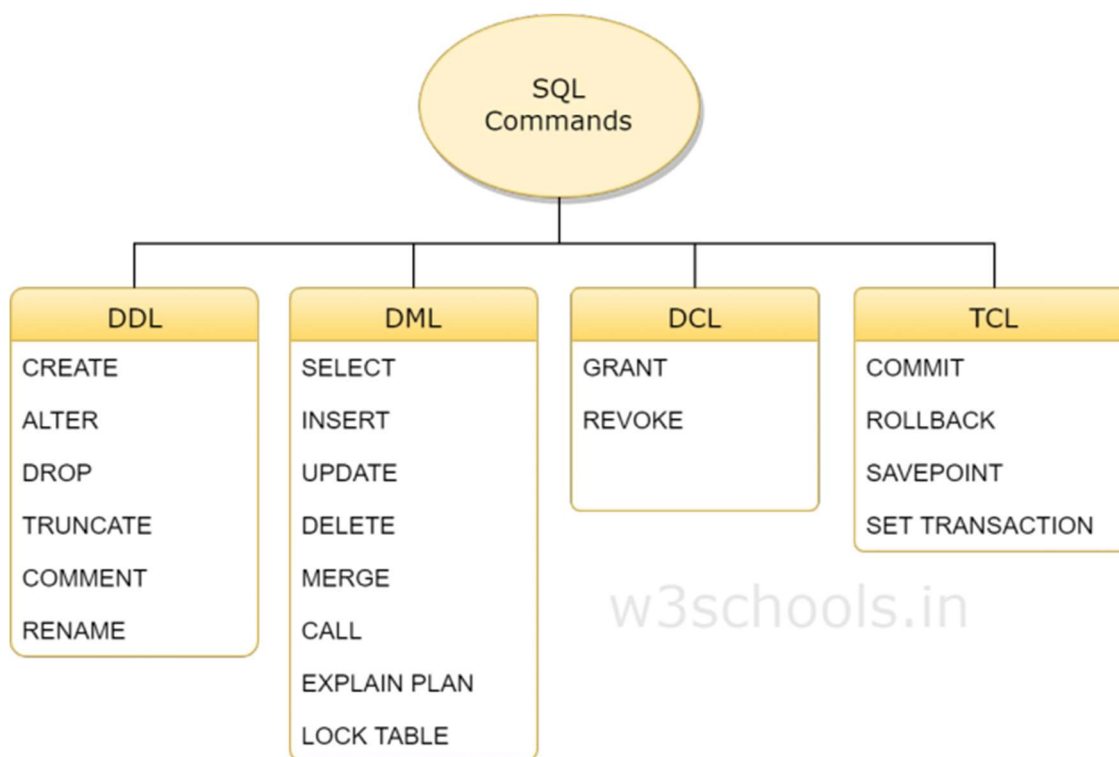


## Introdução a *Structured Query Language* - SQL

Fonte: <https://www.w3schools.in/mysql/ddl-dml-dcl/>  
<https://www.devmedia.com.br/mysql-tutorial/33309>



A **Structured Query Language** ou Linguagem de Consulta Estruturada (também conhecida pela sigla **SQL**) é a linguagem padrão utilizada para manipular bases de dados relacionais; podemos dividi-la em:

- **DDL Data Definition Language ou Linguagem de Definição de Dados:** São os comandos que interagem com os objetos do banco; alguns exemplos: CREATE, ALTER e DROP
- **DML Data Manipulation Language ou Linguagem de Manipulação de Dados:** São os comandos que interagem com os dados dentro das tabelas; alguns exemplos: SELECT, INSERT, DELETE e UPDATE
- **DTL Data Transaction Language - Linguagem de Transação de Dados:** São os comandos para controle de transação; alguns exemplos: BEGIN TRANSACTION, COMMIT E ROLLBACK
- **DCL Data Control Language - Linguagem de Controle de Dados:** São os comandos para controlar a parte de segurança do banco de dados; alguns exemplos: GRANT, REVOKE E DENY.

## Preparando o ambiente para aplicar o SQL no MySQL

Para aplicar as principais variações de consulta no banco de dados MySQL precisamos criar uma base e popular este objeto com algumas informações; **o tópico sobre a criação de bancos de dados e tabelas será abordado em detalhes mais adiante**; segue algumas informações importantes que precisam ser observadas:

- *Todo comando executado deve ser, por boa prática, finalizado com “ponto e vírgula”;*

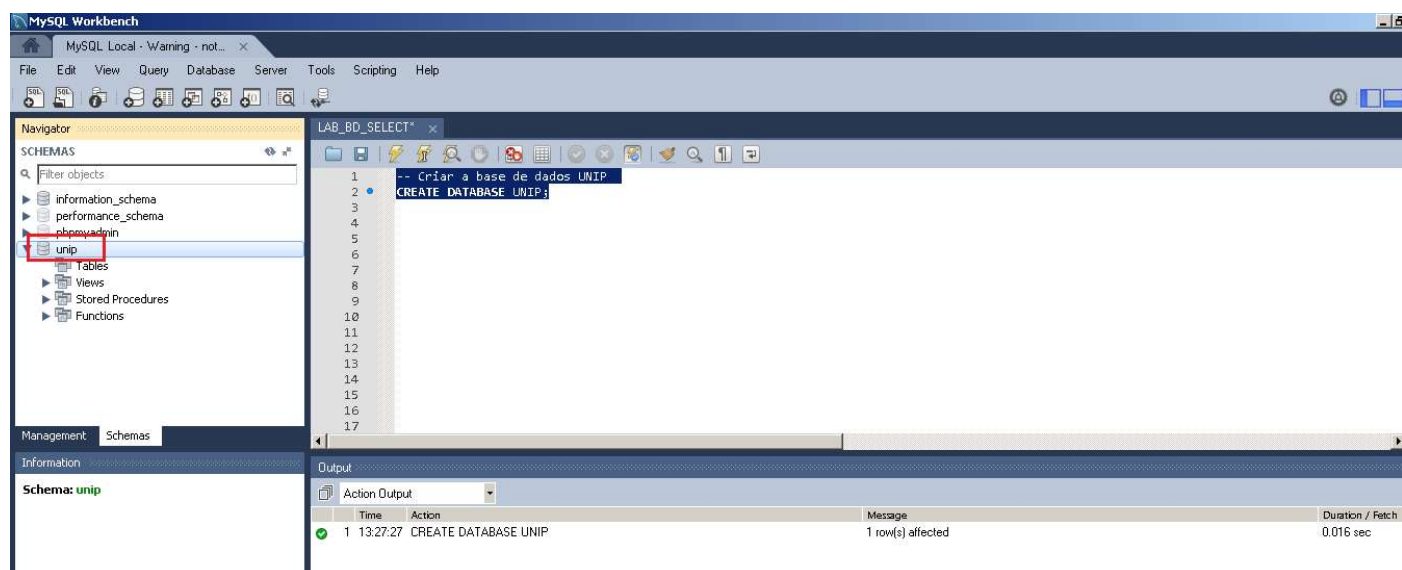
- *Atenção a sintaxe de cada comando; obedeça rigorosamente aos procedimentos abaixo, garantido assim que não ocorram distorções nas etapas do tutorial.*

- O tutorial será preferencialmente executado no ambiente online (<http://ubuntudb.brazilsouth.cloudapp.azure.com/phpmyadmin/index.php>); caso ocorra algum problema com este recurso, será utilizada a VM disponibilizada (Ubuntu + MySQL Workbench).

Vamos iniciar criando um banco de dados; **caso esteja utilizando o ambiente online, a base já esta criada e nomeada com o seu RA (vide orientações).**

-- Criar a base de dados UNIP

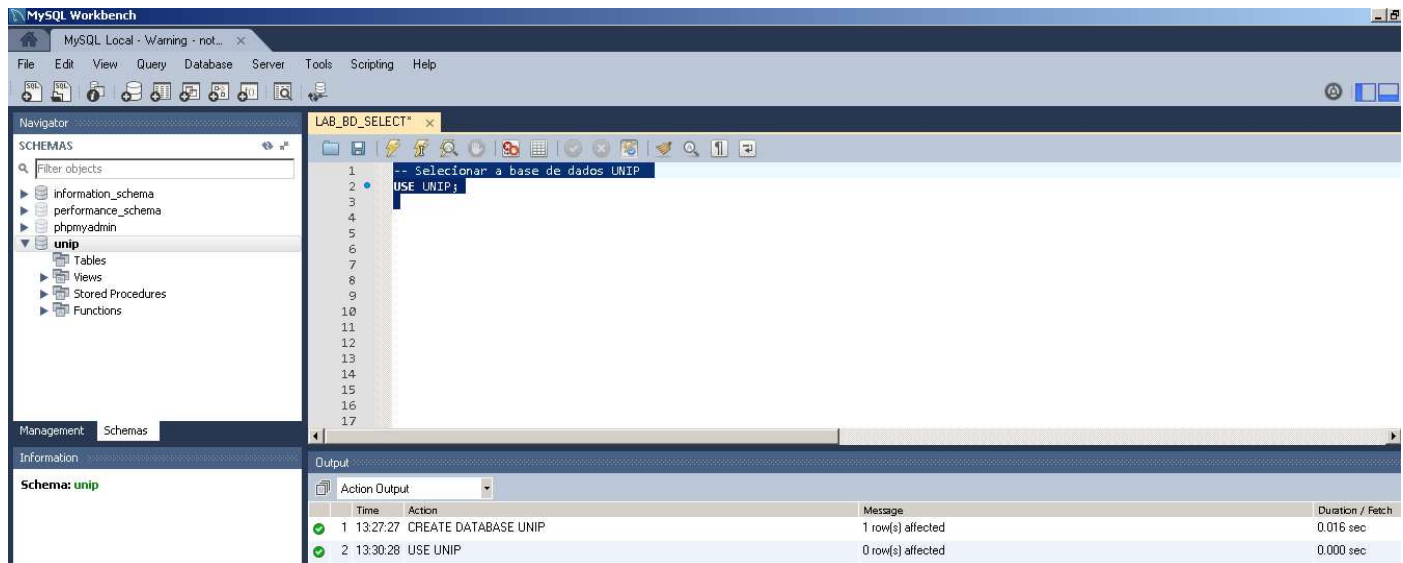
CREATE DATABASE UNIP;



Selecione a base de dados criada no passo anterior; veja que o BD em **NEGRITO** após ser selecionada:

```
-- Selecionar a base de dados UNIP
```

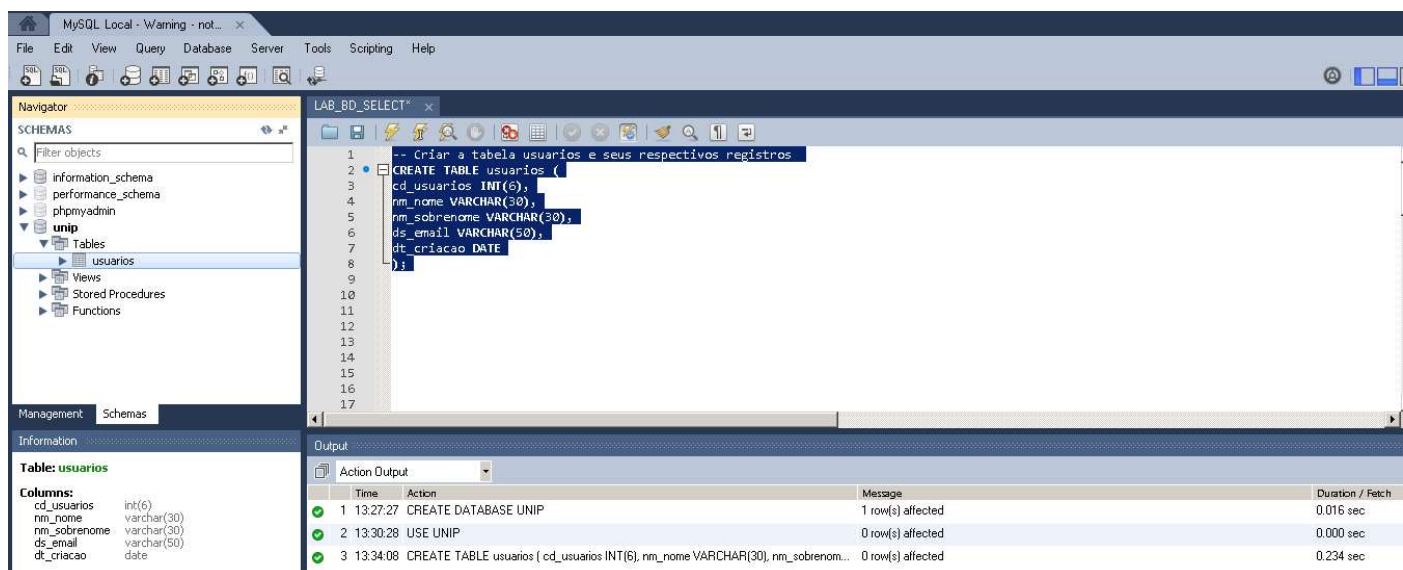
```
USE UNIP;
```



Crie a tabela usuarios; veja que neste primeiro momento não estamos tratando questões sobre chave primária e/ou estrangeira; estas características serão tratadas nas próximas aulas práticas:

```
-- Criar a tabela usuarios e seus respectivos registros
```

```
CREATE TABLE usuarios (
  cd_usuario INT(6),
  nm_nome VARCHAR(30),
  nm_sobrenome VARCHAR(30),
  ds_email VARCHAR(50),
  dt_criacao DATE
);
```



Insira os dados abaixo na tabela criada no passo anterior; a **inserção no campo dt\_criacao** utiliza recursos característicos do MySQL e que serão abordados nas próximas aulas práticas:

-- Inserindo registros na tabela usuarios

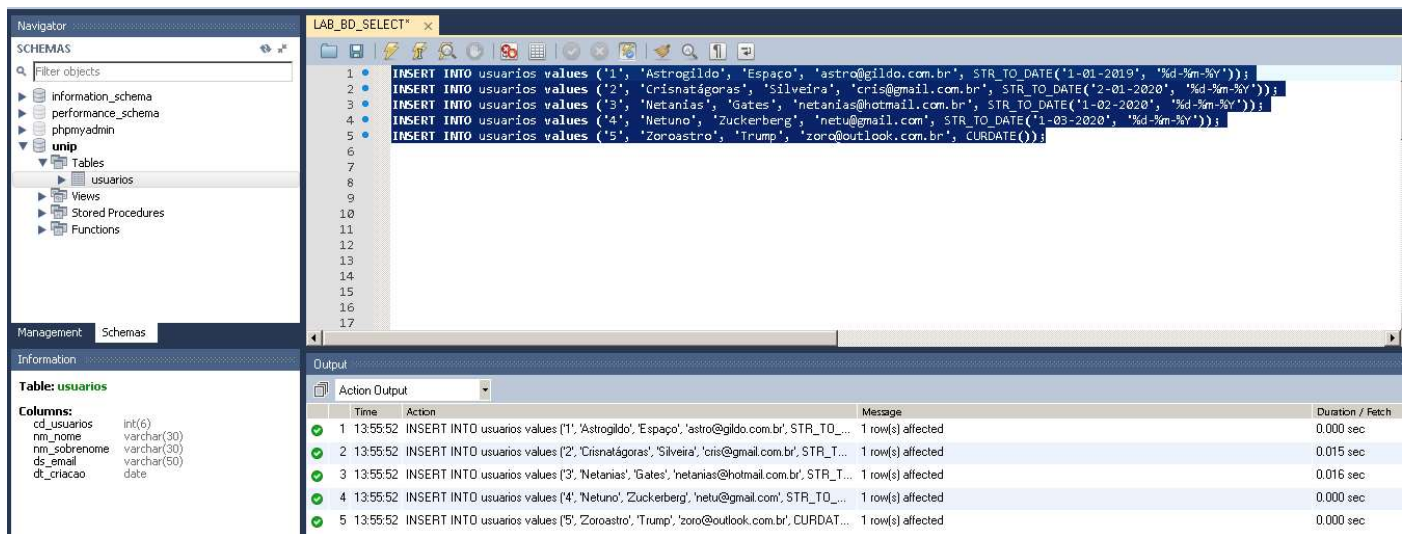
```
INSERT INTO usuarios values ('1', 'Astrogildo', 'Espaço', 'astro@gildo.com.br', STR_TO_DATE('1-01-2019', '%d-%m-%Y'));

INSERT INTO usuarios values ('2', 'Crisnatágoras', 'Silveira', 'cris@gmail.com.br', STR_TO_DATE('2-01-2020', '%d-%m-%Y'));

INSERT INTO usuarios values ('3', 'Netanias', 'Gates', 'netanias@hotmail.com.br', STR_TO_DATE('1-02-2020', '%d-%m-%Y'));

INSERT INTO usuarios values ('4', 'Netuno', 'Zuckerberg', 'netu@gmail.com', STR_TO_DATE('1-03-2020', '%d-%m-%Y'));

INSERT INTO usuarios values ('5', 'Zoroastro', 'Trump', 'zoro@outlook.com.br', CURDATE());
```



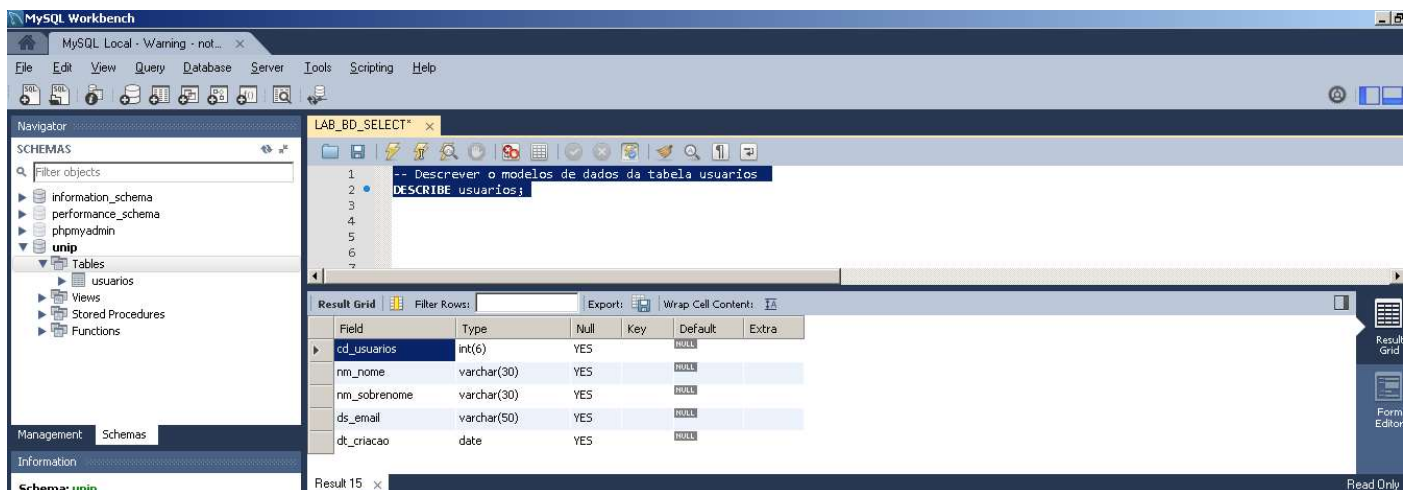
The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'unip' database schema with the 'usuarios' table selected. The main editor window contains five INSERT statements. The bottom Output window shows the results of these statements:

Time	Action	Message	Duration / Fetch
1 13:55:52	INSERT INTO usuarios values ('1', 'Astrogildo', 'Espaço', 'astro@gildo.com.br', STR_TO_DATE('1-01-2019', '%d-%m-%Y'));	1 row(s) affected	0.000 sec
2 13:55:52	INSERT INTO usuarios values ('2', 'Crisnatágoras', 'Silveira', 'cris@gmail.com.br', STR_TO_DATE('2-01-2020', '%d-%m-%Y'));	1 row(s) affected	0.015 sec
3 13:55:52	INSERT INTO usuarios values ('3', 'Netanias', 'Gates', 'netanias@hotmail.com.br', STR_TO_DATE('1-02-2020', '%d-%m-%Y'));	1 row(s) affected	0.016 sec
4 13:55:52	INSERT INTO usuarios values ('4', 'Netuno', 'Zuckerberg', 'netu@gmail.com', STR_TO_DATE('1-03-2020', '%d-%m-%Y'));	1 row(s) affected	0.000 sec
5 13:55:52	INSERT INTO usuarios values ('5', 'Zoroastro', 'Trump', 'zoro@outlook.com.br', CURDATE());	1 row(s) affected	0.000 sec

É possível verificar a estrutura da sua tabela através do comando **DESCRIBE**:

-- Descrever o modelo de dados da tabela usuarios

DESCRIBE usuarios;



The screenshot shows the MySQL Workbench interface. The main editor window contains the command 'DESCRIBE usuarios;'. The bottom Result Grid shows the table structure:

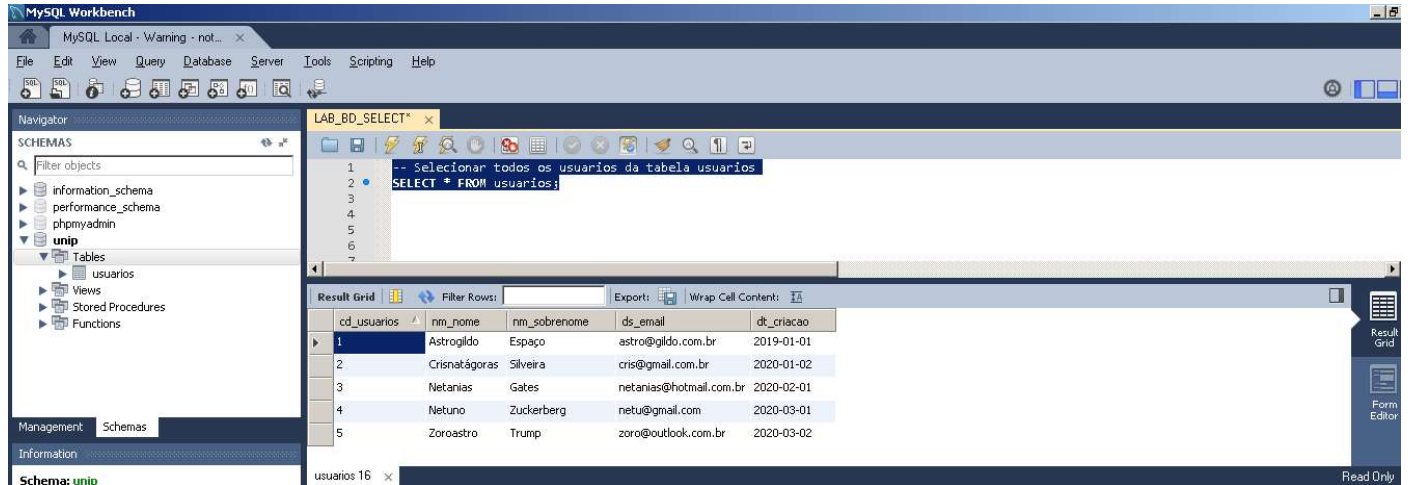
Field	Type	Null	Key	Default	Extra
cd_usuarios	int(6)	YES		NULL	
nm_nome	varchar(30)	YES		NULL	
nm_sobrenome	varchar(30)	YES		NULL	
ds_email	varchar(50)	YES		NULL	
dt_criacao	date	YES		NULL	

## Efetuando consultas no MySQL

Vamos iniciar verificando todos os dados da tabela, independente de condição ou filtro (WHERE):

-- Selecionar todos os registros da tabela usuarios

```
SELECT * FROM usuarios;
```



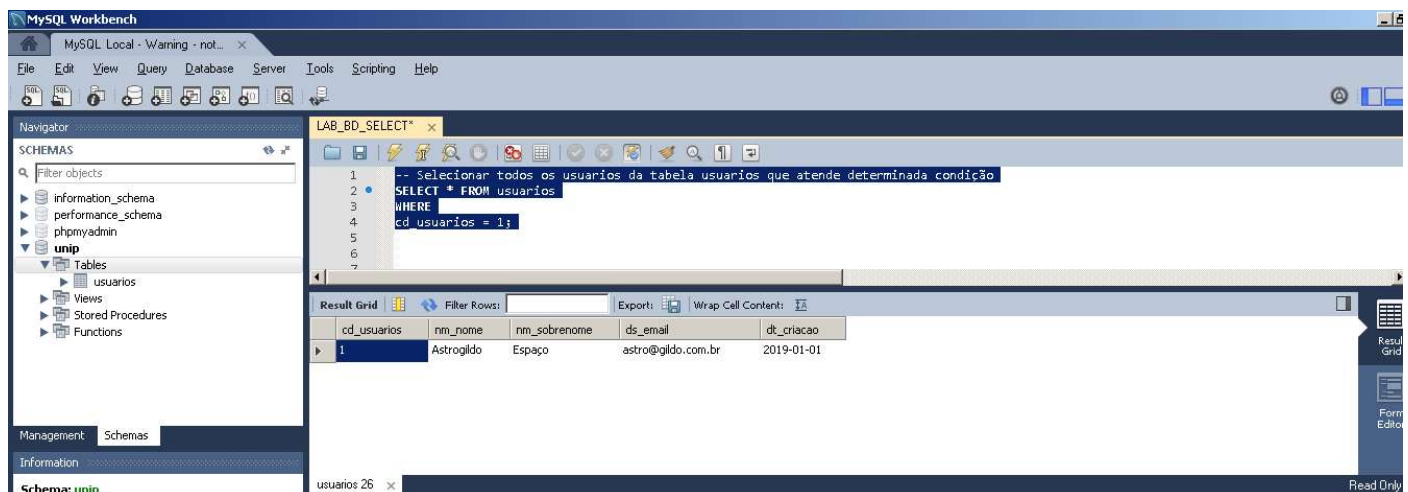
Neste comando vamos aplicar uma condição para que o registro seja encontrado em nossa tabela (WHERE):

-- Selecionar todos os registros da tabela usuarios que atendem determinada condição

```
SELECT * FROM usuarios
```

```
WHERE
```

```
cd_usuario = 1;
```





Neste comando vamos aplicar uma condição com mais de um parâmetro para que o registro seja encontrado em nossa tabela (WHERE); veja que neste caso utilizamos o operador lógico AND para estabelecer a combinação desejada:

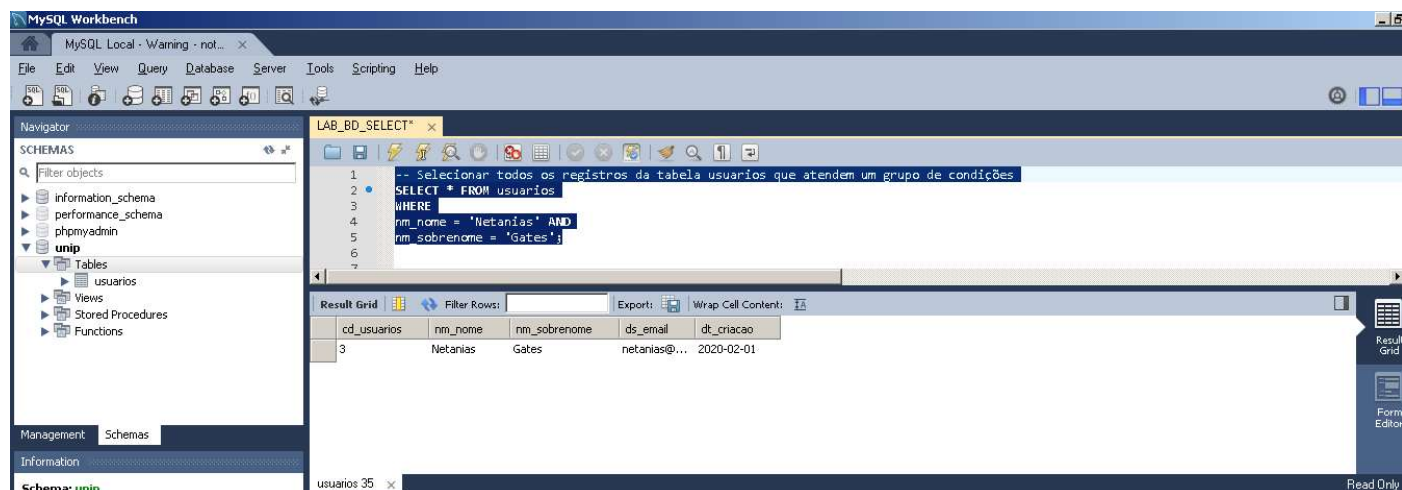
```
-- Selecionar todos os registros da tabela usuarios que atendem um grupo de condições
```

```
SELECT * FROM usuarios
```

```
WHERE
```

```
nm_nome = 'Netanias' AND
```

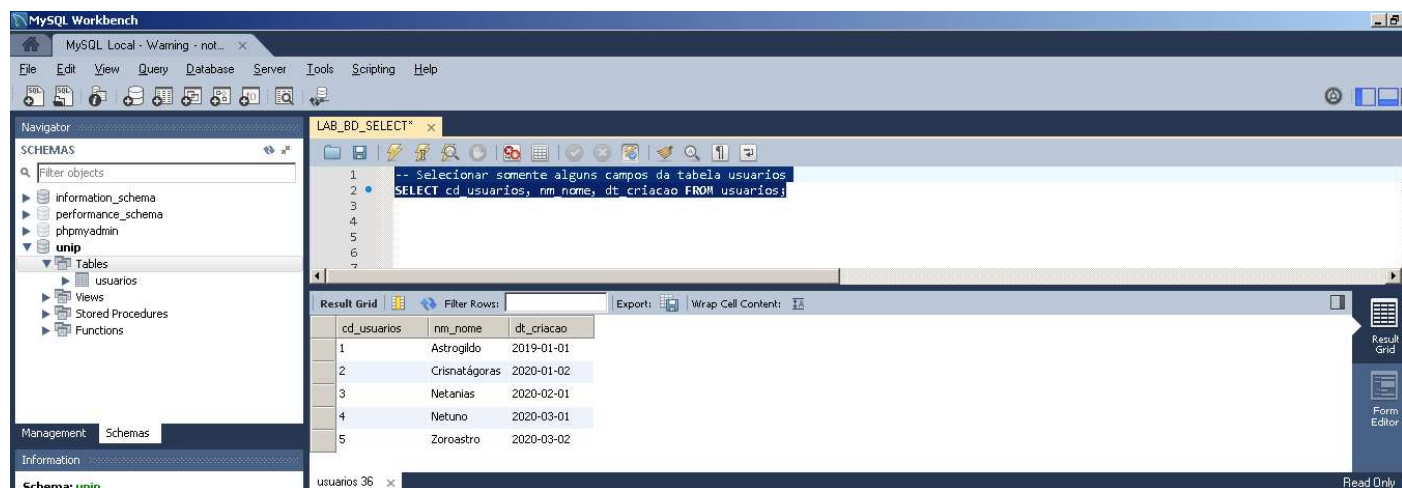
```
nm_sobrenome = 'Gates';
```



Podemos determinar quais são os campos que você deseja retornar na consulta; desta forma otimizamos o resultado, facilitando seu entendimento:

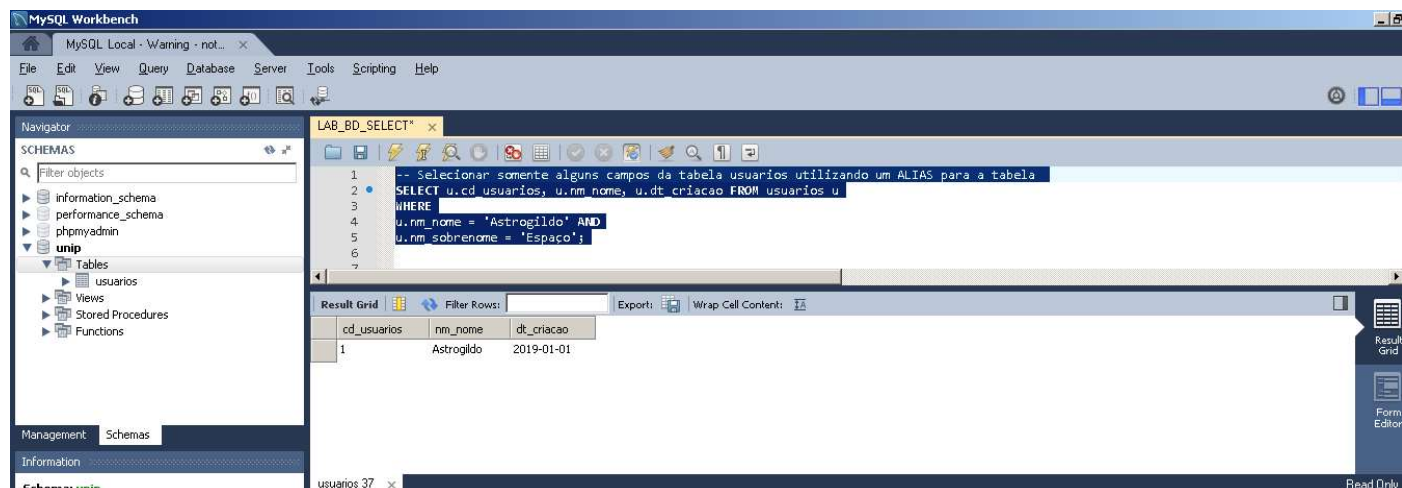
```
-- Selecionar somente alguns campos da tabela usuarios
```

```
SELECT cd_usuario, nm_nome, dt_criacao FROM usuarios;
```



Outro recurso bastante utilizado são os APELIDOS ou ALIAS; este formato permite nomear as tabelas e seus respectivos campos e será de grande serventia quando as consultas precisam retornar valores de múltiplas tabelas simultaneamente (relacionamentos):

```
-- Selecionar somente alguns campos da tabela usuarios utilizando um ALIAS para a tabela
SELECT u.cd_usuario, u.nm_nome, u.dt_criacao FROM usuarios u
WHERE
u.nm_nome = 'Astrogildo' AND
u.nm_sobrenome = 'Espaço';
```



## Exercícios

- 1- Construa uma consulta (SELECT) que retorne somente os campos nm\_sobrenome e dt\_criacao para os usuários que possuem nome igual a Netuno.
- 2- Construa uma consulta (SELECT) que retorne somente os campos cd\_usuarios, nm\_nome e dt\_criacao para os usuários que possuem código igual a 3.
- 3- Utilizando um apelido (*alias*) para a tabela usuarios, efetue uma consulta (SELECT) que retorne todos os registros que possuem código igual a 2.
- 4- Insira os registros abaixo na tabela usuarios:

código	nome	sobrenome	e-mail	data de inclusão
6	Marcus	Pontes	astro@pontes.com.br	17/02/2023
7	Tomas	Edson		01-02-2023
8	Michael	Jackson	michael@jackson.com	2022-12-02
9	Roberto		roberto@carlos.com.br	20221101

- 5- Construa uma consulta (SELECT) que indique os usuarios que não possuem sobrenome cadastrado.
- 6- Construa uma consulta (SELECT) que aponte os usuarios que não possuem e-mail cadastrado.
- 7- Construa uma consulta (SELECT) que aponte a quantidade de usuarios cadastrados.
- 8- Construa uma consulta (SELECT) que aponte a quantidade de usuarios cadastrados e que NÃO possuem e-mail.
- 9- Construa uma consulta (SELECT) que aponte a quantidade de usuarios cadastrados e que possuem e-mail.
- 10- Criar uma tabela departamentos com os seguintes atributos:
  - cd\_departamento
  - nm\_departamento
  - dt\_criacao
- 11- Insira os registros abaixo na tabela departamentos:

código	nome	data de criação
1	Tecnologia da Informação	01/01/2021
2	Financeiro	01-01-2021
3	Gerência	2021-01-01
4	Recursos Humanos	

- 12- Construa uma consulta (SELECT) que aponte todos os departamentos da empresa SEM o respectivo código de cada um no resultado da pesquisa.