MySQL Basics

Project objective

- Understanding that it's a relational base
- Understand how to query a database
- Analyze and design databases with their corresponding tables and rules.
- Understanding how to interact with data stored in the database

Project requirements

1. General analysis

1.1. Install MySQL Server

The first source point of this project is to install MySQL Server on your machine so you can work. It's important that you use a version that has support and is as up-to-date as possible.
It is important that you always keep up to date the software on which your application depends. For this project you need a version of MySQL database server equal to or greater than 5.0.

1.2. Run the local server

Then verify that you have MySQL server up and running on your local server so you can get started.

1.3. Imports a sample database

You will then import a sample database provided by the official MySQL team. The goal of this project is to learn how to work on an existing database, so you don't need to design a database.
Use the following link to import and verify:
https://dev.mysql.com/doc/employee/en/
It's important to analyze and take enough time to understand the design of the database, otherwise you won't be able to make the right queries to get the data you want. In the following link you can see the diagram of the database
https://dev.mysql.com/doc/employee/en/sakila-structure.html

1.4. Execute staked SQL queries

You will then need to perform the following SQL queries:

1.4.1. INSERT DATA

- **Insert** 15 new employees:
  - With salaries between 5,000 and 50,000
  - Different gender
  - 5 employees must have at least two salaries in different date ranges and different amounts
  - 10 employees belong to more than one department
  - 5 employees are managers
  - All employees have a degree and at least 5 degrees are from 2019
  - At least 3 employees have the same name

1.4.2. UPDATE DATA

- **Update** employees:
  - Change an employee's name. To do this, it generates a query that affects only a certain employee based on his or her first name, last name, and date of birth.
- **Update** departments:
  - Rename all departments.

1.4.3. OBTAIN DATA

- **Select** all employees who have a salary above 20,000
- **Select** all employees who have a salary below 10,000
- **Select** all employees who have a salary between 14.00 and 50,000
- **Select** the total number of employees
- **Select** the total number of employees who have worked in more than one department
- **Select** the 2019 titles of the year
- **Select** only the name of employees in uppercase
- **Select** the first name, last name, and first name of each employee's current department
- **Select** the first name, last name, and number of times the employee has worked as a manager
- **Select** the name without any being repeated

1.4.4. DELETE DATA

- **Eliminates** all employees who have a salary above 20,000
- **Eliminate** the department that has the most employees

1.5. CREATE A SQL FILE

Once the above sections are finished, you will need to create a ".sql" file that contains all the querys performed to run the tests. This file must be correctly documented with comments in the code to understand the queries made.

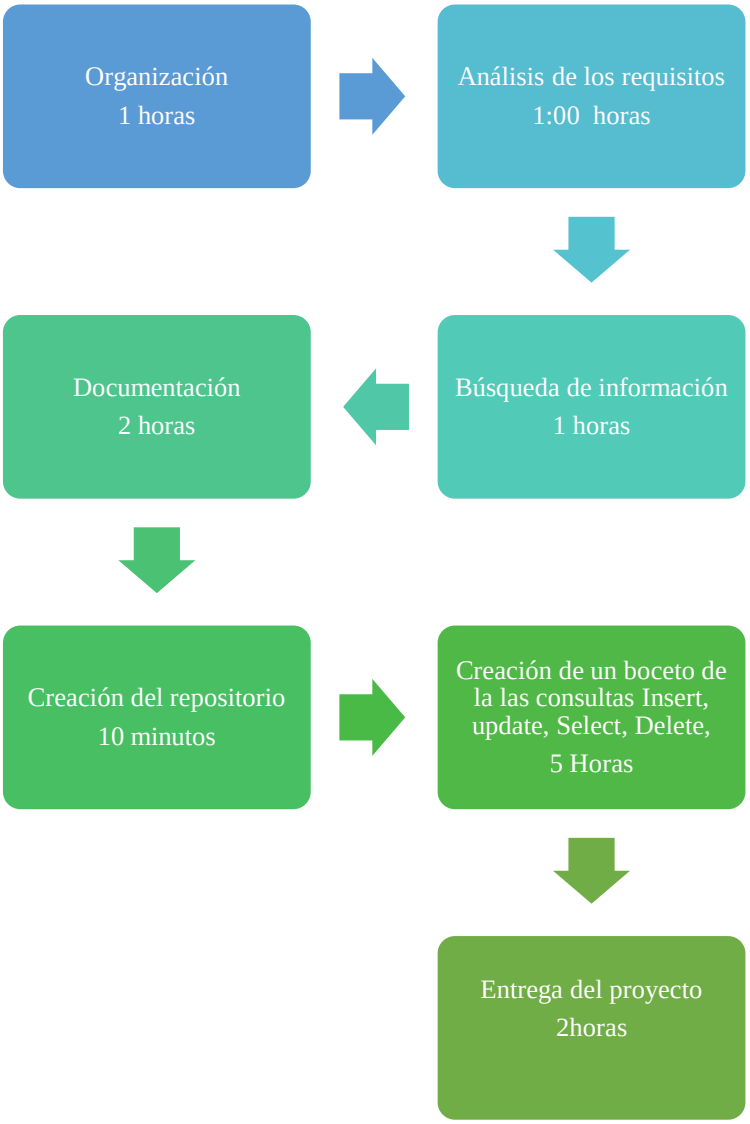
LISTA DE TAREAS A REALIZAR

Task	Priority	Hours	Difficulty
Documentation	High	2,00	High
Organization	High	1,00	High
Pre-search for information	Normal	1,00	Normal
Repository creation	Low	0,15	Low
Documenting Project	Normal	2,00	Normal
Data SQL Query Structure	Normal	1,00	Normal
Sketching the Insert, update, Select, Delete queries, 5 Hours	High	5	High
Project Delivery 2 hours	Normal	2	Normal

We plan to finish the project in 7,15Hours

In which we give 8 more hours to anticipate incidents in the project

Project Calendar Tracking



Quality Metrics

Although the project must adhere to all project *requirements* and *specifications,* there are some conditions that, if properly met, add a sense of quality and robustness to the project itself. These conditions are:

1. HTML and CSS code must be validated by W3C.
2. JavaScript code must be lint-free, as indicated by a trusted JavaScript linter, such as ESLint.
3. The web application must respond.
4. The web application must be compatible with the main browsers on the market:
    - Internet Explorer 11 o superior.
    - Safari in one of its latest versions.
    - Firefox in one of its latest versions.

- Chrome in one of its latest versions.

Risk Documentation

- Project delays.
- Loss or damage to work material

GIT WORKFLOW documentation

- Creating Git Hub https://github.com/robertfox11/MysqlBasics.git
- We make commits of the structure of the main page.
- Probability of a occuring 80%
- Project impact 60%
- Possible alternative (mitigation) Ask colleagues for help
- Chance of it occurring 30%
- Project impact 60%
- Possible alternative (mitigation) Ask colleagues for help
- Not easily finding information related to the project
- Chance of it occurring 30%
- Project impact 60%
- Alternative alternative (mitigation)
- Ask colleagues for help

From the realization of the structure, work continued only on the
"master" branch, through the Workflow "Gitflow".
But information --> https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow

Project tooling

Different tools were used in the development of the project. They are as follows:

1. ***git: A powerful version control system*** that helps track changes in the work tree.

2. ***Visual Studio Code: A code editor*** optimized for creating and debugging modern web applications.
3. ***Google Chrome Developer Tools:*** Used to debug JavaScript code and to test design settings.
4. ***Google Docs:*** Used to write project documentation.
5. ***W3C Validator***– Used to validate HTML and CSS code.
6. ***ESLint***– Used to validate JavaScript code.

Record of lessons learned.

- Making a more detailed documentation improvement
- Class structure in SQL
- Learning advanced sql basic and sql queries.