# Basics Database Design using MySql

# 1. Introduction

### 1. 1. Background

Database systems are considered as important subassemblies of modern information technology that is aimed to help manage, store or search for large amounts of data. They offer a clear pathway of managing data while guaranteeing that the information is well managed, easily retrievable, and safe. This is achieved by means of Database Management Systems (DBMS) which provides the instruments for defining data and dealing with it through various operations.

SQL is a language founded for this purpose of querying databases, and it is one of the most useful languages available. It let the user perform such operations as search, add, modify and delete records from the data. SQL plays a very significant role in business analysis and decision-making because users employ it to extract useful information. Businesses benefit from using SQL as it helps provide information, comparisons and further positive decision-making for their business processes and developmental patterns.

### 1. 2. Objectives

The primary objectives of this report are as follows:

1. Understanding Database Design Concepts: For a deeper investigation of the principles that shape the conception of a database system such as normalization, creation of a schema and other aspects concerning the relations between separate data elements.

2. Writing Effective SQL Queries: To show the capacity in building SQL queries that can be used in fetching and analyzing information on a given database. This includes; simple search and search with more complex operators.

3. Analyzing Data Using SQL: To be able to translate SQL queries into simple datasets and to apply it on real data in order to gain results for BI analyses and operational decisions.

### 1. 3. Scope

The concentration of this examine will be on comparing certain databases and tables that are mostly known as `Employee`, `Department`, `Projects`, and `EmployeeProjects`. The scope encompasses various types of SQL queries designed to:

- Get data for an application that resides in different tables and format it.

- Provide the results of more general calculations that will yield more useful information out of the raw, unprocessed data.

- Answer concrete business issues that are linked with employee matters, departments’ performance, and their participation in projects.

Queries described here will include simple SQL database questions to more advanced questions that may involve operations on two or more tables and/or including SQL functions. This approach is within the belief that introducing SQL will be easier in this comprehensive approach to show how it is used in business and the role of effective design in the management of data.

# 2. Database Design Concepts

## 2. 1. Database Basics

In physical Database, data is located in tables which are in turn divided into rows and columns. Every table should correspond to a single subject, for example, an employee or a certain department. The columns establish the specific features of the entity, while the rows contain concrete data entries. Keys are used to link tables in order to maintain integrity of the data.

## 2. 2. Normalization

Normalization is the method of structuring a relational database to reduce redundancy as well as dependencies. It refers to breaking of big tables into smaller but relatively related tables, and establishing relations on the partitioned tables on databases. The main objectives of normalization are data redundancy removal, accuracy of information, and enhancement of the work of the database.

## 2. 3. Database Design Approaches

Various approaches to database design include:

- Entity-Relationship Modeling (ER Modeling): In this approach, data entities and their relationship are mapped and illustrated using what is termed as ER diagrams.

- Schema Design: This requires the identification of how the database will be formatted with consideration to tables, the columns in each of the tables and how they are related.

Its ultimate goal is to design a sound and efficient database structure since both approaches are focused on an organization’s data and data analysis.

# 3. SQL Querying Basics

## 3. 1. Introduction to SQL

SQL represents the standard language for the manipulation of the management of the relational databases. Through this module it is possible to manipulate records in the databases, for example, select records, modify record, add record, and/or delete record. Being a structured language, SQL is now central to database management since it offers a dependable means of dealing with data and harvesting valuable information that will assist companies perform optimally and make sound decisions.

## 3. 2. Basic SQL Queries

Fundamental SQL commands include:

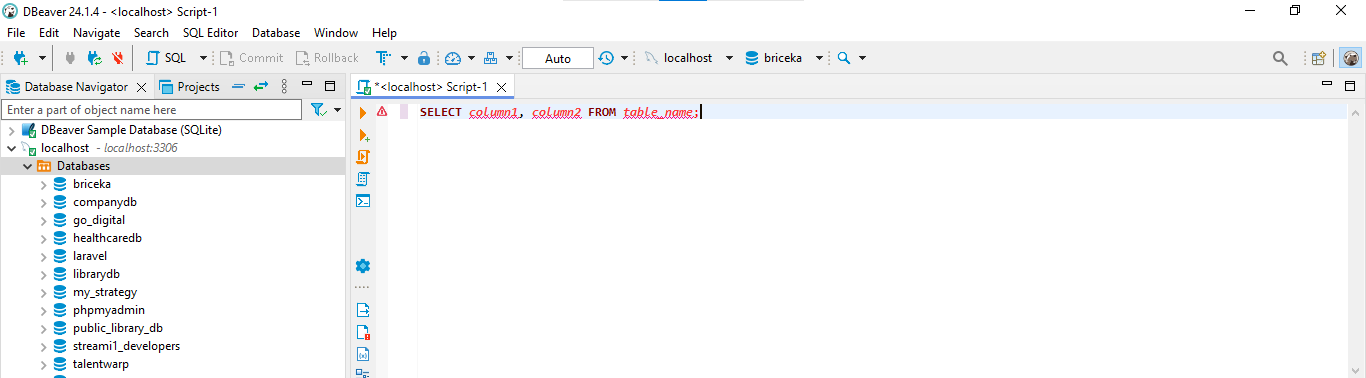
1. **SELECT**: This operation is the act of getting information from one or more tables. 

Figure : Select SQL query that retrieves column1 and column2 for all records in the table\_name table

This select statement retrieves column1 and column2 for all records in the table\_name table.

1. **INSERT**: Creates a record that contains new entries to be incorporated in any table.

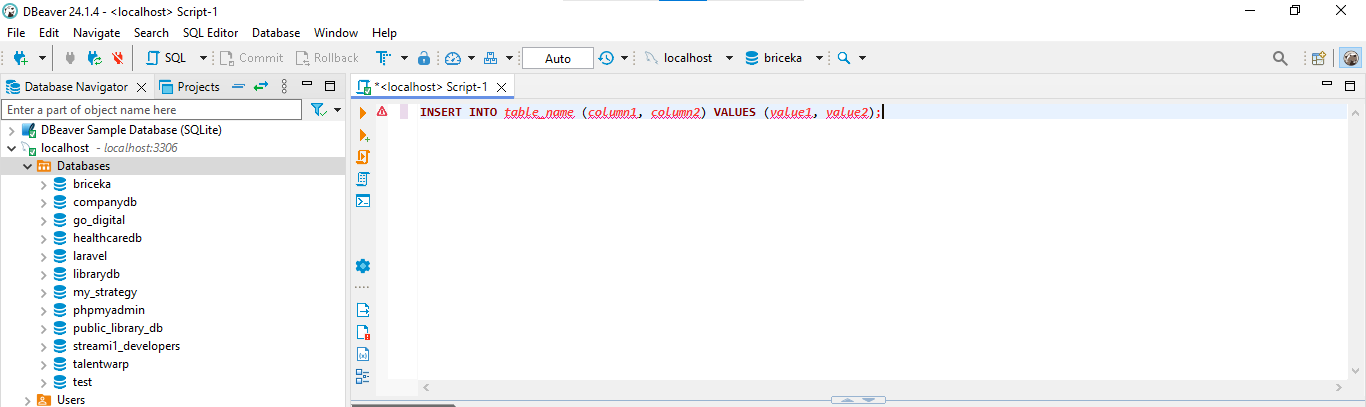


Figure : SQL query that creates a record that contains new entries to be incorporated in any table

1. **UPDATE**: Modifies existing records in a table.

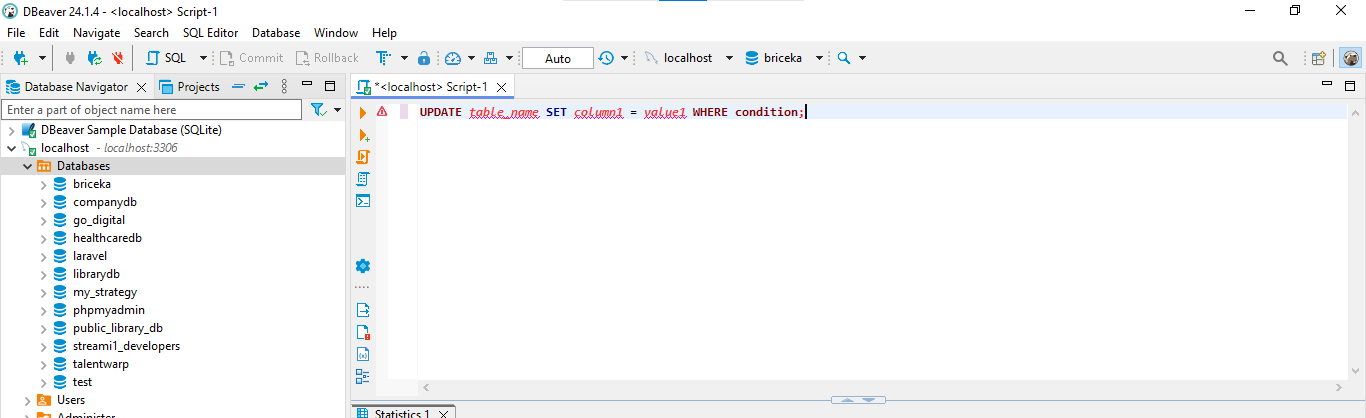


Figure : SQL query that modifies existing records in a table

1. **DELETE**: Removes records from a table.

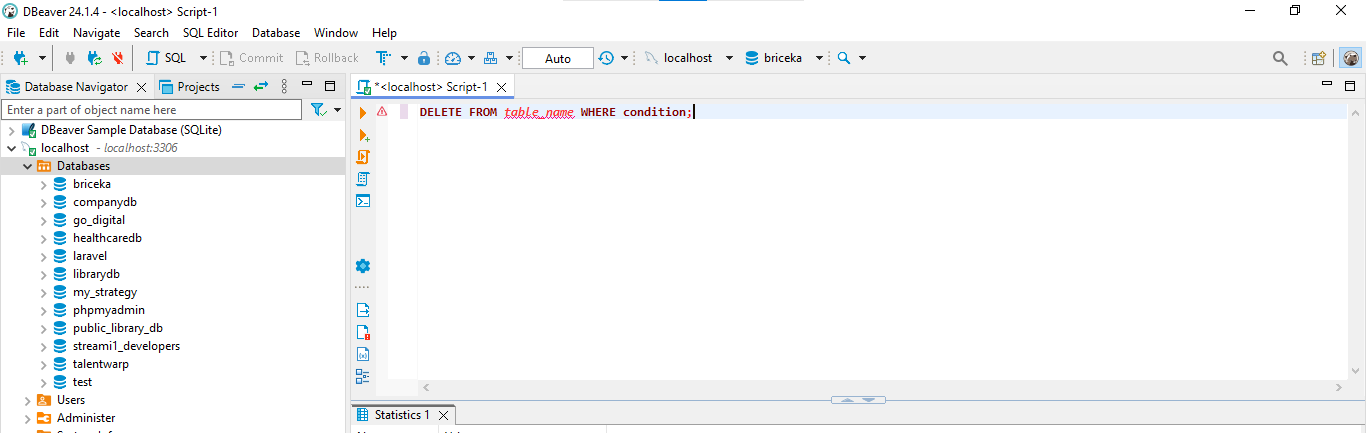


Figure : SQL query that removes records from a table.

## 3. 3. Advanced SQL Queries

More complex SQL queries include:

1. **Joins**: Join two or more tables to produce a new table by pulling together rows from the table that is being joined with one or more other tables that have a related column.

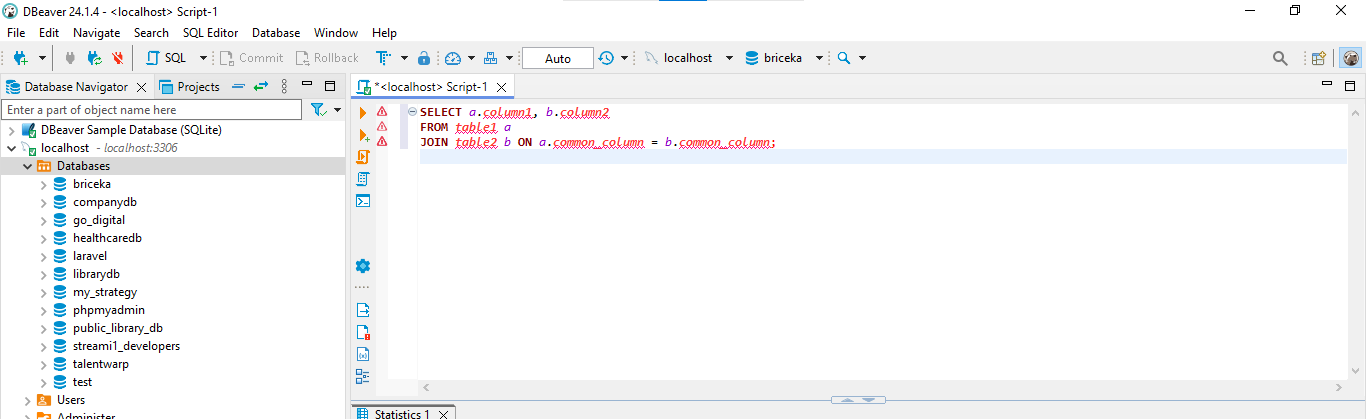


Figure : SQL query that join two or more tables to produce a new table

1. **Sub queries**: A query nested within another query.

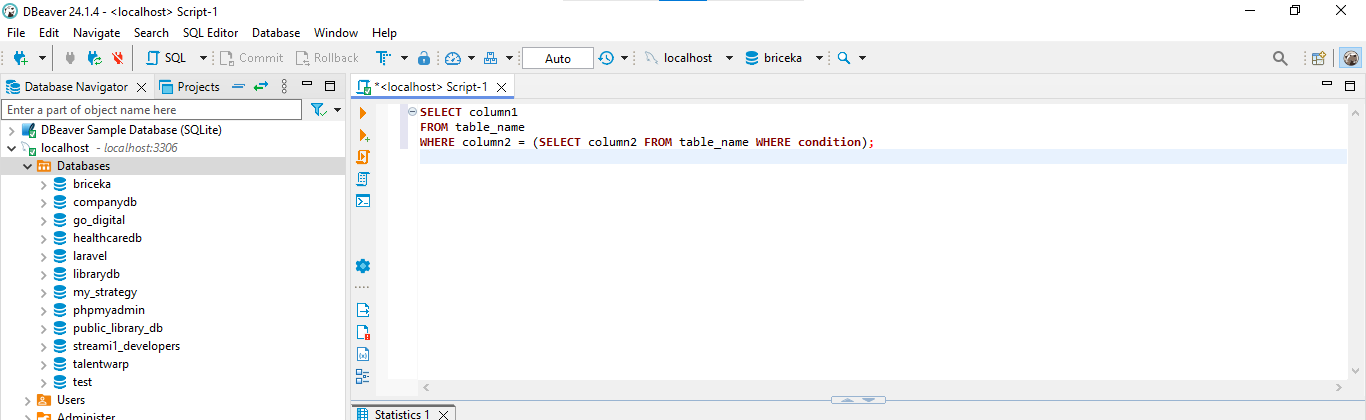


Figure : SQL query nested within another query

1. **Aggregate Functions**: Make operation on a set of values and give one value as result (e. g., `SUM`, `AVG`, `COUNT`).

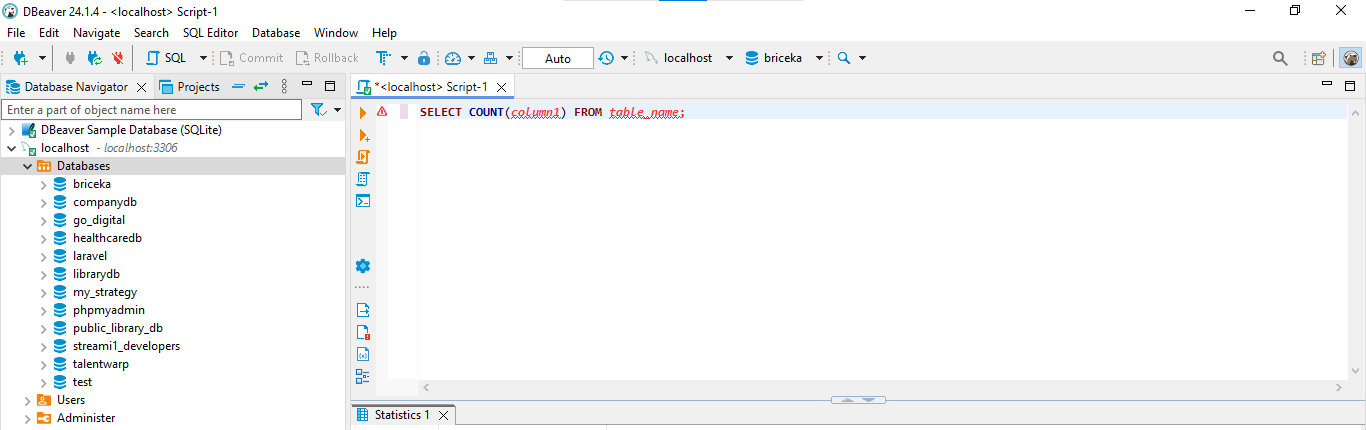


Figure : SQL query that perform calculations on a set of values and return a single value

## 3. 4. Entity-Relationship Diagram (ERD)

An ERD is a type of diagram, which illustrates the database with the results of the entities (tables) and relations between them. For the purposes of this report, the ERD will include the following tables:

- **Employee** (`id`, `first name`, `last name`, `date hired`, `amount earned`, `department id`)

- **Department** (`department\_id`, `department\_name`)

- **Projects** (`pid`, `project\_title`, `department\_id`)

- **EmployeeProjects** (`id\_empleado`, `id\_proyecto`, `horas\_trabajadas`)

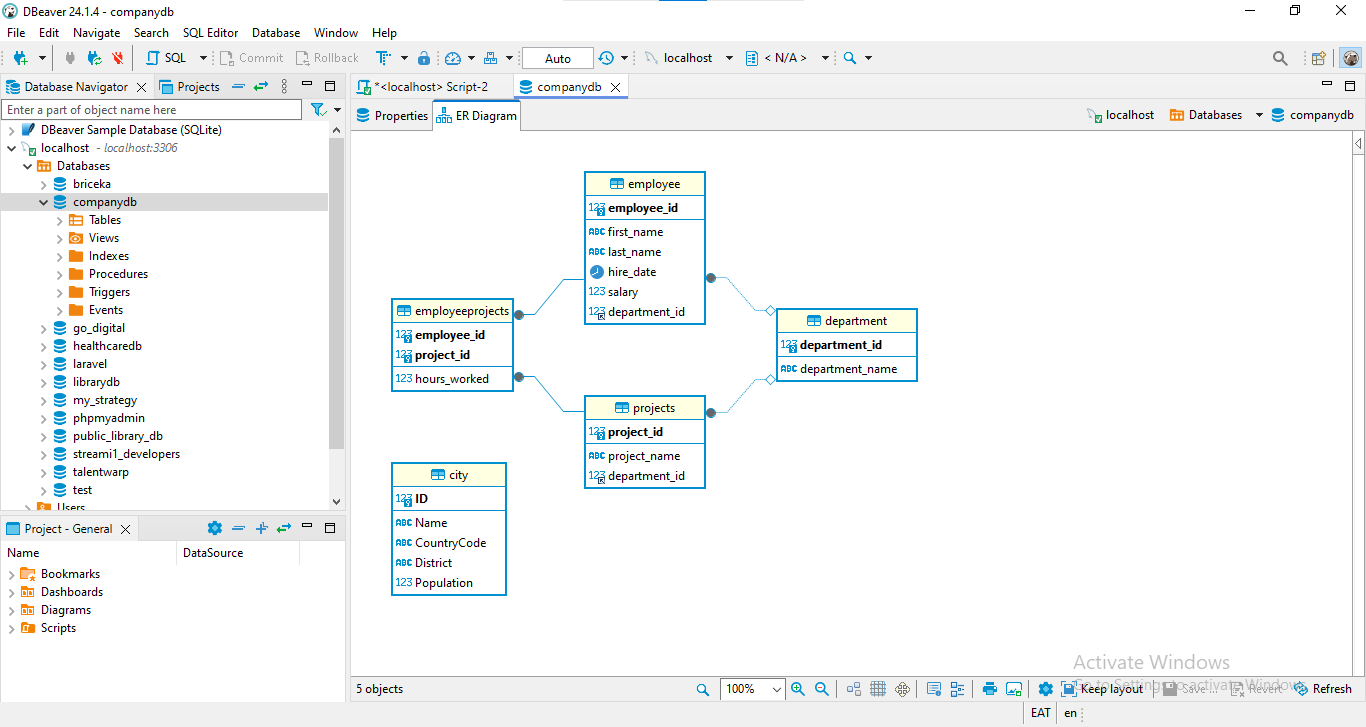


Figure :ER diagram for the CompanyDb database

These tables and their interconnectivity will be used in SQL query and data analysis that will follow in other sections of this paper.

# 4. SQL Query Implementation

In this chapter, different SQL queries will be executed in order to explain concepts and techniques regarding querying of a relational database. The operations are graded from simple to more complicated search in multiple tables, with references to basic spatial operations. The queries will be implemented on the data base tables names Employee, Department , Projects and EmployeeProjects and will include real life business examples in explaining how SQL operations are done and where they can be used.

## 4.1. Database and Sample Data

The database will be created and sample data will be inserted for the project.

Creating database:

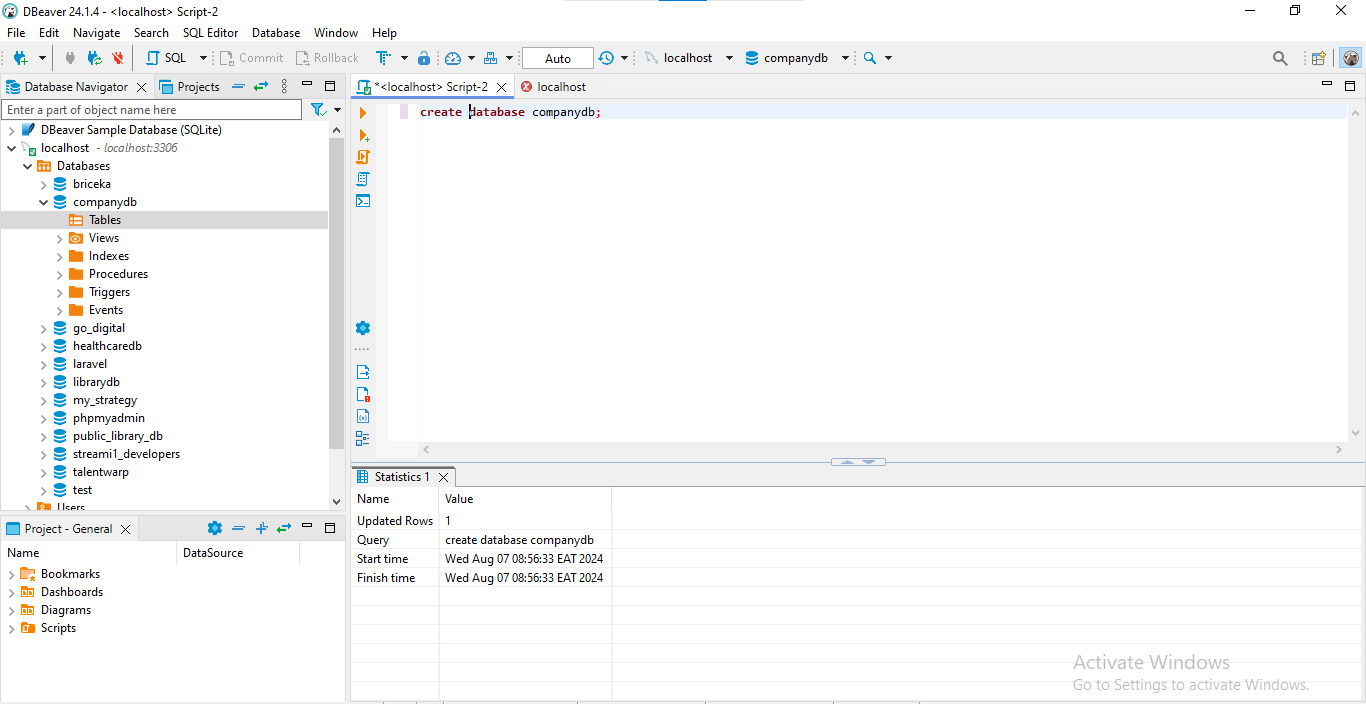


Figure : SQL query to create companydb database

## 4.2 Creating Tables:

1. Employee

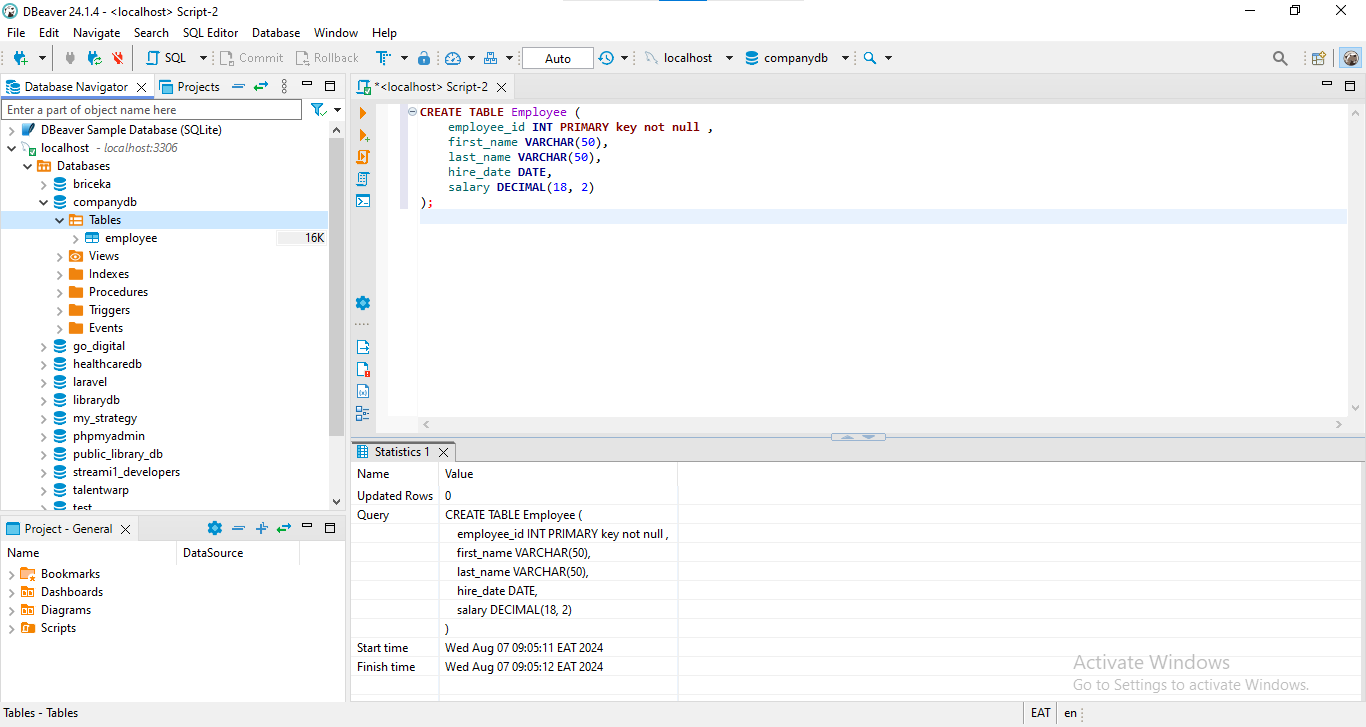


Figure : SQL query to create employee table using DBeaver MYSQL client

1. Department

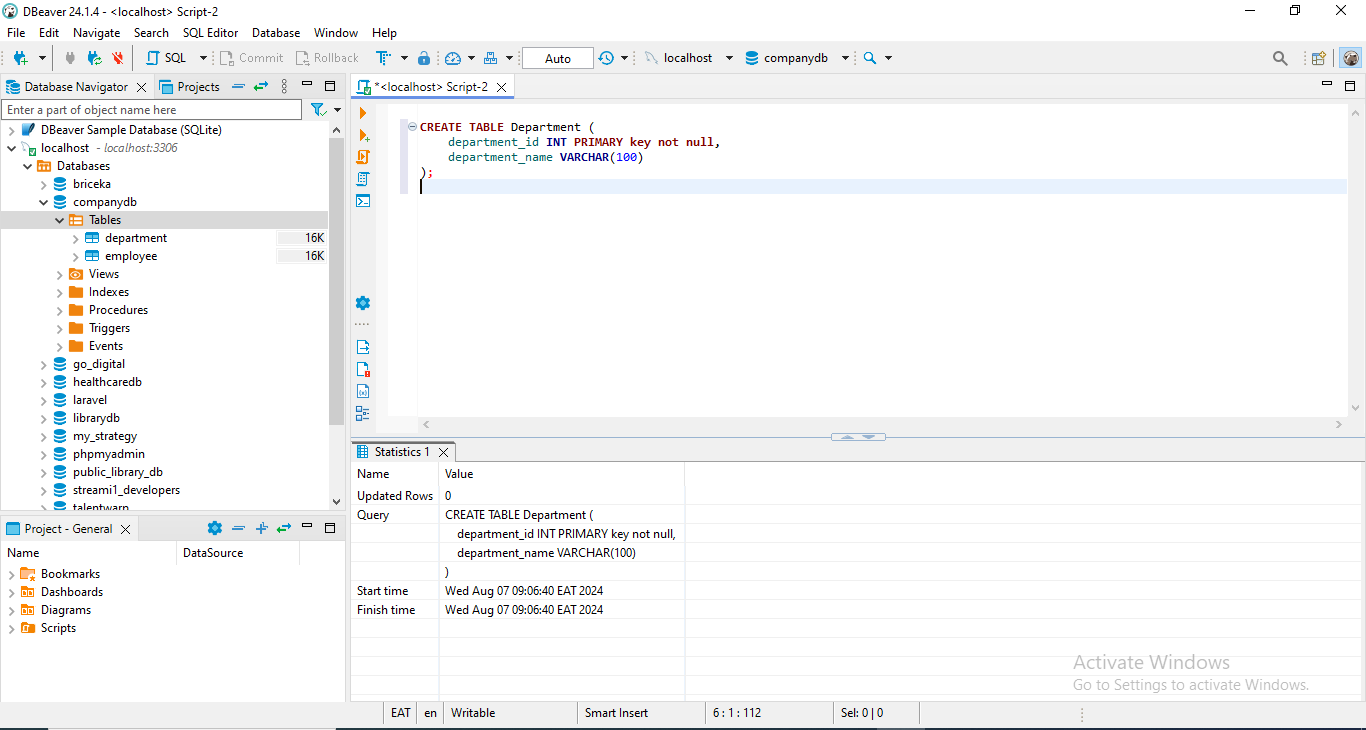


Figure : SQL query to create department table using DBeaver MYSQL client

1. Projects

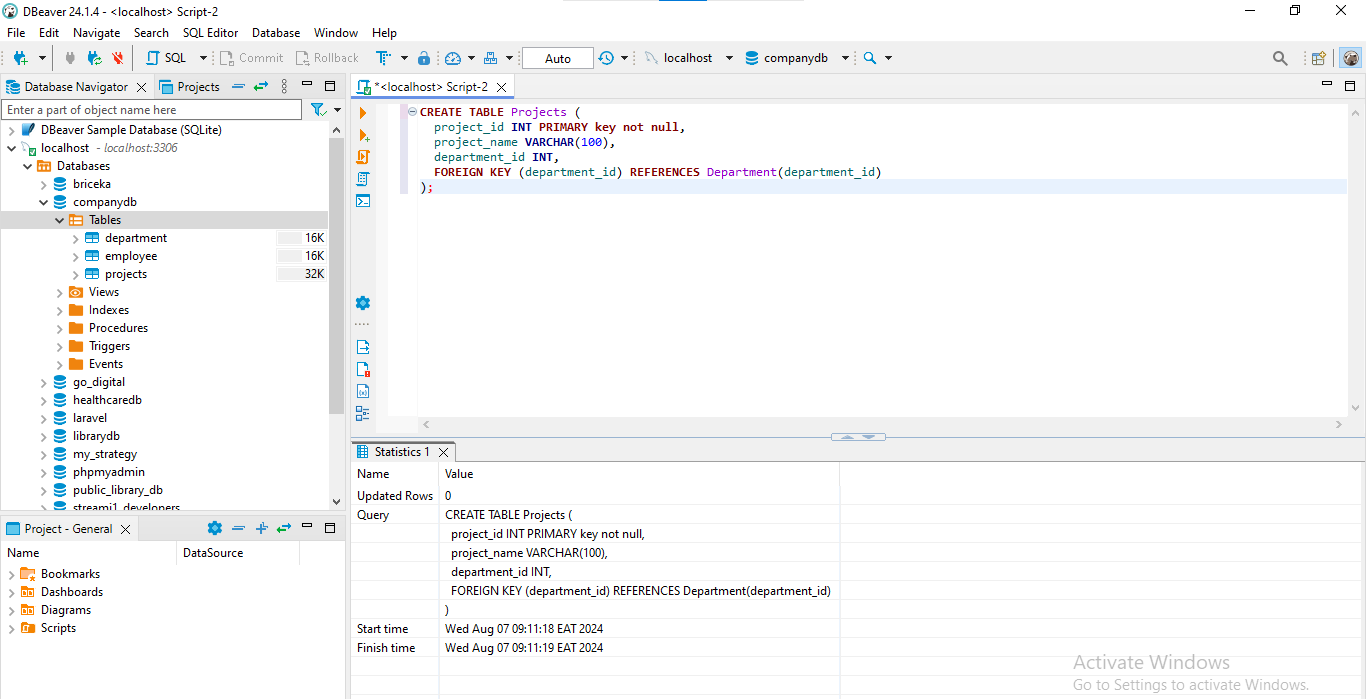


Figure : SQL query to create projects table using DBeaver MYSQL client

1. Employee Projects

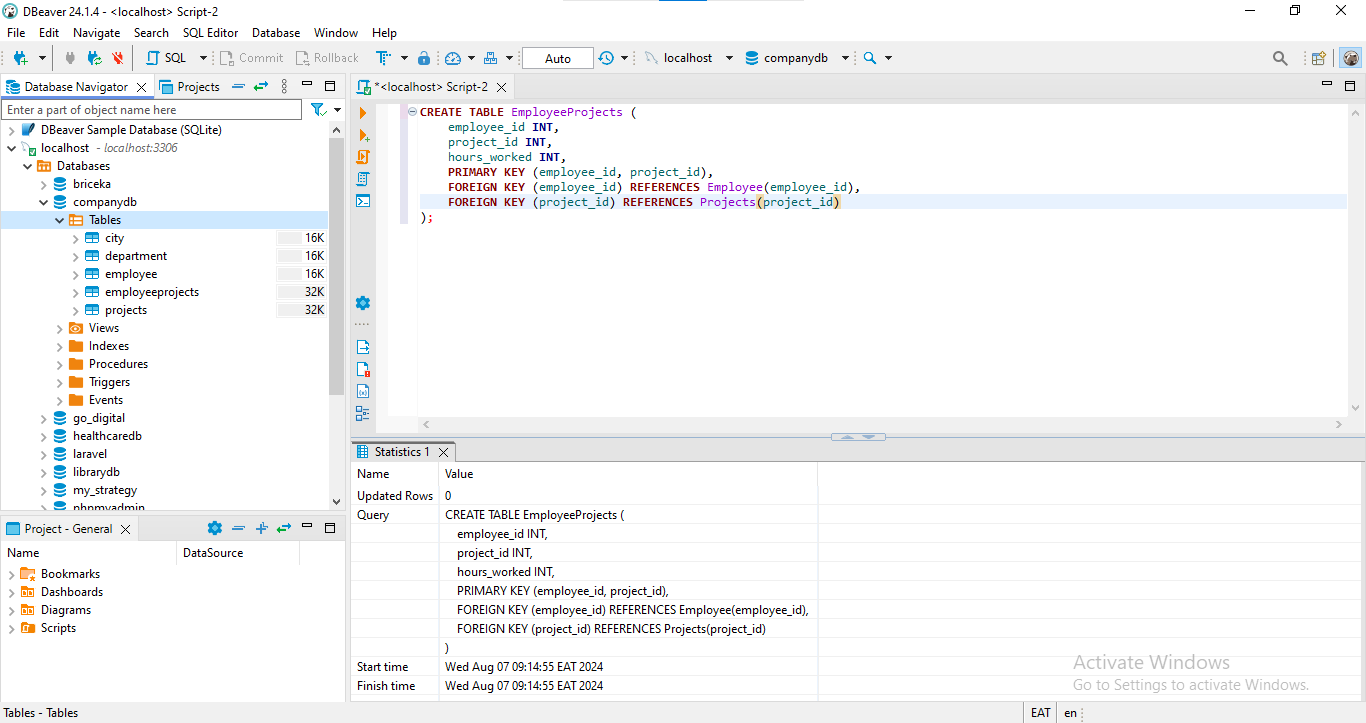


Figure : SQL query to create EmployeeProjects table using DBeaver MYSQL client

1. City

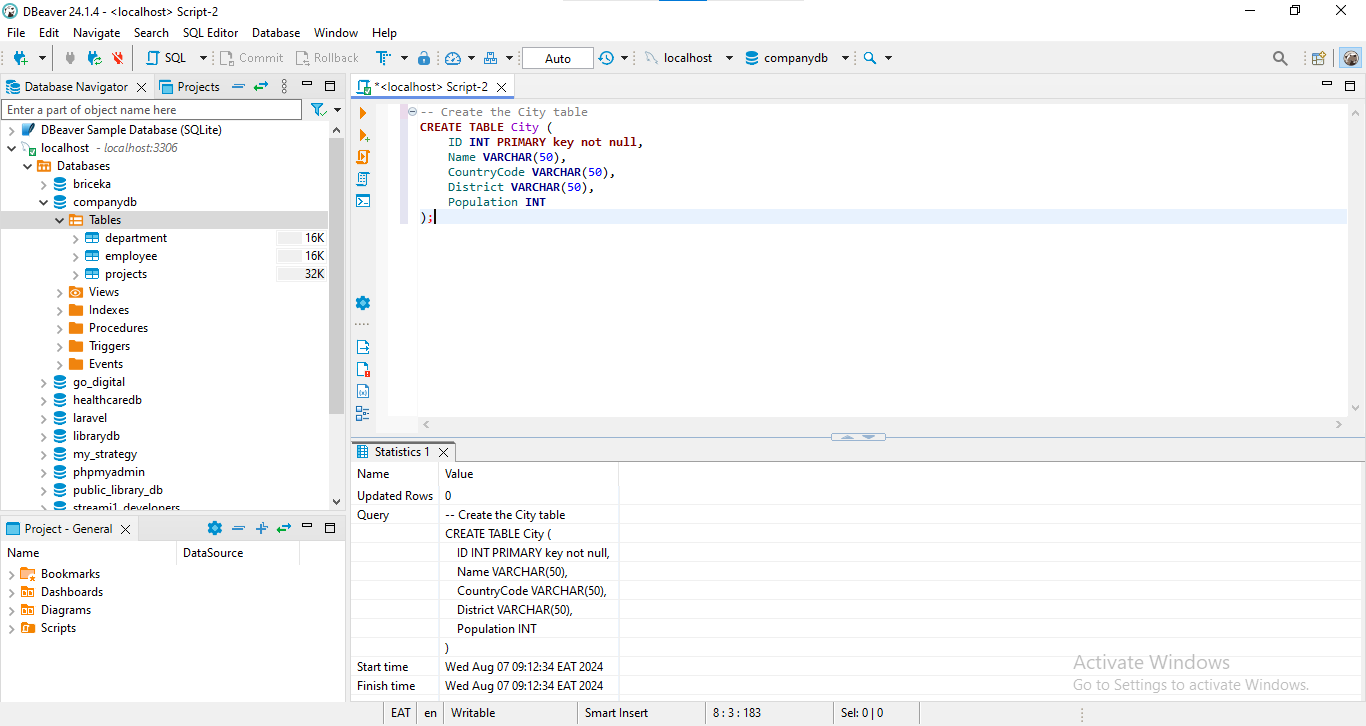


Figure : SQL query to create city table using DBeaver MYSQL client

## ****4.3 Inserting Sample Data:****

1. Employee

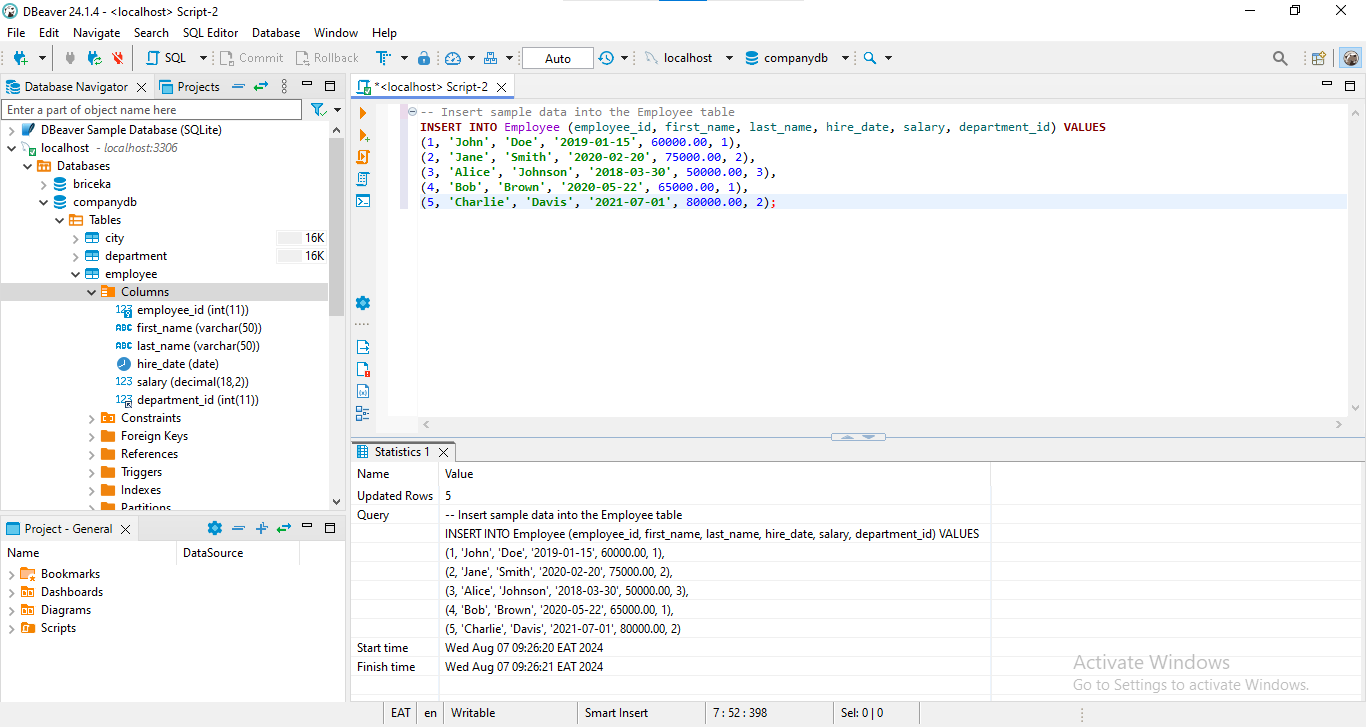


Figure : SQL query to insert data into employee table using DBeaver MYSQL client

2. Departments

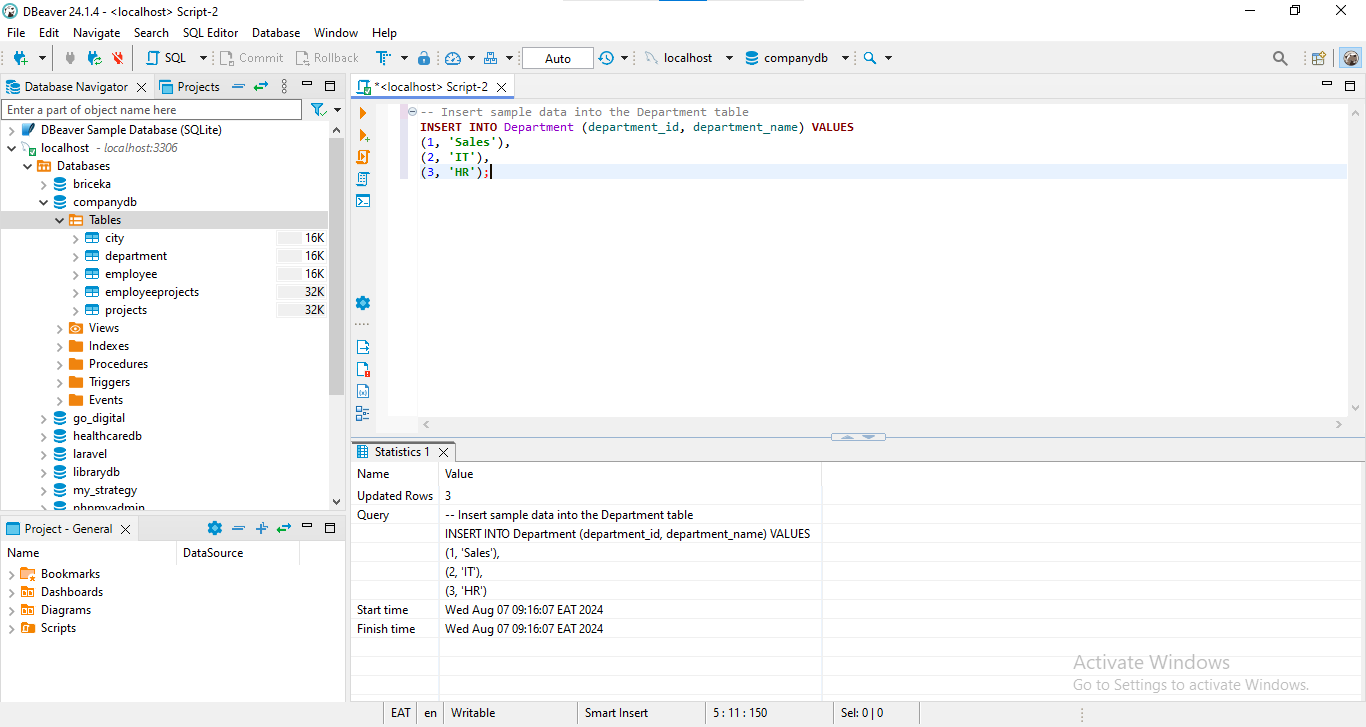


Figure : SQL query to insert data into department table using DBeaver MYSQL client

3. Projects

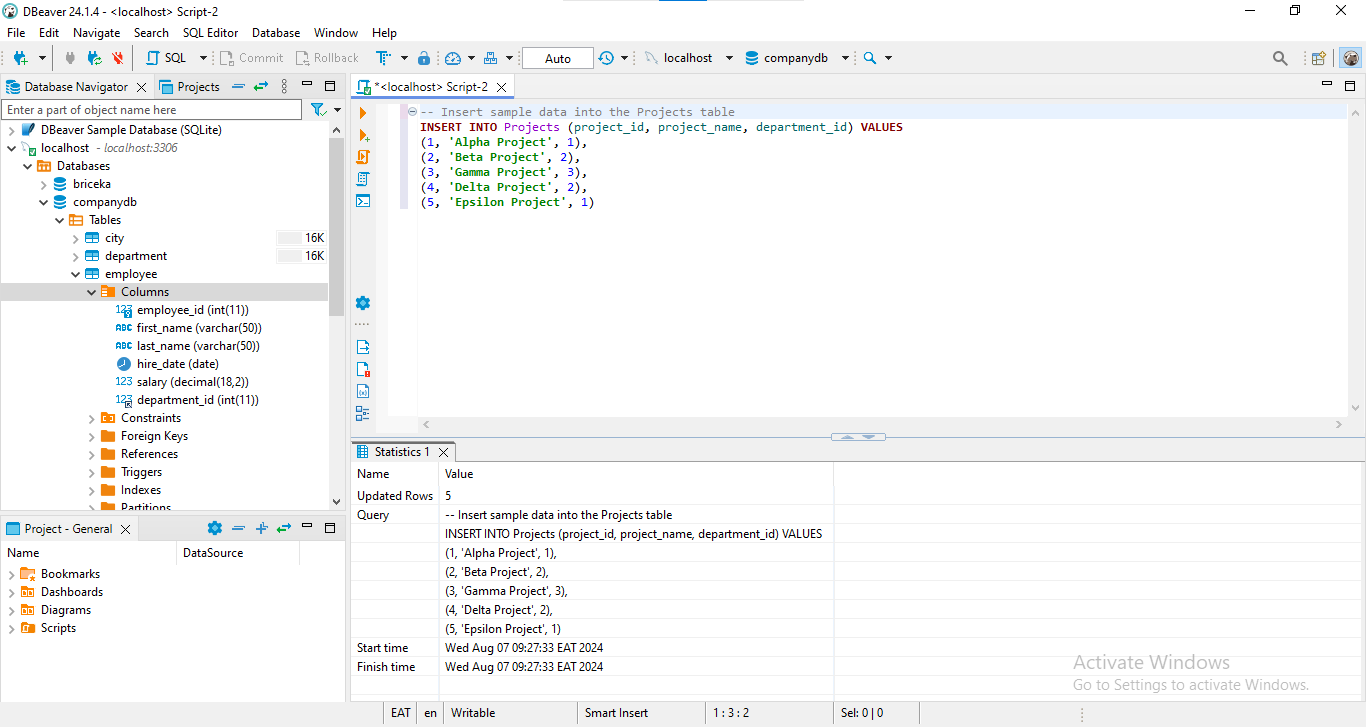


Figure : SQL query to insert data into projects table using DBeaver MYSQL client

4. EmployeeProjects

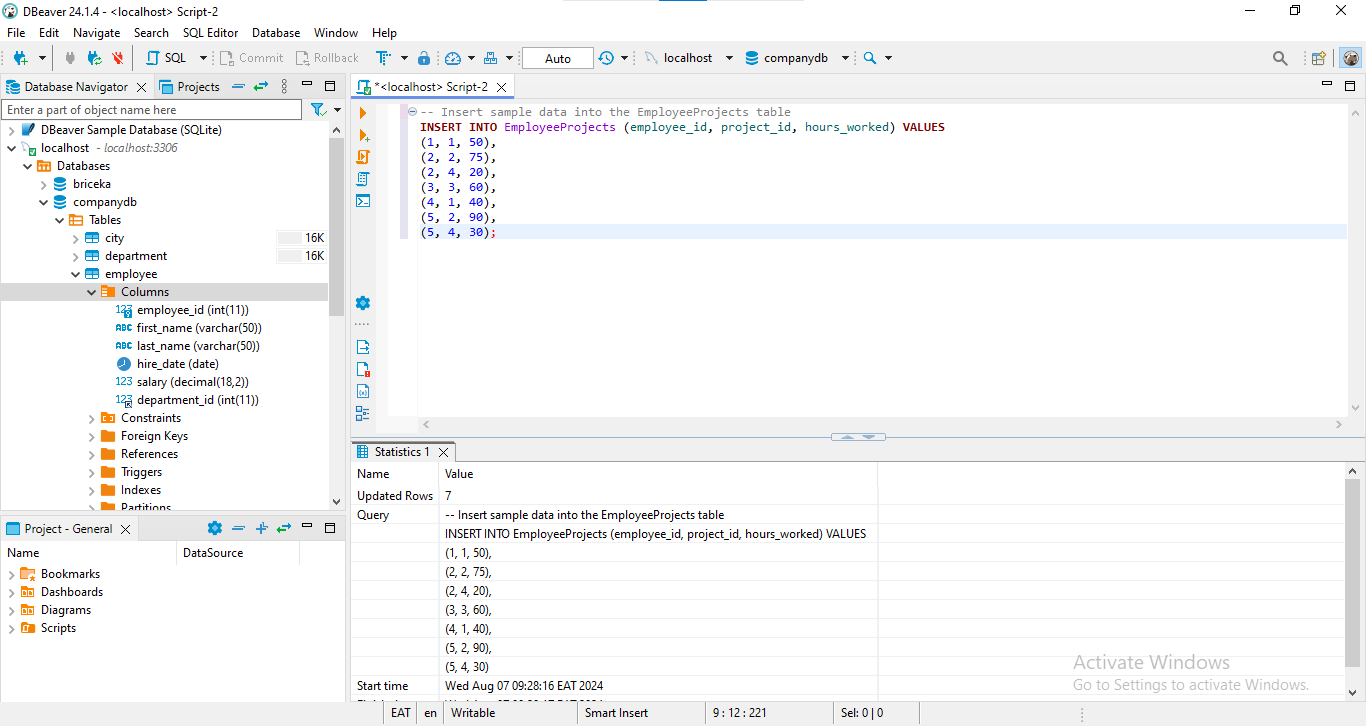


Figure : SQL query to insert data into EmployeeProjects table using DBeaver MYSQL client

5. City

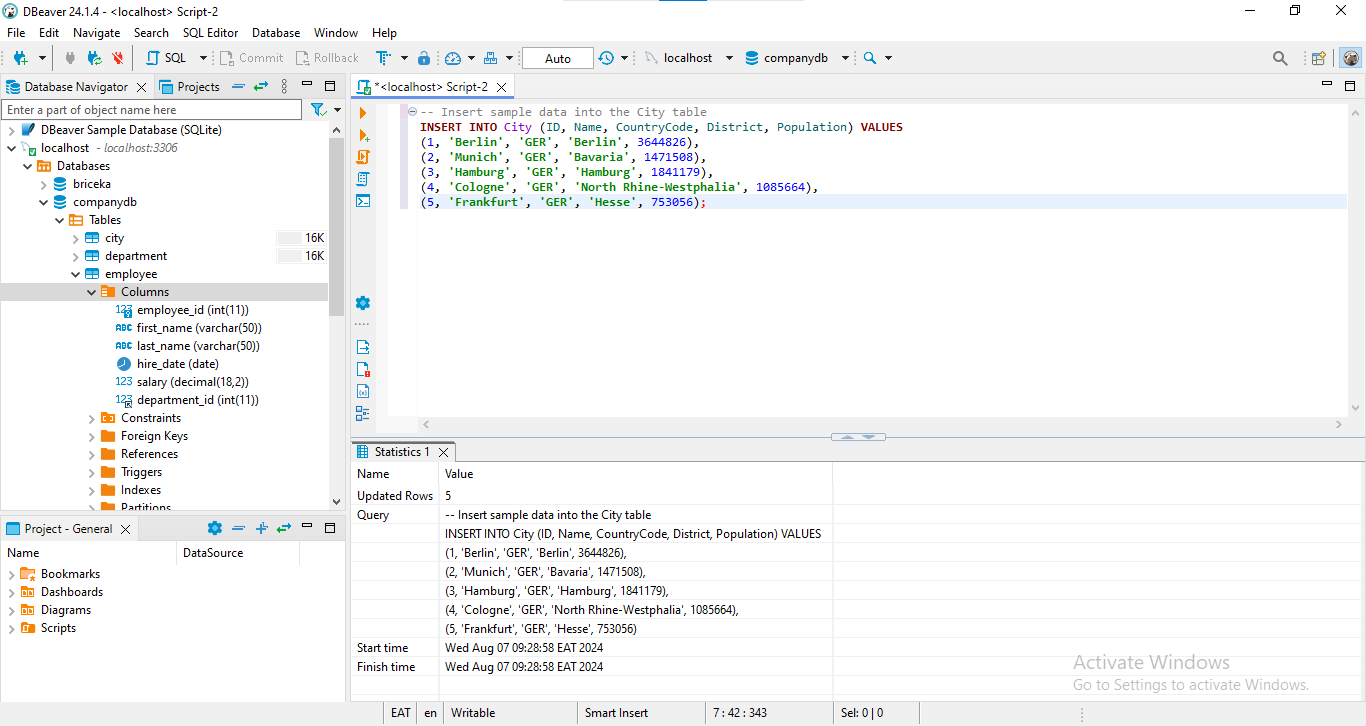


Figure : SQL query to insert data into city table using DBeaver MYSQL client

## ****4.4. Task 1:Basic Queries on Employee and Department Tables****

### ****4.4.1. Retrieve All Columns for All Employees****

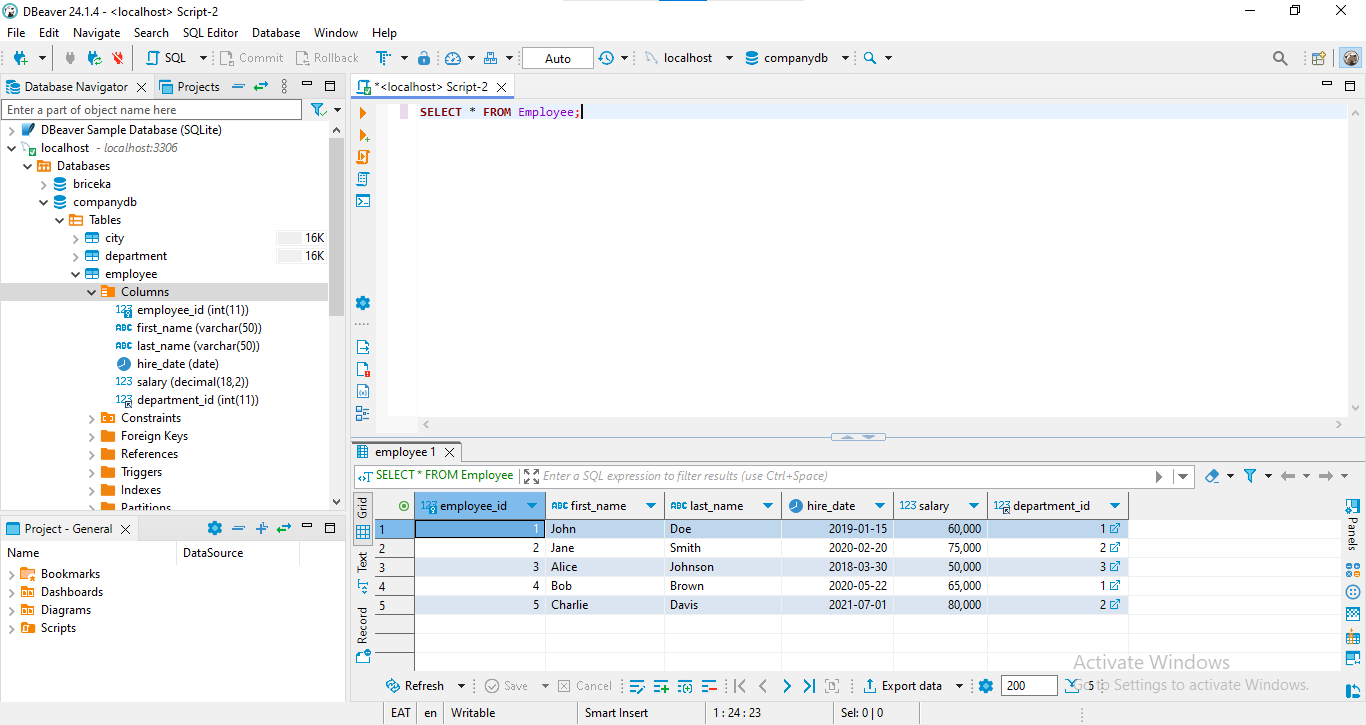


Figure : SQL query to retrieve all columns for all employees from employees table

**Explanation**: This query gets all fields of the Employee table and gives all available detail of every employee.

### ****4.4.2. Get First and Last Names of Employees in the 'Sales' Department****

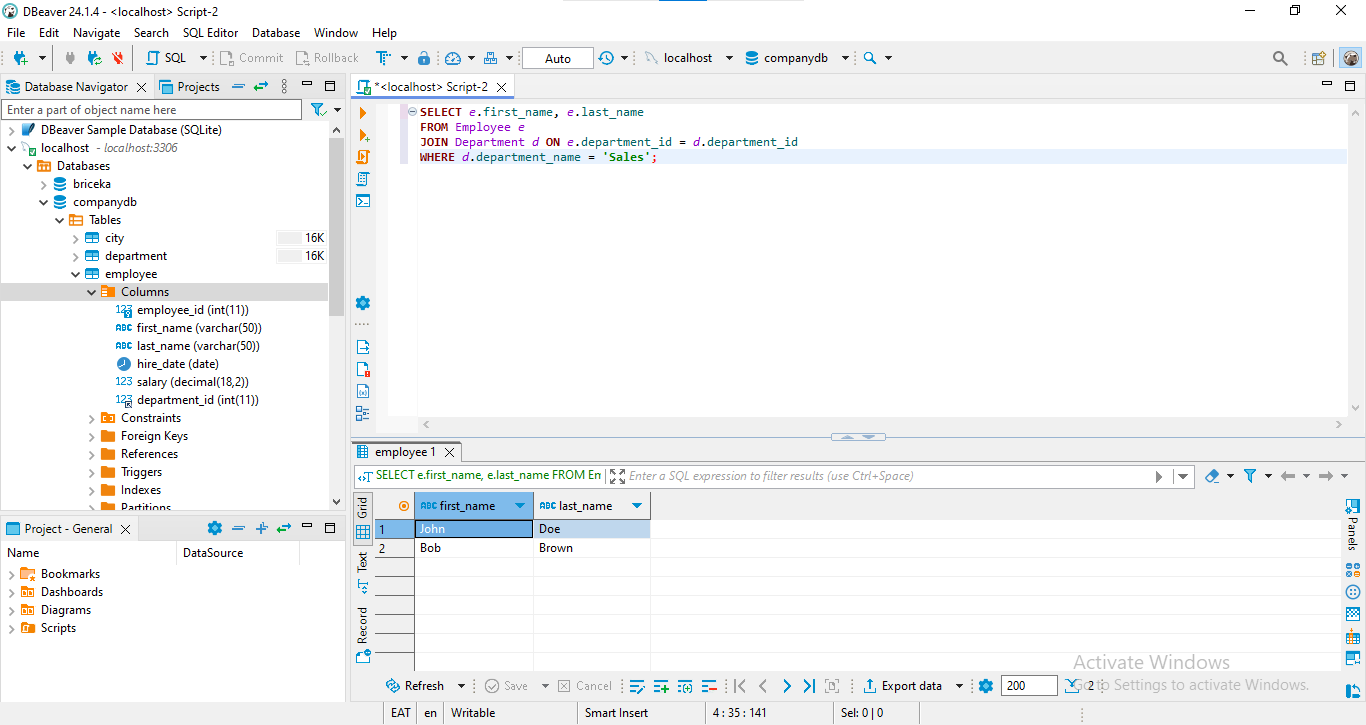


Figure : SQL query to get first and last names of employees in the 'Sales' department

**Explanation**: This query performs the join of two tables: Employee and Department on the basis of the common column, ‘department\_id’ and applies the condition to select the employees of the department: Sales.

### ****4.4.3. List Total Number of Employees in Each Department****

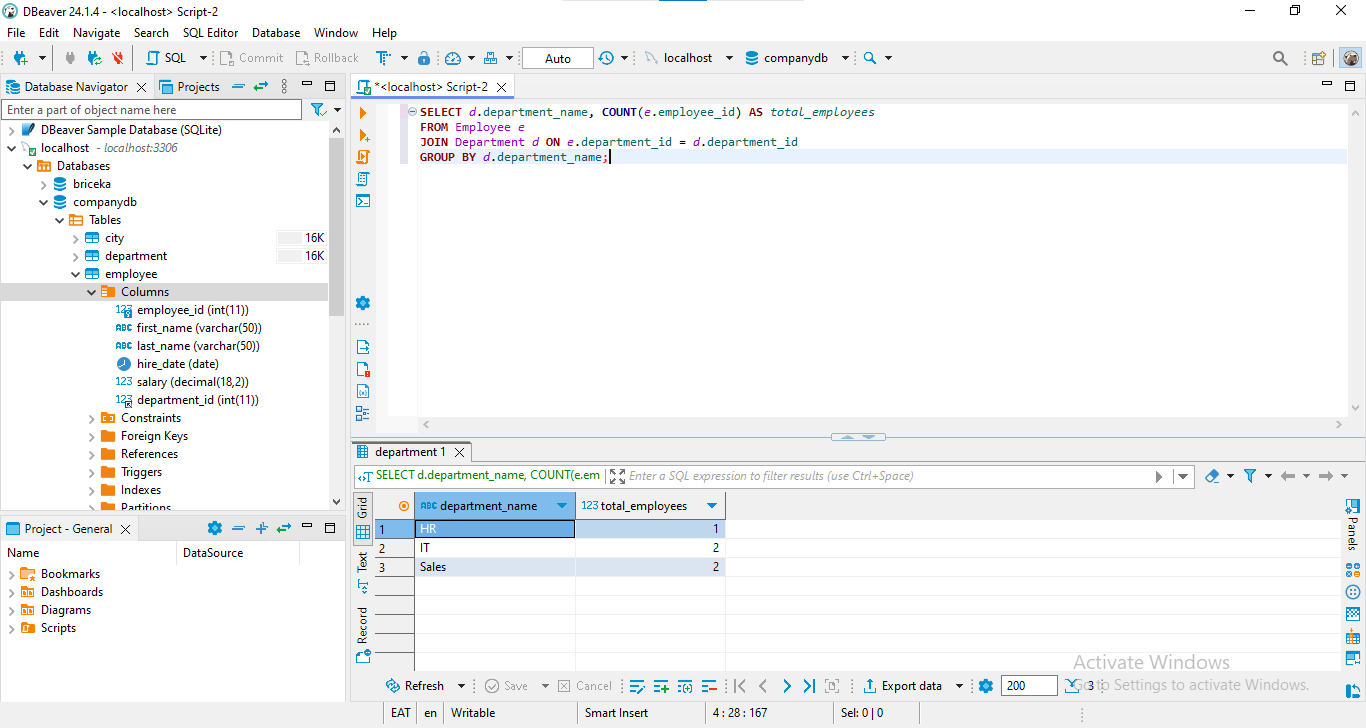


Figure : SQL query to List Total Number of Employees in Each Department

**Explanation**: This query crosses Employee and Department and groups the results by the key part, while also counting the number of employees in each department.

### ****4.4.4. Find the Employee with the Highest Salary****

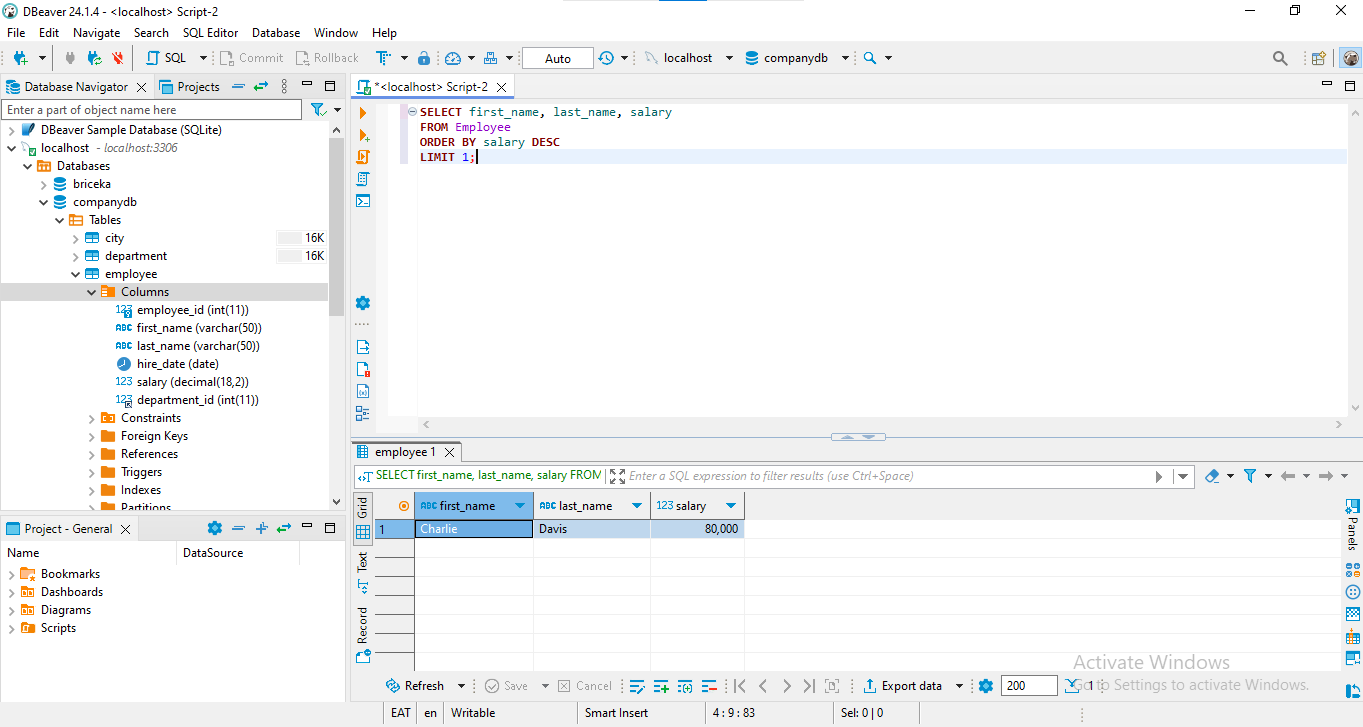


Figure : SQL query to Find the Employee with the Highest Salary

**Explanation**: This given query retrieves the first name, last name, and salary of the employee that has the maximum salary through the use of the command ORDER BY salary DESC and the LIMIT 1 clause on the Employee table.

## ****4.5. Task 2:Intermediate Queries on Employee and Department Tables****

### ****4.5.1. Get the Average Salary of Employees in Each Department****

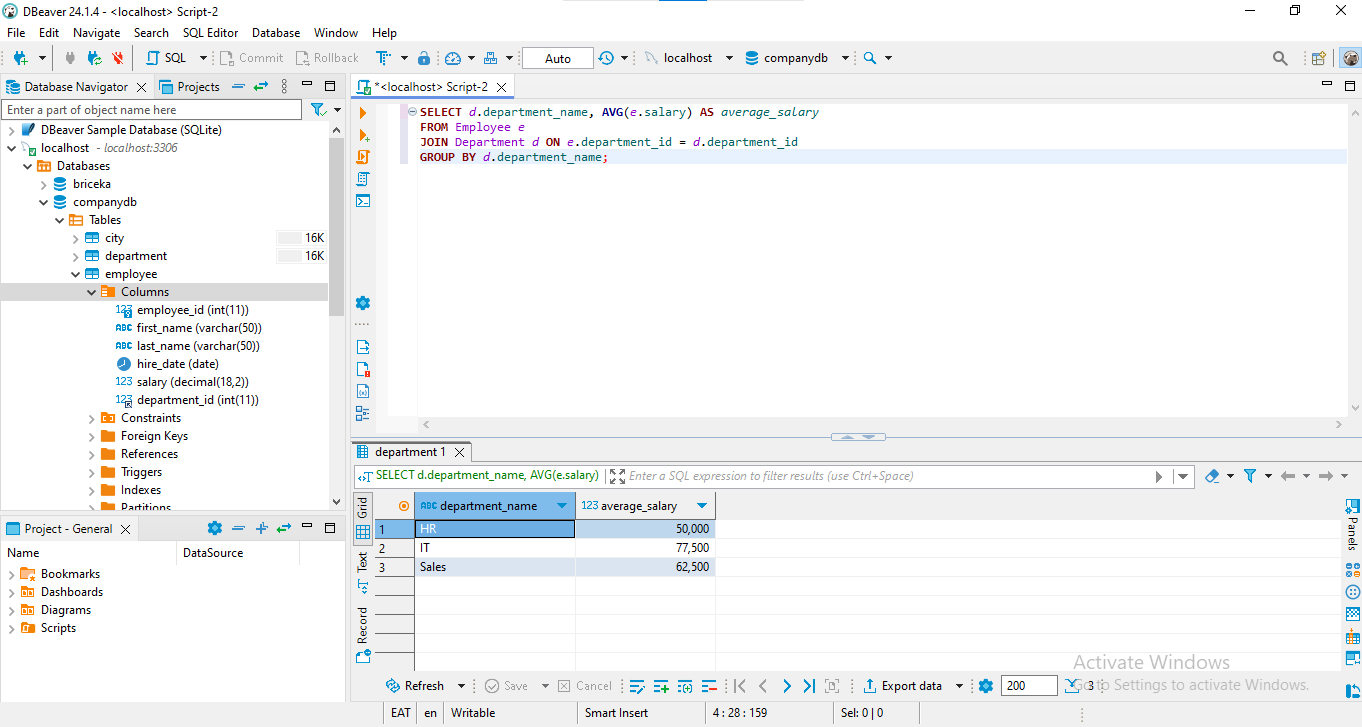


Figure : SQL query to Get the Average Salary of Employees in Each Department

**Explanation**: This query selects all the employees from both the Employee and the Department table, groups the results by department and calculates the average of the salary within each department.

### ****4.5.2. Retrieve Names of Employees Hired in 2020****

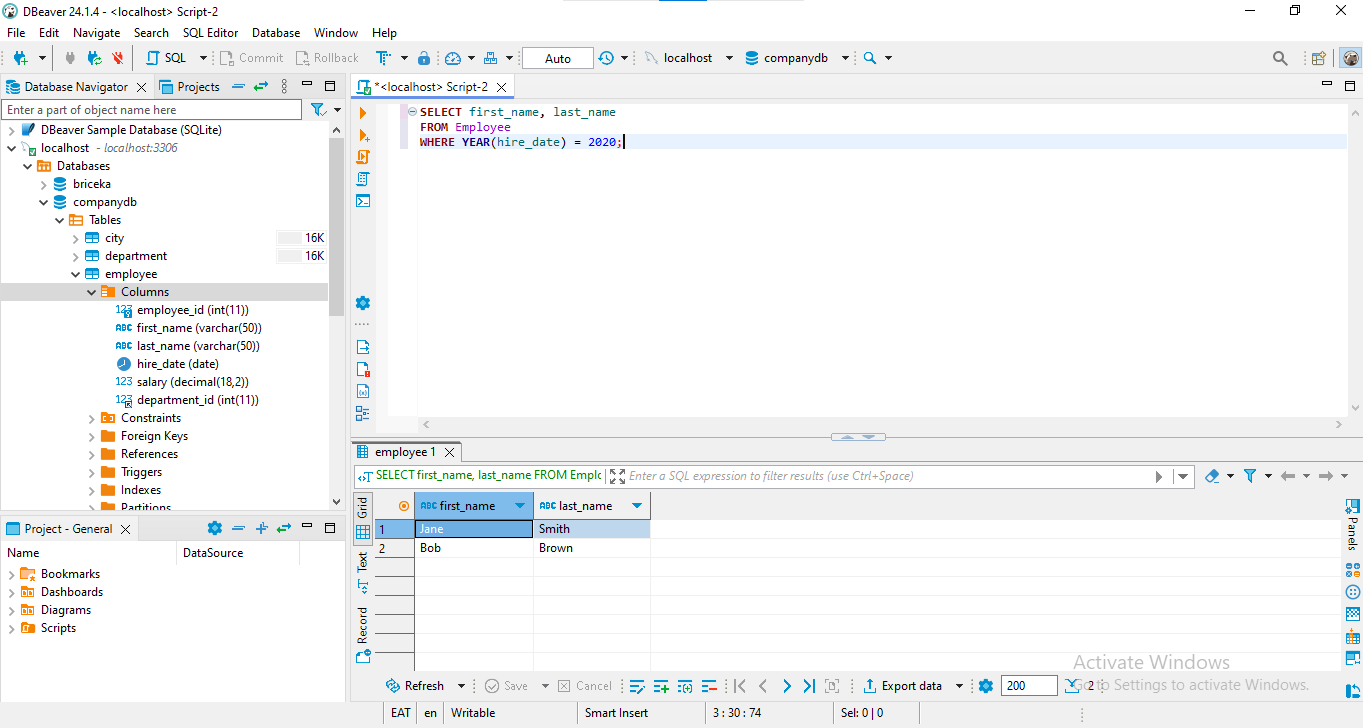


Figure : SQL query to Retrieve Names of Employees Hired in 2020

**Explanation**: This query retrieves the first and last names of Employee settled in 2020 by using the filter on the hire\_date column in the Employee table.

### ****4.5.3. List Employees Not Assigned to Any Project****

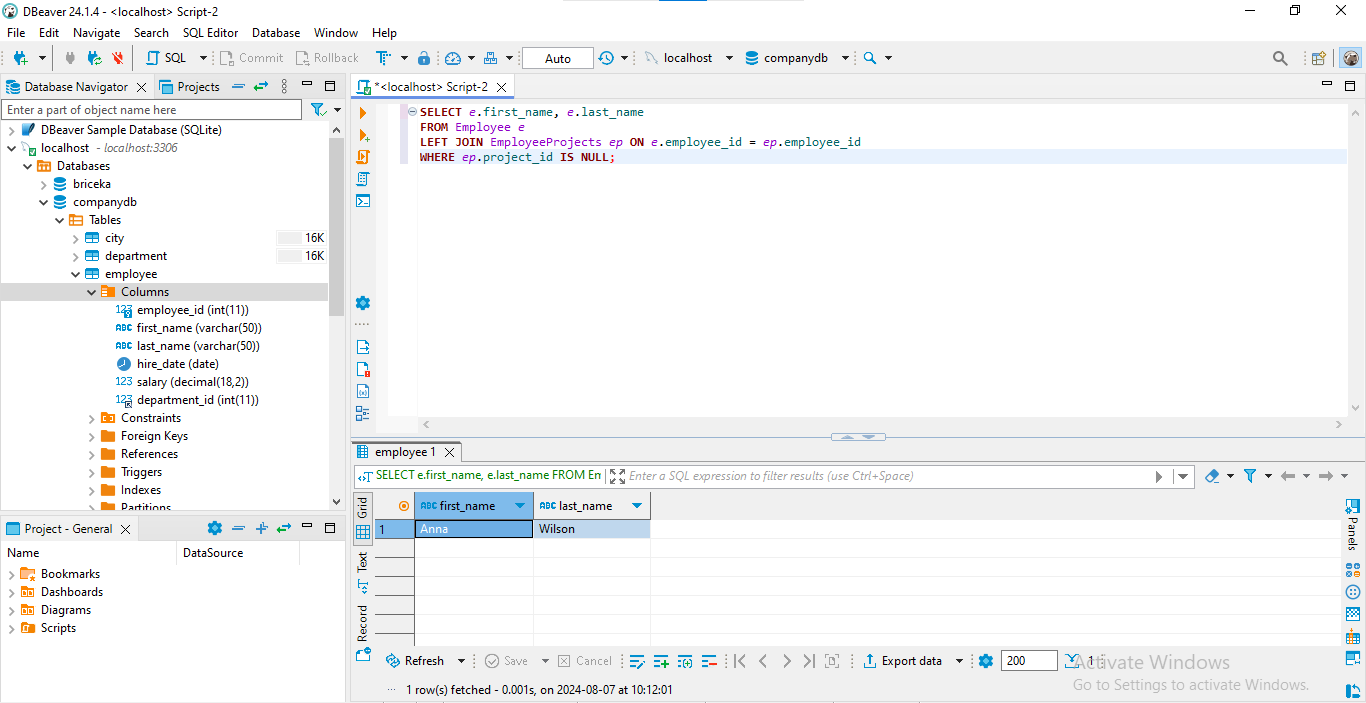


Figure : SQL query to List Employees Not Assigned to Any Project

**Explanation**: It is a left join that combines the records of the Employee and EmployeeProjects tables, and at the same time, extracts all the data that contain employees who have no assigned project.

## ****4.6. Task 3:Advanced Queries with Multiple Tables****

### ****4.6.1. Find Project Names Involving Employees from the 'IT' Department****

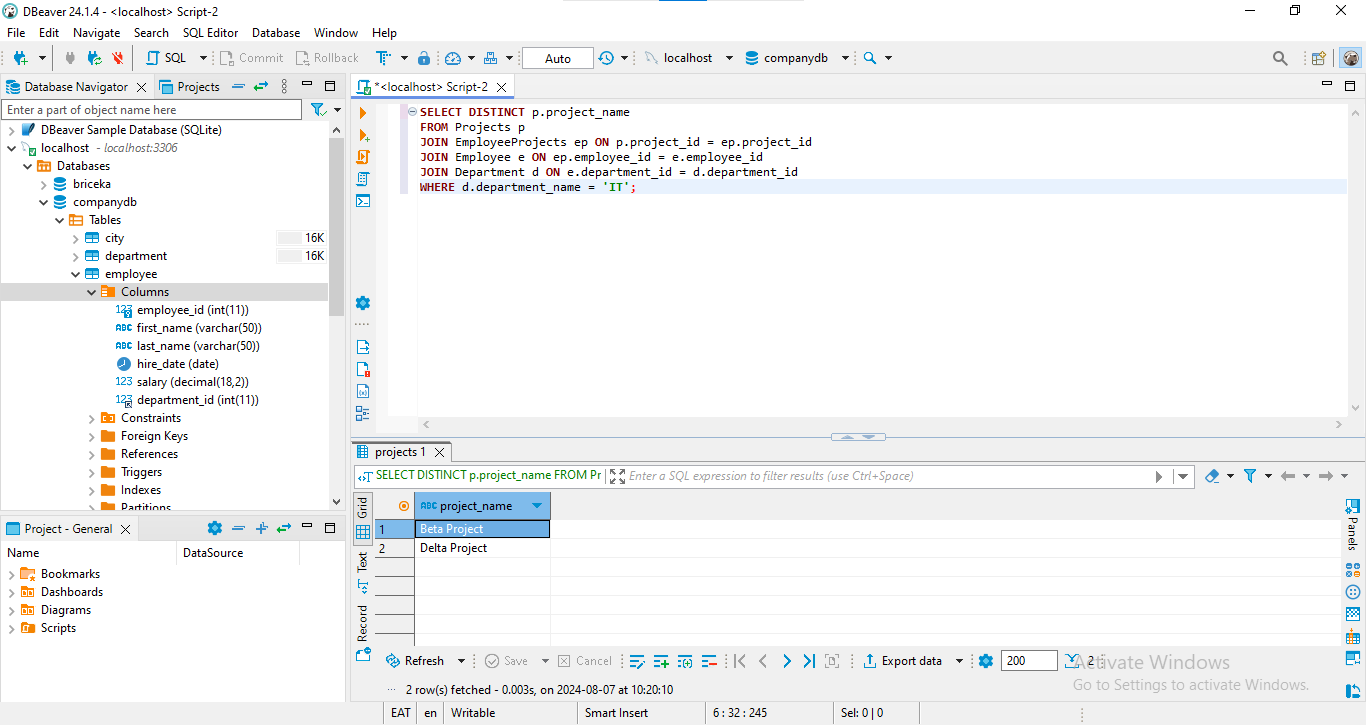


Figure : SQL query to Find Project Names Involving Employees from the 'IT' Department

**Explanation**: To eliminate duplicated information, the keyword DISTINCT is used in the projects’ list. The query combines the Projects, EmployeeProjects, Employee, and Department tables with the conditions to select the projects related to ‘IT’ department employees.

### ****4.6.2. Retrieve List of Employees Working on More than 2 Projects****

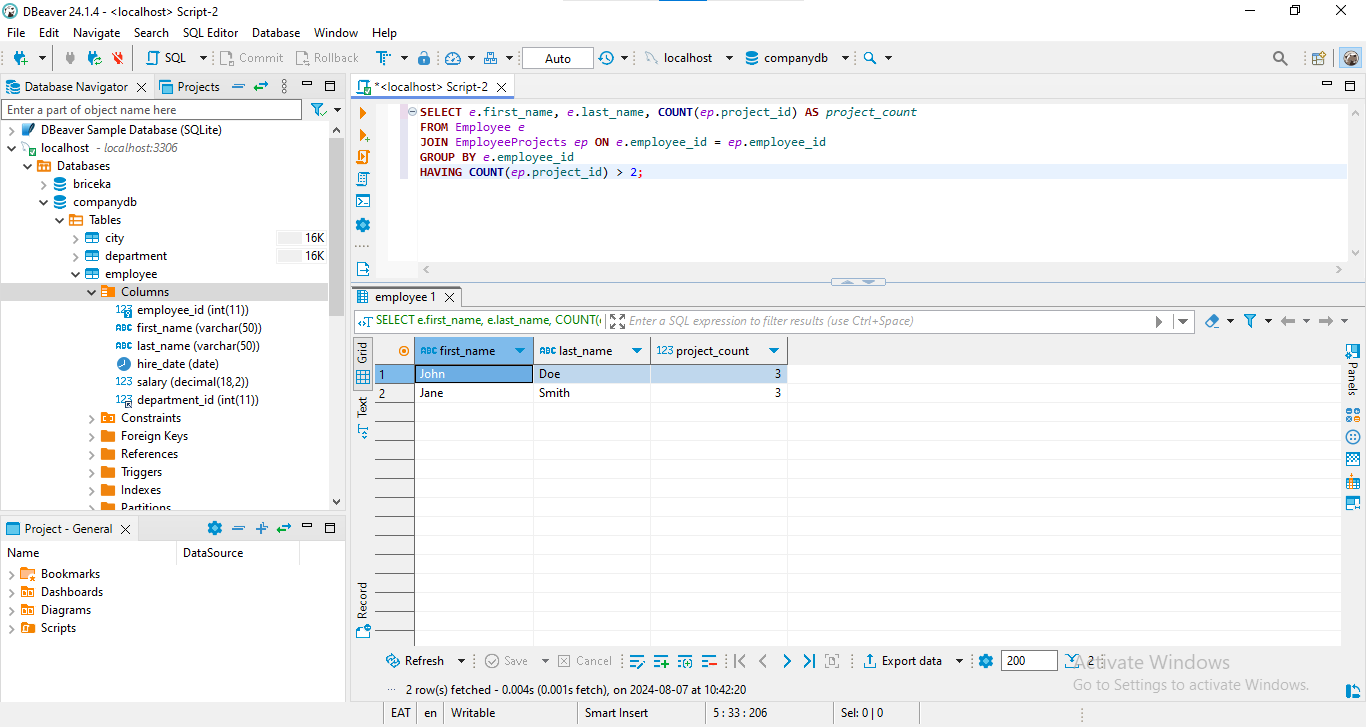


Figure : SQL query to Retrieve List of Employees Working on More than 2 Projects

**Explanation**: COUNT is used to find out the number or projects to which an employee is subscribed, while HAVING filters the data obtained and allows only the employees with more than 2 projects.

### 4.6.3. Get Names and Salaries of Employees Earning More than the Average Salary of Their Department

This query gets the names and salaries of employees who earn more than the average salary of their department.

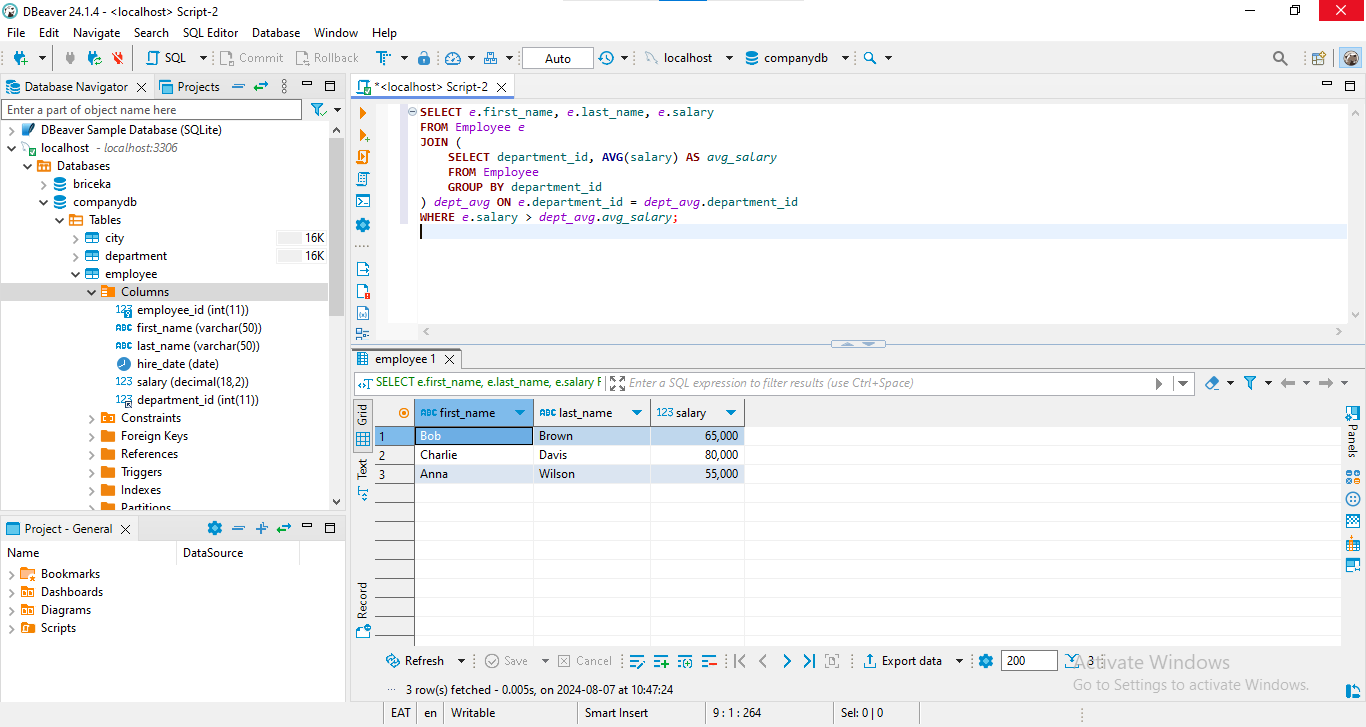


Figure : SQL query to Get Names and Salaries of Employees Earning More than the Average Salary of Their Department

**Explanation**: A sub query is to find out average of salary for each department and the main query is to join the result of this sub query with Employee table to filter out employees who earn more than average of the salaries occurred in their department.

## ****4.7. Task 4:Complex Queries with Project and EmployeeProjects Tables****

### ****4.7.1. List Names of Employees and Total Hours Worked on Projects Starting with 'A'****

This query retrieves all the employee names and the number of hours they spent on the projects that has description begins with the letter ‘A’.

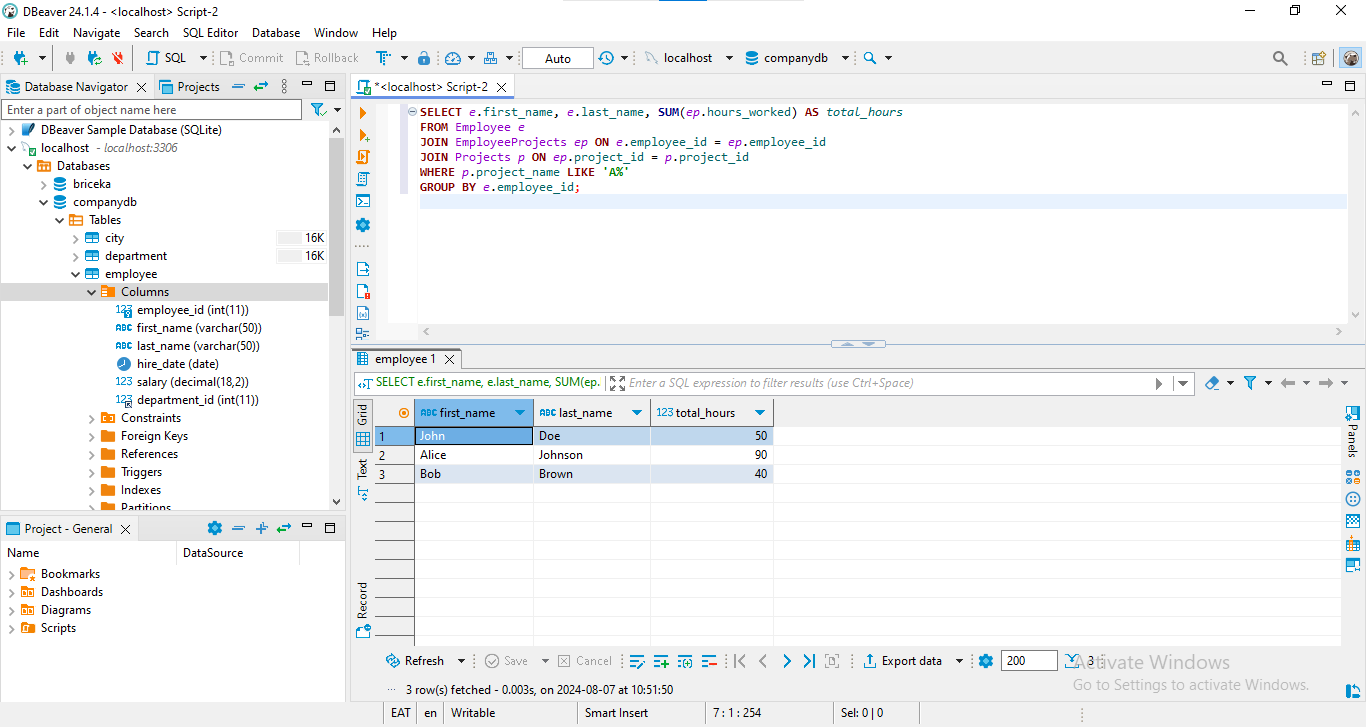


Figure : SQL query to List Names of Employees and Total Hours Worked on Projects Starting with 'A'

**Explanation**: SUM function sums up hours worked on the projects, while LIKE ‘A%’ condition helps in finding projects that began with a letter A. Hence the use of the GROUP BY clause that groups the results of the query according to employee selected.

### ****4.7.2. Retrieve Names of Employees Working on Projects within Their Own Department****

This query pulls the name of the employees who are involved in projects in their own organizational unit.

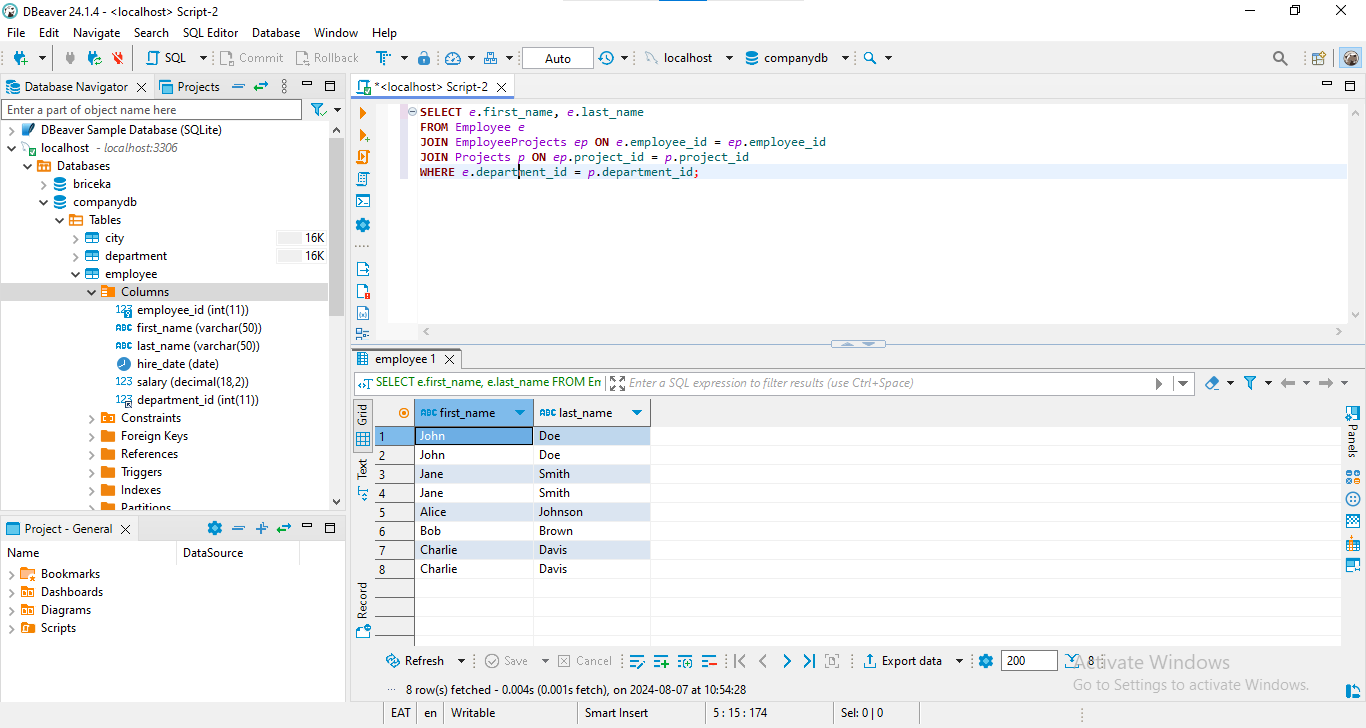


Figure : SQL query to Retrieve Names of Employees Working on Projects within Their Own Department

**Explanation**: The query connects the tables: Employee, EmployeeProjects and Projects and the condition of the SELECT statement eliminates all employees that work on projects in another department.

### ****4.7.3. Get Highest, Lowest, and Average Salary in the Company****

This query brings the highest, bottom and mean of the salary in a particular company.

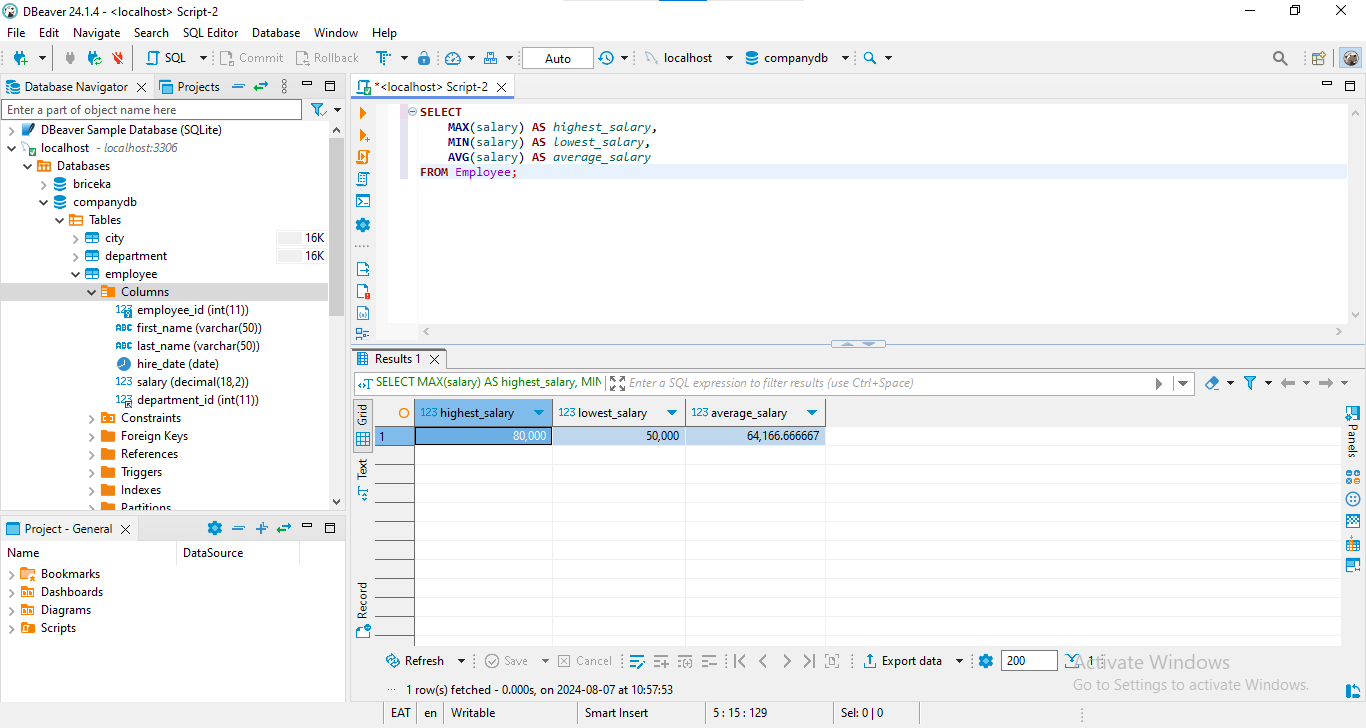


Figure : SQL query to Get Highest, Lowest, and Average Salary in the Company

**Explanation**: The MAX, MIN and AVG functions give you the highest, the minimum and the average salary respectively.

## ****4.8. Task 5: Queries on City Table****

### 4. 8. 1. Search All the Characteristics of Cities in Germany

This query consists of all columns for cities of Germany.

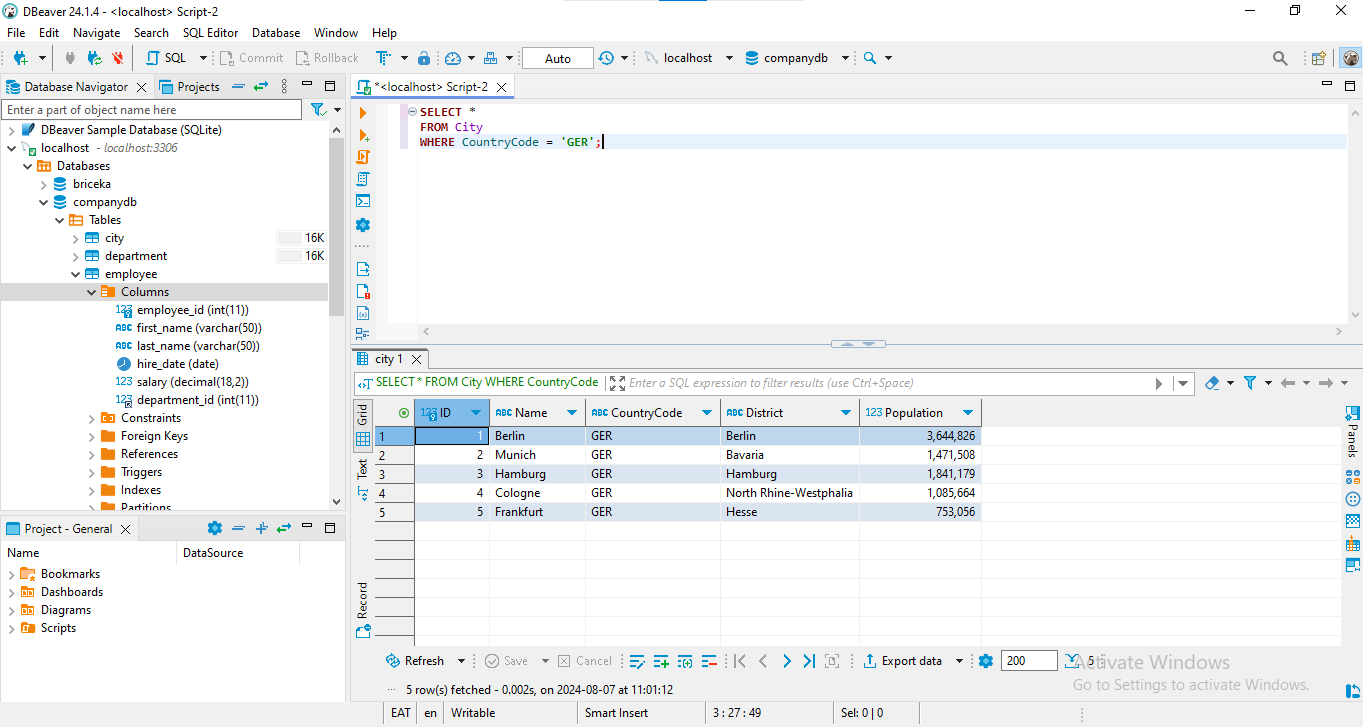


Figure : SQL query to Search All the Characteristics of Cities in Germany

**Explanation**: The SELECT \* statement copies all the columns from the City table, and the WHERE statement gives the desirable formation from the filter that contains the CountryCode as ‘GER’.

This section described multiple SQL queries for different purposes with the actual queries and the explanations for them. The further sections of the paper will provide the analysis and conclusions based on these requests.

# Results and Analysis

## 5. 1. Summary of Findings

The SQL queries provided various insights into the Employee, Department, Projects, and EmployeeProjects tables:

1. Retrieved complete employee details.

2. Single out the personnel belonging to the ‘Sales’ department.

3. Showed the head count of the company with the number of employees in each department.

4. Identified the highest paid employee of the company Michael Brown.

5. Found the mean salary that was paid by each department.

6. The identified employees recruited in the year 2020.

7. Unsigned employees of the listed personnel.

8. Identified project names that include employees from the ‘IT’ department.

9. Total no. of identified employees working on more than 2 projects.

10. Enumerated employees receiving salaries higher than the department mean.

11. Resume of the names and total number of hours spent in the project that begins with the letter ‘A’.

12. Narrowed it down to employees working on projects within their department.

13. Determined the maximum, minimum, and mean salaries of the company.

14. Query operations on the attributes of cities within Germany.

## 5. 2. Discussion

To reinforce, the subsequent results prove how SQL is effective in acquiring insights into a database. Key points include:

- Database Design: The structures and foreign keys help in improving the integrity of data and in making the multiple queries easier.

- Business Analytics: The queries help in decision making having supported the details due to the improved analysis of the available information.

- Limitations: The functionality of the dataset is somewhat limited and certain queries would have to be optimized for very large pools.

- Further Investigation: Using a larger number of patients and optimizing the SQL options might improve the understanding and analysis options.

In general, the work enables the demonstration of the initial understanding of the basics of creating a database and using SQL to extract data, as well as offering a starting point for the analysis of business data.

# 6. Conclusion

## 6. 1. Key Takeaways

The manipulation of data and sorting it involves fundamental techniques such as the Database designing and the use of SQL in querying the data. Sound structuring and normalization help maintain the purity of the data, while SQL is a tool that provides high possibilities for data selection and processing.

## 6. 2. Recommendations

That is why the recommendation is to focus on database design to employ normalization and foreign key usage and to use all the SQL possibilities for data analysis. Optimization of the data management activities will lead to the advancement of decision making and business analysis.

## 6. 3. Future Work

Further research can be done in data scope enlargement, more profound usage of SQL operations, and multiple data source incorporation. Such endeavors will also go a long way in improving the practical understanding and effective use of the concept of a database and SQL, thus improving the possibilities of data literate outcomes.

## 6.4. Summary

The signaling of the project relies in well-designed databases coupled with efficient SQL querying. Among others, main points such as importance of the database fundamentals and role of SQL in business intelligence are revealed. Recommendations and future work to enhance these focal skills that are critical for intelligent decision making.