

Table Of Contents

S.NO	TITLE	PAGE
1	Annexure-1	2
2	Declaration of the Student	3
3	Certificate	4
4	Intoduction of the project	5
5	Introduction of the company	15
6	Description of the work	18
7	Project	20
8	Conclusion	37
9	Reference	39

Annexure – 1

Summer Internship report

DATABASE MANAGEMENT SYSTEM AND SQL From Board Infinity

A training report Submitted in partial fulfillment of the requirements for the
award of degree of

Bachelors of technology (Data Science)

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB



L O V E L Y
P R O F E S S I O N A L
U N I V E R S I T Y

From 05/06/23 to 15/07/23

SUBMITTED BY

Name of student: Gopu Shanmukha Datta

Registration Number: 12104966

Signature of the student:

G. Shanmukha Datta
26/08/23

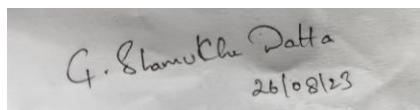
Annexure-II: Student Declaration

To whom so ever it may concern I, GOPU SHANMUKHA DATTA , 12104966 of the student, hereby declare that the work done by me on “DATABASE MANAGEMENT SYSTEM AND SQL” from 05, JUNE , 2023, is a record of original work for the partial fulfillment of the requirements for the award of the degree, B.Tech – IT .

Name of the Student : Gopu Shanmukha Datta

(Registration Number) :12104966

Signature of the student Dated:



CERTIFICATE OF COMPLETION

CERTIFICATE OF COMPLETION

THIS CERTIFICATE IS AWARDED TO

Gopu Shanmukha Datta

for successfully completing Microlearning Program in
Database Management System & Sql

12 July, 2023

ISSUED DATE


CEO, Board Infinity
Sumesh Nair

BI22LPB1345426086

CERTIFICATE NO.

BOARD

CHAPTER 1

INTRODUCTION OF THE PROJECT UNDERTAKEN

CHAPTER -1.1 Objectives of the work undertaken

The objective of a project should clearly communicate what you aimed to achieve through your real-time project using your Database Management System (DBMS) and SQL skills.

1. Objective: Database Design and Implementation

Design and implement a relational database system for a fictional e-commerce platform. Create tables to store product information, user data, order details, and customer reviews. Develop an efficient schema that supports data integrity, and implement SQL queries to retrieve product recommendations based on user preferences.

2. Objective: Performance Optimization and Querying

Optimize the performance of an existing customer database for a subscription-based service. Identify slow-performing queries and use indexing and query optimization techniques to enhance response times. Develop complex SQL queries to generate monthly usage reports for individual customers and provide insights for business decisions.

3. Objective: Data Analysis and Insights

Build a database to store sales data for a retail chain with multiple locations. Develop SQL queries to analyze sales trends, identify the best-selling products, and calculate revenue by

region. Create visualizations to present the findings, enabling the management team to make informed decisions about inventory and expansion.

4. Objective: User Management and Security

Create a database for a social networking platform with user profiles, friend connections, and posts. Implement user authentication and authorization mechanisms using SQL queries to control access to private user data. Develop a user-friendly interface for account management and password reset functionalities.

5. Objective: Data Migration and Transformation

Migrate data from a legacy database system to a modern DBMS for a healthcare institution. Transform and map the data to fit the new schema, ensuring data integrity and consistency. Develop SQL scripts to validate the accuracy of the migrated data and address any discrepancies.

6. Objective: Inventory Management and Restocking

Design a database for a warehouse management system to track inventory levels and automate restocking orders. Implement SQL queries to monitor stock levels in real-time, trigger reorder notifications when items run low, and generate reports to analyze inventory turnover and cost efficiency.

7. Objective: Online Library Catalog

Develop a database for an online library catalog. Design tables to store book details, borrower information, and transaction history. Create SQL queries to check book availability, handle borrowing and returning processes, and generate overdue book notifications.

8. Objective: Event Registration and Attendance

Build a database for an event management platform. Design tables to manage event details, participant registrations, and attendance records. Develop SQL queries to generate event attendance reports and track popular event categories

CHAPTER -1.2 Scope of the Work

The scope of work outlines the boundaries and extent of what your project aims to cover. It helps clarify the project's limits and defines what will and won't be included. Here's is the scope of work for real-time project involving Database Management System (DBMS) and SQL

1. Database Design:

Define the structure of the database, including the tables, relationships, and attributes. Specify the data types and constraints for each field to ensure data integrity.

2. Data Entry and Manipulation:

Populate the database with sample or simulated data that is relevant to the project's context. Perform data insertion, updates, and deletions using SQL statements to demonstrate data manipulation capabilities.

3. Query Development:

Develop a set of SQL queries that showcase various levels of complexity. These queries should include basic SELECT statements, as well as more advanced queries involving JOINs, subqueries, aggregation, and filtering.

4. User Authentication and Authorization:

If applicable, implement a basic user authentication and authorization system using SQL. Define user roles and permissions to restrict access to certain parts of the database.

5. Project-Specific Functionality:

If your project involves a specific application (e.g., e-commerce, inventory management), outline the functionalities you'll demonstrate using the database. This could include features like product search, order placement, or inventory tracking.

6. Performance Optimization:

Apply indexing techniques to improve query performance. Measure and compare query execution times before and after indexing to showcase the optimization's impact.

7. Data Analysis and Reporting:

Develop SQL queries to extract meaningful insights from the data. Generate reports, charts, or graphs that illustrate trends, patterns, or key metrics based on the project's context.

8. Error Handling:

Address potential data validation and error handling scenarios in your project. Show how you manage scenarios such as invalid data entry, duplicate records, or failed queries.

9. Documentation:

Create documentation that includes an explanation of the database schema, table structures, relationships, and explanations of the SQL queries used in the project.

10.Presentation:

Prepare a presentation that summarizes your project's objectives, the database design, key SQL queries, and the outcomes achieved. Use visuals to help explain complex concepts.

CHAPTER -1.3 Importance and Applicability

In today's rapidly evolving technological landscape, proficiency in Database Management System (DBMS) and Structured Query Language (SQL) is of paramount importance. These skills find widespread applicability across various industries and domains, underpinning the efficient handling and analysis of data. Here's a closer look at the significance and practical applications of DBMS and SQL.

1. Efficient Data Management:

DBMS and SQL empower organizations to efficiently manage vast amounts of data. By facilitating structured storage, retrieval, and manipulation of information, they enable streamlined business operations and decision-making processes.

2. Business Intelligence and Data Analysis:

The ability to query and extract insights from data is crucial. SQL's data manipulation capabilities allow analysts and business professionals to generate reports, visualize trends, and derive actionable insights, fostering informed strategic decisions.

3. Application Development:

DBMS and SQL are foundational for developing data-driven applications. From e-commerce platforms to customer relationship management systems, these skills are essential for creating dynamic applications that interact with backend databases.

4. Data-Driven Marketing:

In the marketing realm, DBMS and SQL empower marketers to segment customer data, personalize campaigns, and assess campaign performance. This personalized approach enhances customer engagement and drives marketing success.

5. E-Commerce and Online Transactions:

E-commerce platforms heavily rely on databases to manage products, orders, and customer information. SQL enables seamless online transactions and supports inventory management, enhancing the user experience.

6. Healthcare Informatics:

In healthcare, DBMS and SQL play a pivotal role in managing patient records, medical histories, and treatment plans. These skills ensure secure data storage and quick retrieval of critical patient information.

7. Financial Services and Banking:

Financial institutions leverage DBMS and SQL to manage account information, process transactions, and analyze financial data. These skills contribute to accurate record-keeping and compliance.

8. Supply Chain Management:

Efficient supply chain management requires real-time data visibility. DBMS and SQL facilitate inventory tracking, demand forecasting, and order processing, optimizing the entire supply chain.

9. Educational Institutions:

Educational institutions use DBMS and SQL for student record management, course scheduling, and academic performance analysis. These skills aid in administrative efficiency and student support.

10. Research and Scientific Data:

Researchers utilize DBMS and SQL to organize and analyze vast datasets in fields such as genomics, environmental science, and astronomy, enabling groundbreaking discoveries.

In essence, DBMS and SQL form the backbone of modern data management and utilization. Proficiency in these skills is not only advantageous but often a prerequisite for roles in data analysis, database administration, software development, and other technology-driven domains. The broad applicability across industries underscores the indispensability of DBMS and SQL expertise in today's data-driven world.

CHAPTER -1.4 Role and profile

Professionals skilled in Database Management System (DBMS) and Structured Query Language (SQL) play essential roles across various industries, contributing to effective data handling, analysis, and informed decision-making. Here's an overview of potential roles and profiles for individuals with expertise in DBMS and SQL.

1. Database Administrator

Database administrators oversee the design, implementation, and maintenance of databases. They manage user access, ensure data integrity, and optimize database performance. Proficiency in DBMS and SQL is vital for efficiently managing data storage and retrieval.

2. Data Analyst

Data analysts extract insights from databases using SQL queries. They analyze data trends, patterns, and correlations, and present findings through visualizations or reports. A solid grasp of DBMS and SQL enables accurate data interpretation and reporting.

3. SQL Developer

SQL developers create and optimize SQL queries for various applications. They design databases, write complex queries, and ensure efficient data retrieval. Strong SQL skills are essential for building high-performance, data-centric applications.

4. Business Intelligence Analyst

Professionals in this role use DBMS and SQL to transform raw data into actionable insights. They build data models, design dashboards, and perform data mining to support strategic decision-making across departments.

5. Backend Developer

Backend developers utilize DBMS and SQL to create and maintain server-side logic for applications. They integrate databases, manage data flow, and ensure seamless interaction between the frontend and backend of applications.

6. ETL Developer

ETL (Extract, Transform, Load) developers specialize in moving data between systems. They use SQL to extract data from source systems, transform it to meet target requirements, and load it into data warehouses or databases.

7. Software Engineer

Software engineers with DBMS and SQL skills contribute to application development. They design database schemas, integrate databases with applications, and optimize database interactions for software efficiency.

8. Data Scientist

Data scientists work with large datasets to derive actionable insights. They use SQL to preprocess data, join multiple datasets, and create the foundational structures for machine learning models and advanced analytics.

9. Systems Analyst

Systems analysts leverage DBMS and SQL to understand business requirements and design system solutions. They collaborate with stakeholders to develop systems that optimize processes and data workflows.

10. IT Consultant

IT consultants with DBMS and SQL expertise offer guidance on database design, implementation, and optimization. They help organizations choose appropriate technologies and strategies to meet their data management needs.

In summary, a strong foundation in DBMS and SQL opens doors to a wide range of roles and profiles that rely on effective data management, analysis, and application development. As organizations continue to emphasize data-driven decision-making, professionals with these skills are well-positioned to contribute significantly to their respective industries.

CHAPTER -2

Introduction of the Company/Work: Board Infinity

Board Infinity, a dynamic and innovative educational platform, stands at the intersection of technology and learning. With a steadfast commitment to transforming careers and empowering individuals, Board Infinity has become a beacon of education for the modern world.

CHAPTER -2.1 Company's Vision and Mission

Vision: To revolutionize education by providing accessible and personalized learning experiences that empower individuals to achieve their full potential.

Mission: To bridge the gap between industry demands and traditional education by offering industry-relevant courses and mentorship, thereby nurturing future leaders.

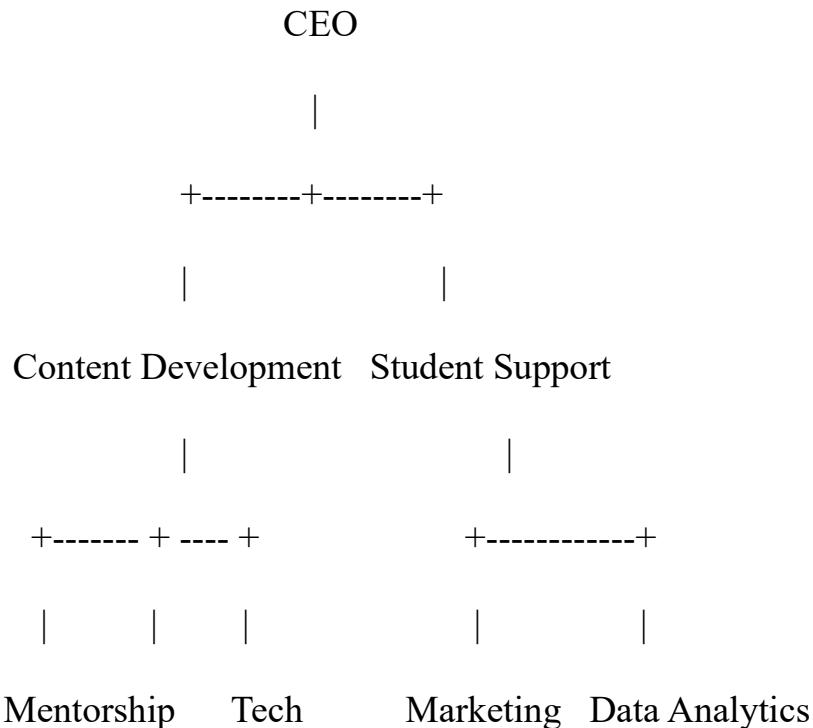
CHAPTER -2.2 Origin and Growth of the Company

Founded in 2017 by Abhay Gupta and Sumesh Nair, Board Infinity has quickly evolved into a leading ed-tech platform that bridges the gap between theoretical knowledge and practical application. The company's journey began with a vision to provide learners with skills that are directly applicable in the professional world. Over the years, the platform has expanded its course offerings and garnered a reputation for equipping learners with the tools they need to excel in their chosen fields.

CHAPTER -2.3 Various Departments and Their Functions

1. Content Development: This department is responsible for creating comprehensive and industry-relevant course content. Subject matter experts collaborate to ensure that courses align with the latest trends and requirements.
2. Student Support: The student support team serves as a lifeline for learners, assisting with enrollment, troubleshooting, and addressing queries promptly to provide a seamless learning experience.
3. Mentorship: Board Infinity's mentorship program connects learners with industry professionals who provide guidance, insights, and real-world context for the skills they are acquiring.
4. Technology and Development: The tech team drives the platform's functionality, ensuring a user-friendly interface and a smooth learning environment. They also integrate advanced learning tools to enhance the educational experience.
5. Marketing and Outreach: This department spreads awareness about the platform's unique approach to education. They engage with potential learners, showcasing the value and benefits of Board Infinity's offerings.
6. Data Analytics: Leveraging data-driven insights, this team monitors user engagement, learning patterns, and course effectiveness. These insights shape decisions to continually enhance the learning journey.

CHAPTER -2.4 Organization Chart of the Company



Board Infinity's organizational structure fosters collaboration and synergy among its diverse departments. This dynamic setup enables the company to deliver exceptional learning experiences that align with its vision and mission.

In essence, Board Infinity's journey from inception to its current stature underscores its dedication to pioneering a transformative approach to education, preparing learners for success in the modern professional landscape.

CHAPTER 3

Brief Description of the Work Done

HR Database Exploration and Querying

During the course with a real-time project at Board Infinity, I had the privilege to delve into the intricacies of the HR Database. This database, originally created by Microsoft, served as an invaluable resource for learning and applying Database Management System (DBMS) and Structured Query Language (SQL) skills.

CHAPTER -3.1 Position of Learning and Roles

In the capacity of a learner, I undertook the role of a database enthusiast. My responsibilities revolved around gaining a deep understanding of the HR Database's structure, querying capabilities, and real-world applicability.

CHAPTER -3.2 Activities/Equipment Handled

- Explored the database schema to comprehend the relationships among tables.
- Executed SQL queries to retrieve specific information from different tables.
- Utilized query optimization techniques to enhance query performance.
- Engaged with SQL tools and interfaces to interact with the database effectively.

CHAPTER -3.3 Challenges Faced and How Those Were Tackled

One of the challenges I encountered was crafting queries that involved complex joins across multiple tables. To overcome this, I thoroughly reviewed SQL JOIN operations, practiced on simpler queries, and gradually applied these techniques to more intricate scenarios.

Additionally, optimizing query performance demanded learning about indexing and evaluating execution plans, which I addressed through research and experimentation.

CHAPTER -3.4 Learning Outcomes

- Developed a strong command over SQL fundamentals, including SELECT statements, JOIN operations, and aggregate functions.
- Enhanced problem-solving skills by tackling complex querying scenarios and optimizing query performance.
- Gained insights into database design principles through hands-on exploration of interrelated tables.
- Learned to transform business questions into actionable SQL queries, fostering a practical approach to data analysis.

CHAPTER -3.5 Data Analysis

Leveraging the HR Database, I conducted data analysis tasks that included:

- Calculating average salaries for different job titles.
- Identifying the departments with the highest number of employees.
- Tracing the job history of specific employees to understand their career trajectory.

These analyses offered insights into employee distribution, salary structures, and career progression within the organization.

In essence, my engagement with the HR Database enabled me to acquire practical skills in querying, data analysis, and database interaction. The exposure to real-world data scenarios, along with the challenges I tackled, enhanced my proficiency in DBMS and SQL, empowering me to apply these skills effectively in diverse contexts.

CHAPTER 4

BOARD INFINITY PROJECT

TITLE : HR CASE STUDY

QUESTION :

The HR sample database has seven tables:

1. The `employees` table stores the data of employees.
2. The `jobs` table stores the job data including job title and salary range.
3. The `departments` table stores department data.
4. The `job_history` table stores the job history of employees.
5. The `locations` table stores the location of the departments of the company.
6. The `countries` table stores the data of countries where the company is doing business.
7. The `regions` table stores the data of regions such as Asia, Europe, America, and the Middle East and Africa. The countries are grouped into regions.

Data Set Link - <https://www.kaggle.com/datasets/sirajahmad/hr-schema-mysql>

Task 1 :

1. Write a query to find the addresses (location_id, street_address, city, state_province, country_name) of all the departments

Ans :

```
SELECT d.department_id, l.street_address, l.city, l.state_province, c.country_name
```

```
FROM departments d
```

```
JOIN locations l ON d.location_id = l.location_id
```

```
JOIN countries c ON l.country_id = c.country_id;
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80.
- INSTANCE:** Startup / Shutdown, Server Logs, Options File.
- PERFORMANCE:** Dashboard, Performance Reports, Performance Schema Setup.
- Administration:** Schemas, Information.
- SQL Editor:** Shows the executed SQL query and its results.
- Result Grid:** Displays the query results in a tabular format.
- Action Output:** Shows the log of actions taken during the session.
- Output:** Shows the log of actions taken during the session.
- Object Info:** Session.
- System Icons:** Weather, Search, Taskbar icons.
- System Status:** ENG IN, 25-06-2023, 13:15.

SQL Editor Content:

```
ALTER TABLE employees ADD FOREIGN KEY (manager_id) REFERENCES employees(employee_id);
ALTER TABLE departments ADD FOREIGN KEY (manager_id) REFERENCES employees(employee_id);
ALTER TABLE job_history ADD FOREIGN KEY (employee_id) REFERENCES employees(employee_id);
ALTER TABLE job_history ADD FOREIGN KEY (job_id) REFERENCES jobs(job_id);
ALTER TABLE job_history ADD FOREIGN KEY (department_id) REFERENCES departments(department_id);
SELECT d.department_id, l.street_address, l.city, l.state_province, c.country_name FROM departments d JOIN locations l ON d.location_id = l.location_id
```

Result Grid:

department_id	street_address	city	state_province	country_name
60	2014 Jabberwocky Rd	Southlake	Texas	United States of America
50	2011 Interiors Blv	South San Francisco	California	United States of America
10	2004 Charade Rd	Seattle	Washington	United States of America
30	2004 Charade Rd	Seattle	Washington	United States of America
90	2004 Charade Rd	Seattle	Washington	United States of America
100	2004 Charade Rd	Seattle	Washington	United States of America
110	2004 Charade Rd	Seattle	Washington	United States of America
120	2004 Charade Rd	Seattle	Washington	United States of America
130	2004 Charade Rd	Seattle	Washington	United States of America
140	2004 Charade Rd	Seattle	Washington	United States of America
150	2004 Charade Rd	Seattle	Washington	United States of America
160	2004 Charade Rd	Seattle	Washington	United States of America

Action Output:

#	Time	Action	Message	Duration / Fetch
3447	13:14:06	ALTER TABLE employees ADD FOREIGN KEY (manager_id) REFERENCES employees(employee_id)	107 row(s) affected Records: 107 Duplicates: 0 Warnings: 0	0.046 sec
3448	13:14:06	ALTER TABLE departments ADD FOREIGN KEY (manager_id) REFERENCES employees(employee_id)	27 row(s) affected Records: 27 Duplicates: 0 Warnings: 0	0.051 sec
3449	13:14:06	ALTER TABLE job_history ADD FOREIGN KEY (employee_id) REFERENCES employees(employee_id)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
3450	13:14:07	ALTER TABLE job_history ADD FOREIGN KEY (job_id) REFERENCES jobs(job_id)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.047 sec
3451	13:14:07	ALTER TABLE job_history ADD FOREIGN KEY (department_id) REFERENCES departments(department_id)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.047 sec
3452	13:15:24	SELECT d.department_id, l.street_address, l.city, l.state_province, c.country_name FROM departments d JOIN...	27 row(s) returned	0.000 sec / 0.000 sec

Task 2 :

2. Write a query to find the name (first_name, last name), department ID and name of all the employees

Ans :

```
SELECT e.first_name, e.last_name, e.department_id, d.department_name
```

```
FROM employees e
```

```
JOIN departments d ON e.department_id = d.department_id;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema "hr-schema-mysqld".
- INSTANCE:** Shows "Local instance MySQL80".
- PERFORMANCE:** Shows "Performance Schema Setup".
- Administration:** Shows "Information".
- Result Grid:** Displays the results of the executed query:

first_name	last_name	department_id	department_name
Jennifer	Whalen	10	Administration
Michael	Hartstein	20	Marketing
Pat	Fay	20	Marketing
Den	Raphaely	30	Purchasing
Alexander	Khoo	30	Purchasing
Shelli	Baida	30	Purchasing
Sat	Tobias	30	Purchasing
Guy	Himura	30	Purchasing
Karen	Colmenares	30	Purchasing
Susan	Mariotti	40	Human Resources
Matthew	Weiss	50	Shipping
Adam	Fripp	50	Shipping
- Action Output:** Shows the history of SQL statements executed:

#	Time	Action	Message	Duration / Fetch
3448	13:14:05	ALTER TABLE departments ADD FOREIGN KEY (manager_id) REFERENCES employees(employee_id)	27 rows(s) affected Records: 27 Duplicates: 0 Warnings: 0	0.063 sec
3449	13:14:05	ALTER TABLE job_history ADD FOREIGN KEY (employee_id) REFERENCES employees(employee_id)	10 rows(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
3450	13:14:07	ALTER TABLE job_history ADD FOREIGN KEY (job_id) REFERENCES jobs(job_id)	10 rows(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.047 sec
3451	13:14:07	ALTER TABLE job_history ADD FOREIGN KEY (department_id) REFERENCES departments(department_id)	10 rows(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.047 sec
3452	13:15:24	SELECT d.department_id, l.street_address, l.city, l.state_province, c.country_name FROM departments d JOIN... 27 rows(s) returned		0.000 sec / 0.000 sec
3453	13:17:30	SELECT e.first_name, e.last_name, e.department_id, d.department_name FROM employees e JOIN departments d ON e.department_id = d.department_id; 106 rows(s) returned		0.000 sec / 0.000 sec

Task 3 :

3. Write a query to find the name (first_name, last_name), job, department ID and name of the employees who works in London

Ans :

```
SELECT e.first_name, e.last_name, j.job_title, e.department_id, d.department_name
```

```
FROM employees e
```

```
JOIN departments d ON e.department_id = d.department_id
```

```
JOIN jobs j ON e.job_id = j.job_id
```

```
JOIN locations l ON d.location_id = l.location_id
```

```
WHERE l.city = 'London';
```

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with tables: departments, job_history, and job.
- SQL Editor:** Contains the query:

```
SELECT e.first_name, e.last_name, j.job_title, e.department_id, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.department_id
JOIN jobs j ON e.job_id = j.job_id
JOIN locations l ON d.location_id = l.location_id
WHERE l.city = 'London';
```
- Result Grid:** Displays the result of the query:

first_name	last_name	job_title	department_id	department_name
Susan	Mavris	Human Resources Representative	40	Human Resources
- Output:** Shows the log of executed statements, including the SELECT query and its execution details.
- System Bar:** Includes icons for file operations, search, and system status.

Task 4:

4. Write a query to find the employee id, name (last_name) along with their manager_id and name (last_name)

Ans :

```
SELECT e.employee_id, e.last_name AS employee_name, e.manager_id, m.last_name AS manager_name
```

```
FROM employees e
```

```
LEFT JOIN employees m ON e.manager_id = m.employee_id;
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL icons for connection, schema, table, etc.
- Navigator:** MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), Administration (Dashboard, Performance Reports, Performance Schema Setup), Schemas, Information.
- SQL Editor:** The query entered is:

```
ALTER TABLE job_history ADD FOREIGN KEY (employee_id) REFERENCES employees(employee_id);
ALTER TABLE job_history ADD FOREIGN KEY (job_id) REFERENCES jobs(job_id);
ALTER TABLE job_history ADD FOREIGN KEY (department_id) REFERENCES departments(department_id);
SELECT e.first_name, e.last_name, j.job_title, e.department_id, d.department_name FROM employees e JOIN departments d ON e.department_id = d.department_id
JOIN locations l ON d.location_id = l.location_id
WHERE l.city = 'London';
SELECT e.employee_id, e.last_name AS employee_name, e.manager_id, m.last_name AS manager_name FROM employees e LEFT JOIN employees m ON e.manager_id = m.employee_id;
```
- Result Grid:** Shows the results of the last query, which returns 11 rows of data:| employee_id | employee_name | manager_id | manager_name |
| --- | --- | --- | --- |
| 100 | King | NULL | NULL |
| 101 | Kochhar | 100 | King |
| 102 | De Haan | 100 | King |
| 103 | Hunold | 102 | De Haan |
| 104 | Ernst | 103 | Hunold |
| 105 | Austin | 103 | Hunold |
| 106 | Pataballa | 103 | Hunold |
| 107 | Lorentz | 103 | Hunold |
| 108 | Greenberg | 101 | Kochhar |
| 109 | Faviet | 108 | Greenberg |
| 110 | Chen | 108 | Greenberg |
| 111 | Sears | 108 | Greenberg |
- Output:** Shows the history of actions taken on the session, including the execution of the previous query and its duration.
- System Bar:** Icons for search, file operations, and system status (language, date).

Task 5:

5. Write a query to find the name (first_name, last_name) and hire date of the employees who was hired after 'Jones'

Ans :

```
SELECT first_name, last_name, hire_date
```

```
FROM employees
```

```
WHERE hire_date > (SELECT hire_date FROM employees WHERE last_name = 'Jones');
```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:

```
2645     JOIN locations l ON d.location_id = l.location_id
2646     WHERE l.city = 'London';
2647 •   SELECT e.employee_id, e.last_name AS employee_name, e.manager_id, m.last_name AS manager_name FROM employees e LEFT JOIN employees m ON e.manager_id = m.employee_id
2648 •   SELECT first_name, last_name, hire_date
2649     FROM employees
2650     WHERE hire_date > (SELECT hire_date FROM employees WHERE last_name = 'Jones'));
2651
```
- Result Grid:** Displays the results of the query, showing 24 rows of employee data:| first_name | last_name | hire_date |
| --- | --- | --- |
| Luis | Popp | 1999-12-07 |
| Karen | Colmenares | 1999-08-10 |
| Kevin | Morgos | 1999-11-16 |
| Steven | Markle | 2000-03-08 |
| TJ | Olson | 1999-04-10 |
| Ki | Gee | 1999-12-12 |
| Hazel | Phantaker | 2000-02-15 |
| Gerald | Cambrault | 1998-10-15 |
| Ebri | Zboray | 2000-11-29 |
| Oliver | Tuxaud | 1999-11-23 |
| Danielle | Grenie | 1999-03-19 |
| Mattea | Marvins | 2000-01-24 |
| Luis | Popp | 1999-12-07 |
| Karen | Colmenares | 1999-08-10 |
| Kevin | Morgos | 1999-11-16 |
| Steven | Markle | 2000-03-08 |
| TJ | Olson | 1999-04-10 |
| Ki | Gee | 1999-12-12 |
| Hazel | Phantaker | 2000-02-15 |
| Gerald | Cambrault | 1998-10-15 |
| Ebri | Zboray | 2000-11-29 |
| Oliver | Tuxaud | 1999-11-23 |
| Danielle | Grenie | 1999-03-19 |
| Mattea | Marvins | 2000-01-24 |
- Output:** Shows the history of actions taken in the session, including the execution of the query.

Task 6:

6. Write a query to get the department name and number of employees in the department

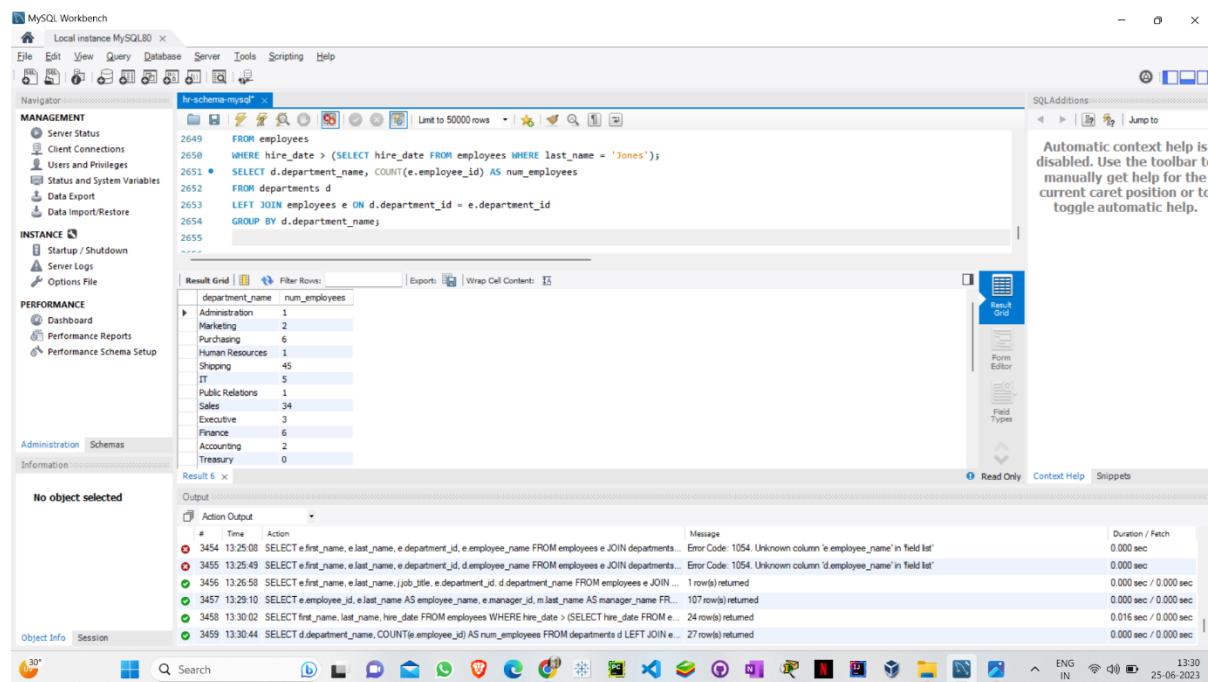
Ans :

```
SELECT d.department_name, COUNT(e.employee_id) AS num_employees
```

```
FROM departments d
```

```
LEFT JOIN employees e ON d.department_id = e.department_id
```

```
GROUP BY d.department_name;
```



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL icons.
- Navigator:** MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup).
- Database:** hr-schema-mysqld*
- SQL Editor:** Contains the following SQL query:

```
2649 FROM employees
2650 WHERE hire_date > (SELECT hire_date FROM employees WHERE last_name = 'Jones');
2651 • SELECT d.department_name, COUNT(e.employee_id) AS num_employees
2652 FROM departments d
2653 LEFT JOIN employees e ON d.department_id = e.department_id
2654 GROUP BY d.department_name;
2655
```
- Result Grid:** Shows the results of the query:

department_name	num_employees
Administration	1
Marketing	2
Purchasing	6
Human Resources	1
Shipping	45
IT	5
Public Relations	1
Sales	34
Executive	3
Finance	6
Accounting	2
Treasury	0
- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
3454	13:25:08	SELECT e.first_name, e.last_name, e.department_id, e.employee_name FROM employees e JOIN departments...	Error Code: 1054. Unknown column 'e.employee_name' in field list'	0.000 sec
3455	13:25:49	SELECT e.first_name, e.last_name, e.department_id, e.employee_name FROM employees e JOIN departments...	Error Code: 1054. Unknown column 'd.employee_name' in field list'	0.000 sec
3456	13:26:58	SELECT e.first_name, e.last_name, e.job_title, e.department_id, d.department_name FROM employees e JOIN ...	1 row(s) returned	0.000 sec / 0.000 sec
3457	13:29:10	SELECT e.employee_id, e.last_name AS employee_name, e.manager_id, m.last_name AS manager_name FR...	107 row(s) returned	0.016 sec / 0.000 sec
3458	13:30:02	SELECT first_name, last_name, hire_date FROM employees WHERE hire_date > (SELECT hire_date FROM e...	24 row(s) returned	0.016 sec / 0.000 sec
3459	13:30:44	SELECT d.department_name, COUNT(e.employee_id) AS num_employees FROM departments d LEFT JOIN e...	27 row(s) returned	0.000 sec / 0.000 sec
- System Bar:** Shows system status (CPU, RAM, Disk), search bar, and system icons.
- Bottom Bar:** Shows the date (25-06-2023), time (13:30), and language (ENG IN).

Task 7:

7. Write a query to display department name, name (first_name, last_name), hire date, salary of the manager for all managers whose experience is more than 15 years

Ans :

```
SELECT d.department_name, CONCAT(e.first_name, ' ', e.last_name) AS manager_name,  
e.hire_date, e.salary  
  
FROM employees e  
  
JOIN departments d ON e.department_id = d.department_id  
  
WHERE e.employee_id IN (  
  
    SELECT manager_id  
  
    FROM employees  
  
    WHERE DATEDIFF(CURDATE(), hire_date) > 15*365  
  
);
```

MySQL Workbench

Local instance MySQL80

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas Information

No object selected

Result Grid | Filter Rows | Export: | Wrap Cell Content: |

Result 7 x

Action Output

#	Time	Action	Message	Duration / Fetch
3455	13:25:49	SELECT e.first_name, e.last_name, e.department_id, d.employee_name FROM employees e JOIN departments... Error Code: 1054. Unknown column 'd.employee_name' in field list'		0.000 sec
3456	13:26:58	SELECT e.first_name, e.last_name, e.job_title, e.department_id, d.department_name FROM employees e JOIN ... 1 row(s) returned		0.000 sec / 0.000 sec
3457	13:29:10	SELECT e.employee_id, e.last_name AS employee_name, e.manager_id, m.last_name AS manager_name FR... 107 row(s) returned		0.000 sec / 0.000 sec
3458	13:30:02	SELECT first_name, last_name, hire_date FROM employees WHERE hire_date > (SELECT hire_date FROM e... 24 row(s) returned		0.016 sec / 0.000 sec
3459	13:30:44	SELECT d.department_name, COUNT(e.employee_id) AS num_employees FROM departments d LEFT JOIN e... 27 row(s) returned		0.000 sec / 0.000 sec
3460	13:32:18	SELECT d.department_name, CONCAT(e.first_name, ' ', e.last_name) AS manager_name, e.hire_date, e.salary... 18 row(s) returned		0.000 sec / 0.000 sec

Object Info Session

Result Grid | Form Editor | Field Types | Read Only | Context Help | Snippets

Output

ENG IN 13:32 25-06-2023

Task 8:

8. Write a query to find the name (first_name, last_name) and the salary of the employees who have a higher salary than the employee whose last_name='Bull'

Ans :

SELECT first_name, last_name, salary

FROM employees

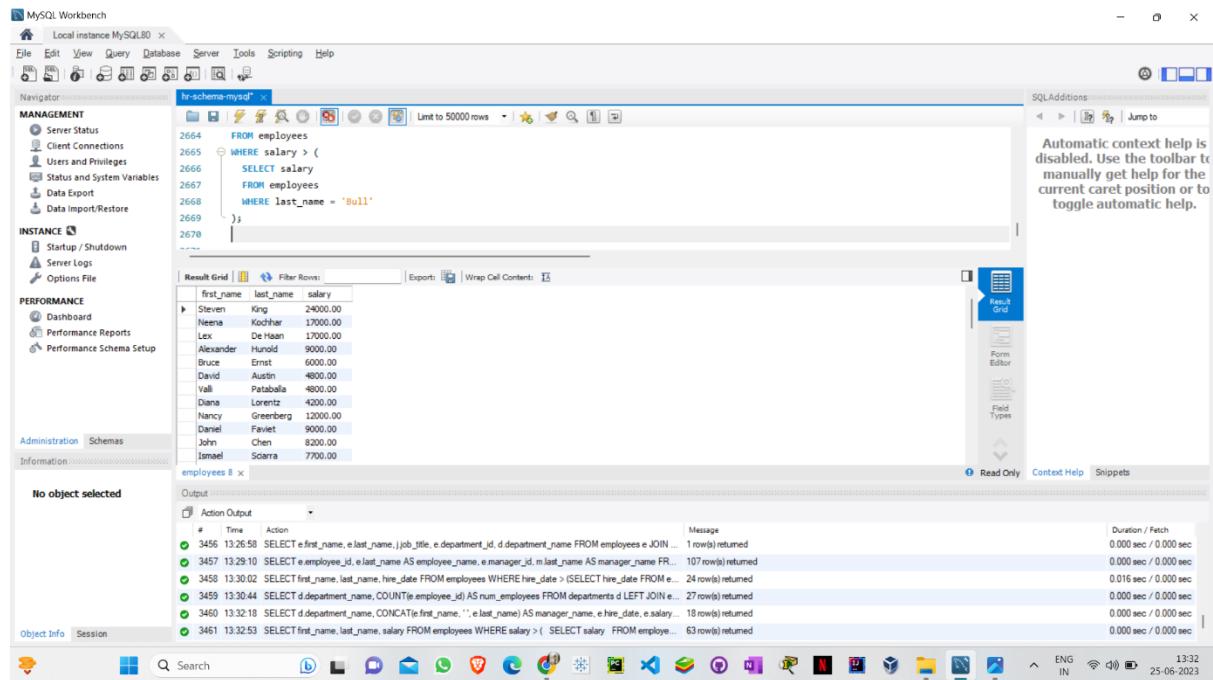
WHERE salary > (

SELECT salary

FROM employees

WHERE last_name = 'Bull'

);



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:

```
2664 FROM employees
2665 WHERE salary > (
2666     SELECT salary
2667     FROM employees
2668     WHERE last_name = 'Bull'
2669 )
2670
```
- Result Grid:** Displays the results of the query, listing employees with salaries higher than those of the employee named 'Bull'.

first_name	last_name	salary
Steven	King	24000.00
Neena	Kochhar	17000.00
Lex	De Haan	17000.00
Alexander	Hunold	9000.00
Bruce	Ernst	6000.00
David	Austin	4800.00
Valli	Pataballa	4800.00
Diana	Lorentz	4200.00
Nancy	Greenberg	12000.00
Daniel	Faviet	9000.00
John	Chen	8200.00
Ismael	Saara	7700.00
- Output Tab:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
3456	13:26:58	SELECT e.first_name, e.last_name, e.job_title, e.department_id, d.department_name FROM employees e JOIN ...	1 row(s) returned	0.000 sec / 0.000 sec
3457	13:29:10	SELECT e.employee_id, e.last_name AS employee_name, e.manager_id, m.last_name AS manager_name FR...	107 row(s) returned	0.000 sec / 0.000 sec
3458	13:30:02	SELECT first_name, last_name, hire_date FROM employees WHERE hire_date > (SELECT hire_date FROM e...	24 row(s) returned	0.016 sec / 0.000 sec
3459	13:30:44	SELECT d.department_name, COUNT(e.employee_id) AS num_employees FROM departments d LEFT JOIN e...	27 row(s) returned	0.000 sec / 0.000 sec
3460	13:32:18	SELECT d.department_name, CONCAT(e.first_name, ' ', e.last_name) AS manager_name, e.hire_date, e.salary...	18 row(s) returned	0.000 sec / 0.000 sec
3461	13:32:53	SELECT first_name, last_name, salary FROM employees WHERE salary > (SELECT salary FROM employee...	63 row(s) returned	0.000 sec / 0.000 sec

Task 9:

9. Write a query to find the name (first_name, last_name) of all employees who works in the IT department

Ans :

```
SELECT first_name, last_name
```

```
FROM employees
```

```
WHERE department_id = (
```

```
SELECT department_id
```

```
FROM departments
```

```
WHERE department_name = 'IT'
```

```
);
```

The screenshot shows the MySQL Workbench interface. The SQL editor window contains the following query:

```
2671 FROM employees
2672 WHERE department_id = (
2673     SELECT department_id
2674     FROM departments
2675     WHERE department_name = 'IT'
2676 )
2677
```

The results grid displays the following data:

first_name	last_name
Alexander	Hunold
Bruce	Eist
David	Austr
Valli	Patabala
Diana	Lorentz

The status bar at the bottom right shows the date and time as 25-06-2023 13:34:04.

Task 10:

10. Write a query to find the name (first_name, last_name) of the employees who have a manager and worked in a USA based department

Ans :

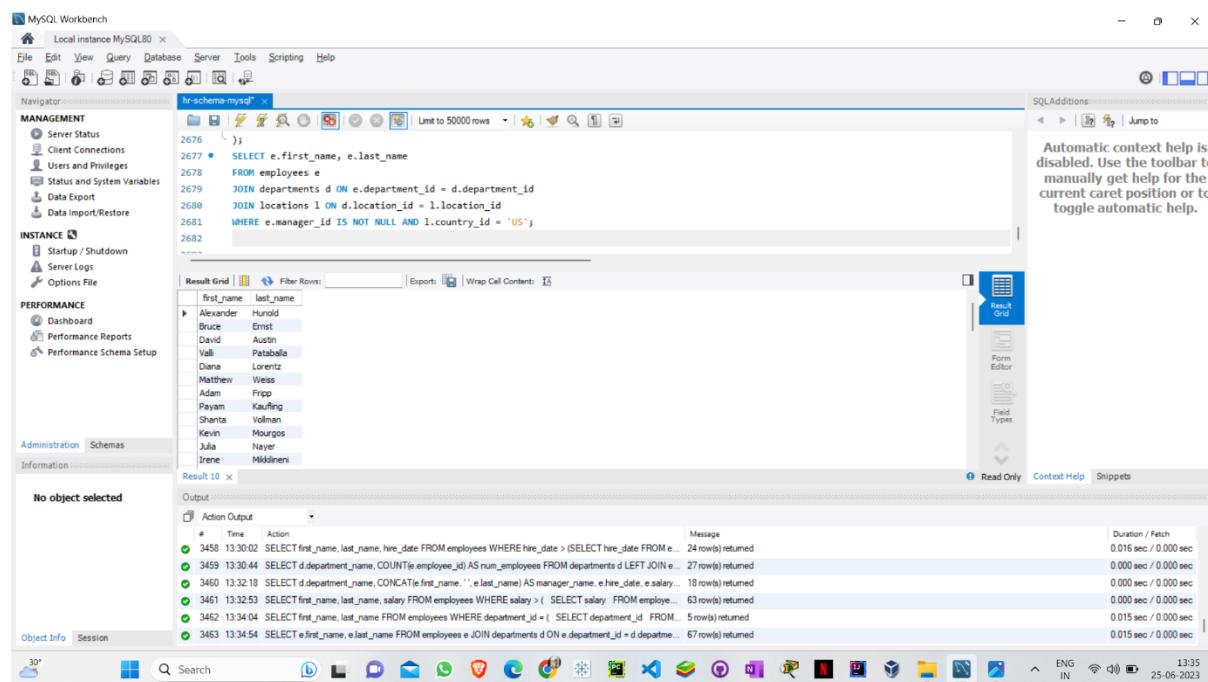
```
SELECT e.first_name, e.last_name
```

```
FROM employees e
```

```
JOIN departments d ON e.department_id = d.department_id
```

```
JOIN locations l ON d.location_id = l.location_id
```

```
WHERE e.manager_id IS NOT NULL AND l.country_id = 'US';
```



The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Contains the query code from Task 10.
- Result Grid:** Displays the results of the query, listing 14 employees with their first and last names.

first_name	last_name
Alexander	Hunold
Bruce	Eist
David	Adler
Valli	Pataballa
Diana	Lorentz
Matthew	Weiss
Adam	Fripp
Payam	Kaufung
Shanta	Vollman
Kevin	Mourgos
Juli	Nayer
Irene	Mikkilineni

- Action Output:** Shows the execution log with 14 entries, each detailing a SELECT or COUNT query and its duration.

#	Time	Action	Message	Duration / Fetch
3458	13:30:02	SELECT first_name, last_name, hire_date FROM employees WHERE hire_date > (SELECT hire_date FROM employees ORDER BY hire_date DESC LIMIT 1)	24 row(s) returned	0.016 sec / 0.000 sec
3459	13:30:44	SELECT d.department_name, COUNT(e.employee_id) AS num_employees FROM departments d LEFT JOIN employees e ON d.department_id = e.department_id	27 row(s) returned	0.000 sec / 0.000 sec
3460	13:32:18	SELECT d.department_name, CONCAT(first_name, ' ', last_name) AS manager_name, e.hire_date, e.salary FROM employees e JOIN departments d ON e.department_id = d.department_id	18 row(s) returned	0.000 sec / 0.000 sec
3461	13:32:53	SELECT first_name, last_name, salary FROM employees WHERE salary > (SELECT salary FROM employees ORDER BY salary DESC LIMIT 1)	63 row(s) returned	0.000 sec / 0.000 sec
3462	13:34:04	SELECT first_name, last_name FROM employees WHERE department_id = (SELECT department_id FROM departments ORDER BY department_id DESC LIMIT 1)	5 row(s) returned	0.015 sec / 0.000 sec
3463	13:34:54	SELECT e.first_name, e.last_name FROM employees e JOIN departments d ON e.department_id = d.department_id	67 row(s) returned	0.015 sec / 0.000 sec

Task 11:

11. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is greater than the average salary

Ans :

SELECT first_name, last_name, salary

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the SQL query:

```
2679 JOIN departments d ON e.department_id = d.department_id
2680 JOIN locations l ON d.location_id = l.location_id
2681 WHERE e.manager_id IS NOT NULL AND l.country_id = 'US'
2682 • SELECT first_name, last_name, salary
2683 FROM employees
2684 WHERE salary > (SELECT AVG(salary) FROM employees);
2685
```
- Result Grid:** Shows the results of the query, listing employees with salaries above the average:

first_name	last_name	salary
Steven	King	24000.00
Neena	Kochhar	17000.00
Lex	De Haan	17000.00
Alexander	Hunold	9000.00
Nancy	Greenberg	12000.00
Daniel	Faviet	9000.00
John	Chen	8200.00
Ismael	Scarra	7700.00
Jose Manuel	Urman	7800.00
Luis	Popp	6900.00
Den	Raphaely	11000.00
Matthew	Weiss	8000.00
- Output Tab:** Shows the execution log with 13 rows of data, each with a timestamp, action, message, and duration/fetch time.
- System Bar:** Includes icons for file operations, search, and system status (language, date).

Task 12:

12. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is equal to the minimum salary for their job grade

Ans :

```
SELECT first_name, last_name, salary
```

```
FROM employees
```

```
WHERE (job_id, salary) IN (
```

```
SELECT job_id, MIN(salary)
```

```
FROM employees
```

```
GROUP BY job_id
```

```
);
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup).
- SQL Editor:** A query window titled "hr-schema-mysqld" containing the following SQL code:

```
2686    FROM employees
2687 WHERE (job_id, salary) IN (
2688     SELECT job_id, MIN(salary)
2689     FROM employees
2690     GROUP BY job_id
2691   );
2692
```
- Result Grid:** A table showing the results of the query:

first_name	last_name	salary
Steven	King	24000.00
Neena	Kochhar	17000.00
Lex	De Haan	17000.00
Diana	Lorentz	42000.00
Nancy	Greenberg	12000.00
Luis	Popp	6900.00
Den	Raphaely	11000.00
Karen	Colmenares	25000.00
Kevin	Morgane	5800.00
TJ	Olson	2100.00
Eleni	Zlotkey	10500.00
Sundita	Kumar	6100.00
- Output Window:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
3460	13:32:18	Action	Message	0.000 sec / 0.000 sec
3461	13:32:53	SELECT d.department_name, CONCAT(e.first_name, ' ', e.last_name) AS manager_name, e.hire_date, e.salary...	18 row(s) returned	0.000 sec / 0.000 sec
3462	13:34:04	SELECT first_name, last_name, salary FROM employees WHERE salary > (SELECT salary FROM employee...	63 row(s) returned	0.015 sec / 0.000 sec
3463	13:34:54	SELECT e.first_name, e.last_name FROM employees e JOIN departments d ON e.department_id = d.department_id	67 row(s) returned	0.015 sec / 0.000 sec
3464	13:36:26	SELECT first_name, last_name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM emplo...	51 row(s) returned	0.000 sec / 0.000 sec
3465	13:37:38	SELECT first_name, last_name, salary FROM employees WHERE (job_id, salary) IN (SELECT job_id, MIN(sal...	21 row(s) returned	0.016 sec / 0.000 sec
- System Bar:** Shows weather (30°C, Mostly cloudy), search bar, and various system icons.
- Bottom Bar:** Shows the date (25-06-2023), time (13:37), and language (ENG IN).

Task 13:

13. Write a query to find the name (first_name, last_name), and salary of the employees who earns more than the average salary and works in any of the IT departments

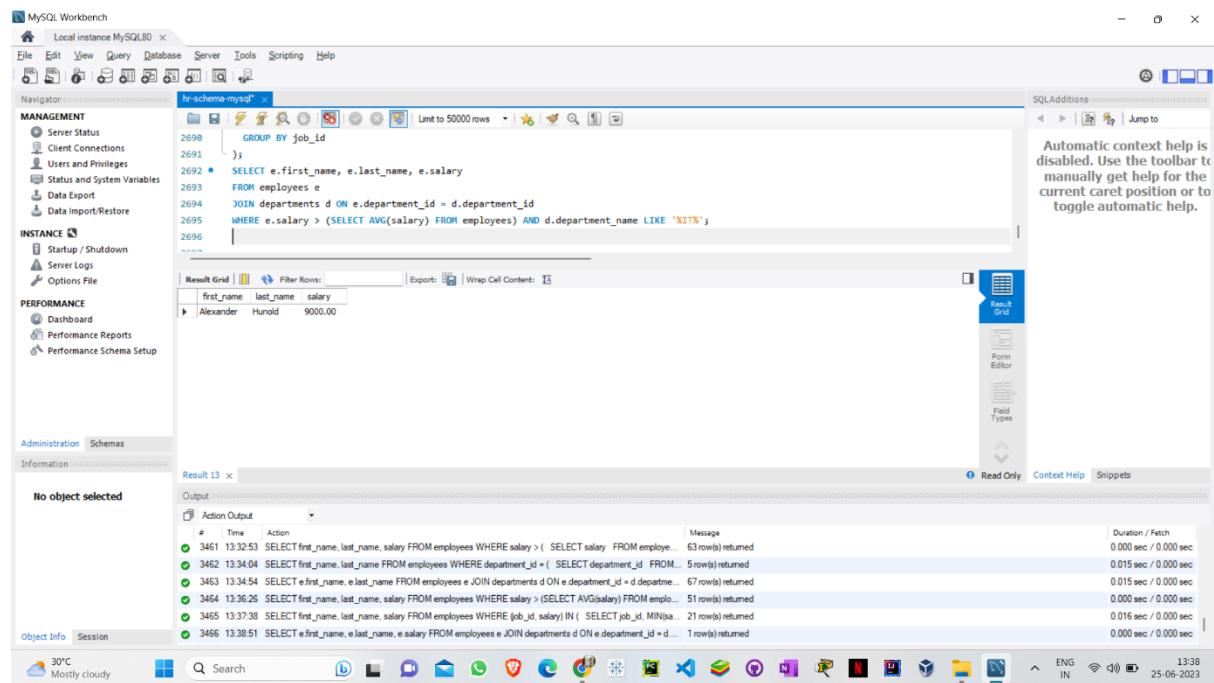
Ans :

```
SELECT e.first_name, e.last_name, e.salary
```

```
FROM employees e
```

```
JOIN departments d ON e.department_id = d.department_id
```

```
WHERE e.salary > (SELECT AVG(salary) FROM employees) AND d.department_name  
LIKE '%IT%';
```



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the SQL query:

```
2698 GROUP BY job_id
2699 );
2700 •   SELECT e.first_name, e.last_name, e.salary
2701   FROM employees e
2702   JOIN departments d ON e.department_id = d.department_id
2703   WHERE e.salary > (SELECT AVG(salary) FROM employees) AND d.department_name LIKE '%IT%';
2704
2705
```
- Result Grid:** Shows the result of the query:

first_name	last_name	salary
Alexander	Hunold	9000.00
- Output Window:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
3461	13:32:53	SELECT first_name, last_name, salary FROM employees WHERE salary > (SELECT salary FROM employees)	63 row(s) returned	0.000 sec / 0.000 sec
3462	13:34:04	SELECT first_name, last_name FROM employees WHERE department_id = (SELECT department_id FROM departments)	5 row(s) returned	0.015 sec / 0.000 sec
3463	13:34:54	SELECT e.first_name, e.last_name FROM employees e JOIN departments d ON e.department_id = d.department_id	67 row(s) returned	0.015 sec / 0.000 sec
3464	13:36:26	SELECT first_name, last_name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM employees)	51 row(s) returned	0.000 sec / 0.000 sec
3465	13:37:38	SELECT first_name, last_name, salary FROM employees WHERE job_id, salary IN (SELECT job_id, MIN(salary) FROM employees)	21 row(s) returned	0.016 sec / 0.000 sec
3466	13:38:51	SELECT e.first_name, e.last_name, e.salary FROM employees e JOIN departments d ON e.department_id = d.department_id	1 row(s) returned	0.000 sec / 0.000 sec

Task 14:

Write a query to find the name (first_name, last_name), and salary of the employees who earn the same salary as the minimum salary for all departments.

Ans :

```
SELECT e.first_name, e.last_name, e.salary  
FROM employees e  
WHERE e.salary = (SELECT MIN(salary) FROM employees);
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** hr-schema-mysq80, MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup).
- SQL Editor:** SQLAdditions, SQL code:

```
2693 FROM employees e  
2694 JOIN departments d ON e.department_id = d.department_id  
2695 WHERE e.salary > (SELECT AVG(salary) FROM employees) AND d.department_name LIKE 'KITS';  
2696 • SELECT e.first_name, e.last_name, e.salary  
2697 FROM employees e  
2698 WHERE e.salary = (SELECT MIN(salary) FROM employees);  
2699
```
- Result Grid:** Result Grid tab, first_name, last_name, salary, data:

first_name	last_name	salary
TJ	Olson	2100.00
- Output Tab:** Action Output, showing log entries for the executed queries.
- System Bar:** 30°C Mostly cloudy, Search, various icons, ENG IN, 25-06-2023, 13:39.

Task 15:

Write a query to find the name (first_name, last_name) and salary of the employees who earn a salary that is higher than the salary of all the Shipping Clerk (JOB_ID = 'SH_CLERK'). Sort the results of the salary of the lowest to highest

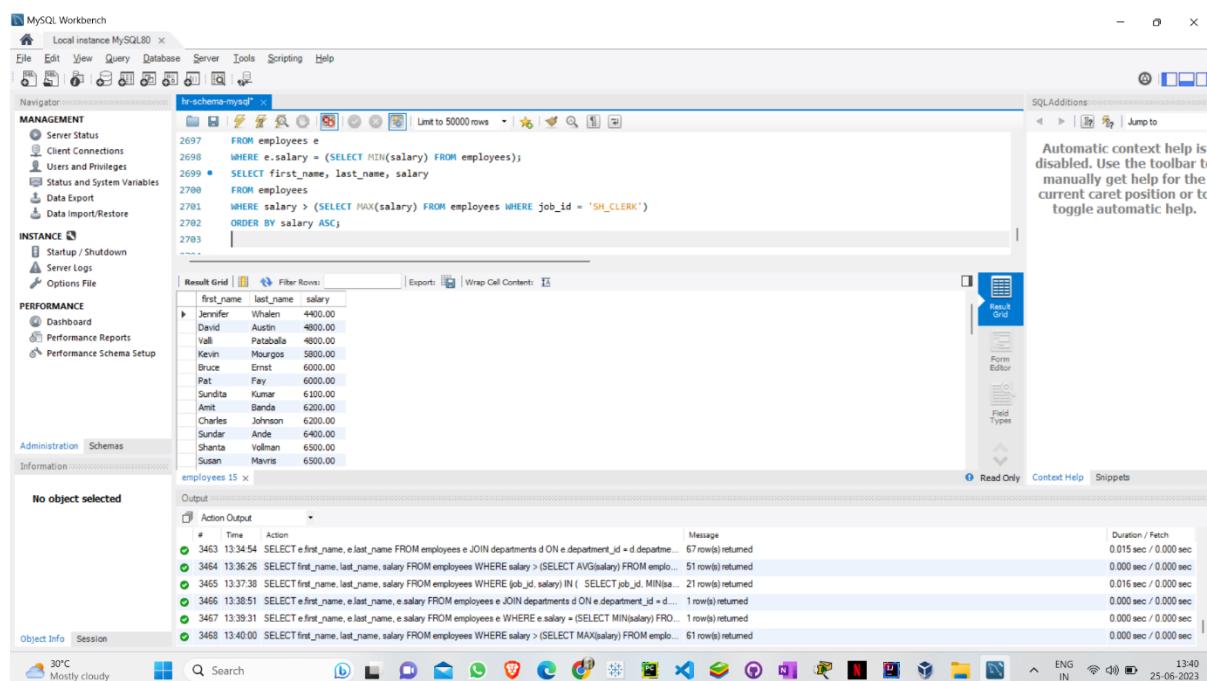
Ans :

```
SELECT first_name, last_name, salary
```

```
FROM employees
```

```
WHERE salary > (SELECT MAX(salary) FROM employees WHERE job_id =  
'SH_CLERK')
```

```
ORDER BY salary ASC;
```



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard database management icons.
- Navigator:** hr-schema-myisam
- SQL Editor:** Contains the query:

```
2697 FROM employees e  
2698 WHERE e.salary = (SELECT MIN(salary) FROM employees);  
2699 • SELECT first_name, last_name, salary  
2700 FROM employees  
2701 WHERE salary > (SELECT MAX(salary) FROM employees WHERE job_id = 'SH_CLERK')  
2702 ORDER BY salary ASC;  
2703 |
```
- Result Grid:** Shows the results of the query:

first_name	last_name	salary
Jennifer	Whalen	4400.00
David	Austin	4800.00
Valli	Pataballa	4800.00
Kevin	Morgulow	5800.00
Bruce	Ernst	6000.00
Pat	Fay	6000.00
Sundita	Kumar	6100.00
Amzi	Banda	6200.00
Charles	Johnson	6200.00
Sundar	Ande	6400.00
Shanta	Vollman	6500.00
Susan	Mavris	6500.00
- Output Tab:** Action Output, showing the execution log:

#	Time	Action	Message	Duration / Fetch
3453	13:34:54	SELECT e.first_name, e.last_name FROM employees e JOIN departments d ON e.department_id = d.department_id	67 row(s) returned	0.015 sec / 0.000 sec
3464	13:36:26	SELECT first_name, last_name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM employees)	51 row(s) returned	0.000 sec / 0.000 sec
3465	13:37:38	SELECT first_name, last_name, salary FROM employees WHERE (job_id, salary) IN (SELECT job_id, MIN(salary) FROM employees GROUP BY job_id)	21 row(s) returned	0.016 sec / 0.000 sec
3466	13:38:51	SELECT e.first_name, e.last_name, e.salary FROM employees e JOIN departments d ON e.department_id = d.department_id	1 row(s) returned	0.000 sec / 0.000 sec
3467	13:39:31	SELECT e.first_name, e.last_name, e.salary FROM employees e WHERE e.salary > (SELECT MIN(salary) FROM employees)	1 row(s) returned	0.000 sec / 0.000 sec
3468	13:40:00	SELECT first_name, last_name, salary FROM employees WHERE salary > (SELECT MAX(salary) FROM employees)	61 row(s) returned	0.000 sec / 0.000 sec
- Object Info and Session Tabs.**
- System Bar:** Weather (30°C), Search, Taskbar icons, Language (ENG IN), Date (25-06-2023), Time (13:40).

CHAPTER 5

CONCLUSION

In traversing the realms of the HR Database through the course with a real-time project at Board Infinity, I've embarked on a transformative journey that has left an indelible mark on my understanding of Database Management System (DBMS) and Structured Query Language (SQL). This expedition has underscored the significance of practical learning and equipped me with an arsenal of skills that transcend the confines of a mere curriculum.

Through the diverse tables of the HR Database, I've witnessed the intricate dance of data relationships, recognizing how entities interweave to form the backbone of organizational information. This experience has illuminated the power of efficient data storage, retrieval, and manipulation, showcasing how SQL queries can unveil insights hidden within the labyrinth of information.

Challenges encountered along the way served as catalysts for growth. From deciphering complex JOINs to optimizing query execution plans, each hurdle propelled me to delve deeper, expanding my problem-solving repertoire. Collaborative engagement with peers, resources, and research became the compass guiding me through uncharted territories of data analysis.

The culmination of this voyage has led to an array of transformative learning outcomes. Proficiency in crafting precise and impactful SQL queries now resides in my toolkit, ready to dissect data puzzles with finesse. The profound understanding of database design principles empowers me to sculpt efficient structures that can withstand the test of time and usage.

As I reflect on this expedition, I am struck by the universality of data's influence across industries. The HR Database is not just a collection of tables, but a testament to the power of data to shape decisions, drive innovation, and elevate operations. With newfound expertise, I am well-prepared to embark on further adventures, tackling data challenges and unraveling insights that illuminate the path forward.

In conclusion, the HR Database has been more than a digital playground; it's been a laboratory of learning and discovery. As the digital landscape continues to evolve, I am armed with the skills to navigate and thrive, harnessing the potential of DBMS and SQL to transform data into knowledge and opportunity. This journey with Board Infinity has been transformative, equipping me to embark on new horizons where data is not just a commodity but a conduit to boundless growth.

CHAPTER 6

REFERENCES

During the course and project completion, I drew inspiration and guidance from a variety of sources that enriched my understanding and skills in Database Management System (DBMS) and Structured Query Language (SQL). I would like to acknowledge the following references that played a significant role in shaping my learning journey:

1. Course Material and Instructors at Board Infinity: The course content and guidance provided by the instructors at Board Infinity were instrumental in enhancing my knowledge of DBMS and SQL. Their expertise and support greatly contributed to my learning experience.

<https://www.boardinfinity.com/> (accessed on June 5)

2. Online Tutorials and Documentation: Various online platforms, such as tutorials on SQL syntax, query optimization, and database design principles, provided valuable insights and solutions to specific challenges encountered during the project.

3. Peer Collaborations: Collaborative discussions with fellow learners allowed for the exchange of ideas, problem-solving approaches, and peer reviews of SQL queries and database designs.

4. Stack Overflow and Online Communities: Online communities like Stack Overflow proved invaluable when seeking solutions to specific technical challenges and gaining insights from experienced professionals.

<https://stackoverflow.com/> (accessed on July 4)

5. LinkedIn Learning and Other E-learning Platforms: Supplementary courses and modules from platforms like LinkedIn Learning provided additional perspectives on database management and SQL practices.

<https://www.linkedin.com/feed/> (accessed on July 5)

6. Project Data Set Source: The HR Database, sourced from Kaggle, served as the foundation for my real-time project. The dataset's creator and contributors provided a rich dataset for hands-on learning

<https://www.kaggle.com/datasets/sirajahmad/hr-schema-mysqls> (accessed on July 1)

7. Industry Professionals: Discussions with professionals in the field of database management and software development provided practical insights into real-world applications and industry trends.

These references collectively shaped my learning journey and provided me with a diverse range of perspectives, resources, and guidance that contributed to my proficiency in DBMS and SQL.