CSE 4308 - DBMS LAB

DBMS LAB 06 TASKS AND SOLUTIONS

Prepared by:
Mohammad Anas Jawad
Lecturer, IUT CSE

Department of Computer Science and Engineering
Islamic University of Technology
July 10, 2021

Note: Write down your commands and errors encountered in a notepad file to be evaluated.

1. What is the maximum salary?

```
SELECT MAX(SALARY) FROM CITIZEN;
```

2. Which male person has the maximum income? [Hint: nested query]

```
SELECT C_NAME, SALARY FROM CITIZEN WHERE GENDER IN 'Male' AND SALARY =
    (SELECT MAX(SALARY) FROM CITIZEN WHERE GENDER IN 'Male');
```

Note: Notice that we have used (`WHERE GENDER IN 'Male'`) condition twice in our implementation. Whenever nested query is involved, the innermost query is executed first. In this case, when the innermost query is executed, suppose the result returned is 100,000. Now, if there isn't any gender condition on the outer query, the outer query essentially becomes

`SELECT C_NAME, SALARY FROM CITIZEN WHERE SALARY = 100000`

Since there is no condition for gender here, it will return the names of both male and female citizens whose salary is equal to 100000. Hence we need to use the condition for both the inner and outer queries.

The desired answer can also be obtained without using aggregate function in this case:

`SELECT C_NAME, SALARY FROM (SELECT * FROM CITIZEN WHERE GENDER IN 'Male'ORDER BY SALARY DESC)WHERE ROWNUM=1;`

3. Who is the oldest male person? [Hint: nested query]

```
SELECT C_NAME, AGE FROM CITIZEN WHERE GENDER IN 'Male' AND AGE =
    (SELECT MAX(AGE) FROM CITIZEN WHERE GENDER IN 'Male');
```

4. Which female person has the least income? [Hint: Nested Query]

```
SELECT C_NAME, SALARY FROM CITIZEN WHERE GENDER IN 'Female' AND SALARY
    = (SELECT MIN(SALARY) FROM CITIZEN WHERE GENDER='Female');
```

5. Which female person is the youngest? [Hint: nested query]

```
SELECT C_NAME, AGE FROM CITIZEN WHERE GENDER IN 'Female' AND AGE =
    (SELECT MIN(AGE) FROM CITIZEN WHERE GENDER IN 'Female');
```

6. Which teacher has the most income?

```
SELECT C_NAME, SALARY, OCCUPATION FROM CITIZEN WHERE OCCUPATION IN
    'Teacher' AND SALARY = (SELECT MAX(SALARY) FROM CITIZEN WHERE
    OCCUPATION IN 'Teacher');
```

7. Who is the most earning doctor?

```
SELECT C_NAME, SALARY, OCCUPATION FROM CITIZEN WHERE OCCUPATION IN
    'Doctor' AND SALARY = (SELECT MAX(SALARY) FROM CITIZEN WHERE
    OCCUPATION IN 'Doctor');
```

8. Which citizens have 'r' in their name?

```
SELECT C_NAME FROM CITIZEN WHERE C_NAME LIKE '%r%';
```

9. Which citizens have 'a' in their name?

```
SELECT C_NAME FROM CITIZEN WHERE C_NAME LIKE '%a%';
```

10. Which citizens have 'y' as the third character in their name?

```
SELECT C_NAME FROM CITIZEN WHERE C_NAME LIKE '__y%';
```

11. What is the average salary of this table?

```
SELECT AVG(SALARY) AS AVG_SAL FROM CITIZEN;
```

12. What is the average age of this table?

```
SELECT AVG(AGE) AS AVG_AGE FROM CITIZEN;
```

13. What is the total income of all the teachers of the table?

```
SELECT SUM(SALARY) FROM CITIZEN WHERE OCCUPATION IN 'Teacher';
```

14. What is the total income of all the doctors of this table?

```
SELECT SUM(SALARY) FROM CITIZEN WHERE OCCUPATION IN 'Doctor';
```

15. What is the average income of the students?

```
SELECT AVG(SALARY) FROM CITIZEN WHERE OCCUPATION IN 'Student';
```

16. Show the name of the least earning student.

```
SELECT C_NAME, SALARY, OCCUPATION FROM CITIZEN WHERE OCCUPATION IN
    'Student' AND SALARY = (SELECT MIN(SALARY) FROM CITIZEN WHERE
    OCCUPATION IN 'Student');
```

17. Show the name of the most earning female student.

```
SELECT C_NAME,GENDER, OCCUPATION, SALARY FROM CITIZEN WHERE GENDER IN
    'Female' AND OCCUPATION IN 'Student' AND SALARY = (SELECT
    MAX(SALARY) FROM CITIZEN WHERE GENDER IN 'Female' AND OCCUPATION
    IN 'Student');
```

18. What is the number of cities with 'aka' in their name?

```
SELECT COUNT(DISTINCT C_HOME) FROM CITIZEN WHERE C_HOME LIKE '%aka%';
--Or:
SELECT COUNT(CITY) FROM (SELECT DISTINCT C_HOME AS CITY FROM CITIZEN
    WHERE C_HOME LIKE '%aka%');
```

19. What is the number of cities with 'tg' in their name?

```
SELECT COUNT(DISTINCT C_HOME) FROM CITIZEN WHERE C_HOME LIKE '%tg%';
```

20. What is the maximum salary of citizens whose name starts with 'A'?

```
SELECT MAX(SALARY) FROM CITIZEN WHERE C_NAME LIKE 'A%';
```

21. Show the description of the maximum earning teacher.

```
SELECT * FROM CITIZEN WHERE OCCUPATION IN 'Teacher' AND SALARY =
    (SELECT MAX(SALARY) FROM CITIZEN WHERE OCCUPATION IN 'Teacher');
```

22. Show the description of the minimum earning student.

```
SELECT * FROM CITIZEN WHERE OCCUPATION IN 'Student' AND SALARY =
    (SELECT MIN(SALARY) FROM CITIZEN WHERE OCCUPATION IN 'Student');
```

23. What is the average age of the students?

```
SELECT AVG(AGE) FROM CITIZEN WHERE OCCUPATION IN 'Student';
```

24. Make a list which will show the number of citizens belonging to each occupation.

```
SELECT OCCUPATION, COUNT(*) FROM CITIZEN GROUP BY OCCUPATION;
```

25. Make a list which will show the number of citizens belonging to each city.

```
SELECT C_HOME, COUNT(*) FROM CITIZEN GROUP BY C_HOME;
```

26. Find the maximum salary of each occupation.

```
SELECT OCCUPATION, MAX(SALARY) FROM CITIZEN GROUP BY OCCUPATION;
```

27. Find the minimum salary of each city.

```sql
SELECT C_HOME, MIN(SALARY) AS MIN_SALARY FROM CITIZEN GROUP BY C_HOME;
```

28. Categorize the average salary of male citizens based on occupation.

```sql
SELECT OCCUPATION, AVG(SALARY) FROM CITIZEN WHERE GENDER IN 'Male'
    GROUP BY OCCUPATION;
```

29. Categorize the average salary of the citizens of Dhaka based on occupation.

```sql
SELECT OCCUPATION, AVG(SALARY) AS AVG_SAL_DHAKA FROM CITIZEN WHERE
    C_HOME IN 'Dhaka' GROUP BY OCCUPATION;
```

30. Categorize the average salary of male citizens based on occupation where the average salary is at least 10000.

```sql
SELECT OCCUPATION, AVG(SALARY) FROM CITIZEN WHERE GENDER IN 'Male'
    GROUP BY OCCUPATION HAVING AVG(SALARY)>=10000;
```

31. Categorize the average salary of female citizens based on occupation where the maximum salary is at least 10000.

```sql
SELECT OCCUPATION, AVG(SALARY) FROM CITIZEN WHERE GENDER IN 'Female'
    GROUP BY OCCUPATION HAVING MAX(SALARY)>=10000;
```

32. What is the number of citizens whose salary is not within the range 40000-50000?

```sql
SELECT COUNT(*) FROM CITIZEN WHERE SALARY NOT BETWEEN 40000 AND 50000;
```

33. If the state decided to oust the citizens whose salary is less than 20000, what will be the remaining number of citizens in Ctg? [Hint: You can use nested query]

```sql
SELECT COUNT(*) AS REMAINING_CHATGAIYYA FROM CITIZEN WHERE C_HOME IN
    'Ctg' AND SALARY >= 20000;
```

```
--Or, if you're feeling fancy and want to show off your subquery
    skills:
SELECT (SELECT COUNT(*) FROM CITIZEN WHERE C_HOME IN 'Ctg') - (SELECT
    COUNT(*) FROM CITIZEN WHERE C_HOME IN 'Ctg' AND SALARY<20000) AS
    REMAINING_CHATGAIYYA FROM DUAL;
```

34. If the state decided to oust the citizens whose salary is greater than 20000 but less than 50000, what will be the remaining number of citizens in Ctg? [Hint: You can use nested query]

```
SELECT COUNT(*) AS REMAINING_CHATGAIYYA FROM CITIZEN WHERE C_HOME IN
    'Ctg' AND NOT (SALARY>20000 AND SALARY<50000);
--Or:
SELECT (SELECT COUNT(*) FROM CITIZEN WHERE C_HOME IN 'Ctg' AND
    SALARY<=20000) + (SELECT COUNT(*) FROM CITIZEN WHERE C_HOME IN
    'Ctg' AND SALARY>=50000) AS REMAINING_CHATGAIYYA FROM DUAL;
```

35. Find the average salary of upper middle-class citizens (i.e. salary >= 55000) under each occupation if there are at least two citizens in that occupation.

```
SELECT OCCUPATION, AVG(SALARY) FROM CITIZEN WHERE SALARY>=55000 GROUP
    BY OCCUPATION HAVING COUNT(*)>=2;
```

36. Find the average salary of upper middle-class citizens (i.e. salary >= 55000) under each city if there are at least two citizens in that city.

```
SELECT C_HOME, AVG(SALARY) FROM CITIZEN WHERE SALARY>=55000 GROUP BY
    C_HOME HAVING COUNT(*)>=2;
```

Take a look at the following schema. You don't need to show any data entry for these tables. Create these tables manually and then write down the necessary SQL queries to answer the subsequent questions (37-40).

Districts(name(pk),size, weather)

Citizens(ID(pk), name, dob, dist(fk[District]))

CovidInjected(Citizen(pk) and fk[Citizens],date of vaccination,place of vaccination)

37. List the district as Name of district, size of district,total number of citizen.

```
SELECT DISTRICTS.NAME, DISTRICTS.SIZE, COUNT(*) AS TOT_CITIZEN FROM
    DISTRICTS INNER JOIN CITIZENS ON DISTRICTS.NAME=CITIZENS.DIST
    GROUP BY DISTRICTS.NAME, DISTRICTS.SIZE;
```

38. List the district names that have at most 50000 people in it.

```
SELECT DISTRICTS.NAME, COUNT(*) AS TOT_CITIZEN FROM DISTRICTS INNER
    JOIN CITIZENS ON DISTRICTS.NAME=CITIZENS.DIST GROUP BY
    DISTRICTS.NAME HAVING COUNT(*)<=50000;
```

39. List the top 3 districts based on the least number of vaccinated citizens.

```
SELECT NAME,TOT_VAC_CITIZEN FROM (SELECT DISTRICTS.NAME, COUNT(*) AS
    TOT_VAC_CITIZEN FROM CITIZENS INNER JOIN DISTRICTS ON
    CITIZENS.DIST=DISTRICTS.NAME INNER JOIN COVIDINJECTED ON
    CITIZENS.ID = COVIDINJECTED.CITIZEN GROUP BY DISTRICTS.NAME ORDER
    BY COUNT(*) ASC) WHERE ROWNUM<=3;
```

40. List the 3rd, 5th and 7th districts based on the number of vaccinated citizens.

    This was a trick question, mostly to see how you would use ROWNUM to solve it. If you have tried ROWNUM how you always do, you probably noticed that it doesn't work. The reason ROWNUM doesn't work normally is because ROWNUM is a number that is assigned to each row returned, indicating the order in which Oracle selects the row from a table or set of joined rows. If the rows returned are not discarded, then the first row selected has a ROWNUM of 1, the second has 2, and so on. Let's say you wrote

"WHERE ROWNUM=3". This will always return false, simply because when the first row is returned, it will have a ROWNUM of 1, which does not satisfy the condition and hence will be discarded. When the second row is returned, it will also have a ROWNUM of 1, because no other row has been generated and kept before it. And so, since this row also has a ROWNUM of 1, it will also be discarded. This has been explained in detail in Oracle's website: `https://docs.oracle.com/cd/B19306_01/server.102/b14200/pseudocolumns009.htm`

A proper way of using ROWNUM to solve this problem is by retrieving everything and assigning the ROWNUM to a "real" column (the column being temporary), then using that real column number to filter the results, like we do in the following:

```
1  SELECT * FROM (SELECT NAME,TOT_VAC_CITIZEN, ROWNUM AS POSITION FROM
       (SELECT DISTRICTS.NAME, COUNT(*) AS TOT_VAC_CITIZEN FROM CITIZENS
       INNER JOIN DISTRICTS ON CITIZENS.DIST=DISTRICTS.NAME INNER JOIN
       COVIDINJECTED ON CITIZENS.ID = COVIDINJECTED.CITIZEN GROUP BY
       DISTRICTS.NAME ORDER BY COUNT(*) ASC)) WHERE POSITION IN (3,5,7);
```