COEN 178 Intro to Database Systems Spring 2020

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 **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 first||last as FullName, salary
from Staff_2010
where salary >= ALL (Select salary from Staff_2010);
```

Approach 2: Complete the subquery below and run it.

```
Select first||last as FullName, salary
from Staff_2010
where salary = (Select MAX(salary) from Staff_2010);
```

What is the output?

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');
```

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

```
SELECT last, salary
FROM Staff_2010
WHERE salary = (SELECT salary FROM Staff_2010 WHERE UPPER(last) =
'ZICHAL');
```

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

It doesn't work because there are 2 people with the last name "Young".

Fix the query and re run the query.

```
SELECT last, salary
FROM Staff_2010
WHERE salary = ANY(SELECT salary FROM Staff_2010 WHERE UPPER(last) =
'YOUNG');
```

What is the output?

LAST	SALARY
Metcalf	42000

Mrazek	42000
Northern	42000
Ocampo	42000
Oxtoby	42000
Page	42000
Pope	42000
Portilla	42000
Posey	42000
Pulliam	42000
Robertson	42000

LAST	SALARY
Schaub	42000
Schmuck	42000
Sinha	42000
Sirbu	42000
Smith	42000
Smith	42000
Stickel	42000
Swanson	42000
Tennison	42000
Thomas	42000
Tranchin	42000

LAST	SALARY
Alley	42000
Asen	42000
Attili	58511
Ayling	42000
Baggetto	42000
Bates	42000
Bisi	42000
Block	42000
Campbell	42000
Campos	42000
Canery	58511

LAST	SALARY
Carden	42000
Castillo	42000
Chhabra	42000
Claude	58511
Cobbina	42000

Conrad	42000
Cuba	42000
Cunningham	42000
Donovan	42000
Dorsett	42000
Faulman	42000

LAST	SALARY
Fenn	42000
Fergenson	42000
Ford	42000
Frank	42000
Godfrey	42000
Gottlieb	42000
Grant	42000
Hanson	42000
Harris	42000
Hegde	42000
Hernandez	42000

LAST	SALARY
Hiatt	42000
Hughes	42000
Jackson	42000
Jones	42000
Kennedy	42000
Ledbetter	42000
Lee	42000
Lewin	42000
Lewis	42000
Limon	42000
Young	42000

LAST	SALARY
Young	58511

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
```

```
SELECT COUNT(salary) as SALARIES_100K_ABOVE
FROM Staff_2010
WHERE salary > 100000;
```

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
```

```
SELECT salary, COUNT(salary) as SALARIES_100K_ABOVE
FROM Staff_2010
WHERE salary > 100000
GROUP BY salary;
```

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
```

```
SELECT salary, COUNT(salary) as SALARIES_100K_ABOVE
FROM Staff_2010
WHERE salary > 100000
GROUP BY salary
HAVING COUNT(salary) >= 10;
```

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?

The option "i" is for case-insensitive matching, which doesn't distinguish between upper and lowercase letters.

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.

```
SELECT deptid AS deptno, COUNT(*) AS empcount
FROM L_EMP
GROUP BY deptid;
```

```
DEPTN    EMPCOUNT
----- ----------
d1          4
d2          1
d3          2
```

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 (SELECT deptid AS deptno, COUNT(*) AS empcount
```

```
            FROM L_EMP
            GROUP BY deptid), L_DEPT
      where deptno = L_DEPT.deptid
```

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

The query does give us the correct results.

```
DEPTN DEPTNAME      EMPCOUNT
----- ---------- ----------
d1    Research          4
d2    Devt              1
d3    Testing           2
```

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.

```
SELECT deptNo, deptName, empcount
FROM (SELECT deptid as deptNo, COUNT(*) AS empcount FROM L_EMP GROUP BY
deptid), L_DEPT
WHERE deptNO = L_DEPT.deptId
ORDER BY empcount;
```

```
DEPTN DEPTNAME      EMPCOUNT
----- ---------- ----------
d2    Devt              1
d3    Testing           2
d1    Research          4
```

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;
```

The query does not work because of the "not a single-group function". This is because MAX() returns a single value and can't be a column attribute in the SELECT statement.

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);
```

The query does not work because of the "single-row subquery returns more than one row". This is because SELECT COUNT(*) will return multiple rows.

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

```
SELECT deptid, COUNT(*) as NumberOfEmployees
FROM L_EMP
GROUP BY deptId
HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM L_EMP GROUP BY deptid);
```

b) **Find the dept.name of the department with maximum number of employees.**

```
SELECT deptname FROM L_DEPT
WHERE deptId = (SELECT deptid FROM L_EMP
                GROUP BY deptId
                HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM L_EMP
                GROUP BY deptid));
```

DEPTNAME

```
----------
Research
```

The department name with the maximum number of employees is "Research".

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.

```
DEPTI      EMPNO EMPNAME    DEPTNAME
----- ---------- ---------- ----------
d1          1 smith      Research
d2          2 jones      Devt
d1          3 wayne      Research
d3          4 moor       Testing
d1          5 king       Research
d1          6 chen       Research
d3          7 winger      Testing
```

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 L_EMP.deptid, empNo, empName, deptName FROM L_EMP, L_DEPT
WHERE L_EMP.deptID = L_DEPT.deptId;
```

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