COEN 178   Intro to Database Systems   Spring 2018

Lab 3 (100 pts)

Objectives: Learn

- SQL queries with subqueries.
- Aggregate functions

--

Run the queries and capture the results in lab3_output.lst, using *spool*.

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 **queries_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);
```

What is the output?

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');
```

a) Rewrite and run the query so that the last name comparison will work, whether it is stored in uppercase, lowercase or mixed case.

b) Substitute the last name “Young” for “Zichal” and run the query again. Did it work? If it did not work, why?

Fix the query and re run the query.

What is the output?

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.

```
SELECT last

FROM Staff_2010

WHERE REGEXP_LIKE (last, '([aeiou])\1', 'i');
```

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

Regex – A reference

Part 2

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

Please note that you may have created these tables in your Lab2.

Creating Tables

Create the tables, L_EMP and L_DEPT using the DDL statements below:

```
Create table L_EMP (empNo Integer Primary Key, empname CHAR(10),deptId CHAR(5));
```

```
Create table L_DEPT (deptId CHAR(5) Primary Key, deptname CHAR(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 **queries_part2.sql.** This file will contain the SQL statements that you want to execute.

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.

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.

a) 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 query here),L_DEPT
where deptno = L_DEPT.deptid
```

Execute the query. Does it give you the correct results?

b) 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;
```

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);
```

a) What is the problem with the above query? Fix the query in approach 2 and run it.

b) Find the dept.name 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

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

```
      Select * from L_EMP NATURAL JOIN L_DEPT;
```

Show the output.

b) 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;
```