# 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)

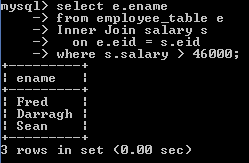
WHERE et.NextOfKin IS NOT NULL

* 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.



* + 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))