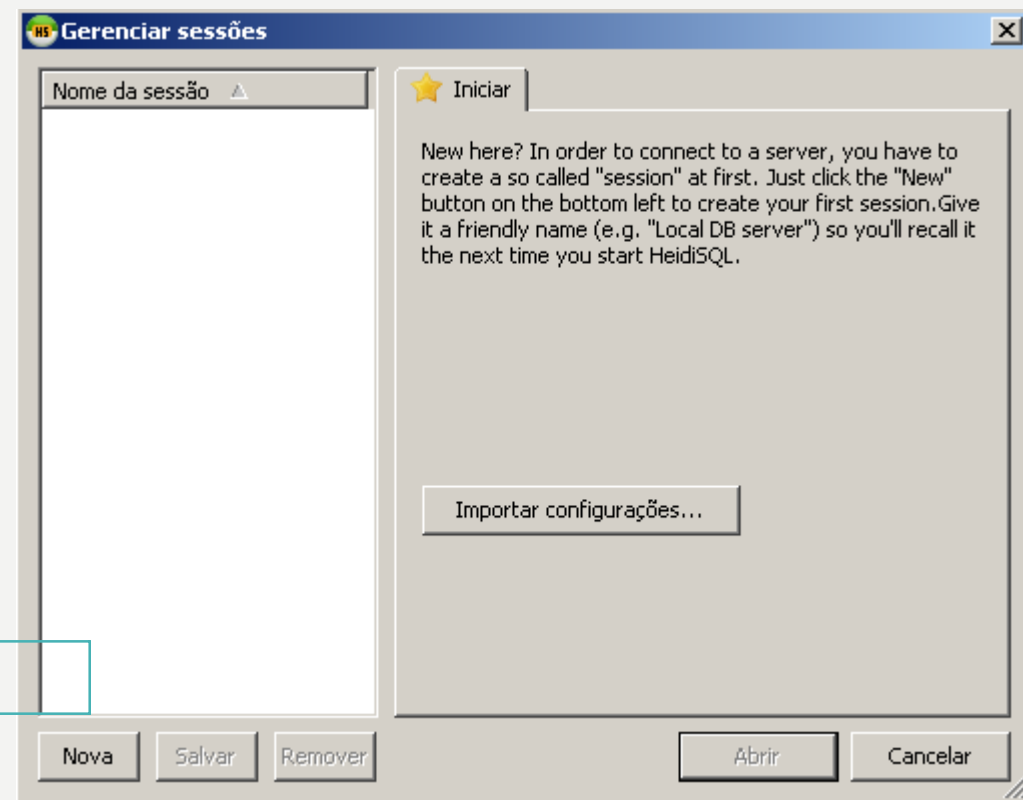
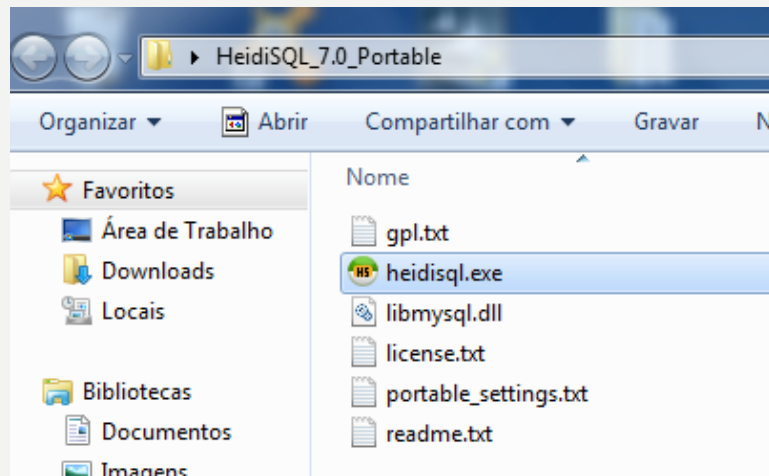
Value	CHAR (4)	Storage Required	VARCHAR (4)	Storage Required
' '	'   '	4 bytes	' '	1 byte
'ab'	'ab  '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

# SQL - STRUCTURED QUERY LANGUAGE

- SQL é padronizado para a definição e manipulação de bancos de dados relacionais
- Exemplos de SGBD que utilizam SQL
  - Oracle
  - **MySQL**
  - PostgreSQL
  - SQL Server
  - Interbase

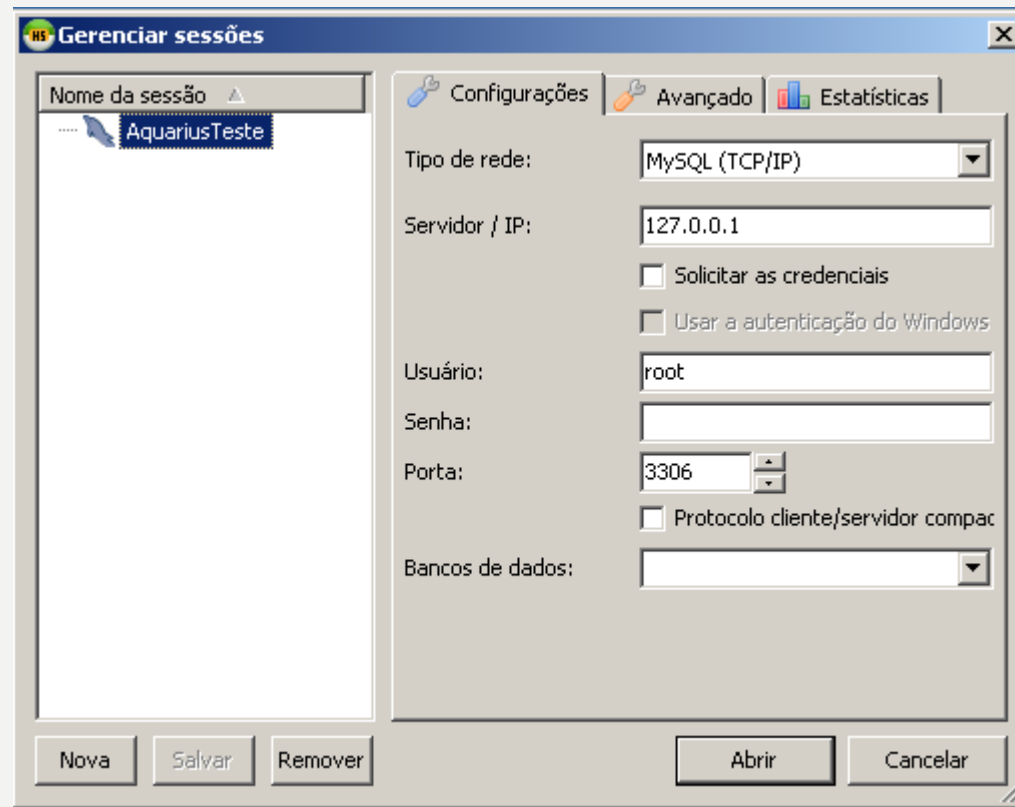
# INICIANDO O PROGRAMA E CRIANDO UMA NOVA SESSÃO

- A sessão cria uma conexão com o servidor mysql



# CRIANDO UMA NOVA SESSÃO

- Clique onde esta escrito “AquariusTeste” e escolha um nome para sessão.
- Como o servidor mysql está rodando no localhost é utilizado o ip “127.0.0.1”.
- **Usuário:** root e não possui senha
- Clique em abrir



# CRIAR UM NOVO BANCO DE DADOS

AquariusTeste - HeidiSQL 7.0.0.4331

Arquivo Editar Pesquisar Ferramentas Ajuda

AquariusTeste 693,9 KB

Servidor: 127.0.0.1 Consulta

Bancos de dados (4) Variáveis Status Processos Estatísticas

Banco de dados	Tamanho	Itens	Últim...	Tabelas	Views	Fu
information_schema	9,0 KB	37	2013...	37	0	0
mysql	654,7 KB	24	2012...	24	0	0
performance_schema	0 B	17		17	0	0
phpmyadmin	30,2 KB	12	2012...	12	0	0

- Editar Alt+Enter
- Excluir...
- Esvaziar tabela(s)...
- Executar rotina(s)...
- Criar novo**
  - Banco de dados**
  - Tabela
  - Copiar tabela
  - View
  - Stored Routine
  - Trigger
  - Evento
- Exportar banco de dados como SQL
- Manutenção
- Encontrar texto no servidor Shift+Ctrl+F
- Editor de tabela em massa
- Expandir todos
- Recolher todos
- Tree style options
- Imprimir... Ctrl+P
- Atualizar F5
- Desconectar

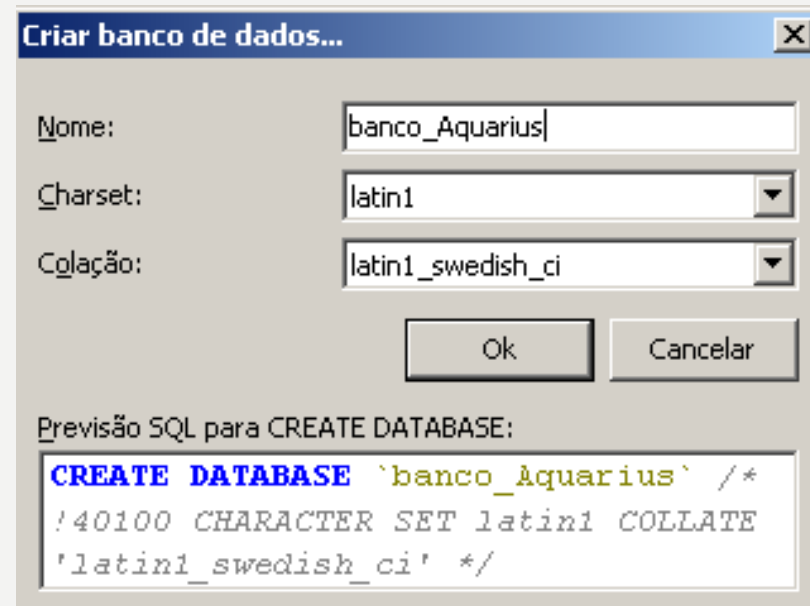
Botão direito do mouse

```
42 SHOW FUNCTION STATUS WHERE `Db`='phpmyadmin';
43 SHOW PROCEDURE STATUS WHERE `Db`='phpmyadmin';
44 SHOW TRIGGERS FROM `phpmyadmin`;
45 SHOW EVENTS FROM `phpmyadmin`;
```

Connected: 00:02 h MySQL 5.5.25 Ativo durante: 00:06 h Idle.

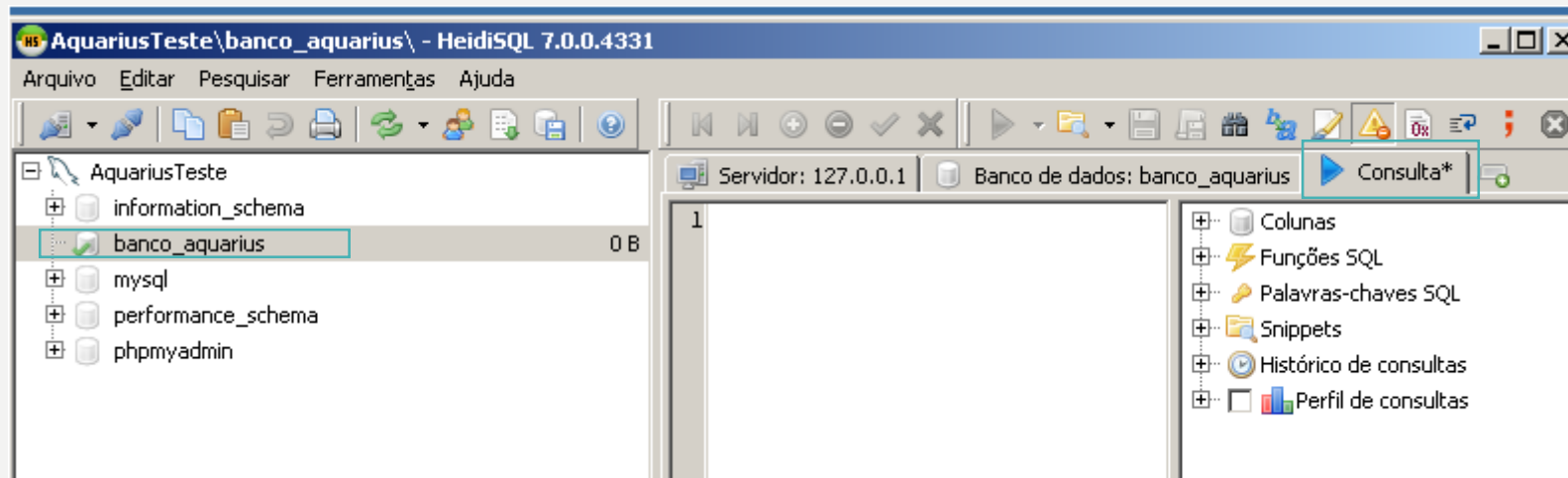
# CRIAR UM NOVO BANCO DE DADOS

- Defina um nome para a base de dados
- Clique em ok



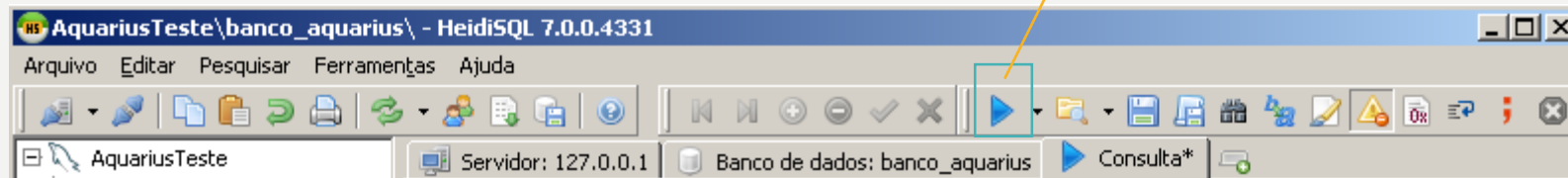
# O AMBIENTE DE DESENVOLVIMENTO

- Pressione f5 para atualizar e marque o banco de dados criado “banco\_aquarius”.
- Clique em “consulta” ou “query”, a área em branco é reservada para inserir código SQL.



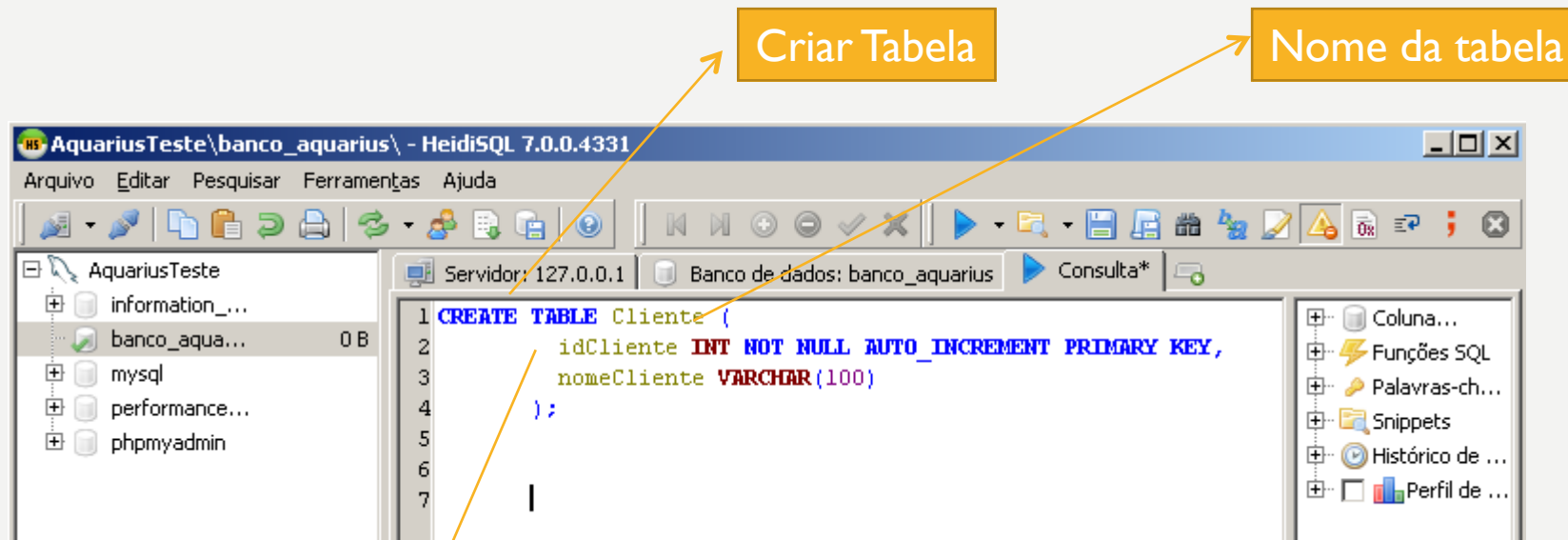
# RODANDO O CÓDIGO SQL

Roda o código





# CRIANDO TABELAS



Campos e tipos separados por vírgula

# CRIANDO UMA TABELA

```
CREATE TABLE `cliente` (  
  `idCliente` INT(10) NOT NULL AUTO INCREMENT PRIMARY KEY ,  
  `Nome` VARCHAR(50) NULL DEFAULT NULL,  
  `Sobrenome` VARCHAR(80) NULL DEFAULT NULL,  
  `idade` TINYINT NULL DEFAULT NULL  
)
```

Não permite nulos

Chave primária

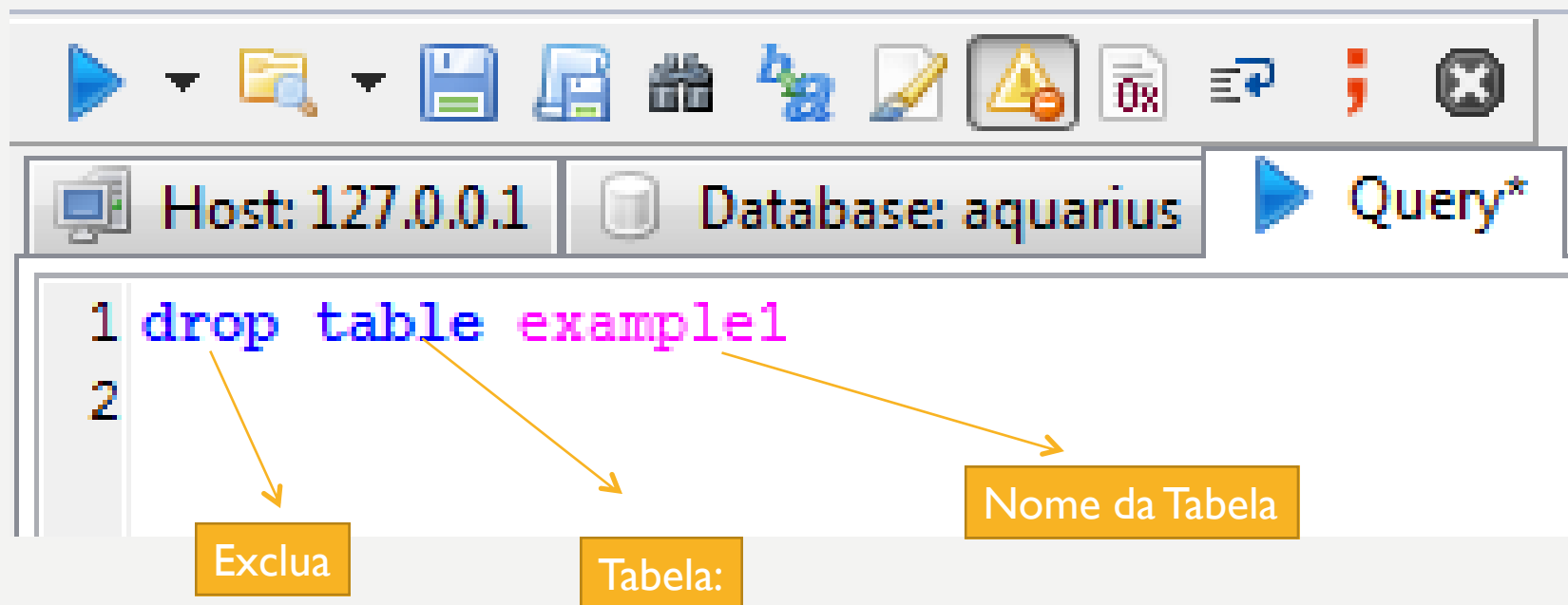
Auto Incremento

Valor padrão igual a nulo

Permite que o campo seja nulo

# EXCLUINDO TABELAS

- Cuidado esta operação não pode ser desfeita.
- O banco de dados não pede confirmação para excluir a tabela.
- Todos os registros da tabela são excluídos também.



# INSERINDO DADOS EM UMA DETERMINADA TABELA

Insira em:

Nome da tabela

Campos separados por virgula

Valores que serão inseridos

```
insert into cliente (nomeCliente,e_mail) values  
("Helio Esperidião","helio@univap.br");
```

Valores que serão inseridos separados por virgula  
Em campos do tipo texto utiliza-se aspas.

Field	Type
idCliente	int(11)
nomeCliente	varchar(100)
e_mail	varchar(100)

```
insert into cliente (nomeCliente,e_mail) values  
("Maria paula","magia@gmail.com");
```

```
insert into cliente (nomeCliente,e_mail) values  
("Andreia Silva","Asilva@gmail.com");
```

# INSERINDO REGISTROS

```
Insert into cliente (nome,sobrenome,idade) values ("hélío","Esperidião",18);  
Insert into cliente (nome,sobrenome,idade) values ("Antonio","Silva",22);  
Insert into cliente (nome,sobrenome,idade) values ("Marina","Duarte",25);  
Insert into cliente (nome,sobrenome,idade) values ("João","Ferreira",57);  
Insert into cliente (nome,sobrenome,idade) values ("João","Ferreira",28);  
Insert into cliente (nome,sobrenome,idade) values ("Anderson","Barata",24);  
Insert into cliente (nome,sobrenome,idade) values ("Adrielle","Andrade",23);  
Insert into cliente (nome,sobrenome,idade) values ("Adrielle","Santana",23);  
Insert into cliente (nome,sobrenome,idade) values ("Andressa","Ferrari",19);  
Insert into cliente (nome,sobrenome,idade) values ("Adrielle","Andrade",33);  
Insert into cliente (nome,sobrenome,idade) values ("Paloma","Duarte",23);
```

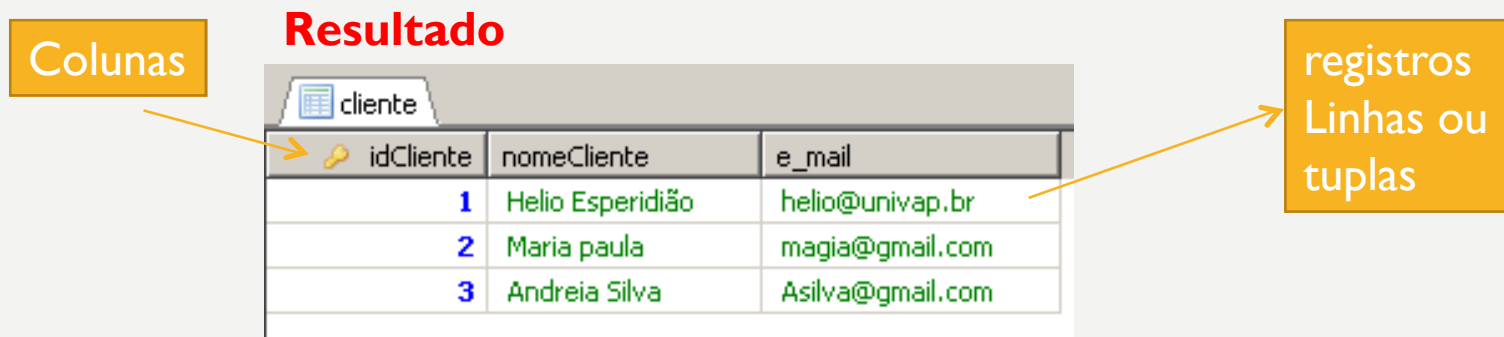
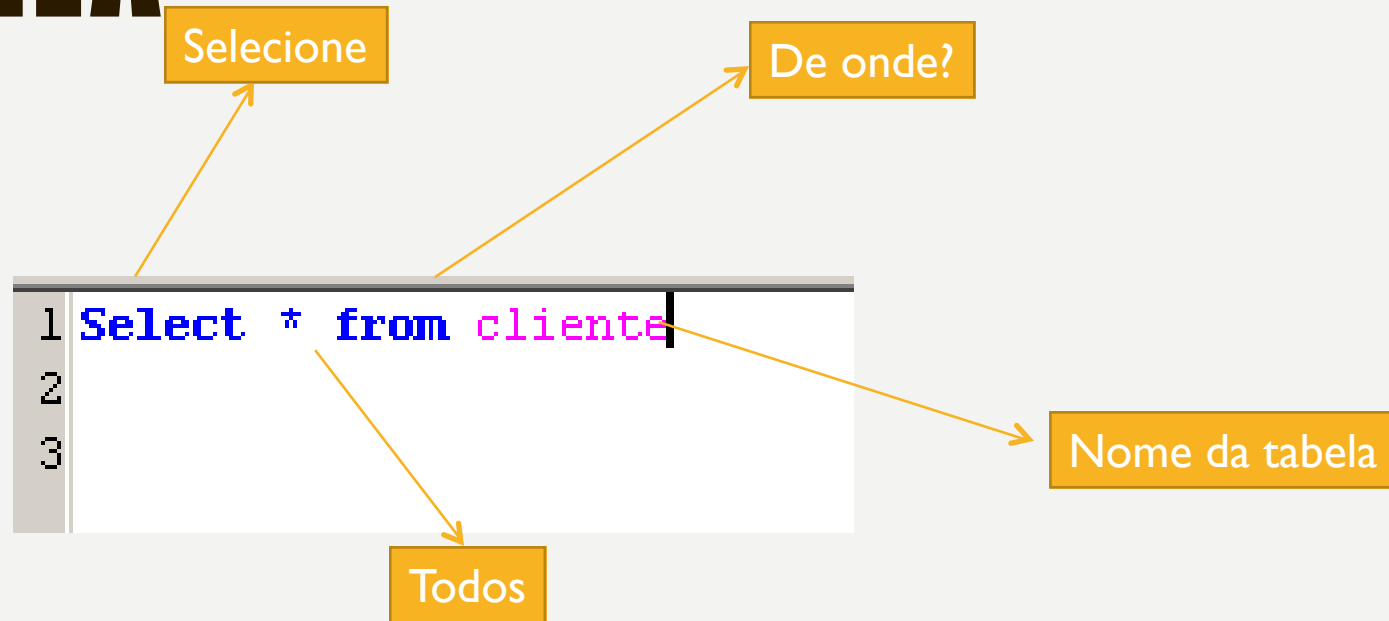
Inserir em:

Nome da Tabela

Campos separados por virgula

Valores separados por virgula;  
Utilize a mesma seqüência dos campos

# SELECIONANDO DADOS EM UMA TABELA



# SELECIONANDO DADOS EM UMA TABELA

Nome dos campos separados por virgula

```
select nomeCliente from cliente
```

**Resultado**

cliente
nomeCliente
Helio Esperidião
Maria paula
Andreia Silva

# SELECIONANDO REGISTROS CAMPOS ESPECÍFICOS

```
select nome, idade from cliente;
```

Selecione

Campos escolhidos  
Separados por  
virgula

da tabela

Nome da tabela

Resultado:

cliente	
nome	idade
hélio	18
Antonio	22
Marina	25
João	57
João	28
Anderson	24
Adrielle	23
Adrielle	23
Andressa	19
Adrielle	33
Paloma	23



# SELECIONANDO DADOS EM UMA TABELA

```
select * from cliente where nomeCliente="Helio Esperidiao";
```

Onde?

Condição lógica

## Resultado

cliente		
idCliente	nomeCliente	e_mail
1	Helio Esperidião	helio@univap.br

# SELECIONANDO DADOS EM UMA TABELA

```
select * from cliente where idCliente>1
```

Onde?

Condição lógica

## Sintaxe de comparação

>	Maior
<	Menor
>=	Maior ou igual
<=	Menor ou igual
<>	Diferente

## Resultado

cliente		
idCliente	nomeCliente	e_mail
2	Maria paula	magia@gmail.com
3	Andreia Silva	Asilva@gmail.com

# SELECIONANDO DADOS EM UMA TABELA CONCATENANDO INSTRUÇÕES

```
select * from cliente where  
nomeCliente ="Helio esperidiao" and  
e_mail = "helio@univap.br";
```

## instrução

And

e lógico

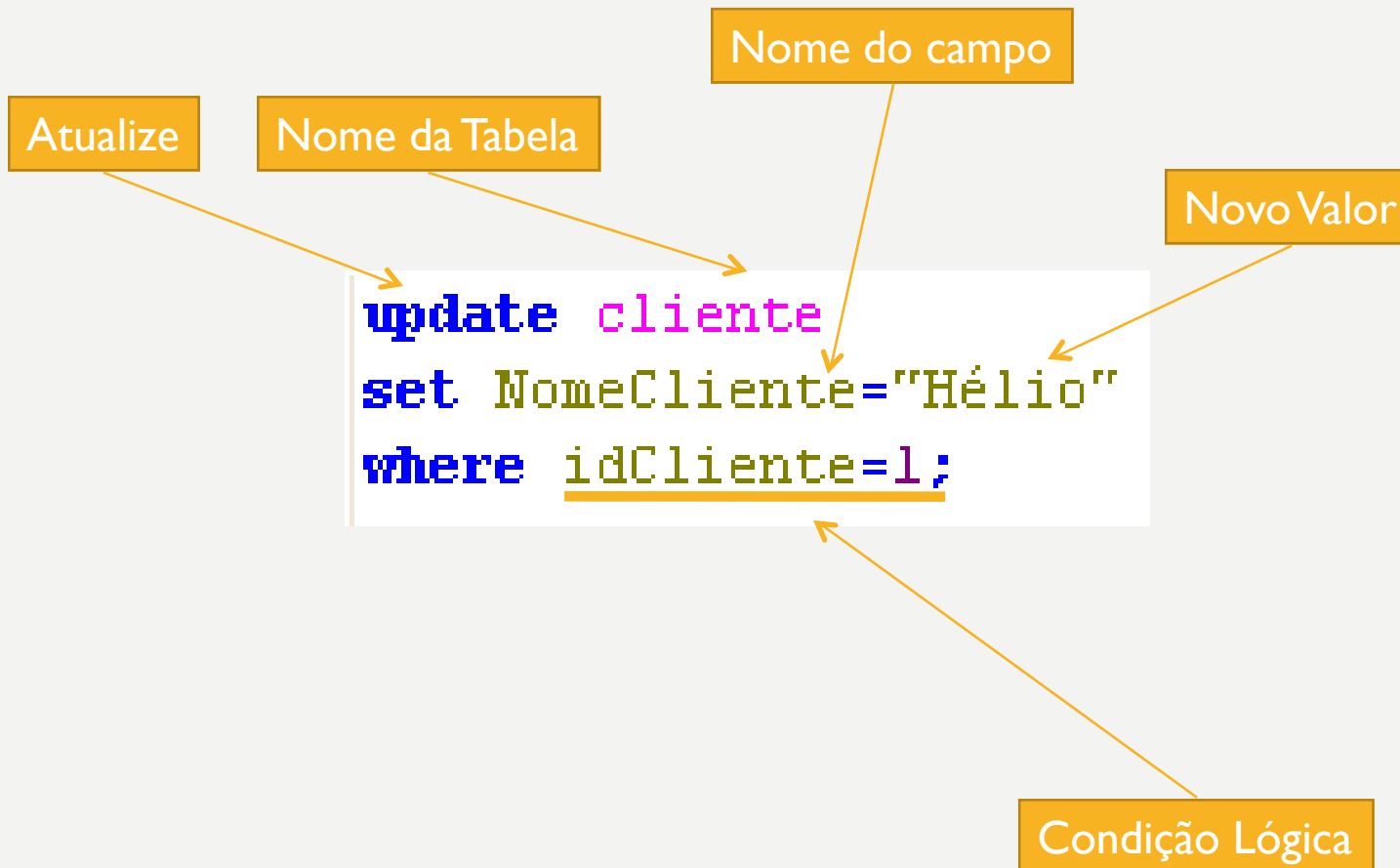
Or

Ou lógico

## Resultado

cliente		
idCliente	nomeCliente	e_mail
1	Helio Esperidião	helio@univap.br

# ATUALIZANDO REGISTROS



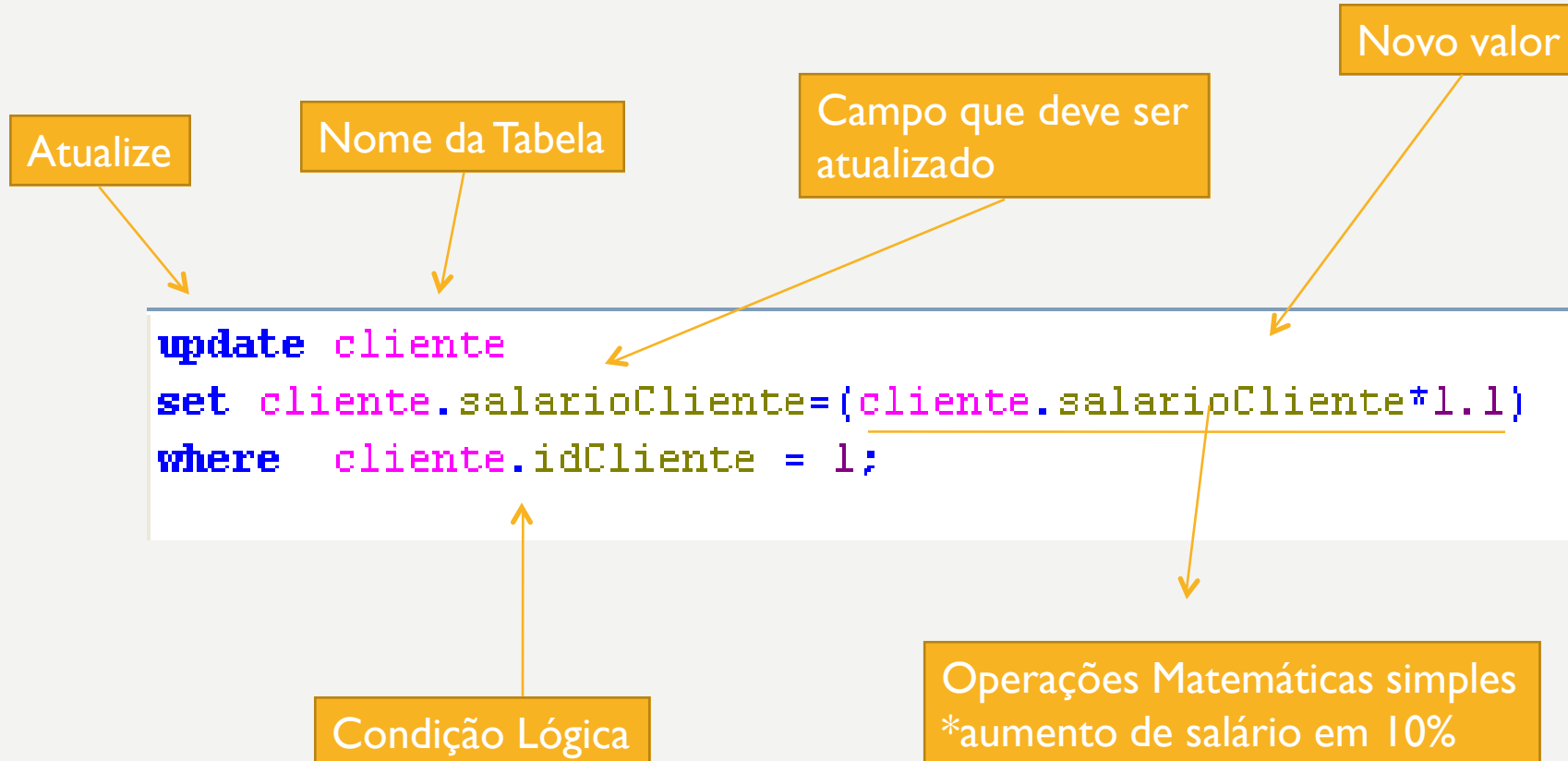
# ATUALIZANDO REGISTROS

Atualizações separadas por virgula

```
update cliente
set NomeCliente="Hélio",
sobreNomeCliente="Esperidião"
where NomeCliente="Héli" and sobreNomeCliente="Experidião"
and idCliente = 1;
```

Condições lógicas

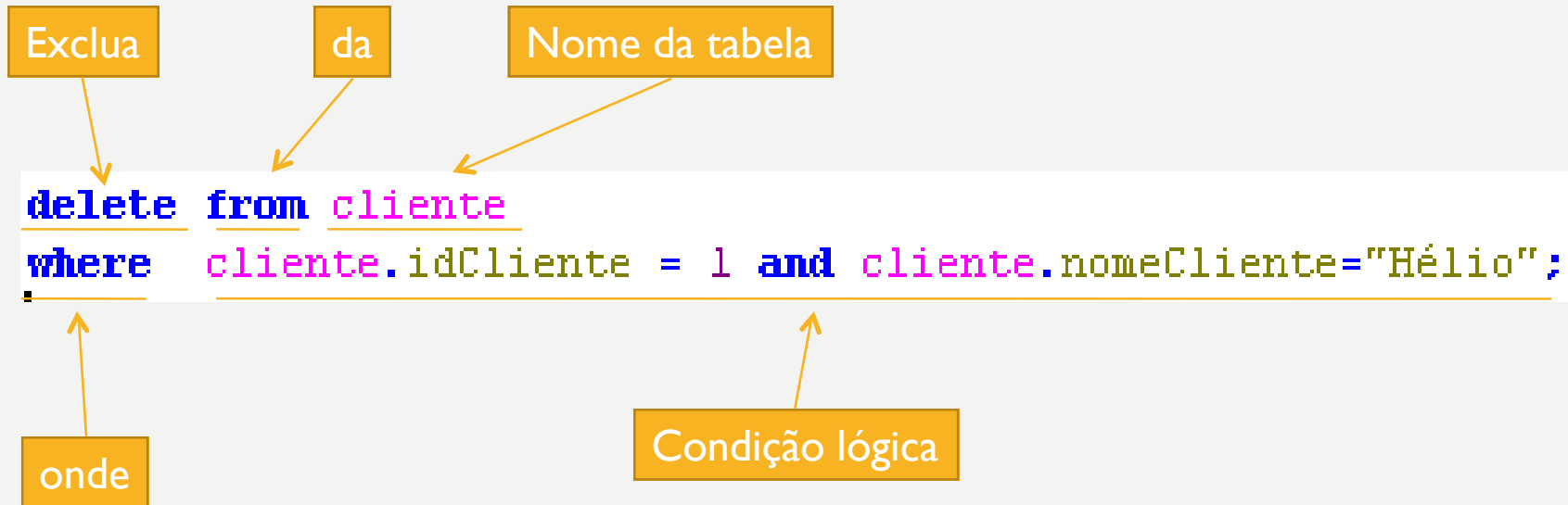
# ATUALIZANDO REGISTROS



## Cuidado:

Caso Não defina uma condição lógica  
• todos os registros serão atualizados

# EXCLUINDO REGISTROS



## Cuidado:

Caso Não defina uma condição lógica  
• todos os registros serão excluídos

# EXCLUINDO REGISTROS

Prioridade de exclusão.

[LOW\_PRIORITY] [QUICK] [IGNORE]

```
delete QUICK from cliente  
where cliente.idCliente = 1 and cliente.nomeCliente="Hélio";
```



# IMPORTAÇÕES

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;
```

# ATRIBUTOS

```
private Connection conexao;
```

```
private final String URLDB = "jdbc:mysql://localhost:3306/agenda";
```

```
private final String usuario = "root";
```

```
private final String senha = "";
```

# CONSTRUTOR

```
public ConexaoBancoDeDados() {  
    try {  
        Class.forName("com.mysql.jdbc.Driver");  
        conexao = DriverManager.getConnection(URLDB, usuario,  
senha);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

# MÉTODO INSERIR

```
public void inserir() {  
    Statement inserir;  
  
    try {  
        inserir = conexao.createStatement();  
        inserir.execute("INSERT INTO estado(idEstado,estadoNome) VALUES('2','Rio de janeiro')");  
    } catch (SQLException ex) {  
  
    }  
}
```

# MÉTODO UPDATE()

```
public void update() {  
    Statement comando;  
  
    try {  
        comando = conexao.createStatement();  
        comando.executeUpdate("UPDATE estado SET estadoNome ='Rio de Janeiro' WHERE idEstado = 2");  
    } catch (SQLException ex) {  
  
    }  
}
```

# MÉTODO DELETE

```
public void delete() {  
    Statement comando;  
  
    try {  
        comando = conexao.createStatement();  
        comando.execute("delete from estado WHERE idEstado = 3");  
    } catch (SQLException ex) {  
  
    }  
}
```

# MÉTODO SELECIONAR

```
public void selecionar() {  
    try {  
        Statement comando;  
        comando = conexao.createStatement();  
        ResultSet res = comando.executeQuery("SELECT * FROM estado");  
  
        while (res.next()) {  
            System.out.println(res.getString("estadoNome"));  
        }  
    } catch (SQLException ex) {  
  
    }  
}
```

# MÉTODO PREPARESELECT

```
public void prepareSelect(){
    try {
        PreparedStatement psSelect= this.conexao.prepareStatement("Select * from estado where
idEstado=?");
        psSelect.setString(1, "1");
        ResultSet res = psSelect.executeQuery();
        while (res.next()) {
            System.out.println(res.getString("estadoNome"));
        }

    } catch (SQLException ex) {

    }

}
```