E/16/156

H.A.D.T.T. Jayathilaka

Department of Computer Engineering

University of Peradeniya

CO527 Advanced Database Systems

Lab Task :

Refer to the Company ER diagram shown in Figure 1 and create a database named Company. All the questions listed below are based on this database.

1. **Load data to each of the tables from the given .sql files. It should have the following mentioned number of records for each table if the import was successful.**

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| test               |
+--------------------+
5 rows in set (0.00 sec)
```

Initially there were five databases. Created company database using following commands and it is shown in below figure.

- Create database company;
- Show databases;

```
MariaDB [(none)]> Create database company;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| company            |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| test               |
+--------------------+
6 rows in set (0.00 sec)
```

To use this database for further implementation, use below code.

- Use company;

Then tables were created according to the given ER diagram.

```
MariaDB [(none)]> use company;
Database changed
MariaDB [company]> CREATE TABLE employees (
    ->         emp_no int,
    -> birth_date date,
    -> first_name varchar(14),
    -> last_name varchar(16),
    -> sex enum('M','F'),
    -> hire_date date,
    -> primary key (emp_no)
    -> );
Query OK, 0 rows affected (0.25 sec)
```

```
MariaDB [company]>
MariaDB [company]> CREATE TABLE departments (
    ->         dept_no char(4),
    -> dept_name varchar(40),
    -> primary key (dept_no)
    -> );
Query OK, 0 rows affected (0.28 sec)
```

```
MariaDB [company]>
MariaDB [company]> CREATE TABLE dept_manager (
    ->         emp_no int,
    -> dept_no char(4),
    -> from_date date,
    -> to_date date,
    ->         primary key (dept_no,emp_no),
    -> foreign key (dept_no) references departments (dept_no),
    -> foreign key (emp_no) references employees (emp_no)
    -> );
Query OK, 0 rows affected (0.23 sec)
```

```
MariaDB [company]>
MariaDB [company]> CREATE TABLE titles (
    ->         emp_no int,
    -> title varchar(50),
    -> from_date date,
    -> to_date date,
    ->         primary key (emp_no,title,from_date,to_date),
    -> foreign key (emp_no) references employees (emp_no)
    -> );
Query OK, 0 rows affected (0.23 sec)
```

```
MariaDB [company]>
MariaDB [company]> CREATE TABLE salaries (
    ->         emp_no int,
    -> salary int,
    -> from_date date,
    -> to_date date,
    ->         primary key (emp_no,from_date,to_date),
    -> foreign key (emp_no) references employees (emp_no)
    -> );
Query OK, 0 rows affected (0.27 sec)
```

```
MariaDB [company]>
MariaDB [company]> CREATE TABLE dept_emp (
    ->         emp_no int,
    -> dept_no char(4),
    -> from_date date,
    -> to_date date,
    ->         primary key (dept_no,emp_no),
    -> foreign key (dept_no) references departments (dept_no),
    -> foreign key (emp_no) references employees (emp_no)
    -> );
Query OK, 0 rows affected (0.28 sec)
```

Then stored all the .sql files in C:\xampp\mysql\bin directory. Using below codes, load the data in to tables.

- mysql -u root company < load_employees.sql
- mysql -u root company < load_departments.sql
- mysql -u root company < load_dept_emp.sql
- mysql -u root company < load_dept_manager.sql
- mysql -u root company < load_dept_salaries1.sql
- mysql -u root company < load_dept_salaries2.sql
- mysql -u root company < load_titles.sql

```
MariaDB [company]>
MariaDB [company]> select count(*) from employees;
+----------+
| count(*) |
+----------+
|   300024 |
+----------+
1 row in set (0.48 sec)

MariaDB [company]> select count(*) from dept_manager;
+----------+
| count(*) |
+----------+
|       24 |
+----------+
1 row in set (0.05 sec)

MariaDB [company]> select count(*) from dept_emp;
+----------+
| count(*) |
+----------+
|   331603 |
+----------+
1 row in set (0.38 sec)

MariaDB [company]> select count(*) from titles;
+----------+
| count(*) |
+----------+
|   443308 |
+----------+
1 row in set (0.38 sec)

MariaDB [company]> select count(*) from salaries;
+----------+
| count(*) |
+----------+
|  1876717 |
+----------+
1 row in set (0.91 sec)
```

```
MariaDB [company]> select count(*) from departments;
+----------+
| count(*) |
+----------+
|        9 |
+----------+
1 row in set (0.02 sec)
```

2. **Find the top 10 family names(last_name) in the company.**

```
11      -- Question 2
12      SELECT last_name,COUNT(last_name) AS last_name_count
13      FROM employees
14      GROUP BY last_name
15      ORDER BY last_name_count DESC
16      LIMIT 10;
17
```

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

last_name	last_name_count
Baba	226
Coorg	223
Gelosh	223
Sudbeck	222
Farris	222
Adachi	221
Osgood	220
Mandell	218
Masada	218
Neiman	218

3. **List the number of Engineers each department has.**

```
18      -- Question 3
19      SELECT  departments.dept_name,
20      count(employees.emp_no) AS 'Number of Engineers'
21      FROM departments , dept_emp ,employees ,titles
22      WHERE departments.dept_no=dept_emp.dept_no AND
23      dept_emp.emp_no=employees.emp_no AND
24      employees.emp_no=titles.emp_no AND
25      titles.title='Engineer' AND
26      titles.to_date > curdate()
27      GROUP BY departments.dept_no;
```

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

dept_name	Number of Engineers
Production	13325
Development	15677
Quality Management	3744
Research	830
Customer Service	627

4. **List all the female employees who are department managers and have worked as a senior engineer.**

```
29      -- Question 4
30  •   SELECT employees.emp_no as ID,concat(first_name," ",last_name) AS emp_name
31      FROM employees,dept_manager,titles
32      WHERE employees.emp_no = dept_manager.emp_no
33      AND employees.sex="F" AND titles.title = "Senior Engineer" ;
```

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

ID	emp_name
110344	Rosine Cools
110800	Sanjoy Quadeer

5. **Display the departments and titles of employees who have a salary greater than 115000. Display how many of such employees work for each department.**

```
36      -- Question 5
37      -- Display the departments and titles of employees who have a salary greater than 115000
38  •  SELECT departments.dept_no , dept_name , title
39     FROM departments, salaries, employees, titles, dept_emp
40     WHERE dept_emp.emp_no = salaries.emp_no
41     AND salaries.salary > 115000
42     AND dept_emp.emp_no = titles.emp_no
43     AND departments.dept_no = dept_emp.dept_no
44     AND employees.emp_no = dept_emp.emp_no
45     AND salaries.to_date>=CURDATE()
46     AND titles.to_date>=CURDATE()
47     AND dept_emp.to_date>=CURDATE()
48     ORDER BY departments.dept_no;
```

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

dept_no	dept_name	title
d001	Marketing	Senior Staff
d001	Marketing	Senior Staff
d001	Marketing	Senior Staff
d001	Marketing	Senior Staff
d001	Marketing	Senior Staff
d001	Marketing	Senior Staff
d001	Marketing	Senior Staff
d001	Marketing	Senior Staff

```
49
50      -- Display how many of such employees work for each depart- ment.
51 •    SELECT  dept_emp.dept_no AS 'Department_Number' ,
52      COUNT(dept_emp.emp_no) AS 'No of Employees'
53      FROM employees,salaries,dept_emp,titles,departments
54      WHERE employees.emp_no=salaries.emp_no AND
55      salaries.salary>115000 AND
56      salaries.to_date >= curdate() AND
57      dept_emp.to_date >=curdate() AND
58      employees.emp_no=titles.emp_no AND
59      employees.emp_no=dept_emp.emp_no
60      GROUP BY Department_Number
61      ORDER BY departments.dept no ASC;
```

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

Department_Number	No of Employees
d001	4374
d002	3222
d003	171
d004	1269
d005	1566
d006	144
d007	29430
d008	441

6. **Assume that the company wants to reward the most senior employees who are more than 50 years of age and have contributed to the company for more than 10 years. Who is on the list? Display employee name, age, years of service in the company and joined date.**

```
43      -- Question 6
44 •    SELECT concat(employees.first_name," ",employees.last_name) AS employee_name,
45      TIMESTAMPDIFF(YEAR,employees.birth_date,CURDATE()) AS age ,
46      TIMESTAMPDIFF(YEAR,employees.hire_date,CURDATE()) as years_of_service ,
47      employees.hire_date AS joined_date
48      FROM employees,titles
49      WHERE TIMESTAMPDIFF(YEAR,employees.birth_date,CURDATE()) > 50 AND
50      employees.emp_no=titles.emp_no AND
51      TIMESTAMPDIFF(YEAR,employees.hire_date,CURDATE())>10;
```

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

employee_name	age	years_of_service	joined_date
Georgi Facello	69	36	1986-06-26
Bezalel Simmel	58	36	1985-11-21
Parto Bamford	62	36	1986-08-28
Chirstian Koblick	68	35	1986-12-01
Chirstian Koblick	68	35	1986-12-01
Kyoichi Maliniak	67	33	1989-09-12
Kyoichi Maliniak	67	33	1989-09-12
Anneke Preusig	69	33	1989-06-02
Tzvetan Zielinski	65	33	1989-02-10
Tzvetan Zielinski	65	33	1989-02-10

7. **Find all the names (first name + last name) of employees in the database who do not work in the Human Resources department. Assume that all the people work for exactly one department.**

```
53      -- Question 7
54  •   SELECT concat(employees.first_name," ",employees.last_name) AS employees_doesnot_work_in_HR_dept
55      FROM employees,departments,dept_emp
56      WHERE departments.dept_name!='Human Resources' AND
57      departments.dept_no=dept_emp.dept_no AND
58      dept_emp.emp_no=employees.emp_no;
```

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

employees_doesnot_work_in_HR_dept
Cristinel Bouloucos
Georgy Dredge
Berhard McFarlin
Lunjin Giveon
Yucel Auria
Aleksandar Ananiadou
Xiping Klerer
Karoline Cesareni
Nikolaos Llado
Susanna Vesel
Djelloul Laventhal
Phule Hammerschmidt

8. **Find the names of all employees in the database who earn more than every employee in the Finance department. Assume that all people work for at most one company.**

```
60      -- Question 8
61  •  SELECT employees.first_name , employees.last_name
62     FROM employees,salaries
63     WHERE employees.emp_no IN
64     (SELECT salaries.emp_no FROM salaries )
65     AND salaries.salary >
66     (SELECT MAX(salaries.salary) FROM
67     salaries,departments,dept_emp,employees
68     WHERE departments.dept_name='Finance' AND
69     departments.dept_no=dept_emp.dept_no AND
70     dept_emp.emp_no=employees.emp_no AND
71     employees.emp_no=salaries.emp_no);
```

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

first_name	last_name
Guenter	Hatcliff
Krassimir	Linares
Luise	Tchuente
Luerbio	Itschner
Nobuyoshi	Arlazarov
Wonhee	Perl
Kshitij	Kropp
Reinhard	Vitiello
Franziska	Pardalos

9. **Find the names of all employees who earn more than the average salary of all employees of their company.**

```
73      -- Question 9
74 •    SELECT DISTINCT employees.first_name , employees.last_name
75      FROM employees,salaries
76      WHERE salaries.to_date >=curdate() AND
77      employees.emp_no=salaries.emp_no AND
78      salaries.salary > (SELECT AVG(salaries.salary) FROM salaries);
```

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

first_name	last_name
Krassimir	Linares
Wonhee	Perl
Nidapan	Provine
Margareta	Petersohn
Urs	Krone
Franziska	Marrevee
Eishiro	Garigliano
Mary	Gente
Chinhyun	Hiyoshi
Shmuel	Sudkamp
Zengping	Poehlman
Toshiki	Szilard
Matt	Benner
Ortrun	Bolsens
Waiman	Genin

10. **Compute the difference between the average salary of a Senior Engineer and the average salary of all employees (including Senior Engineers).**

```
80      -- Question 10
81 •    SELECT ((SELECT AVG(salaries.salary) FROM salaries) -
82      (SELECT AVG(salaries.salary) FROM employees,salaries,titles
83      WHERE titles.title='Senior Engineer' AND
84      titles.emp_no=employees.emp_no AND employees.emp_no=salaries.emp_no)) AS diff;
```

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

diff
3297.7505

11. **Create a view current dept emp (emp no, fromdate, todate) to show only the current department for each employee. You may have to use two views for this.**

```
106      -- Question 11
107 •   CREATE VIEW current_dept_emp AS
108     SELECT e.emp_no , de.from_date , de.to_date
109     FROM employees e
110     INNER JOIN dept_emp de
111     ON e.emp_no = de.emp_no;
112
```

12. **Write a normal SQL query to do the above task in problem 11.**

```
109      -- Question 12
110 •   SELECT e.emp_no, de.from_date, de.to_date
111     FROM employees e
112     INNER JOIN dept_emp de
113     ON de.emp_no = e.emp_no;
```

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

emp_no	from_date	to_date
10017	1993-08-03	9999-01-01
10055	1992-04-27	1995-07-22
10058	1988-04-25	9999-01-01
10108	1999-12-06	2001-10-20
10140	1991-03-14	9999-01-01
10175	1988-09-24	1995-05-24
10208	1995-02-05	1999-05-15
10228	1993-01-28	9999-01-01
10239	1996-05-04	9999-01-01
10259	1987-07-25	1994-08-15
10340	1988-03-30	9999-01-01
10353	1989-08-24	9999-01-01

13. **Create a trigger to print salary changes of employees. For example, if you enter an SQL statement such as UPDATE salaries SET salary = salary + 1000 WHERE emp no = 1500, the trigger should fire once for each row that is updated and it should print the new and old salaries, and the difference.**

```
119      -- Question 13
120  •   create table emp_salary_change
121      (
122          old_salary int,
123          new_salary int,
124          difference int,
125          action VARCHAR(50) DEFAULT NULL
126      );
127
128      delimiter $
129  •   create trigger after_salaries_update
130      after update on salaries
131      for each row
132      begin
133      insert into emp_salary_change
134      SET action = 'update',
135      old_Salary = old.salary,
136      new_Salary = new.salary,
137      difference = new.salary-old.salary;
138      end $
139      delimiter ;
```

14. **Create a trigger that will cause an error when an update occurs that would result in a salary increase greater than 10% of the current salary**

```
141      -- Question 14
142      delimiter $
143
144  •   create trigger error_salary_update
145      before update on salaries
146      for each row
147      begin
148      declare msg varchar(50);
149      if(new.salary-old.salary)>(old.salary*0.1)then
150          set msg ="Error : Salary increment > 10%";
151          signal sqlstate '45000' set message_text = msg;
152      end if;
153      end $
154      delimiter ;
```