# Introduction

## Background and Overview

Relational databases are an important component of modern data management and provide methods for effective storage, search, and analysis of the information. This report is purely implemented in MySQL, using a set of simulated tables and concentrating on the SQL queries only, in the context of employees, departments, projects and cities. The purpose is to demonstrate the basic and the advanced way of using different queries to search a structured database for business information.

## Scope

The coverage of this report includes a sequence of activities aimed at showcasing several basic SQL querying abilities. It includes:

- Creation of simple operations that enable the user to retrieve and modify information regarding the employees and departments.

- Application of filter and group by and even advanced conditions of comparison.

- Complex operations that use several tables in order to obtain additional data on projects and assignments of employees.

- Interactions with a city table to review global city information.

## Objectives

The primary objectives of this report are to:

- Emphasize the capability of creating basic SQL statements for selecting and filtering the data stored in relational systems.

- Explain the combined use of SQL join, aggregations, and sub query.

- Outline real-life scenarios where SQL can be used to solve business and data analysis problems.

## Purpose

The need for this report is to show simple and complex ways of performing research in a structured database with business data using SQL queries. In view of this, the report will demonstrate different forms of querying that is applicable in MS SQL with an intention of establishing how data processing can be accelerated and information generated therefrom further used in making appropriate business decisions.

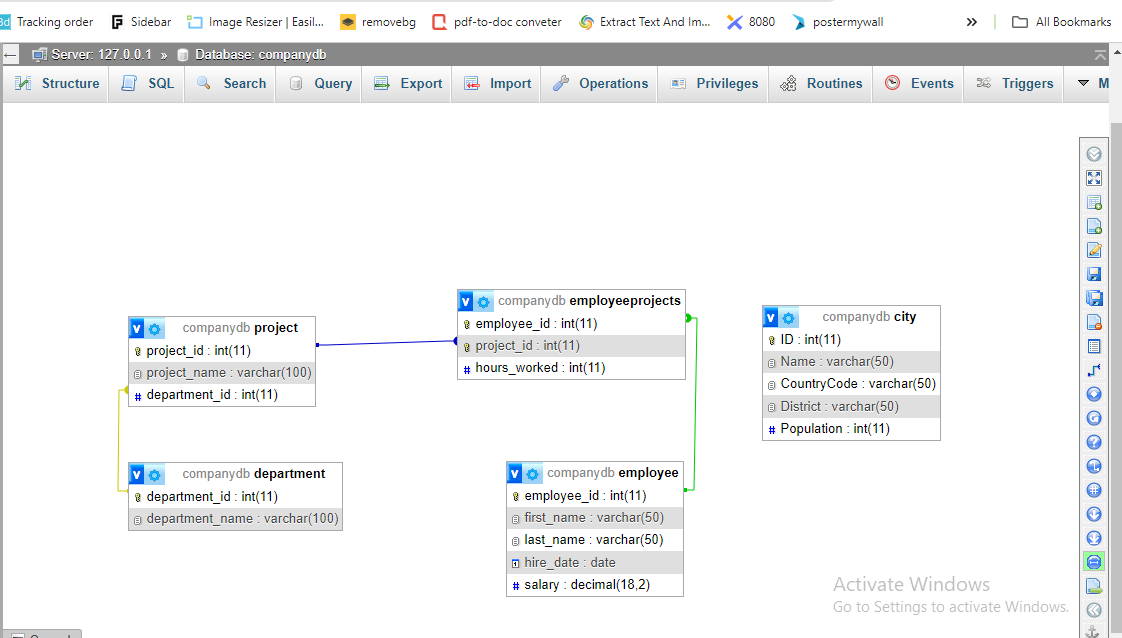
# Task 1: Basic Employee and Department Queries

Task 1 is related to basic employee and department queries where the goal is to present an elegant, simple, and intuitive design for the user along with outstanding performance and scalability.

In this subtask, select and simple SQL commands are shown to pull over and process initial data from the Employee and Department tables. Focus of these queries is to extract particular data, to sum up provided information and to state the key results regarding metrics of employees.

## Entity-Relationship (ER) Diagram

Enterprise to Relationship (ER) diagram is a graphical representation of the data base structure outlay. It depicts various tables that form the database and their associated fields and the connections between the tables. ER diagrams are important since it simplify and clarify the database structure thus enhancing understanding and designing of the database.



Title: Entity-relational diagram illustrating table relationships within the Company dB database.

- Employee: Spans the attributes like `employee\_id`, `first\_name`, `last\_name`, `hire\_date` ,`salary’ and ` department\_id` for the employees.

- Department: Contacts departments, where each department has, for instance, attributes that include `department\_id` and `department\_name`.

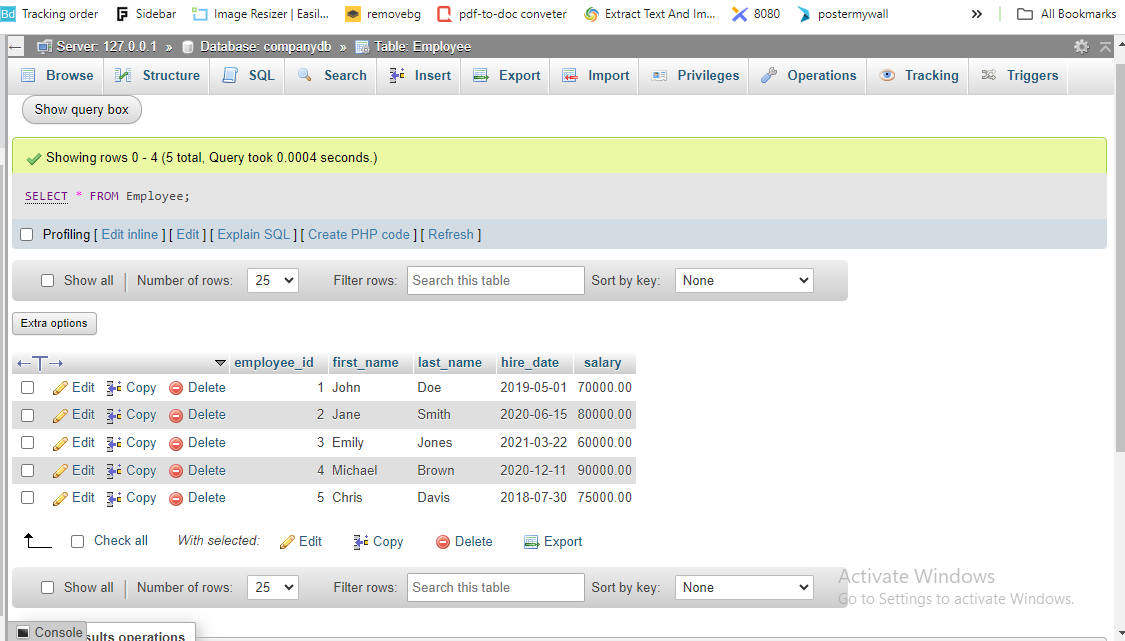
- Project: Retains project data that has features like project ID, project name, and department ID.

- EmployeeProjects: Connects the employees and projects together; It has fields like `employee\_id`, `project\_id`, and `hours\_worked`.

- City: A sample of representing cities with characteristics that include `ID`, `Name`,`CountryCode`, `District`, and `Population`.

## 1. Retrieve all columns for all employees

Query:



Title: Selecting all data from employees table using PHPMyadmin

### Description

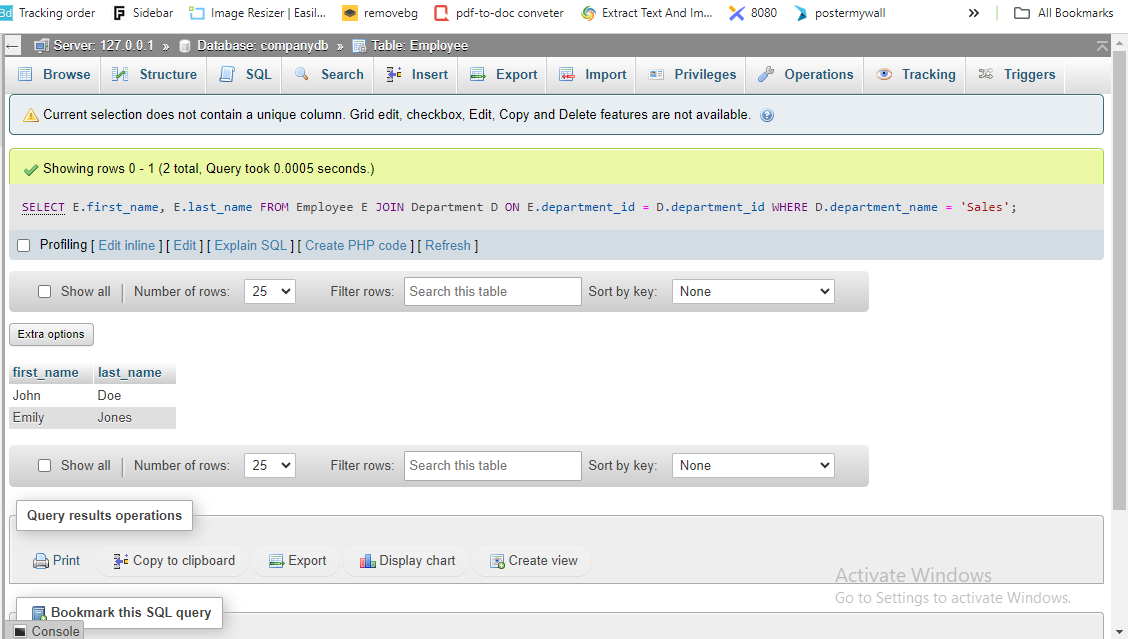
The following query returns all columns of the Employee table and shows every record of all the employees. This operation gives summary information of all the employee information stored in the given table.

### Additional Note

The output set will consist of all attributes for the employees and thus, a detailed analysis of the employee’s particulars will be feasible.

## 2. Get the first and last names of all employees who work in the 'Sales' department

Query:



Title: Getting the first and last names of all employees who work in the 'Sales' department using PHPMyadmin.

### Description

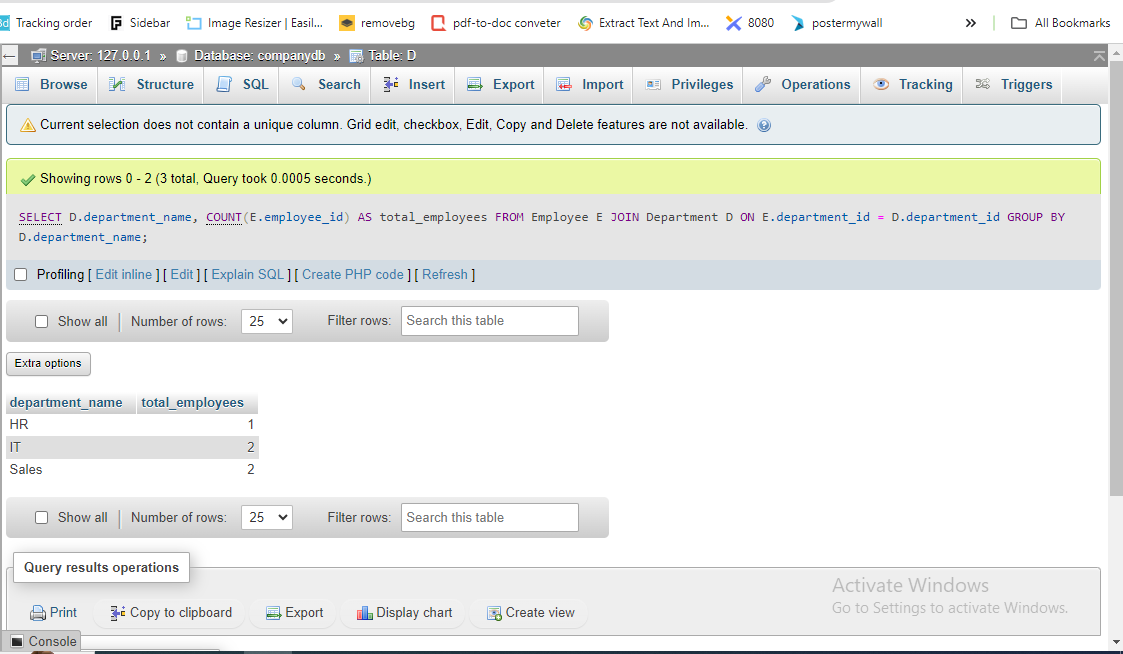
This query performs an inner join between the Employee and Department tables on the field department\_id, then restricts the given result set to the records of the employees belonging to the department ‘Sales’ only and lastly, only the first name and the last name of the employees are selected. To an extent, it gives knowledge on which employees are in the Sales team.

### Additional Note

The query also succeeds in disentangling and enumerating the employees by name that are directly with the Sales department only – handy for focused departmental comparisons.

## 3. List the total number of employees in each department

Query:



Title: Listing the total number of employees in each department using PHPMyadmin.

### Description

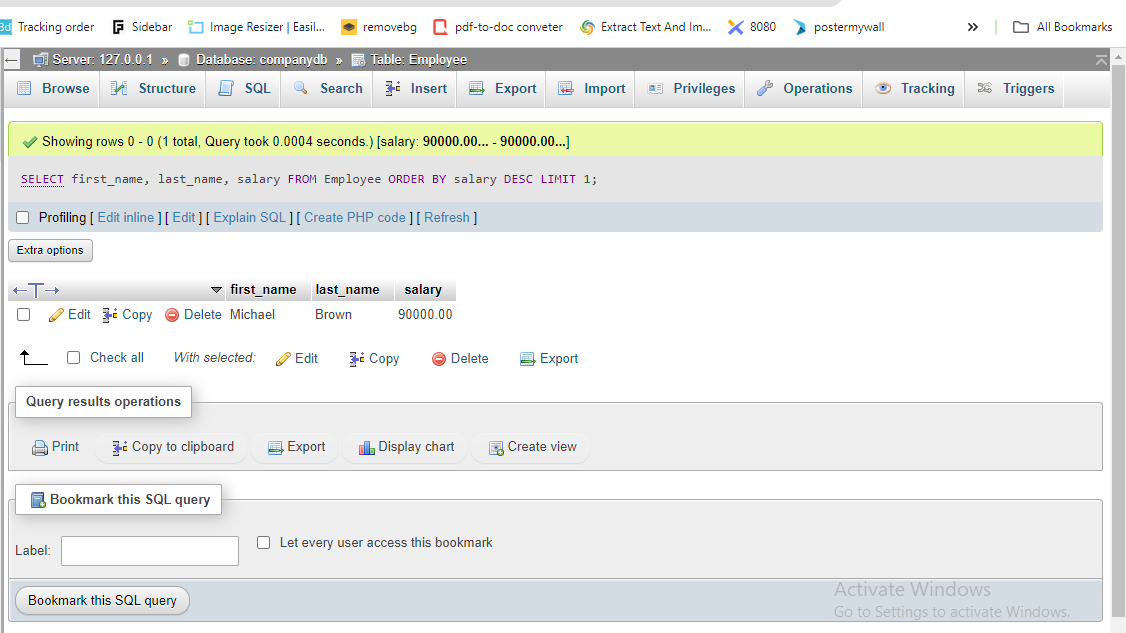
This query first retrieves records on the join of the Employee and Department tables then it sources the records grouped by department\_name where the latter specifies the number of employees in the department. It accumulates numbers of employees for a vision of the departmental summary of employees.

### Additional Note

The aggregation results enable one to signal the possibility of the employees’ distribution across the departments which could be useful in resource forecasting.

### ****4. Find the employee with the highest salary****

**Query:**

****

Title: Finding the employee with the highest salary using PHPMyadmin.

### Description

This query sorts employees by the salary decreasingly and returns only record which the first one is thereby showing the employee with the highest salary. It finds out the employee that has earned the most within the sample data.

### Additional Note

The query analyzes the information from the table by mentioning the maximum possible salary and the specific details of the employee.

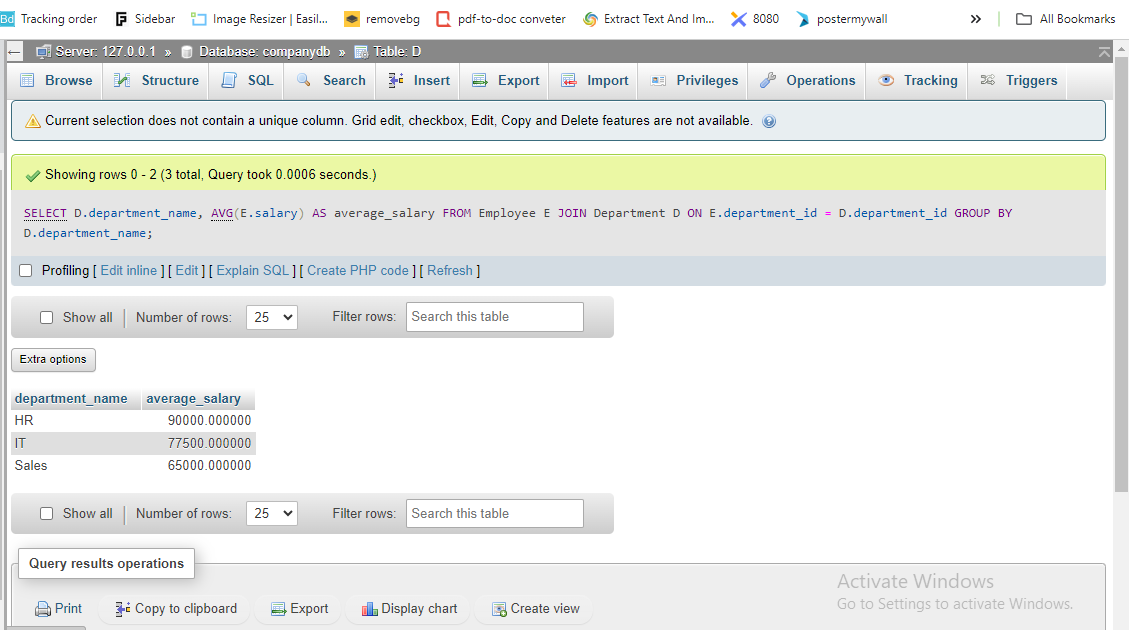
As for Task 1, basic SQL statements were used in order to select and operate data at the tables Employee and Department. This was done by the following query which provides all the information about the employees, that is, all employees and listing of employees in sales department, count of the number of employees in each department and determination of the employee with highest salary. Also, provided with the help of the ER diagram, the nature of the internal database structure and the connections between the entities were easily explained. These elementary searches form the major building blocks for other data mining activities as well as business intelligence processes.

# ****Task 2: Advanced Employee and Department Queries****

This is an activity that entails applying more complex SQL statements in an attempt to gain deeper insights into data about employees and departments. Such queries include those which help one arrive at group total and subtotals, those used to narrow down the result set according to certain characteristics, and those used to recognize people in the establishment based on the projects they are working on.

## ****1. Get the average salary of employees in each department****

**Query:**



### Description

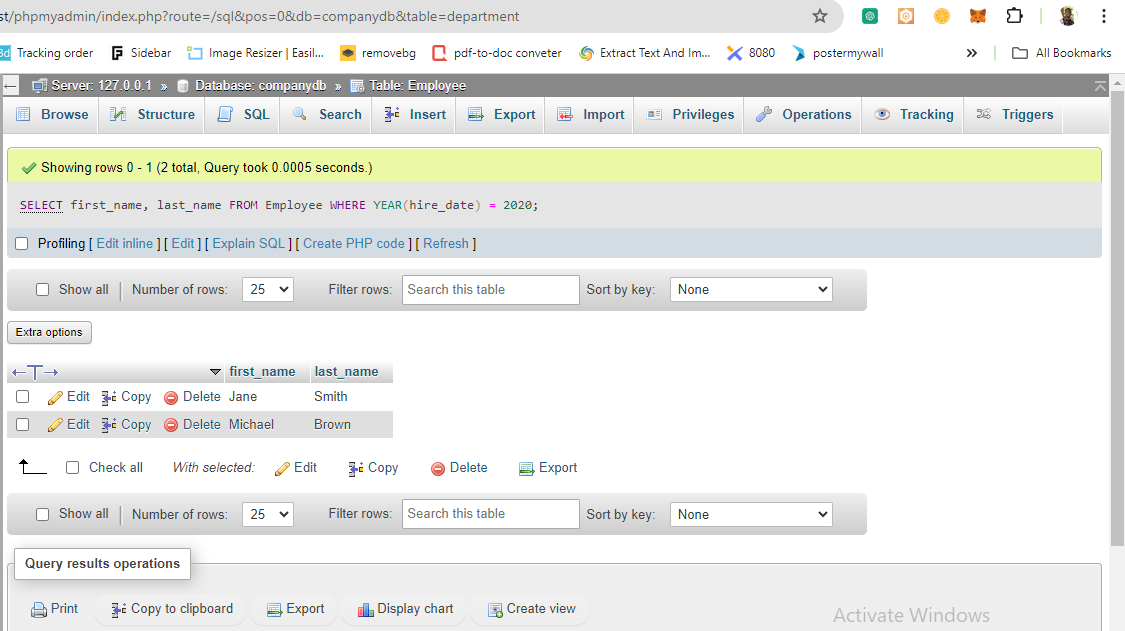
This query generates the mean salary for each of the departments in the organization. It combines the Department table with Employee table on Common field department\_id, then, it group the result by department\_name and at the same time give the average salary in each department.

### Additional Note

The information provided will guide compensation strategies for various departments within an organization and is relevant for budget planning and comparing employees’ salaries.

## ****2. Retrieve the names of employees who were hired in the year 2020****

Query:



### Description

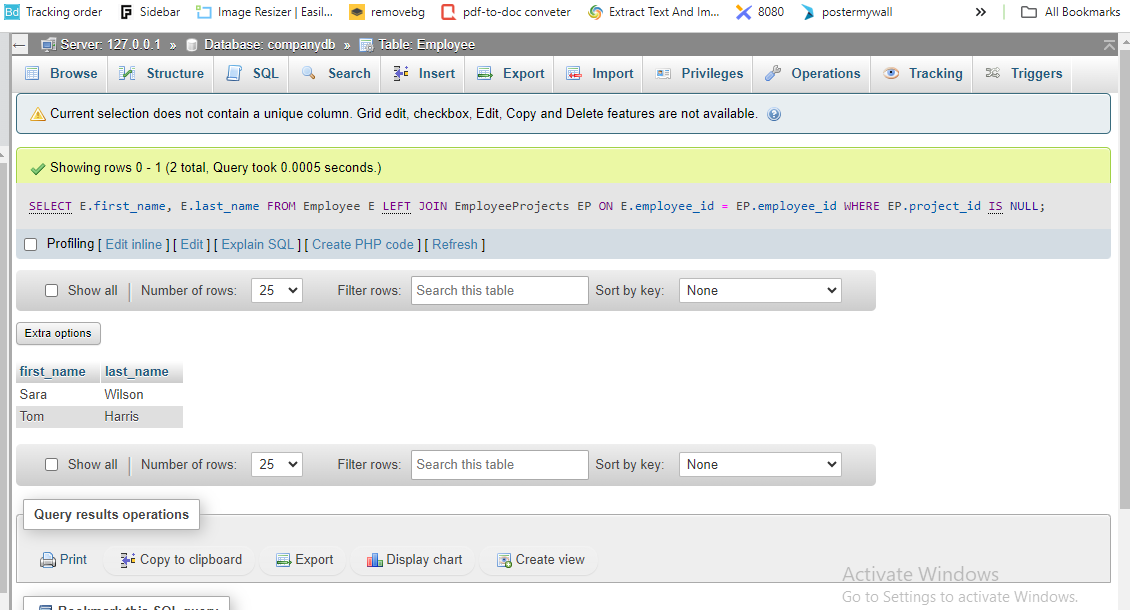
This query seeks to pull out the first and last names of employees that were hired in the year 2020. It gets the Employee table and in the subsequent line it restricts the data based on the hire\_year.

### Additional Note

Knowing which employees joined the company in a particular year might be helpful in reviewing the trends and strategies of staff recruitment, the procedures used in the staff onboarding, or studying the company’s growth in terms of its staff.

## ****3. List all employees who have not been assigned to any project****

Query:



### Description

This query retrieves information of those employees who have not been allocated to any project. It uses the Join operation to combine the data from the Employee and EmployeeProjects tables, and where the table EmployeeProjects is a left join to the Employee table, to select all rows where the project\_id is equal to null.

### Additional Note

This query assists in filtering out employees who might not have any project to work on at the moment, which is useful in case of resource management and project scheduling.

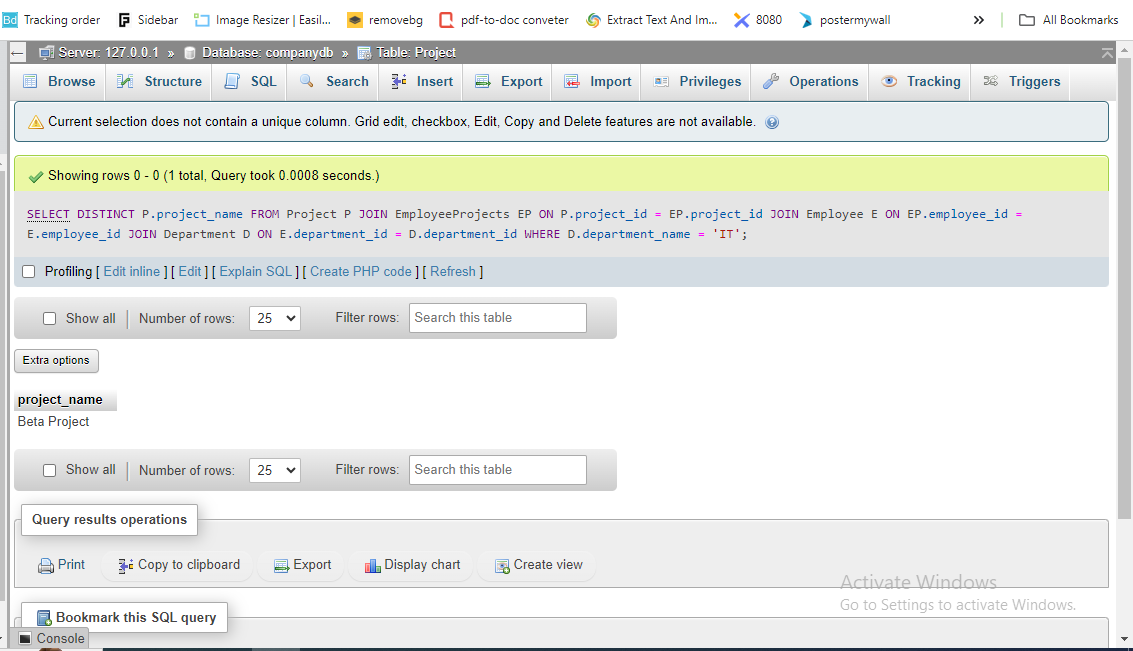
More specifically, in Task 2, SQL query commands were expanded and used to address more sophisticated analysis related to the known data on employees and departments. The queries generated averages in terms of salary per department, determined employees joined in a given year, and those employees who are not attached to any project. More so, these elaborate queries offer a richer dataset to probe into the organizational workforce and the nature of its projects implying better decision-making.

# ****Task 3: Complex Queries Involving Projects****

This task is also based on the combination of different tables and sharp manners for working with them with the help of sub queries and other functions. These queries’ objective is to obtain more detailed information about the connections employees/departments and projects.

## ****1. Find the names of projects that involve employees from the 'IT' department****

**Query:**



### Description

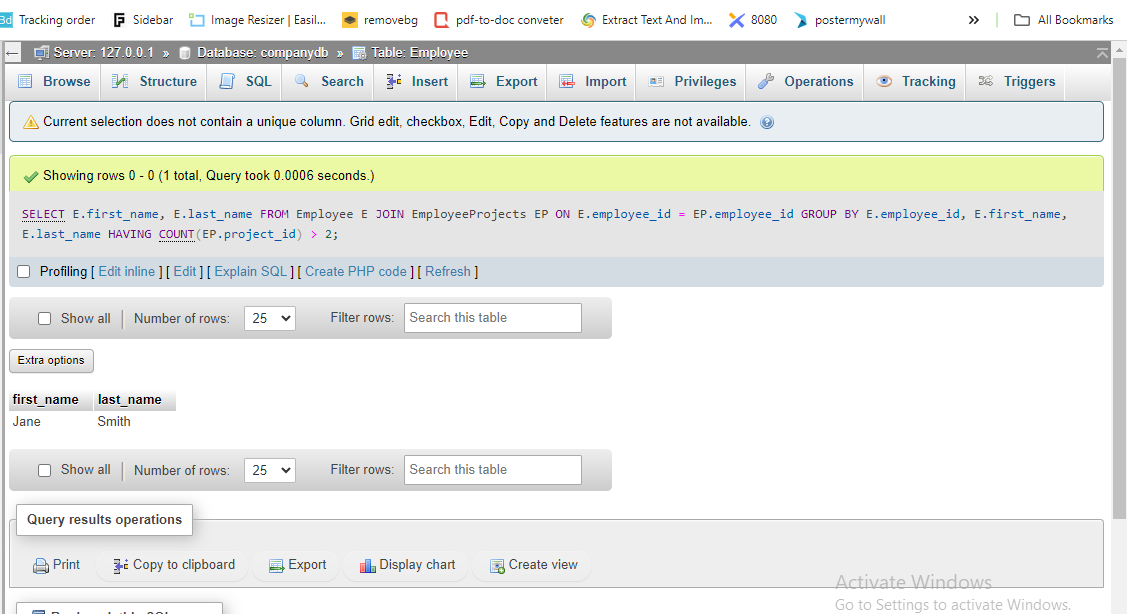
This query allows the identification of project names known to engage employees of the ‘IT’ department. This also links to the `Project`, `EmployeeProjects`, `Employee`, and `Department` tables and adds a condition for the department name.

### Additional Note:

Specifying projects by departments assists in figuring out the department’s commitments to various projects and thus improves the management of resources.

## ****2. Retrieve the list of employees who work on more than 2 projects****

**Query:**



### Description

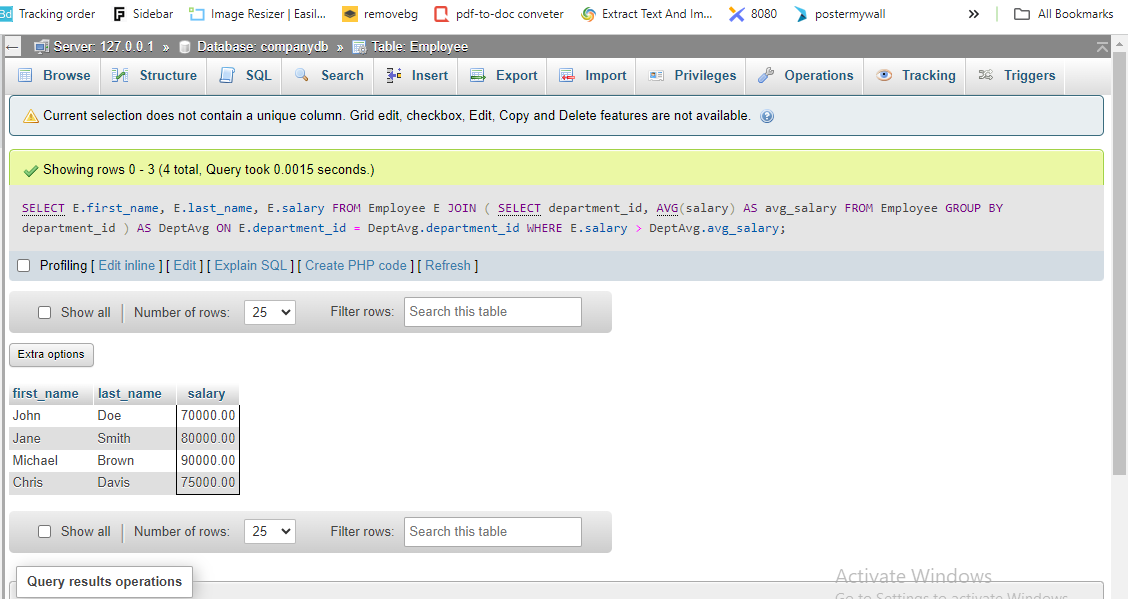
This query returns the set of employees, which have more than 2 projects allocated to them. Thus, it connects the `Employee` and `EmployeeProjects` tables, aggregates the results by the number of projects assigned to each employee, and the `HAVING` clause is added to filter the final output.

### Additional Note

By identifying which of the employees are heavily connected to other employees, project managers can effectively manage on workload and avoid inflating an employee’s workload.

## 3. Get the names and salaries of employees who earn more than the average salary of their department

Query:



### Description

This query lists all the employees’ name and their salary if the employee’s salary is greater than the average of his department. It employs the sub query to determine the average of salary per department and then applies a join with the `Employee` table with the purpose of filtering employees through their salary.

### Additional Note

It is useful to search for those workers who have wages above the averaged department rate as it helps to consider various compensation relationships and to define possible tendencies in the salaries increase.

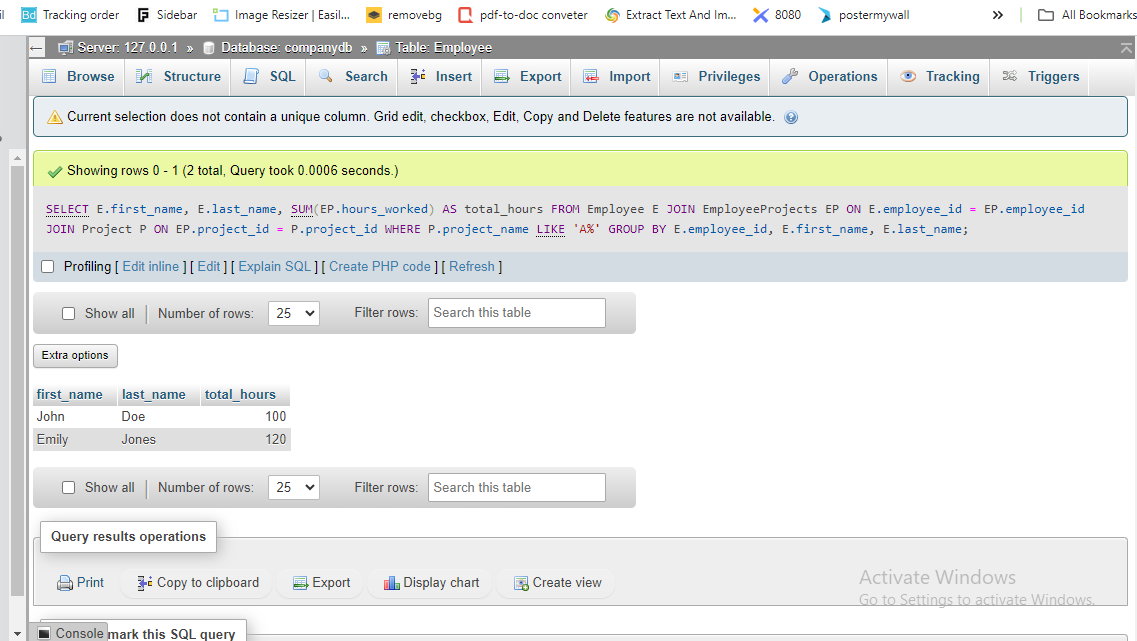
In Task 3, to analyze the interconnection between the employee, department, and project complex SQL command was applied. The queries Just outlined projects that included the ‘IT’ department; listed employees undertake more than two projects; and those employees earning more than the average FY salary in their respective departments. These latter types of complex queries provide capabilities for insight into organizational phenomena and allow for better decision-making in regards to project management and compensation policy.

# Task 4: Queries Involving Employee Projects

This task requires the ability to construct and run SQL statements that analyze the correlation between employees and projects. It involves questions that are meant to provide a detailed and comprehensive insight into employees’ contribution to the projects, the relationship between projects and departments and salary structure of the company in general.

## ****1. List the names of employees and the total hours they worked on projects where the project name starts with 'A'****

**Query:**



### Description

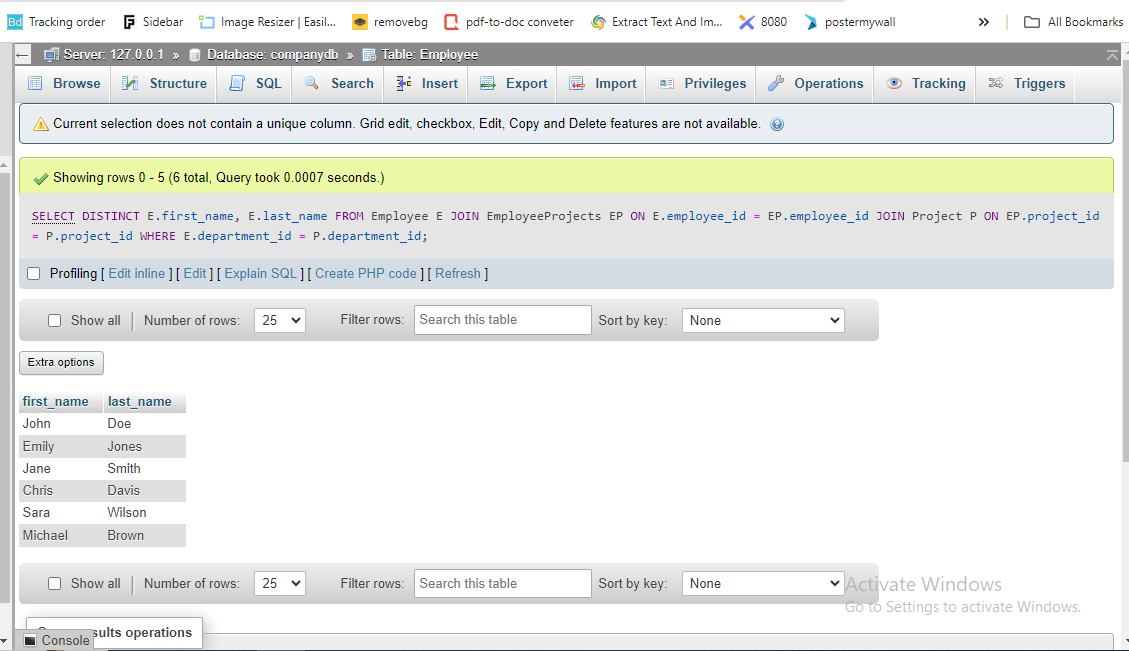
This query executes with the aim of bringing forth the names of employees and total working hours done on projects which have the Project Name beginning with the alphabet ‘A’. It is a join query between the `Employee`, `EmployeeProjects`, and `Project` tables where the client uses the `LIKE` operator to narrow down on projects.

### Additional Note

This query proves more useful in defining the contribution of the employees in certain projects as it assists in appraisals and project development.

## ****2. Retrieve the names of employees who work on projects within their own department****

**Query:**



### Description

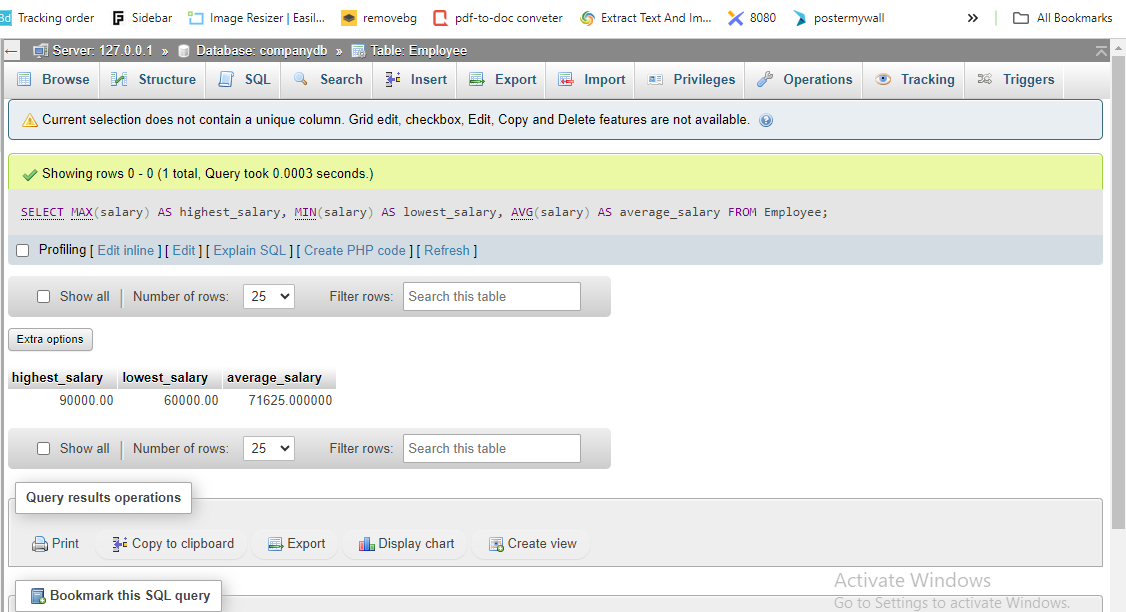
Here is the query, which returns the names of the employees who are engaged in the projects within the department that employs them. This table connects the `Employee` and `Project` tables and only selects data based on the department match.

### Additional Note

Frequently used types of this query shed light on how departments work together and show which employees are concerned with the projects of their department only.

## ****3. Get the highest, lowest, and average salary in the company****

**Query:**



### Description

This query selects the maximum, minimum, and average of the salary in the company. It applies group by operations on the `salary’ field of the `Employee’ table.

### Additional Note

By evaluating salary data one can get to know the salary level within the organization, which in turn assists in decisions concerning a change of pay structures.

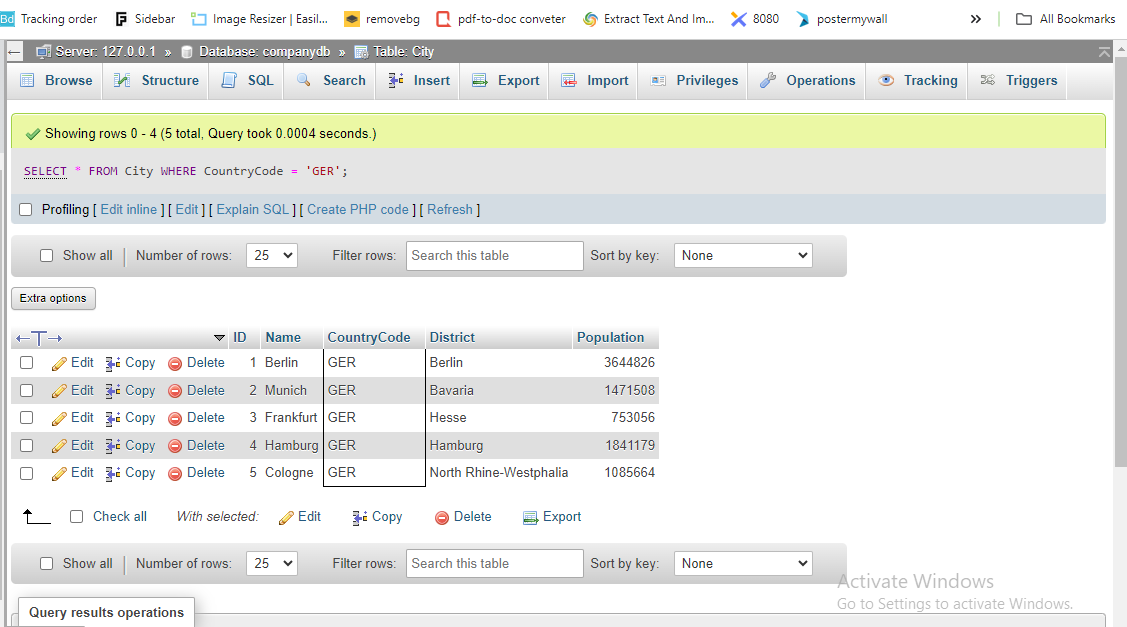
When analyzing the results of Task 4, it was possible to gain information about the actions of employees in relation to particular projects, and their distribution within departments as well as the general salary survey results in the company. These advanced queries improve analysis and help in making strategic decisions on project management and compensations.

# Task 5: City Table Queries

This task is related to writing and running SQL queries to analyze data stored in the City table. The queries’ purpose is to search for information about cities only with some requirements.

## ****1. Write a SQL query that selects all columns for cities in Germany****

**Query:**



### Description

This query pulls out all columns for the reference of cities located in Germany. From the results, it filters those that only have the value of ‘GER’ for the `CountryCode` field.

### Additional Note

This query is helpful in the extraction of detailed information concerning cities situated in Germany to enable geographical and demographical analysis.

The specifics for Task 5 involved using the provided query to extract information about all the cities situated in Germany by using all the columns for ‘CountryCode’ equal to ‘GER’. This makes it easy to evaluate and understand all the geographic and demographic factors peculiar to the Germany market.

# Conclusion

## Summary

This report provided a detailed description of how to use the SQL commands and the MySQL database to interrogate data stored in different tables. Here are some specific operations that we did use or would use with the given tables: pulling information out of the Employee, Department, and City tables; and updating/selecting data from the EmployeeProjects and Project tables. Every assignment focused on different forms of SQL, from the simplest query to more difficult and complicated joins and aggregations.

## Reflection

This report gave and extensive coverage of what SQL is, its use in managing the database and analyzing data. This activity of writing and implementing these queries was indeed very useful to recap the basic relational database mechanism as well as the capability of SQL to seek specific information out of tabular data. Each of them was following the previous one which step by step introduced the advanced level of the real SQL application starting from basic query and finishing by the most complex data operation which demonstrated the wide range of uses of the SQL in different aspects of business.