SIT103/SIT772: Database Fundamentals

5.1P: Basic SQL – SELECT Queries

Overview

In this task, you will start learning SQL using MySQL Workbench environment you set up in Week 1 (Task 1.2P). You will learn using the Data Manipulation Language (DML) command of SELECT query to retrieve required data from single tables. It is assumed that you have successfully installed MySQL server and MySQL Workbench and set up the environment in Week 1 (Task 1.2P). If you have not done so already, please complete it first, Task 1.2P task sheet helps you with setting up the MySQL environment.

After completing Task 1.2P, you have a database called 'SIT103' or 'SIT772' in you MySQL environment. First, you will create some tables in the database and populate them with some data to practice SELECT queries to retrieve data from the tables. You can open two '.sql' files provided in the '*Task5_1.zip*' file as part of the task resources in MySQL Workbench.

- *tables.sql* – has SQLs to create tables; and
- *data.sql* – has SQLs to add records in the tables
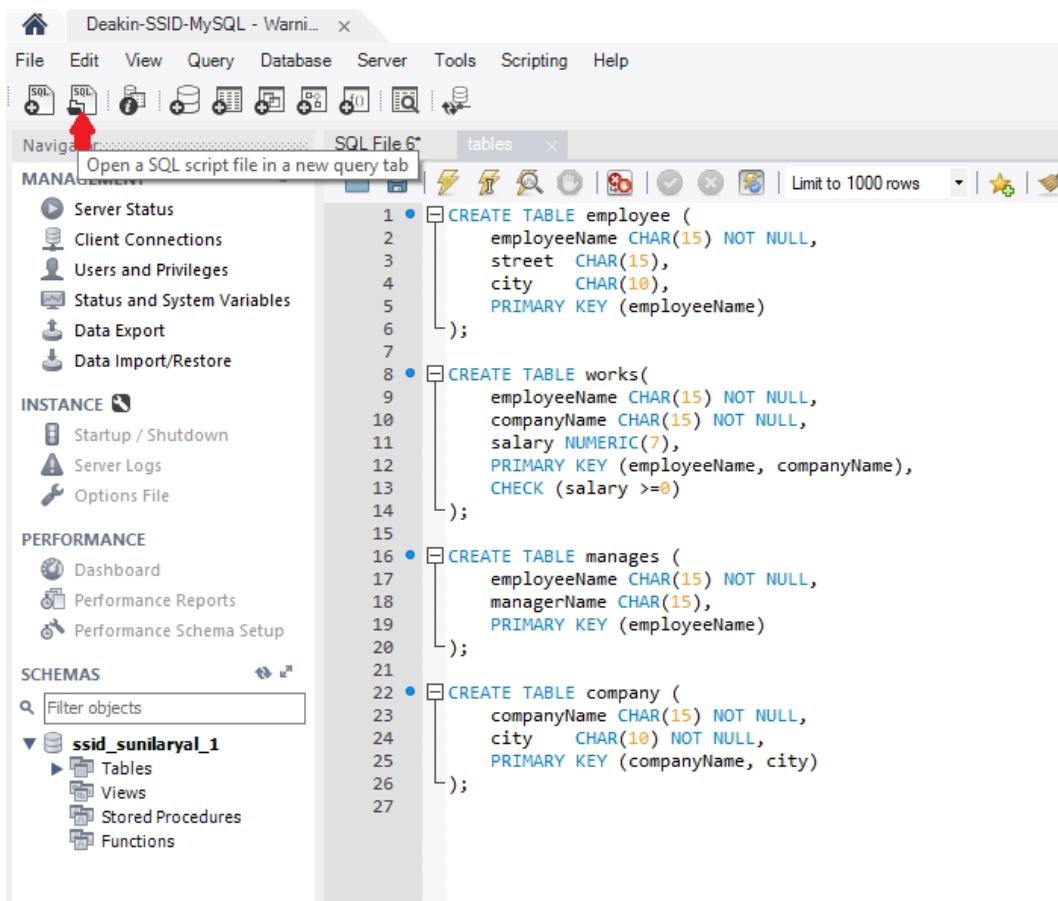
SQL commands in the two files are also provided at the end of this task sheet just in case if you can't open the zip file. First, run SQLs in 'tables.sql' to create tables and then populate data in the tables using SQLs in data.sql. Now you will have the following four new tables in your database.

Employee Table

employeeName	street	city
Jones	Main	Harrison
Smith	North	Rye
Hayes	Main	Harrison
Curry	North	Rye
Lindsay	Park	Pittsfield
Turner	Putnam	Stamford
Williams	Nassau	Princeton
Adams	Spring	Pittsfield

Company Table

CompanyName	City
Waltons	Harrison
Meyer	Rye
Waltons	Rye
Woolworths	Pittsfield
Tweeties	Harrison
Firebrand	Woodside

Works Table

EmployeeName	CompanyName	Salary
Jones	Tweeties	21000
Smith	Waltons	22000
Hayes	Woolworths	19000
Curry	Meyer	25000
Lindsay	Meyer	9000
Turner	Firebrand	20000
Williams	Tweeties	18000
Adams	Meyer	22000

Manages Table

EmployeeName	ManagerName
Jones	Collins
Smith	Collins
Hayes	Wills
Curry	Wills
Lindsay	Mulhare
Turner	Mulhare
Williams	Bond
Adams	Bond

Note that if you have tables already in your database with the same names, you will get error messages. In that case, you have to delete existing tables first using DROP TABLE commands. For example:

DROP TABLE Employee;

Tasks to do

1. Write a SQL query to retrieve names (displayed as "Employee Name") and salary of employees. [*Relevant table: Works*]

2. Write a SQL query to list name, street, and city of employees in descending order by their names. [*Relevant table: Employee*]

3. Write a SQL query to get a list of unique streets from the Employee table. [*Relevant table: Employee*]

4. Write a SQL query to list all records in the works table in descending order of company names and within a company in ascending order by employee name. [*Relevant table: Works*]

5. Write a SQL query to list name and salary of all employees who work in Meyer and sort the records in

ascending order by their incomes. [*Relevant table: Works*]

6. Assuming that the salary in the Works table is annual salary, write a SQL query to retrieve names (displayed as "Employee Name") and monthly salary as "Monthly Salary" of employees. [*Relevant table: Works*]

7. Write a SQL query to list names and salaries of all employees who work in Meyer and earn more than 20000. [*Relevant table: Works*]

8. Write a SQL query to list names and companies of the employees who earn in the range of 20000 to 25000 (inclusive). [*Relevant table: Works*]

9. Write a SQL query to list names of employees whose managers have "ll" (double ls) in their names. [*Relevant table: Manages*]

10. Write a SQL query to list company names and the average salary of their employees. [*Relevant table: Works*]

11. Write a SQL query to list the name of the companies with average salary of employees more than or equal to 20000. [*Relevant table: Works*]

12. Write a SQL query to select details of the employees who works in companies located in Rye. [*Relevant tables: Works and Company; Hint: use a subquery*]

13. Write a SQL query find the number of rows in the Manages table. [*Relevant tables: Manages; Hint: use* `COUNT()`]

14. Write a SQL query to find the name and company of the employee earning the highest salary. [*Relevant tables: Works*; *Hint: use a subquery using* `max()` *to find the highest salary. Please do not use 'WHERE salary=25000' as it is the highest salary in this case. Hope you can understand that it is not possible if there are millions of records. We want you to learn how to find it with a query.*]

Submission Requirements:

Submit one PDF/WORD file with the **SQL statement and screenshot of result** for each question above.

Submission Due

The due for each task has been stated via its OnTrack task information dashboard.

Appendix

1. tables.sql

```sql
CREATE TABLE employee (
    employeeName CHAR(15) NOT NULL,
    street      CHAR(15),
    city CHAR(10),
    PRIMARY KEY (employeeName)
);

CREATE TABLE works(
    employeeName CHAR(15) NOT NULL,
    companyName CHAR(15) NOT NULL,
    salary NUMERIC(7),
    PRIMARY KEY (employeeName, companyName),
    CHECK (salary >=0)
);

CREATE TABLE manages (
    employeeName CHAR(15) NOT NULL,
    managerName      CHAR(15),
    PRIMARY KEY (employeeName)
);

CREATE TABLE company (
    companyName CHAR(15) NOT NULL,
    city CHAR(10) NOT NULL,
    PRIMARY KEY (companyName, city)
);
```

2. data.sql

```sql
-- insert data in the employee table
INSERT INTO employee VALUES ('Jones', 'Main', 'Harrison');
INSERT INTO employee VALUES ('Smith', 'North', 'Rye');
INSERT INTO employee VALUES ('Hayes', 'Main', 'Harrison');
INSERT INTO employee VALUES ('Curry', 'North', 'Rye');
INSERT INTO employee VALUES ('Lindsay', 'Park', 'Pittsfield');
INSERT INTO employee VALUES ('Turner', 'Putname', 'Stamford');
INSERT INTO employee VALUES ('Williams', 'Nassus', 'Princeton');
INSERT INTO employee VALUES ('Adams', 'Spring', 'Pittsfield');

-- insert data in the company table
INSERT INTO company VALUES('Waltons', 'Harrison');
INSERT INTO company VALUES('Meyer', 'Rye');
```

```
INSERT INTO company VALUES('Waltons', 'Rye');
INSERT INTO company VALUES('Woolworths', 'Pittsfield');
INSERT INTO company VALUES('Tweeties', 'Harrison');
INSERT INTO company VALUES('Firebrand', 'Woodside');

-- insert data in the works table
INSERT INTO works VALUES ('Jones', 'Tweeties', 21000);
INSERT INTO works VALUES ('Smith', 'Waltons', 22000);
INSERT INTO works VALUES ('Hayes', 'Woolworths', 19000);
INSERT INTO works VALUES ('Curry', 'Meyer', 25000);
INSERT INTO works VALUES ('Lindsay', 'Meyer', 9000);
INSERT INTO works VALUES ('Turner', 'Firebrand', 20000);
INSERT INTO works VALUES ('Williams', 'Tweeties', 18000);
INSERT INTO works VALUES ('Adams', 'Meyer', 22000);

-- insert data in the works table
INSERT INTO manages VALUES ('Jones', 'Collins');
INSERT INTO manages VALUES ('Smith', 'Collins');
INSERT INTO manages VALUES ('Hayes', 'Wills');
INSERT INTO manages VALUES ('Curry', 'Wills');
INSERT INTO manages VALUES ('Lindsay', 'Mulhare');
INSERT INTO manages VALUES ('Turner', 'Mulhare');
INSERT INTO manages VALUES ('Williams', 'Bond');
INSERT INTO manages VALUES ('Adams', 'Bond');
```