

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE SÃO PAULO - IFSP

RAISSA PEREIRA MIRANDA

MYSQL

CAMPOS DO JORDÃO

2025

1. INTRODUÇÃO

MySQL é um sistema de gerenciamento de banco de dados relacional (SGBDR) criado em 1995 por Michael "Monty" Widenius e sua equipe na empresa sueca MySQL AB. Em 2008, a Sun Microsystems adquiriu a MySQL AB e, posteriormente, em 2010, a Oracle Corporation comprou a Sun, tornando-se a responsável pelo desenvolvimento e manutenção do MySQL. Essa trajetória o consolidou como um dos bancos de dados mais utilizados no mundo.

Trata-se de um sistema de código aberto, multiusuário e multitarefa, que utiliza a linguagem SQL (Structured Query Language) para manipulação de dados. Ele permite que desenvolvedores e administradores de banco de dados realizem operações como inserção, atualização e consulta de informações de forma eficiente. Além disso, é compatível com diversas plataformas e linguagens de programação, como PHP, Java e Python, facilitando sua integração em projetos variados.

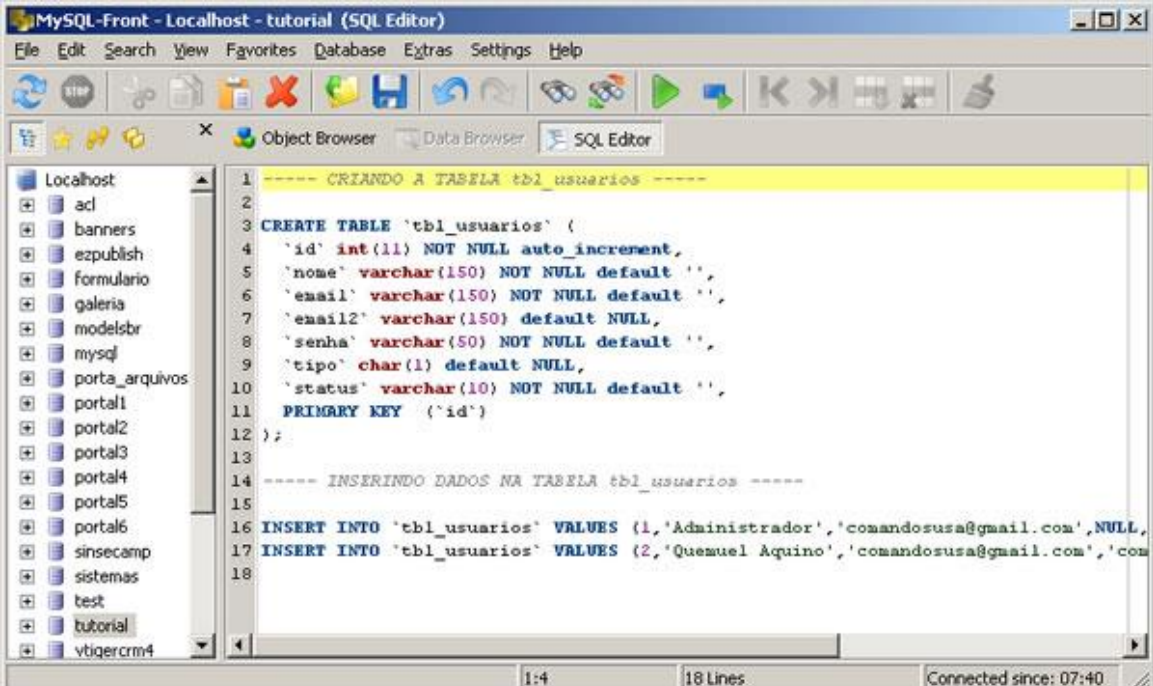
2. DESENVOLVIMENTO

2.1 ONDE É UTILIZADO

A capacidade do MySQL de lidar com grandes conjuntos de dados e consultas complexas o torna amplamente utilizado em setores, como:

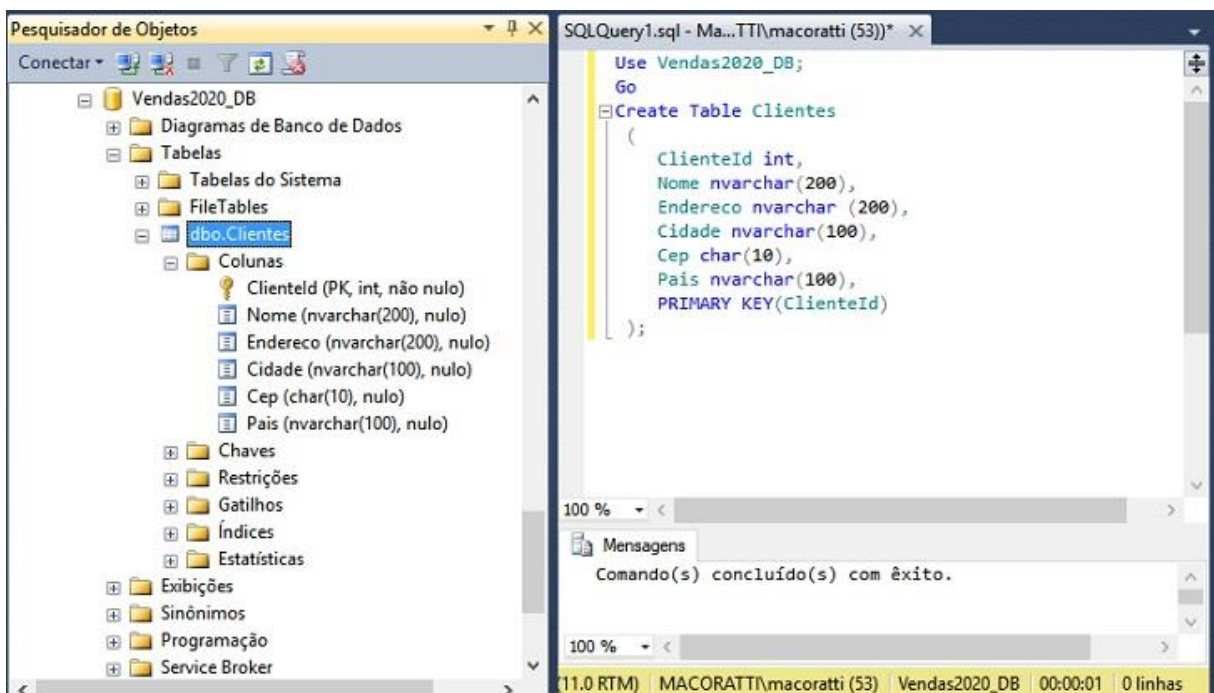
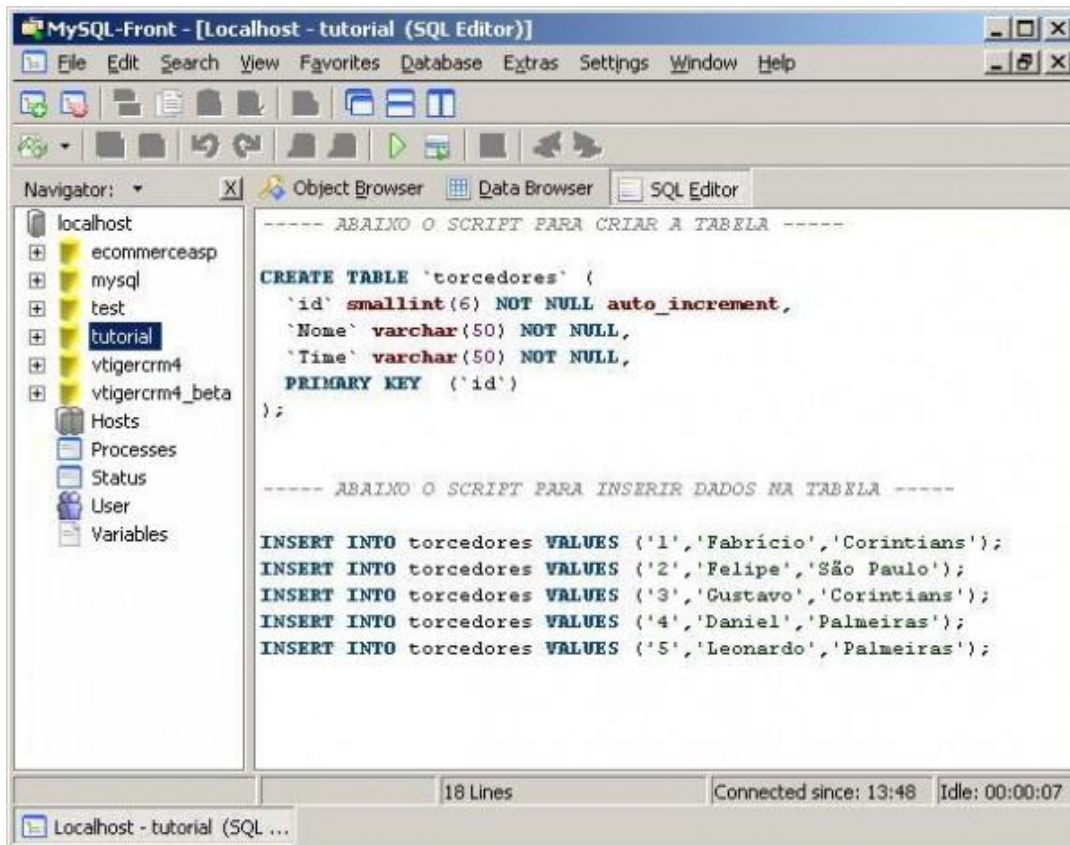
- **E-commerce:** Usado por empresas como Uber e Booking.com para gerenciar transações, perfis de usuário, pagamentos e detecção de fraudes.
- **Redes sociais:** Plataformas como Facebook, X (Twitter) e LinkedIn utilizam MySQL em sua infraestrutura.
- **Gerenciamento de conteúdo:** Suporta operações SQL e NoSQL, com recursos para consultar documentos JSON, facilitando relatórios e análises.
- **SaaS e ISVs:** Empresas como Zendesk, Ericsson e IBM integram MySQL em seus produtos e serviços para ganhar competitividade e agilidade.
- **Aplicações locais (on-premises):** O MySQL Enterprise Edition oferece recursos avançados de segurança, escalabilidade e suporte técnico, sendo ideal para aplicações críticas de negócios.

2.2 EXEMPLOS DE CONSULTAS



The screenshot shows the MySQL-Front application window titled "MySQL-Front - Localhost - tutorial (SQL Editor)". The interface includes a menu bar (File, Edit, Search, View, Favorites, Database, Extras, Settings, Help), a toolbar with various icons, and a tabbed interface with "Object Browser", "Data Browser", and "SQL Editor" tabs. The "Object Browser" on the left shows a tree view of the database structure, including "Localhost" and several databases like "ad", "banners", "ezpublish", etc. The "SQL Editor" tab is active, displaying SQL code. The code is divided into two sections: "CRIANDO A TABELA tbl_usuarios" (Creating the table) and "INSERINDO DADOS NA TABELA tbl_usuarios" (Inserting data into the table). The first section uses the CREATE TABLE statement to define the table structure with columns: id (int(11) NOT NULL auto_increment), nome (varchar(150) NOT NULL default ''), email (varchar(150) NOT NULL default ''), email2 (varchar(150) default NULL), senha (varchar(50) NOT NULL default ''), tipo (char(1) default NULL), and status (varchar(10) NOT NULL default ''), with 'id' as the PRIMARY KEY. The second section uses two INSERT INTO statements to add data to the table. The status bar at the bottom indicates "1:4", "18 Lines", and "Connected since: 07:40".

```
1  ----- CRIANDO A TABELA tbl_usuarios -----
2
3  CREATE TABLE `tbl_usuarios` (
4    `id` int(11) NOT NULL auto_increment,
5    `nome` varchar(150) NOT NULL default '',
6    `email` varchar(150) NOT NULL default '',
7    `email2` varchar(150) default NULL,
8    `senha` varchar(50) NOT NULL default '',
9    `tipo` char(1) default NULL,
10   `status` varchar(10) NOT NULL default '',
11   PRIMARY KEY (`id`)
12 );
13
14 ----- INSERINDO DADOS NA TABELA tbl_usuarios -----
15
16 INSERT INTO `tbl_usuarios` VALUES (1,'Administrador','comandosusa@gmail.com',NULL,
17 INSERT INTO `tbl_usuarios` VALUES (2,'Queemuel Aquino','comandosusa@gmail.com','com
```



2.2.1 CONSULTAS BÁSICAS

As consultas serão baseadas na tabela **employees**:

id	first_name	last_name	department	salary
1	Paul	Garrix	Corporate	3,547.25
2	Astrid	Fox	Private Individuals	2,845.56
3	Matthias	Johnson	Private Individuals	3,009.41
4	Lucy	Patterson	Private Individuals	3,547.25
5	Tom	Page	Corporate	5,974.41
6	Claudia	Conte	Corporate	4,714.12
7	Walter	Deer	Private Individuals	3,547.25
8	Stephanie	Marx	Corporate	2,894.51
9	Luca	Pavarotti	Private Individuals	4,123.45
10	Victoria	Pollock	Corporate	4,789.53

Consulta 01: Seleção de todas as colunas da tabela.

Code

```
SELECT *  
FROM employees;
```

Consulta 02: Seleção de três colunas da tabela e filtragem usando comparação numérica em WHERE.

```
Code

SELECT
    first_name,
    last_name,
    salary
FROM employees
WHERE salary > 3800;
```

Consulta 03: Seleção de três colunas com uma condição lógica complexa em WHERE.

```
Code

SELECT
    first_name,
    last_name,
    salary
FROM employees
WHERE salary > 5000 OR salary < 3000;
```

2.2.1 CONSULTAS AVANÇADAS

As consultas serão baseadas na tabela **employee**:

employee_id	first_name	last_name	dept_id	manager_id	salary	expertise
100	John	White	IT	103	120000	Senior
101	Mary	Danner	Account	109	80000	junior
102	Ann	Lynn	Sales	107	140000	Semisensor
103	Peter	O'connor	IT	110	130000	Senior
106	Sue	Sanchez	Sales	107	110000	Junior
107	Marta	Doe	Sales	110	180000	Senior
109	Ann	Danner	Account	110	90000	Senior
110	Simon	Yang	CEO	null	250000	Senior
111	Juan	Graue	Sales	102	37000	Junior

Consulta 01: Listagem de todos os funcionários ordenados por salário (primeiro o salário mais alto). O relatório incluirá a posição de cada funcionário no ranking.

Code

```
SELECT
    employee_id,
    last_name,
    first_name,
    salary,
    RANK() OVER (ORDER BY salary DESC) as ranking
FROM employee
ORDER BY ranking
```

Consulta 02: Criação de um relatório com os dados dos funcionários para os 5 salários mais altos da empresa.

Code

```
WITH employee_ranking AS (
    SELECT
        employee_id,
        last_name,
        first_name,
        salary,
        RANK() OVER (ORDER BY salary DESC) as ranking
    FROM employee
)
SELECT
    employee_id,
    last_name,
    first_name,
    salary
FROM employee_ranking
WHERE ranking ≤ 5
ORDER BY ranking
```

Consulta 03: Criação de um ranking que atribua a cada linha um número indicando a posição daquela linha no ranking

Code

```
SELECT
  employee_id,
  last_name,
  first_name,
  salary,
  ROW_NUMBER() OVER (ORDER BY employee_id) as ranking_position
FROM employee
```


3. CONCLUSÃO

Em conclusão, esta pesquisa permitiu compreender a eficiência do MySQL como um dos sistemas de gerenciamento de banco de dados mais utilizados no mundo. Sua compatibilidade com as diferentes linguagens de programação e plataformas, além da capacidade de lidar com grandes volumes de dados e realizar consultas complexas, o torna uma escolha confiável para aplicações em diversos setores. Além disso, por ser um software de código aberto, o MySQL oferece um ambiente acessível para aprendizado e inovação, sendo uma ótima ferramenta para a área de tecnologia.

4. REFERÊNCIAS

BABIC, Tihomir. 20 exemplos de consultas SQL básicas para iniciantes: Uma visão geral completa. Learn SQL, 2023. Disponível em: <https://learnsql.com.br/blog/20-exemplos-de-consultas-sql-basicas-para-iniciantes-uma-visao-geral-completa/>. Acesso em: 12 abr. 2025.

BISSO, Ignacio. 25 Exemplos avançados de consulta SQL. Learn SQL, 2023. Disponível em: <https://learnsql.com.br/blog/25-exemplos-avancados-de-consulta-sql/>. Acesso em: 12 abr. 2025.

CESCONETTO, Juliano. O que é: MySQL. Drift Web, 2024. Disponível em: <https://driftweb.com.br/glossario/o-que-e-mysql-entenda-o-banco-de-dados/>. Acesso em: 11 abr. 2025.

ERICKSON, Jeffrey. MySQL: Entendendo o que é e como é usado. Oracle, 2024. Disponível em: <https://www.oracle.com/br/mysql/what-is-mysql/>. Acesso em: 12 abr. 2025.