

# **CRIANDO, ALTERADO E EXCLUINDO TABELAS, INSERINDO, SELECIONANDO, ALTERANDO E EXCLUINDO DADOS**

**Prof. Me. Hélio Esperidião**

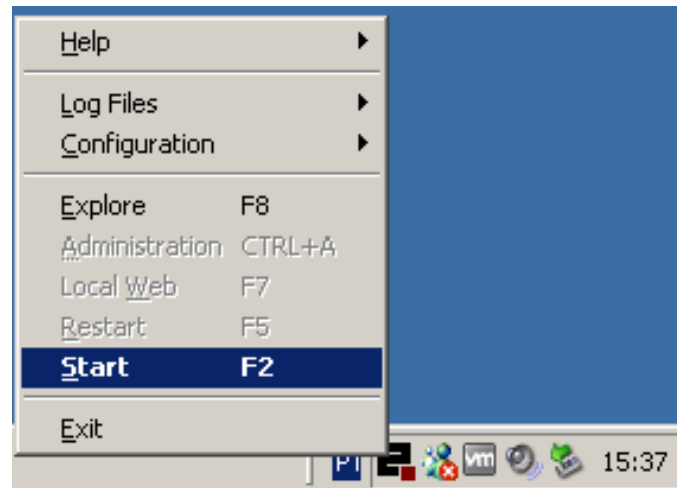
# SQL - STRUCTURED QUERY LANGUAGE

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



# INICIANDO O EASYPHP E SGBD MYSQL

- É um pacote de programas que incluem
  - Mysql
  - Apache
  - Php
  - Phpmyadmin



HEIDISQL

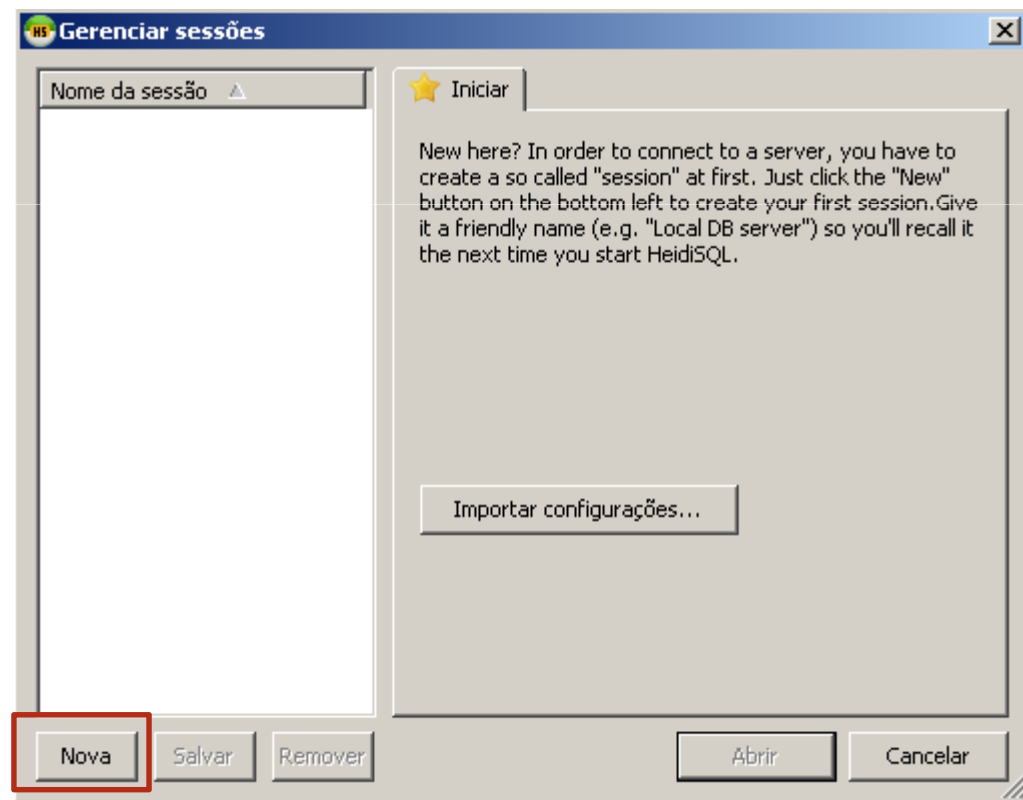
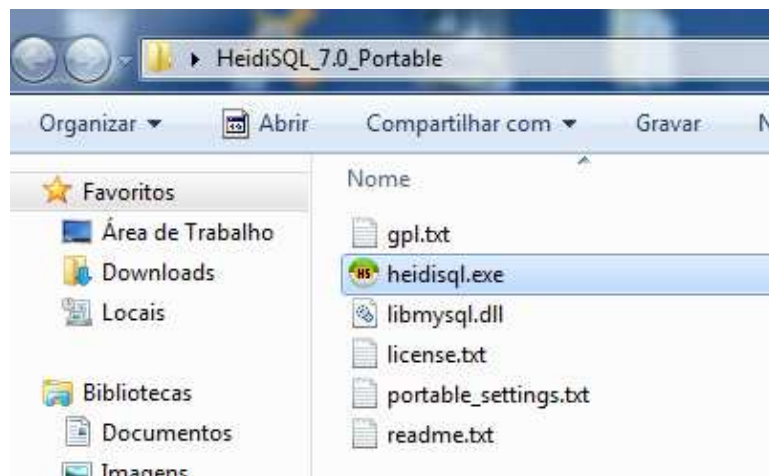


- É um programa que permite acessar o SGDB Mysql.
- Pode ser adquirido gratuitamente na internet.
- Não é necessário instalar.
  - Possui versão portátil.
- <http://www.heidisql.com/>



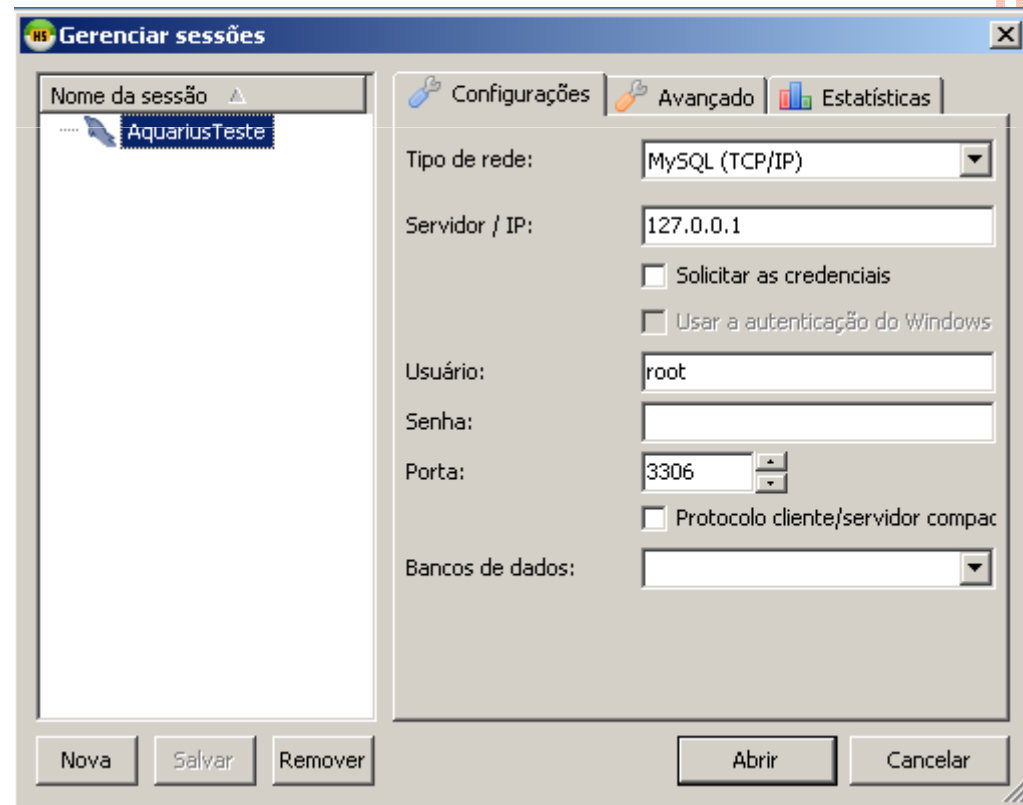
# INICIANDO O PROGRAMA E CRIANDO UMA NOVA SESSÃO

- 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

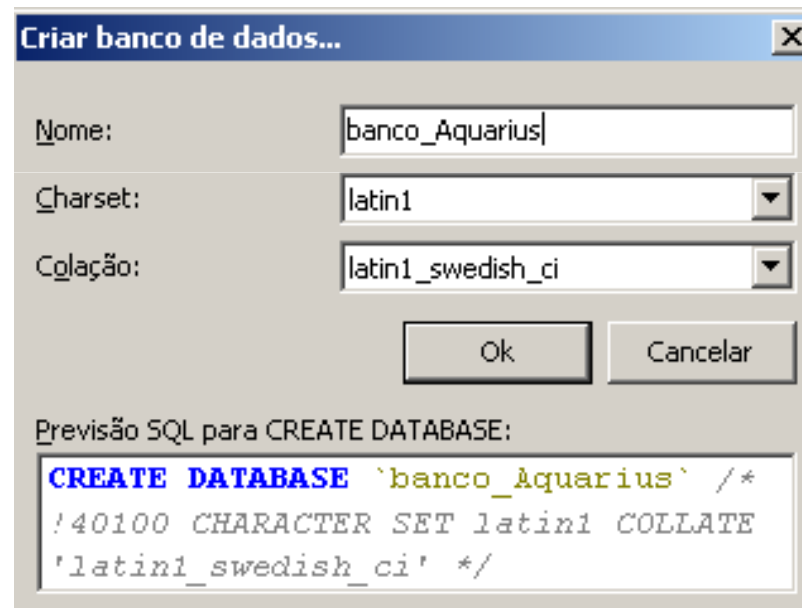
The screenshot shows the HeidiSQL interface with a context menu open over the 'Bancos de dados (4)' table. The menu is triggered by a right-click, as indicated by an orange arrow and a callout box labeled 'Botão direito do mouse'. The 'Criar novo' option is selected, and its sub-menu is visible, with 'Banco de dados' highlighted. The table below shows the existing databases on the server.

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

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

# CRIAR UM NOVO BANCO DE DADOS

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



**Criar banco de dados...**

Nome:

Charset:

Colação:

Previsão SQL para CREATE DATABASE:

```
CREATE DATABASE `banco_Aquarius` /*  
!40100 CHARACTER SET latin1 COLLATE  
'latin1_swedish_ci' */
```



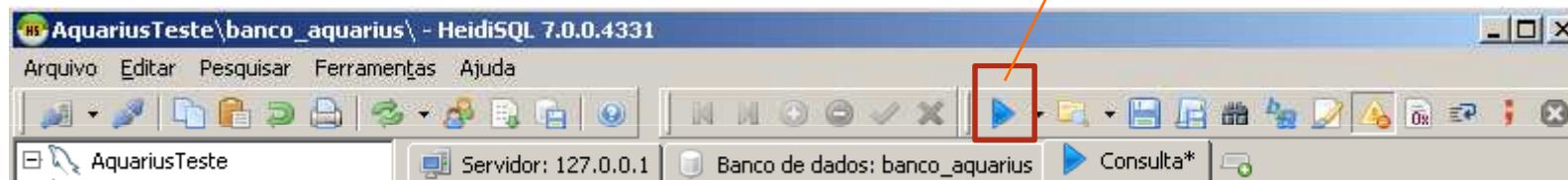
## ○ AMBIENTE DE DESENVOLVIMENTO

- 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

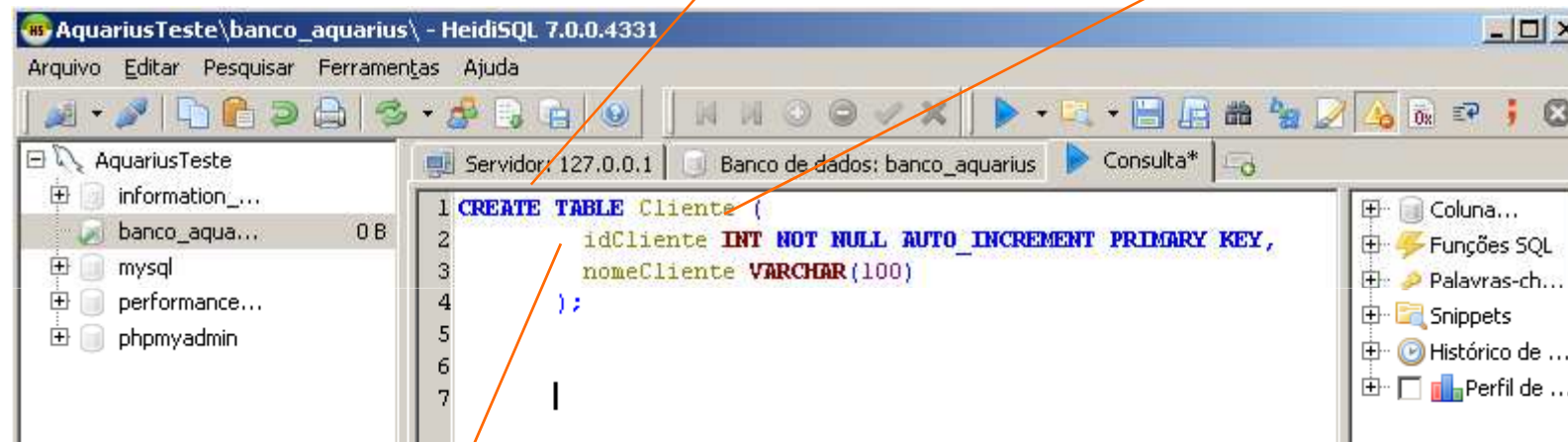
Roda o código



# CRIANDO TABELAS

Criar Tabela

Nome da tabela



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

# ALTERANDO TABELAS

## ADD

- O comando **add** permite adicionar campos a uma tabela criada anteriormente

The screenshot shows the HeidiSQL interface with the following SQL command in the editor:

```
1 ALTER TABLE Cliente
2 ADD email VARCHAR(100) AFTER nomeCliente;
3
4
```

Callouts from the image explain the parts of the command:

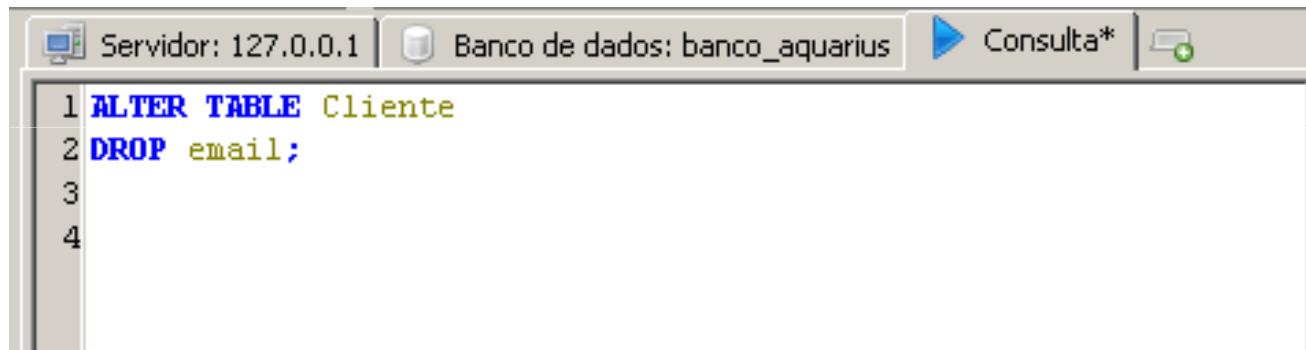
- Adicione**: Points to the `ADD` keyword.
- Altere a tabela**: Points to the `ALTER TABLE` keywords.
- Nome da tabela**: Points to the `Cliente` table name.
- Nome do campo**: Points to the `email` field name.
- Tipo do campo**: Points to the `VARCHAR(100)` data type.
- Depois de:**: Points to the `AFTER` keyword.
- Nome do campo**: Points to the `nomeCliente` existing field name.

A solid orange circle is located in the bottom right corner of the slide.

# ALTERANDO TABELAS

## DROP

- O comando **drop** permite excluir campos de uma tabela



Servidor: 127.0.0.1 | Banco de dados: banco\_aquarius | Consulta\*

```
1 ALTER TABLE Cliente
2 DROP email;
3
4
```



# ALTERANDO TABELAS

## CHANGE

- O comando **change** permite alterar nomes e tipos de dados de um determinado atributo.

```
1 ALTER TABLE cliente
2 CHANGE email e_mail varchar(100);
3
4 |
```



## O COMANDO **SHOW**

- Lista todas as tabelas presentes em um banco de dados

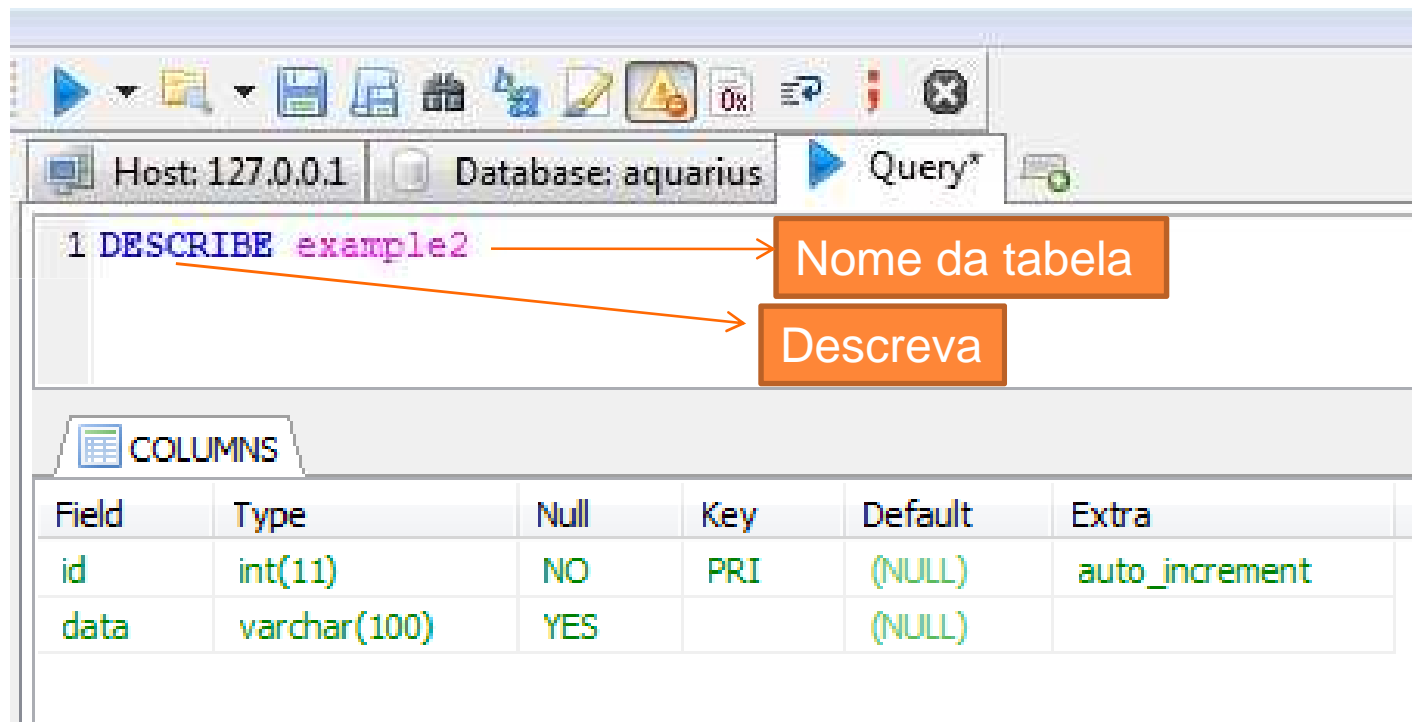
```
1 SHOW TABLES
```

TABLE_NAMES
Tables_in_aquarius
example1
example2
example_increment



# O COMANDO **DESCRIBE**

- Descreve a estrutura de uma tabela.



The screenshot shows a MySQL query editor window. The top toolbar contains various icons for execution and editing. Below the toolbar, the connection information is displayed: Host: 127.0.0.1, Database: aquarius, and Query\*. The query editor contains the following SQL command:

```
1 DESCRIBE example2
```

Two orange arrows point from the command to labels: one from 'example2' to 'Nome da tabela' and another from 'DESCRIBE' to 'Descreva'.

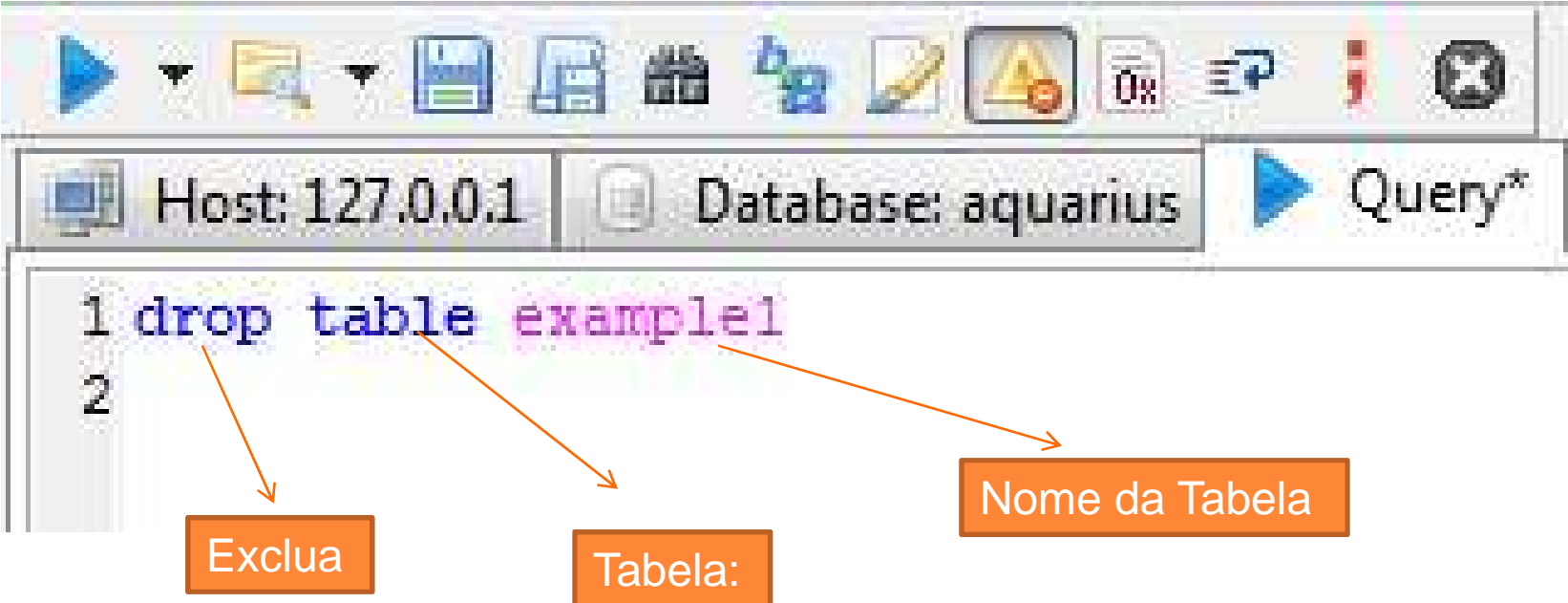
Below the query editor, the 'COLUMNS' tab is active, displaying the following table structure:

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	(NULL)	auto_increment
data	varchar(100)	YES		(NULL)	



## 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.



The image shows a screenshot of a database query editor interface. The toolbar at the top contains various icons for navigation and execution. Below the toolbar, the connection information is displayed: Host: 127.0.0.1 and Database: aquarius. The main area shows a query window with the following SQL command:

```
1 drop table example1
2
```

Annotations with arrows point to the components of the command:

- An arrow points from the word "drop" to a box labeled "Exclua".
- An arrow points from the word "table" to a box labeled "Tabela:".
- An arrow points from the word "example1" to a box labeled "Nome da Tabela".

A solid orange circle is located in the bottom right corner of the slide.

# INSERINDO DADOS EM UMA DETERMINADA TABELA

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

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.

```
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 ("h elio","Esperidi o",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 ("Jo o","Ferreira",57);
Insert into cliente (nome,sobrenome,idade) values ("Jo o","Ferreira",28);
Insert into cliente (nome,sobrenome,idade) values ("Anderson","Barata",24);
Insert into cliente (nome,sobrenome,idade) values ("Adriele","Andrade",23);
Insert into cliente (nome,sobrenome,idade) values ("Adriele","Santana",23);
Insert into cliente (nome,sobrenome,idade) values ("Andressa","Ferrari",19);
Insert into cliente (nome,sobrenome,idade) values ("Adriele","Andrade",33);
Insert into cliente (nome,sobrenome,idade) values ("Paloma","Duarte",23);
```

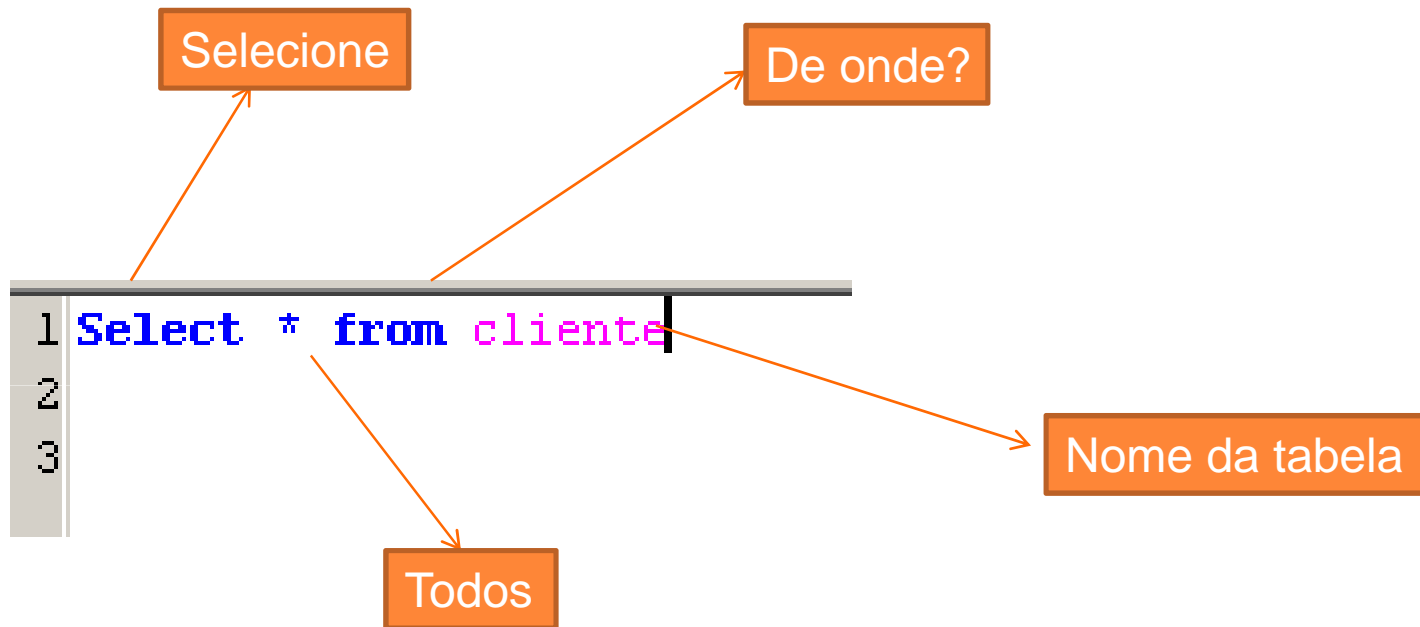
Inserir em:

Nome da Tabela

Campos separados por virgula

Valores separados por virgula;  
Utilize a mesma seq encia dos campos

# SELECIONANDO DADOS EM UMA TABELA




Colunas

**Resultado**

cliente		
idCliente	nomeCliente	e_mail
1	Helio Esperidião	helio@univap.br
2	Maria paula	magia@gmail.com
3	Andreia Silva	Asilva@gmail.com

registros  
Linhas ou  
tuplas



# 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

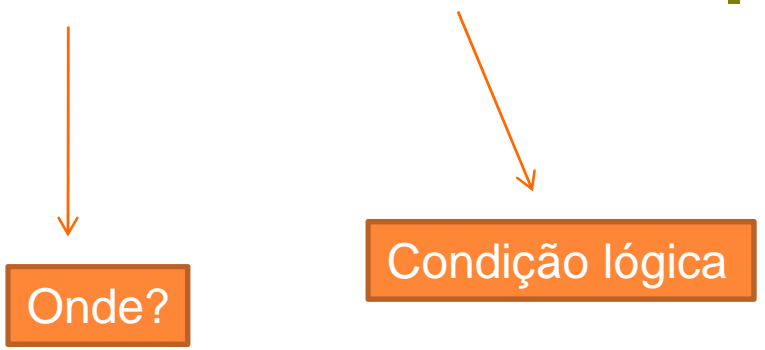
Nome da tabela

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";
```



## 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";
```



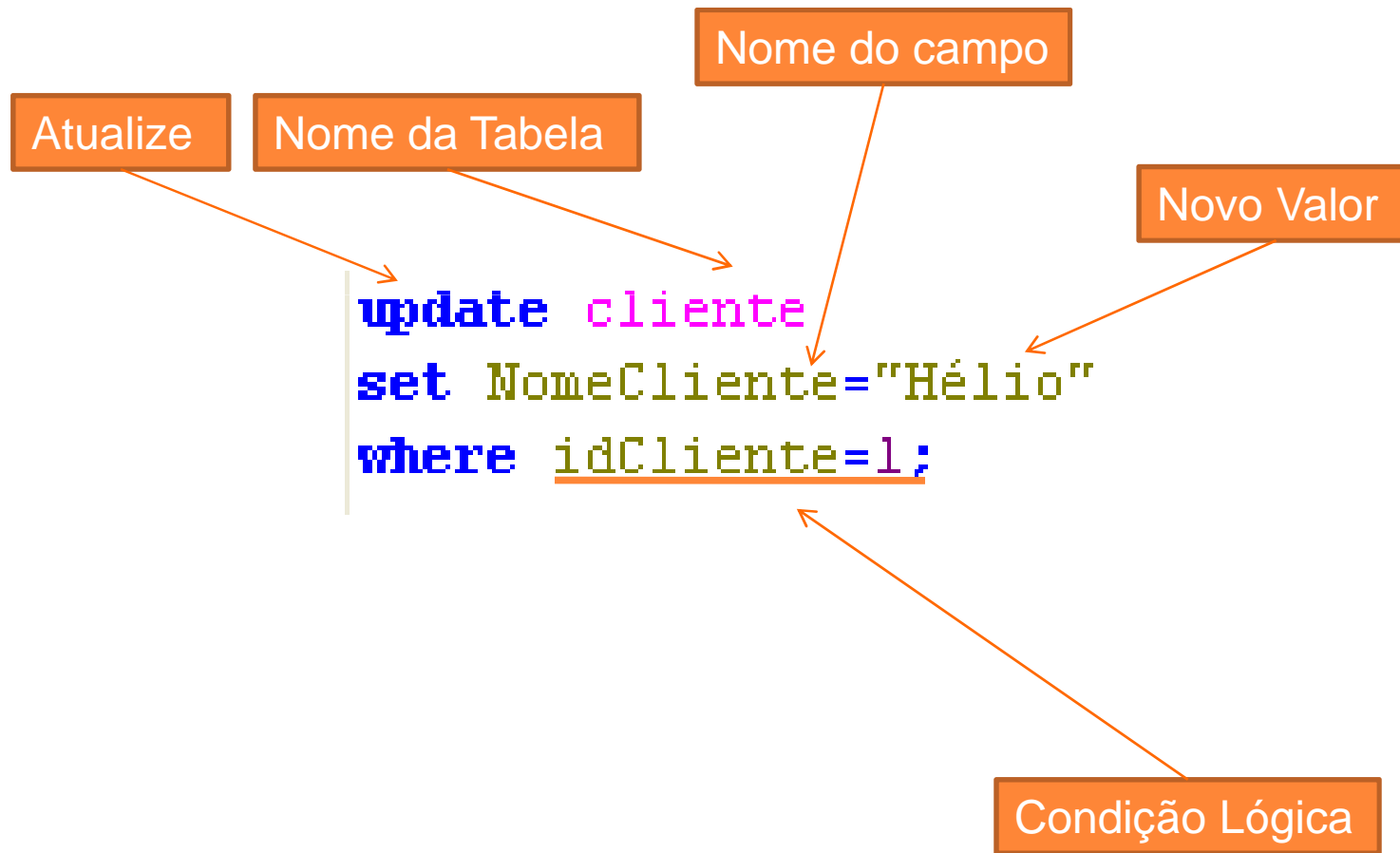
instrução	
And	e lógico
Or	Ou lógico

## Resultado

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



# ATUALIZANDO REGISTROS



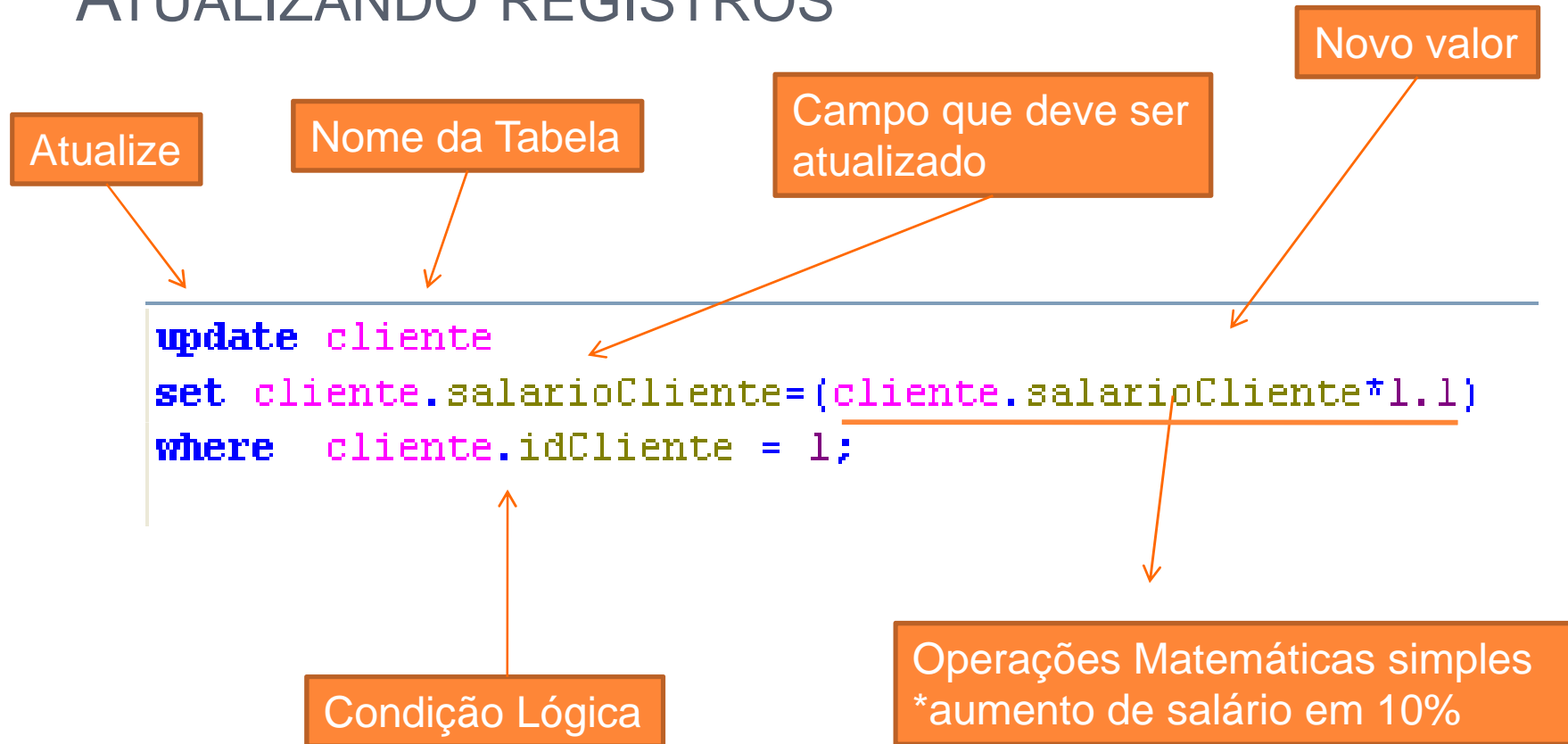
# ATUALIZANDO REGISTROS

Atualizações separadas por virgula

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

Condições lógicas

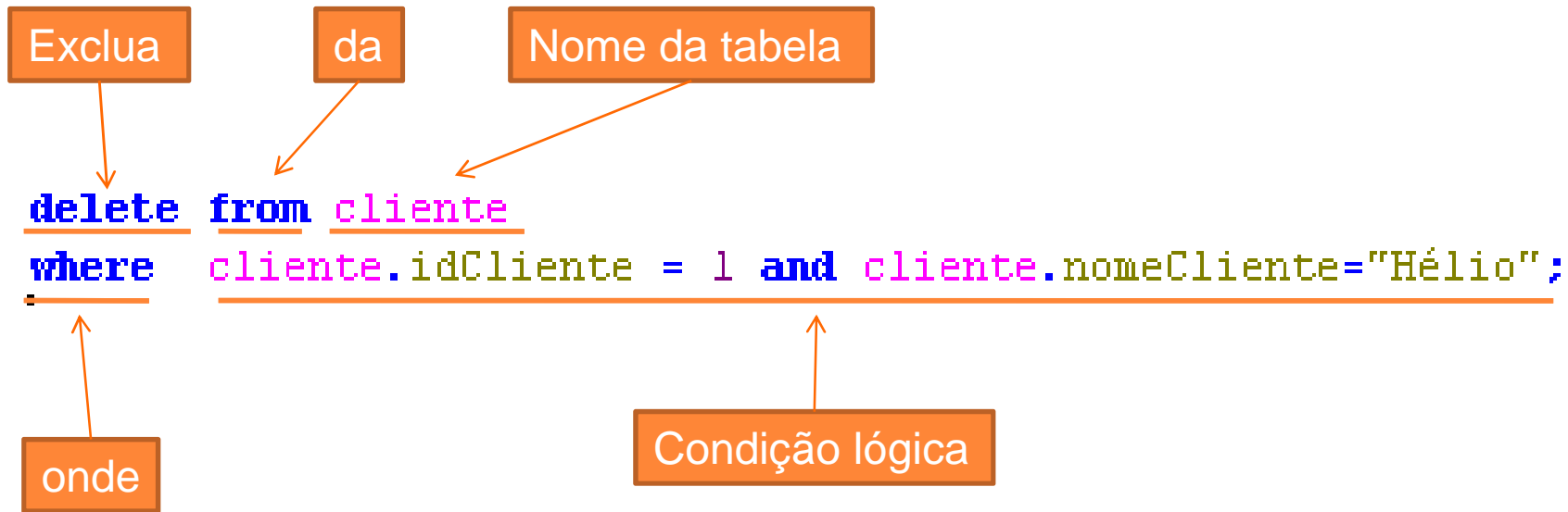
# ATUALIZANDO REGISTROS



## 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";
```