**COEN 178 Intro to Database Systems Winter 2018**

**Lab 3 (100 pts)**

**Objectives: Learn**

* SQL queries with subqueries.
* Aggregate functions

**------------------------------------------------------------------------------------------------**

**Part 1**

In this part, you will use **staff\_2010** table with the data you have loaded in Lab1.

**Step 1**: Create a folder structure called COEN178\labs\lab3.

**Step 2**: Create a text file called **part1.sql.** This file will contain the SQL statements that you want to execute.

**Exercise 1 (10 pts)**

**Write a query to show the full names of employees with maximum salary.**

**Approach 1**: Complete the subquery below and run it.

**Select \_\_\_\_\_\_\_\_\_, salary**

**from Staff\_2010**

**where salary \_\_\_\_\_\_\_\_\_ (Select salary from Staff\_2010);**

**Approach 2**: Complete the subquery below and run it.

**Select \_\_\_\_\_\_\_\_\_\_\_\_, salary**

**from Staff\_2010**

**where salary = (Select \_\_\_\_\_\_\_salary from Staff\_2010);**

**Q: What is the output?**

**A: FULLNAME SALARY**

**---------------------------------------- ----------**

**Michael M.Hash 179700**

**Timothy P.Love 179700**

**FULLNAME SALARY**

**---------------------------------------- ----------**

**Timothy P.Love 179700**

**Michael M.Hash 179700**

**Exercise 2 (10 pts)**

**Using the query below, find the last names of people with the same salary as “Zichal”.**

**Select last, salary**

**from Staff\_2010**

**where salary = (select salary from Staff\_2010 where last = 'Zichal');**

1. Rewrite and run the query so that the last name comparison will work, whether it is stored in uppercase, lowercase or mixed case.
2. Substitute the last name “Young” for “Zichal” and run the query again.

**Q: Did it work? If it did not work, why?**

**A: 3 people named Young so it returns more than one result and cannot use = operator for multi-line results**

Fix the query to match any “Young” salary and re-run the query.

**Q: What is the output?**

**A: Changed '=' to '=ANY' that then changes the query to find anyone who has a salary the same as any of the 3 youngs**

**Exercise 3 (5 pts)**

**Write and run a query to find the number of people with salaries greater than 100,000.**

**Note:** the output should be like something given below (the count may vary for your table)

SALARIES\_100K\_ABOVE

-------------------

140

**Exercise 4 (10 pts)**

**Write and run a query to find the number of people with salaries greater than 100,000 and grouped by a salary number. See the example output below:**

SALARY SALARIES\_100K\_ABOVE

-------- -------------------

140000 2

120000 8

105211 1

179700 2

150000 3

110000 2

102829 1

144868 1

107770 1

**Exercise 5 (15 pts)**

**Write and run a query to find the number of people with salaries greater than 100,000, grouped by a salary number, where the no. of people in the group is >= 10. See the example output below:**

SALARY SALARIES\_100K\_ABOVE

---------- -------------------

130500 27

172200 23

**Exercise 6 (5 pts)**

**Examine the query below. It uses regular expressions (regex) to show the last names where the same vowel repeats itself. Link:** [Regex – A reference](https://www.regular-expressions.info/)

**SELECT last**

**FROM Staff\_2010**

**WHERE REGEXP\_LIKE (last, '([aeiou])\1', 'i');**

**Q: Examine the output. What is the option “i” for?**

**A: Means that the lettering is non-case sensitive**

**Part 2**

**In this part, you will create two tables and load them with values given.**

**Please note, these tables are from Lab 2. Run init.sql to create and populate new instances of these tables.**

**Creating Tables**

Create the tables, L\_EMP and L\_DEPT using the DDL statements below:

Create table L\_EMP (empNo Integer Primary Key, empname VARCHAR(10),deptId VARCHAR(5));

Create table L\_DEPT (deptId VARCHAR(5) Primary Key, deptname VARCHAR(10));

**Note**: We have not defined any foreign key constraint in these tables.

**Inserting Tuples**

Add the following tuples into the tables (use a script file to add the data).

insert into L\_EMP values(1,'smith','d1');

insert into L\_EMP values(2,'jones','d2');

insert into L\_EMP values(3,'wayne','d1');

insert into L\_EMP values(4,'moor','d3');

insert into L\_EMP values(5,'king','d1');

insert into L\_EMP values(6,'chen','d1');

insert into L\_EMP values(7,'winger','d3');

insert into L\_DEPT values('d1','Research');

insert into L\_DEPT values('d2','Devt');

insert into L\_DEPT values('d3','Testing');

insert into L\_DEPT values('d4','Advert');

Create a text file called **part2.sql.** This file will contain the SQL statements to execute for Exercises 7-10.

**Exercise 7 (15 pts)**

In this query, we want to **show the deptid and the number of employees in each dept**. This information comes from L\_EMP table. Write the Select query to show deptid and count(\*) from L\_EMP. Make sure that you group by deptid. **Name deptid column as deptno and the count(\*) column as empcount .** Show the results of query in your spool file.

**Exercise 8 (10 pts)**

In this query, we want to **show the deptname (note the change from the previous exercise) and the number of employees in each dept**. This information comes from both L\_EMP and L\_DEPT tables.

To write this query, we will use the fact that a subquery can be given in the FROM clause.

1. Use the query in exercise 7, as the **subquery below**. This will go in to the from clause of the query below:

**Select deptno,deptname,empcount**

**from (include your subquery here),L\_DEPT**

**where deptno = L\_DEPT.deptid**

**Q: Execute the query. Does it give you the correct results?**

**A: YES it does**

1. Add the statement to show the rows displayed in ascending order, sorted by empcount (think of **order by** in the outer clause). Execute the statement.

**Exercise 9 (10 pts)**

In this exercise, we will **find the deptid of the department with maximum number of employees.**

**Attempt 1: Try the query below. Will it work?**

Select deptid, max(count(\*)) from L\_EMP

Group by deptid;

**Q: What is wrong with the above query?**

**A: The max function wants to return a single value and there maybe multiple deptid's that have the same count**

**Attempt 2: Try the query below. Will it work?**

Select deptid from L\_EMP

Group by deptid

Having count(\*) = (Select count(\*) from L\_EMP

Group by deptid);

**Q: What is wrong with the above query?**

**A: Trying to compare a single variable to a bunch of variables at 'count(\*) = (Select count(\*)...'**

1. **Fix the query in approach 2 and run it.**
2. **Find the deptname of the department with maximum number of employees.**

**Exercise 10 (10 pts)**

Write a query, to show the employee and dept. information only where there are employees working in a dept. Include only those tuples that have a common **deptid** in both relations

1. Run the query ( using natural join) below.

**Select \* from L\_EMP NATURAL JOIN L\_DEPT;**

Show the output in your spool.

1. The query (incomplete) query below accomplishes the same thing using cartesian product. Complete it and run to display the same output as shown by the query in a).

**Select \* from L\_EMP, L\_DEPT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;**

Run the queries and capture the results in **lab3\_output.lst**, using *spool*.