Data Centric RAD

Lab 4 MySQL Review II

Part 1

- Get `employee_kin.sql` from Moodle.

- Import it into MySQL as described in Lab 1 Exercises.

- `use employee_kin;`

- Display the Employee Name and Next of Kin name of ALL employees.

  - SELECT et.ename, nk.NOK_Name FROM employee_table et
    LEFT JOIN next_of_kin_table nk on (et.NextOfKin = nk.NOK_ID);

- Display the Employee Name and Next of Kin name only of employees who have a Next of Kin.
  - SELECT et.ename, nk.NOK_Name FROM employee_table et
    INNER JOIN next_of_kin_table nk on (et.NextOfKin = nk.NOK_ID)
    WHERE et.NextOfKin IS NOT NULL

- Display the Employee ID as ‘Employee ID’, the Employee Name as ‘Employee Name’ and the Employee Salary as ‘Employee Salary’ for all employees.
  - SELECT et.eid as \`Employee ID\`, et.ename \`Employee Name\`, s.salary \`Employee Salary\` FROM employee_table et INNER JOIN salary s on (et.eid = s.eid)

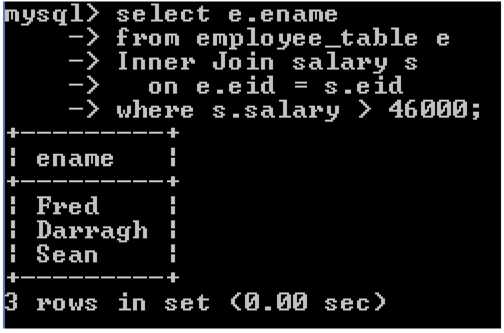
- Display the Employee Name as ‘Employee Name’ and the Next of Kin’s phone number as ‘Emergency Contact’ only for employees with a Next of Kin.
  - SELECT et.ename \`Employee Name\`, nk.phone \`Emergency phone\` FROM employee_table et INNER JOIN next_of_kin_table nk on (et.NextOfKin = nk.NOK_ID)

- Display the Next of Kin’s name as ‘NOK Name’ and the salary of the associated employee as ‘Associated Salary’ for next of kins.

- SELECT nk.NOK_Name `Employee Name`, s.salary `Assosiated Salary` FROM next_of_kin_table nk INNER JOIN employee_table et on (et.NextOfKin = nk.NOK_ID) join salary s on (s.eid = et.eid)

- Display the Employee Name as ‘Employee’, his salary as ‘Salary’, and his next of Kin’s phone number as ‘Emergency Contact’ for ALL employees.
    - SELECT et.ename `Empoloyee`, s.salary `Salary`, nk.phone `Emergency Contact` FROM employee_table et JOIN salary s on (s.eid = et.eid)
      LEFT JOIN next_of_kin_table nk on (nk.NOK_ID = et.NextOfKin)

- Rewrite the following query that uses Inner Joins as a Subquery.



```
mysql> select e.ename
    -> from employee_table e
    -> Inner Join salary s
    ->   on e.eid = s.eid
    -> where s.salary > 46000;
+----------+
| ename    |
+----------+
| Fred     |
| Darragh  |
| Sean     |
+----------+
3 rows in set (0.00 sec)
```

    - SELECT et.ename FROM employee_table et
      WHERE et.eid IN(SELECT s.eid from salary s where s.salary > 46000)

Part 2

- Get `employeesDB14.sql` from Moodle.

- Import it into MySQL as described in Lab 1 Exercises.

- `use employeesDB14;`

- Display the employee name as 'Name' and department location as 'Location' of the employee 7566.

  - SELECT e.ENAME `Name`, d.LOC `Location` from emp e join dept d on (e.DEPTNO = d.DEPTNO) where e.EMPNO = 7566;

- Display the name, job and hiredate of all employees in department 20.
  - SELECT e.ENAME `Name`, e.JOB, e.HIREDATE from emp e
    where e.DEPTNO = 20;

- Display the employee number, employee name, job, department number and department location of all employees.
  - SELECT e.EMPNO, e.ENAME, e.JOB, e.DEPTNO, d.LOC from emp e JOIN dept d on d.DEPTNO = e.DEPTNO;

- Display the Customer ID, Name, Address, City and State of all customers who bought the *RH: "GUIDE TO TENNIS"* product.
  - SELECT c.CUSTID, c.NAME, c.ADDRESS, c.CITY,c.STATE from customer c join ord o on (o.CUSTID = c.CUSTID)
    WHERE o.ORDID IN(SELECT i.ORDID from item i JOIN product p on (i.PRODID = p.PRODID)
    WHERE p.DESCRIP = 'RH: "GUIDE TO TENNIS"')
    group by c.CUSTID

Part 3

- Get `studentDB3.sql` from Moodle.

- Import it into MySQL as described in Lab 1 Exercises.

- Show the Student Name, and whether or not he/she attends an NUI university.

  -
    ```
    SELECT st.student_name, co.college_name from
    student_table st LEFT JOIN course_table cot on
    (cot.course_ID = st.course_id)
    join college_table co on (co.college_id =
    cot.college_id)
    ```

- Show college name and the number of students attending each college as 'Attending Students'.

  -
    ```
    select cot.college_name, count(st.student_id) from
    college_table cot
    JOIN course_table co on (cot.college_id =
    co.college_id)
    JOIN student_table st on (st.course_id =
    co.course_ID)
    Group by cot.college_id;
    ```

- Show the college name and the population of the county where the college is.

  -
    ```
    SELECT  co.college_name, cot.population `County
    Population` from college_table co
    join county_table cot on cot.county_name = co.county
    ```

- Show the Student name, the course he/she is doing, the name of the college they are attending, and the main town and population of the county in which the college is.

  -
    ```
    SELECT s.student_name, co.course_name,
    cot.college_name, cty.main_town, cty.population from
    student_table s
    join course_table co on (co.course_ID = s.course_id)
    JOIN college_table cot on(co.college_id =
    cot.college_id)
    JOIN county_table cty on (cty.county_name =
    cot.county);
    ```

- Show the Names of the students doing the longest course:
    - SELECT s.student_name from student_table s
      join course_table co on (co.course_ID = s.course_id)
      WHERE co.course_ID IN(SELECT co.course_ID from course_table co having MAX(co.duration))