CO527: Advanced Database Systems
Lab 01 - Review on SQL

Ranage R.D.P.R.
E/19/310

1. Load data to each of the tables from the given .sql files.

```
MariaDB [Company]> select (
    -> select count(*) from employees) as employees,
    -> (select count(*) from dept_manager) as dept_manager,
    -> (select count(*) from dept_emo) as dept_emp,
    -> (select count(*) from titles) as titles,
    -> (select count(*) from salaries) as salaries,
    -> (select count(*) from departments) as departments;
ERROR 1146 (42S02): Table 'company.dept_emo' doesn't exist
MariaDB [Company]> select (
    -> select count(*) from employees) as employees,
    -> (select count(*) from dept_manager) as dept_manager,
    -> (select count(*) from dept_emp) as dept_emp,
    -> (select count(*) from titles) as titles,
    -> (select count(*) from salaries) as salaries,
    -> (select count(*) from departments) as departments;
+-----------+--------------+----------+--------+----------+-------------+
| employees | dept_manager | dept_emp | titles | salaries | departments |
+-----------+--------------+----------+--------+----------+-------------+
|    300024 |           24 |   331603 | 443306 |  1876717 |           9 |
+-----------+--------------+----------+--------+----------+-------------+
1 row in set (0.949 sec)

MariaDB [Company]>
```

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

```
MariaDB [Company]> select last_name, count(*) as count
    -> from employees
    -> group by last_name
    -> order by count DESC
    -> limit 10;
+-----------+-------+
| last_name | count |
+-----------+-------+
| Baba      |   226 |
| Gelosh    |   223 |
| Coorg     |   223 |
| Sudbeck   |   222 |
| Farris    |   222 |
| Adachi    |   221 |
| Osgood    |   220 |
| Masada    |   218 |
| Neiman    |   218 |
| Mandell   |   218 |
+-----------+-------+
10 rows in set (0.318 sec)

MariaDB [Company]>
```

3. List the number of Engineers each department has.

```
MariaDB [Company]> select de.dept_no, count(*) as count
    -> from dept_emp de
    -> join titles t ON de.emp_no = t.emp_no
    -> where t.title = 'Engineer'
    -> group by de.dept_no;
+---------+-------+
| dept_no | count |
+---------+-------+
| d004    | 49649 |
| d005    | 58135 |
| d006    | 13852 |
| d008    |  2986 |
| d009    |  2362 |
+---------+-------+
5 rows in set (2.423 sec)

MariaDB [Company]>
```

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

```
MariaDB [Company]> select e.emp_no
    -> from employees e
    -> join titles t ON e.emp_no = t.emp_no
    -> join dept_manager dm ON dm.emp_no = t.emp_no
    -> where t.title = 'Senior Engineer' AND sex = 'F';
+--------+
| emp_no |
+--------+
| 110344 |
| 110800 |
+--------+
2 rows in set (0.002 sec)
```

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.

```
MariaDB [Company]> select de.dept_no, t.title, count(*)
    -> from dept_emp de
    -> join titles t ON t.emp_no = de.emp_no
    -> join salaries s ON s.emp_no = t.emp_no
    -> where salary > 115000
    -> group by de.dept_no,t.title;
+---------+--------------------+----------+
| dept_no | title              | count(*) |
+---------+--------------------+----------+
| d001    | Senior Staff       |     1061 |
| d001    | Staff              |      855 |
| d002    | Senior Staff       |      771 |
| d002    | Staff              |      653 |
| d003    | Senior Staff       |       42 |
| d003    | Staff              |       27 |
| d004    | Assistant Engineer |       16 |
| d004    | Engineer           |      183 |
| d004    | Senior Engineer    |      207 |
| d004    | Technique Leader   |       32 |
| d005    | Assistant Engineer |       52 |
| d005    | Engineer           |      258 |
| d005    | Senior Engineer    |      277 |
| d005    | Senior Staff       |        2 |
| d005    | Technique Leader   |       21 |
| d006    | Assistant Engineer |        1 |
| d006    | Engineer           |       17 |
| d006    | Senior Engineer    |       17 |
| d006    | Technique Leader   |        6 |
| d007    | Senior Staff       |     8638 |
| d007    | Staff              |     7117 |
| d008    | Assistant Engineer |        4 |
| d008    | Engineer           |       12 |
| d008    | Senior Engineer    |       17 |
| d008    | Senior Staff       |       54 |
| d008    | Staff              |       51 |
| d008    | Technique Leader   |        7 |
| d009    | Engineer           |       11 |
| d009    | Senior Engineer    |       16 |
| d009    | Senior Staff       |      447 |
| d009    | Staff              |      356 |
+---------+--------------------+----------+
31 rows in set (5.440 sec)
```

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.

```
MariaDB [Company]> SELECT first_name, last_name,
    -> TIMESTAMPDIFF(YEAR, birth_date, CURDATE()) AS age,
    -> TIMESTAMPDIFF(YEAR, hire_date, CURDATE()) AS years_of_service,
    -> hire_date
    -> FROM employees
    -> WHERE TIMESTAMPDIFF(YEAR, birth_date, CURDATE()) > 50
    -> AND TIMESTAMPDIFF(YEAR, hire_date, CURDATE()) > 10
    -> LIMIT 20;
+------------+-------------+------+------------------+------------+
| first_name | last_name   | age  | years_of_service | hire_date  |
+------------+-------------+------+------------------+------------+
| Georgi     | Facello     |   70 |               37 | 1986-06-26 |
| Bezalel    | Simmel      |   59 |               38 | 1985-11-21 |
| Parto      | Bamford     |   64 |               37 | 1986-08-28 |
| Chirstian  | Koblick     |   69 |               37 | 1986-12-01 |
| Kyoichi    | Maliniak    |   69 |               34 | 1989-09-12 |
| Anneke     | Preusig     |   70 |               34 | 1989-06-02 |
| Tzvetan    | Zielinski   |   66 |               35 | 1989-02-10 |
| Saniya     | Kalloufi    |   66 |               29 | 1994-09-15 |
| Sumant     | Peac        |   71 |               39 | 1985-02-18 |
| Duangkaew  | Piveteau    |   60 |               34 | 1989-08-24 |
| Mary       | Sluis       |   70 |               34 | 1990-01-22 |
| Patricio   | Bridgland   |   63 |               31 | 1992-12-18 |
| Eberhardt  | Terkki      |   60 |               38 | 1985-10-20 |
| Berni      | Genin       |   68 |               37 | 1987-03-11 |
| Guoxiang   | Nooteboom   |   64 |               36 | 1987-07-02 |
| Kazuhito   | Cappelletti |   62 |               29 | 1995-01-27 |
| Cristinel  | Bouloucos   |   65 |               30 | 1993-08-03 |
| Kazuhide   | Peha        |   69 |               36 | 1987-04-03 |
| Lillian    | Haddadi     |   71 |               24 | 1999-04-30 |
| Mayuko     | Warwick     |   71 |               33 | 1991-01-26 |
+------------+-------------+------+------------------+------------+
20 rows in set (0.002 sec)
```

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.

```
MariaDB [Company]> select first_name, last_name
    -> from employees e
    -> join dept_emp de ON de.emp_no = e.emp_no
    -> where dept_no != 'd003'
    -> LIMIT 20;
+------------+---------------+
| first_name | last_name     |
+------------+---------------+
| Cristinel  | Bouloucos     |
| Georgy     | Dredge        |
| Berhard    | McFarlin      |
| Lunjin     | Giveon        |
| Yucel      | Auria         |
| Aleksandar | Ananiadou     |
| Xiping     | Klerer        |
| Karoline   | Cesareni      |
| Nikolaos   | Llado         |
| Susanna    | Vesel         |
| Djelloul   | Laventhal     |
| Phule      | Hammerschmidt |
| Hyuckchul  | Gini          |
| Feiyu      | Luft          |
| Candida    | Porotnikoff   |
| Aleksandar | Sudkamp       |
| Garnik     | Narahari      |
| Maik       | Luft          |
| Ramalingam | Gunderson     |
| Dietrich   | Journel       |
+------------+---------------+
20 rows in set (0.014 sec)
```

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.

```
MariaDB [Company]> select distinct first_name, last_name
    -> from employees e
    -> join salaries s ON s.emp_no = e.emp_no
    -> WHERE s.salary > (select max(salary)
    -> from salaries s
    -> join dept_emp de ON de.emp_no = s.emp_no
    -> where de.dept_no = 'd002');
+------------+-------------+
| first_name | last_name   |
+------------+-------------+
| Charmane   | Griswold    |
| Boalin     | Rosen       |
| Nikolaus   | Businaro    |
| JoAnne     | Matheson    |
| Wonhee     | Pagter      |
| Tadanori   | Sudbeck     |
| Weicheng   | Hatcliff    |
| Chaitali   | Baik        |
| Mitsuyuki  | Stanfel     |
| Dines      | Giaccio     |
| Arnd       | Junot       |
| Heping     | Brender     |
| Sanjai     | Luders      |
| Honesty    | Mukaidono   |
| Honglan    | Otillio     |
| Satoru     | Gruenwald   |
| Rance      | Chinin      |
| Eberhardt  | Gubsky      |
| Weijing    | Chenoweth   |
| Florina    | Tchuente    |
| Jungsoon   | Brendel     |
| Shin       | Birdsall    |
| Mohammed   | Moehrke     |
| Eldridge   | Heiserman   |
| Huiqun     | Birke       |
| Leaf       | Menyhert    |
| Zhanqiu    | Muntz       |
| Guenter    | Ranai       |
| Seongbin   | Mitsuhashi  |
| Qingxiang  | Piancastelli |
| Izaskun    | Wissmann    |
| Akemi      | Warwick     |
| Masato     | Heering     |
| Ghassan    | Birta       |
| Taiji      | Malinowski  |
| Lidong     | Meriste     |
+------------+-------------+
36 rows in set (1.620 sec)
```

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

```
MariaDB [company]> select distinct first_name,last_name
    -> from employees e
    -> join salaries s ON s.emp_no = e.emp_no
    -> where s.salary > (
    -> select avg(salary)
    -> from salaries)
    -> LIMIT 20;
+------------+------------+
| first_name | last_name  |
+------------+------------+
| Krassimir  | Linares    |
| Wonhee     | Perl       |
| Nidapan    | Provine    |
| Margareta  | Petersohn  |
| Urs        | Krone      |
| Franziska  | Marrevee   |
| Eishiro    | Garigliano |
| Mary       | Gente      |
| Chinhyun   | Hiyoshi    |
| Shmuel     | Sudkamp    |
| Vivian     | Chachaty   |
| Zengping   | Poehlman   |
| Toshiki    | Szilard    |
| Matt       | Benner     |
| Ortrun     | Bolsens    |
| Waiman     | Genin      |
| Jaana      | Besselaar  |
| Alagu      | Kabayashi  |
| Fay        | Genin      |
| Elvia      | Jenevein   |
```

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

```
MariaDB [company]> select (select avg(salary)
    -> from salaries) -
    -> (select avg(salary)
    -> from salaries s
    -> join titles t ON t.emp_no = s.emp_no
    -> where title = 'Senior Engineer') as Difference;
+-------------+
| Difference |
+-------------+
|  3297.7505 |
+-------------+
1 row in set (1.616 sec)
```

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.

```
XAMPP for Windows - mysql
+--------+---------+------------+------------+
20 rows in set (0.654 sec)

MariaDB [company]> drop view current_dept_emp
    -> ;
Query OK, 0 rows affected (0.002 sec)

MariaDB [company]> CREATE VIEW current_dept_emp AS
    -> SELECT de.emp_no, de.dept_no
    -> FROM dept_emp de
    -> JOIN current_department cd ON de.emp_no = cd.emp_no AND de.to_date = cd.current_to_date;
Query OK, 0 rows affected (0.004 sec)

MariaDB [company]> select * from current_dept_emp
    -> limit 20;
+--------+---------+
| emp_no | dept_no |
+--------+---------+
|  10001 | d005    |
|  10002 | d007    |
|  10003 | d004    |
|  10004 | d004    |
|  10005 | d003    |
|  10006 | d005    |
|  10007 | d008    |
|  10008 | d005    |
|  10009 | d006    |
|  10010 | d006    |
|  10011 | d009    |
|  10012 | d005    |
|  10013 | d003    |
|  10014 | d005    |
|  10015 | d008    |
|  10016 | d007    |
|  10017 | d001    |
|  10018 | d004    |
|  10019 | d008    |
|  10020 | d004    |
+--------+---------+
20 rows in set (0.330 sec)

MariaDB [company]>
```

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

```
MariaDB [company]> select de.emp_no, de.dept_no
    -> from dept_emp de
    -> where de.to_date IN (
    -> select max(to_date)
    -> from dept_emp
    -> group by emp_no)
    -> group by de.emp_no
    -> limit 20;
+--------+---------+
| emp_no | dept_no |
+--------+---------+
|  10001 | d005    |
|  10002 | d007    |
|  10003 | d004    |
|  10004 | d004    |
|  10005 | d003    |
|  10006 | d005    |
|  10007 | d008    |
|  10008 | d005    |
|  10009 | d006    |
|  10010 | d004    |
|  10011 | d009    |
|  10012 | d005    |
|  10013 | d003    |
|  10014 | d005    |
|  10015 | d008    |
|  10016 | d007    |
|  10017 | d001    |
|  10018 | d004    |
|  10019 | d008    |
|  10020 | d004    |
+--------+---------+
20 rows in set (0.217 sec)
```

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.

```
MariaDB [company]>
MariaDB [company]> DELIMITER ;
MariaDB [company]> DELIMITER //
MariaDB [company]>
MariaDB [company]> CREATE TRIGGER print_salary_changes
    -> AFTER UPDATE ON salaries
    -> FOR EACH ROW
    -> BEGIN
    ->     DECLARE message VARCHAR(255);
    ->
    ->     -- Concatenate the message with old and new salaries, and the difference
    ->     SET message = CONCAT_WS(' ',
    ->         'Employee', NEW.emp_no, ':',
    ->         'Old Salary = $', OLD.salary,
    ->         ', New Salary = $', NEW.salary,
    ->         ', Salary Difference = $', NEW.salary - OLD.salary
    ->     );
    ->
    ->     -- Raise an error with the concatenated message
    ->     SIGNAL SQLSTATE '45000'
    ->     SET MESSAGE_TEXT = message;
    -> END;
    -> //
Query OK, 0 rows affected (0.011 sec)

MariaDB [company]>
MariaDB [company]> DELIMITER ;
MariaDB [company]> Update salaries
    -> set salary = salary + 1000
    -> where emp_no = 201774;
ERROR 1644 (45000): Employee 201774 : Old Salary = $ 40000 , New Salary = $ 41000 , Salary Difference = $ 1000
```

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.

```
MariaDB [company]> DELIMITER //
MariaDB [company]> CREATE TRIGGER prevent_salary_increase
    -> BEFORE UPDATE ON salaries
    -> FOR EACH ROW
    -> BEGIN
    ->     DECLARE max_increase DECIMAL(10, 2);
    ->     DECLARE current_salary DECIMAL(10, 2);
    ->     SET max_increase = OLD.salary * 0.10;
    ->     SET current_salary = OLD.salary;
    ->     IF (NEW.salary - current_salary) > max_increase THEN
    ->         SIGNAL SQLSTATE '45000'
    ->         SET MESSAGE_TEXT = 'Salary increase cannot exceed 10% of the current salary.';
    ->     END IF;
    -> END;
    -> //
Query OK, 0 rows affected (0.006 sec)

MariaDB [company]> DELIMITER ;
MariaDB [company]> update salaries
    -> set salary = salary + 10000
    -> where emp_no = 201774;
ERROR 1644 (45000): Salary increase cannot exceed 10% of the current salary.
MariaDB [company]>
```