

INTRODUÇÃO AO MYSQL



MySQL®

PROF. ME. HÉLIO ESPERIDIÃO



Universidade do Vale do Paraíba

HeidiSQL



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 bancos de dados mais populares do mundo

- Entre os usuários do banco de dados MySQL estão:
 - 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 datos 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

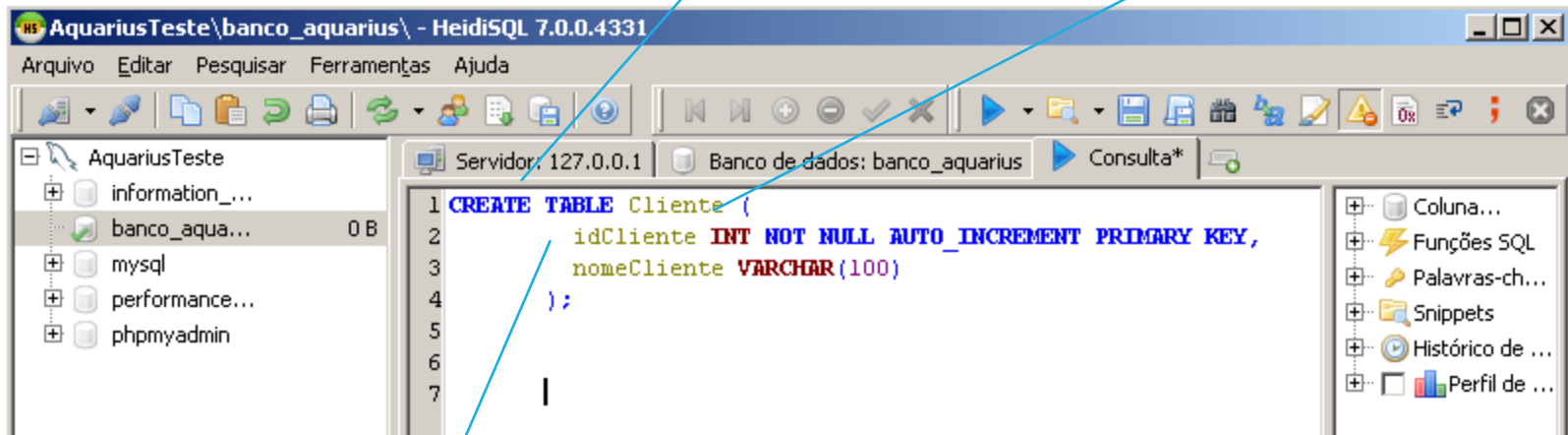
Exemplos de SGBD que utilizam SQL

- Oracle
- **MySQL**
- PostgreSQL
- SQL Server
- Interbase

Criando tabelas

Criar Tabela

Nome da tabela



The screenshot shows the HeidiSQL interface with a SQL query editor. The query is as follows:

```
1 CREATE TABLE Cliente (  
2     idCliente INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
3     nomeCliente VARCHAR(100)  
4 );  
5  
6  
7 |
```

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 query editor:

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

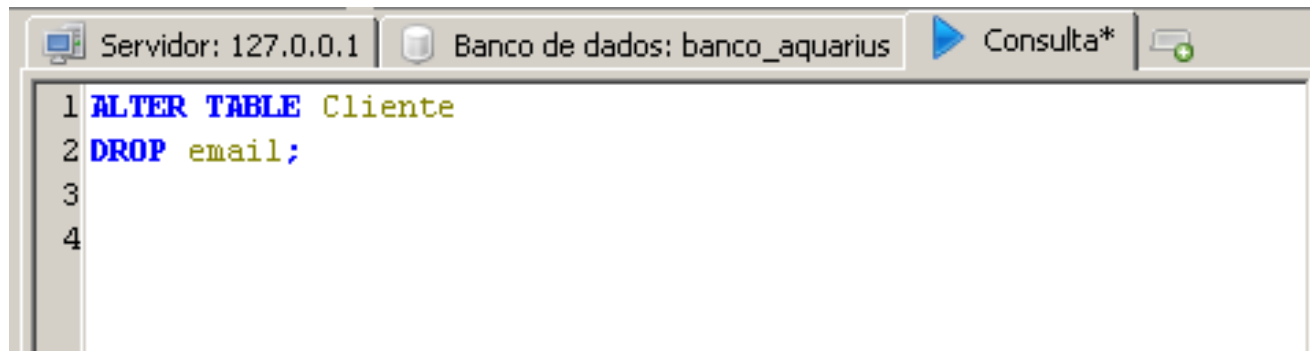
Annotations with arrows pointing to parts of the SQL command:

- Adicione** points to the `ADD` keyword.
- Nome do campo** points to `email`.
- Tipo do campo** points to `VARCHAR(100)`.
- Depois de:** points to `AFTER`.
- Nome do campo** points to `nomeCliente`.
- Altere a tabela** points to `ALTER TABLE`.
- Nome da tabela** points to `Cliente`.

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
```

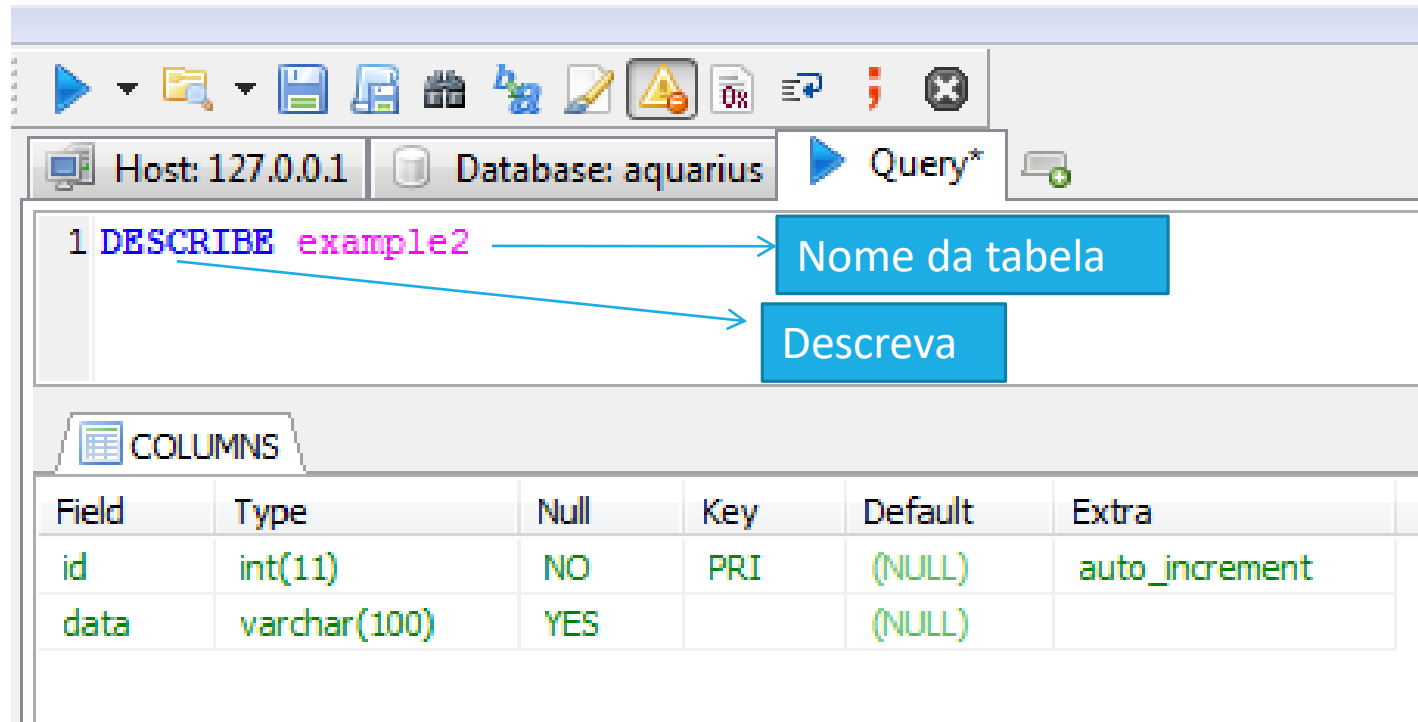
```
example1
```

```
example2
```

```
example_increment
```

O comando **Describe**

Descreve a estrutura de uma tabela.



The screenshot shows a database client interface with a toolbar at the top. Below the toolbar, the connection information is displayed: Host: 127.0.0.1 and Database: aquarius. A query editor window contains the command `1 DESCRIBE example2`. Two blue arrows point from the text in the query editor to labels: one points from `example2` to the label "Nome da tabela", and another points from `DESCRIBE` to the label "Descreva". Below the query editor, a tab labeled "COLUMNS" is active, showing a table with the following structure:

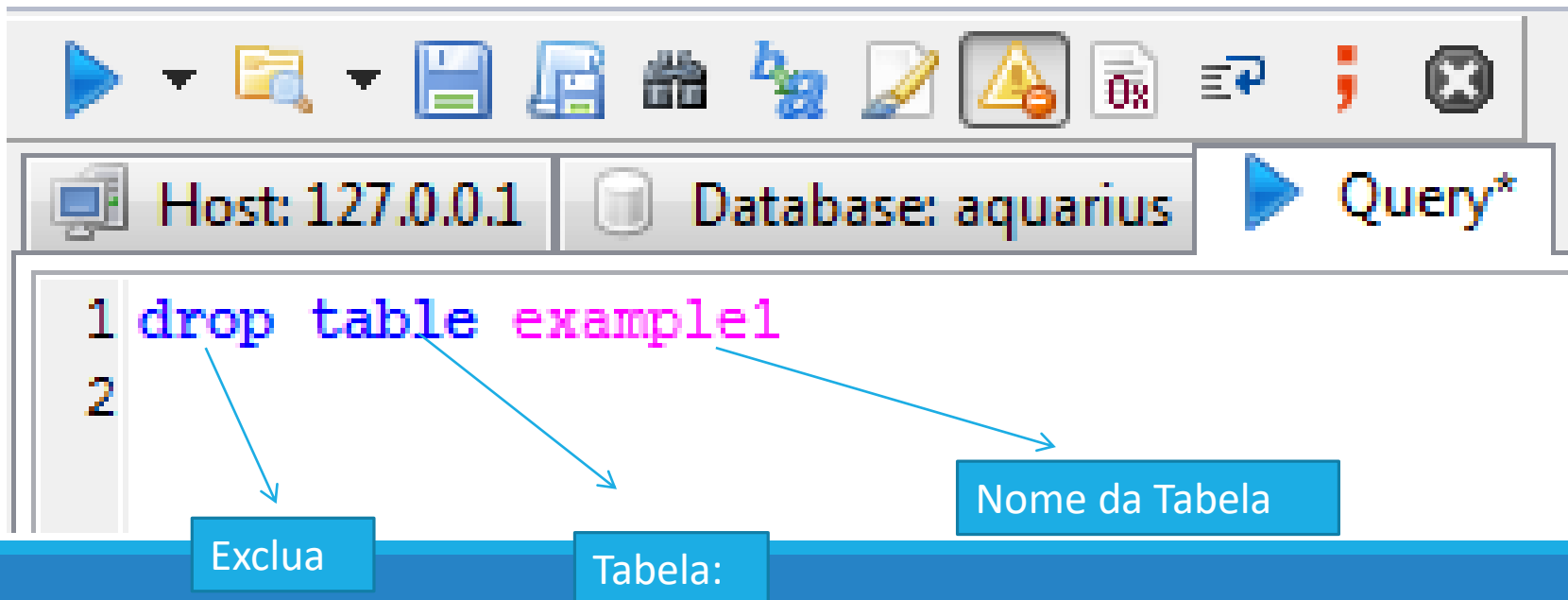
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	(NULL)	auto_increment
data	varchar(100)	YES		(NULL)	

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 screenshot shows a database management tool interface. At the top, there is a toolbar with various icons. Below the toolbar, the connection information is displayed: Host: 127.0.0.1 and Database: aquarius. A blue play button icon is followed by the text "Query*". The main area shows a SQL query:

```
1 drop table example1
2
```

Three blue arrows point from the query to labels below:

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

Inserindo dados em uma determinada ta

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 ("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);
```

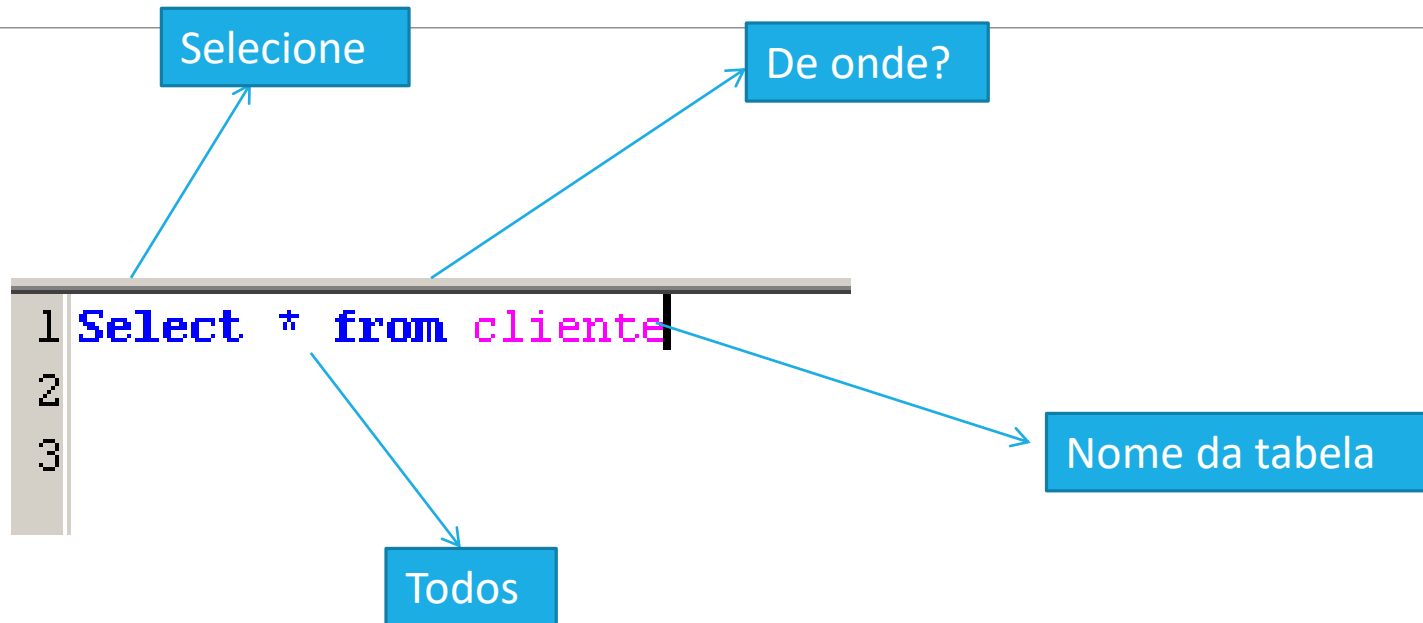
Inserir em:

Nome da Tabela

Campos separados por virgula

Valores separados por virgula;
Utilize a mesma seqüência dos campos

Selecionando dados em uma tabela



Colunas

Resultado

	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";
```

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

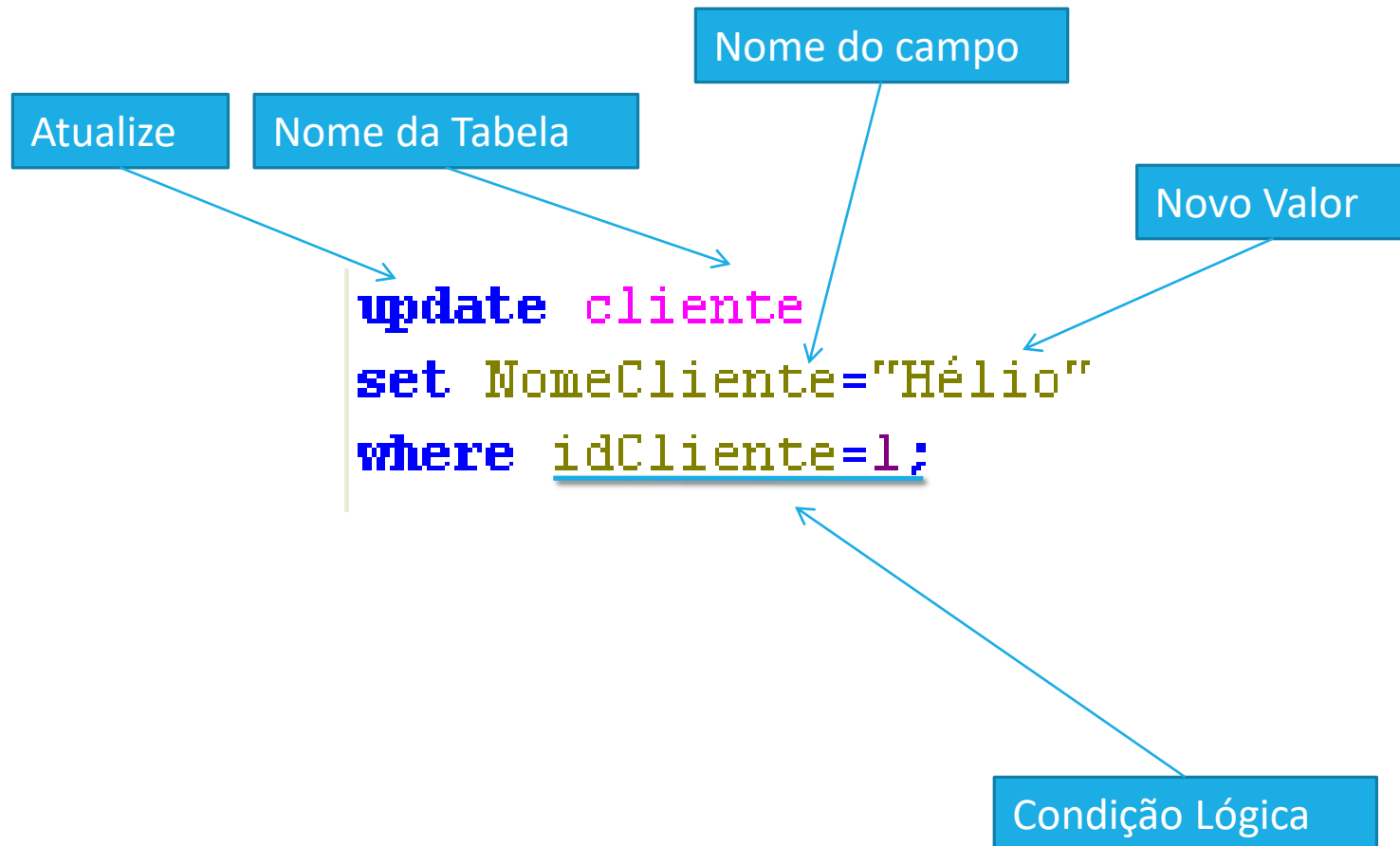
Or

Ou lógico

Resultado

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

Atualizando registros



Atualizando registros

Atualizações separadas por virgula

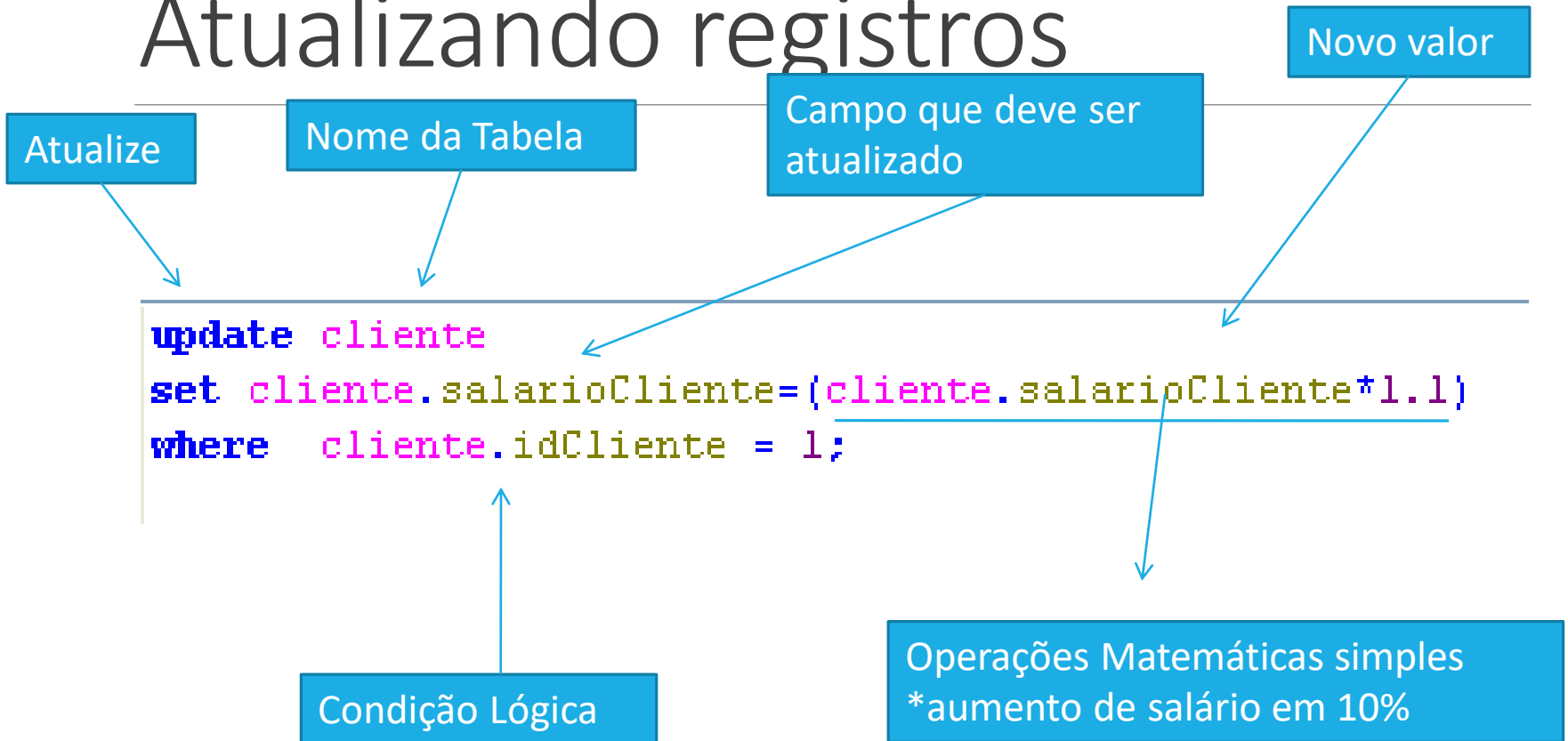


```
update cliente
set NomeCliente="Hélio",
sobreNomeCliente="Esperidião"
where NomeCliente="Héli" and sobreNomeCliente="Experidião"
and idCliente = 1;
```

Condições lógicas



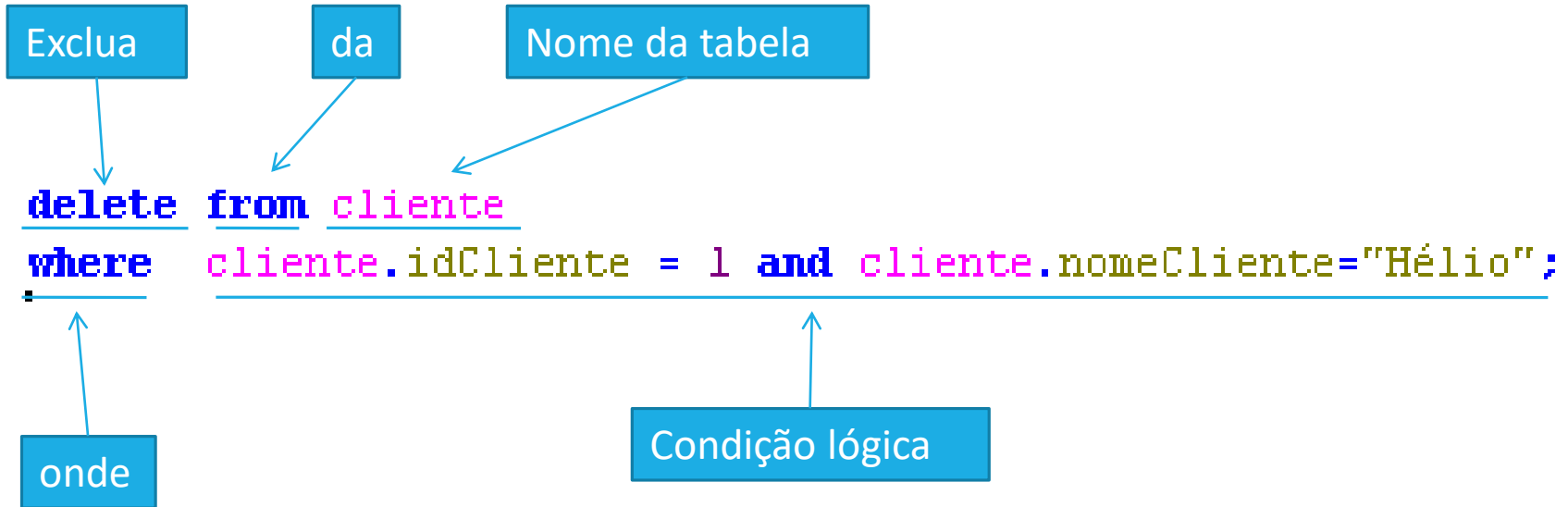
Atualizando registros



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";
```