Faculty of Computer Science
Dalhousie University

DALHOUSIE UNIVERSITY
FACULTY OF COMPUTER SCIENCE

CSCI 2141 Introduction to Database Systems

Assignment 4

The Due Date is as listed in the ***Schedule of Due Dates -CSCI 2141 Winter 2019.PDF*** document located in the module ***Syllabus and Admin Support*** on Brightspace. As the assignment was posted on Sunday, March 3 and I stated that you will have a week to work on it, *there will not be any penalty if you submit this assignment by March 10, 8pm. After that time, there will be a late penalty of 5% for each of the next two days after which no more submissions will be permitted as solutions will be posted.*

Write name, student ID, and LAB section below and then write your answer directly into this document and submit its pdf version while making sure that the name of your file is of the format "A4-B00XXXXXX-YourLastName-2141.pdf", where "B00XXXXXX" is your student ID and "YourLastName" is your last name.

Note that this assignment is not asking you to use any DBMS and hence no screenshots should be submitted. Of course, you are not forbidden to create your own DB (e.g., using MySQL and Workbench and MySQL) and use it to check your answers. In fact, the SQL script files to create the DB schema and insert tuples into tables of this assignment have been provided for you for Assignment 3 and appear on Brightspace in the module ***Assignments*** (file ***TablesCreateInsert for A3 CSCI 2141 Winter 2019.zip****).*

Please, consider both single or double quotes to be correct for string literals.

First Name:	Yansong
Last Name:	Li
Student ID:	B00755354
LAB Section:	01

1. Part 1 – Write SQL queries to retrieve data from given tables

Consider tables, of a Supplier-Parts DB, shown in Figure 1 at the end of this paper. Names of tables, columns and identification of primary and foreign keys are shown. As the content of the rows/tuples of tables/relations is transient (it changes over time) and the following questions ask you to write SQL statements to retrieve certain data/information from the tables, sample tuples/rows are not shown. *Questions in this part ask you* ***to write SQL queries*** that would retrieve certain information from the tables.

Recall that this assignment is not asking you to use any DBMS and hence no screenshots should be submitted. Write your answers directly in this doc.

1. Display information about suppliers, such that the result is ordered by city (sCity) and then by status (sStatus).

```
SELECT *
FROM s
ORDER BY sCity, sStatus;
```

2. Find the number of suppliers.

```
SELECT COUNT(sNo) AS 'Number of Suppliers'
FROM s;
```

3. Find the number of suppliers located in each city. Display the information in reverse order by city.

```
SELECT sCity, COUNT(sNo) AS 'Number of Suppliers'
FROM s
GROUP BY Scity
ORDER BY scity DESC;
```

4. For each part, find the count, sum, minimum, maximum, and average qty on order. Display information in the ascending order by the sum of quantities.

```
SELECT       pNo, COUNT(pNo), SUM(qty), MIN(qty), MAX(qty), AVG(qty)
FROM         sp
GROUP BY     pNo
ORDER BY     SUM(qty) ASC;
```

5. For each red part, find the count, sum, minimum, maximum, and average qty on order. Display information in the ascending order by the sum of quantities.

```
SELECT       pNo, COUNT(pNo), SUM(qty), MIN(qty), MAX(qty), AVG(qty)
FROM         sp JOIN p USING (pNo)
WHERE        p.pColor = 'Red'
GROUP BY     pNo
ORDER BY     SUM(qty) ASC;
```

6. For each part find the total number of orders, total quantity on order, minimum, maximum, and average qty on order from suppliers located in ‘London’.

```
SELECT          sp.pNo, COUNT(sp.sNo) AS 'Total number of orders', SUM(qty), MIN(qty), MAX(qty),
                AVG(qty)
FROM            sp JOIN s USING (sNo)
WHERE           s.sCity = 'London'
GROUP BY        sp.pNo;
```

7. For each red part find the total number of orders, total quantity on order, minimum, maximum, and average qty on order from suppliers located in ‘London’.

```
SELECT          sp.pNo, COUNT(sp.sNo) AS 'Total number of orders', SUM(qty), MIN(qty), MAX(qty),
                AVG(qty)
FROM            sp JOIN p USING (pNo) JOIN s USING (sNo)
WHERE           s.sCity = 'London' AND p.pColor = 'Red'
GROUP BY        sp.pNo;
```

8. Display the total quantity on order for each part that has more than 500 of total quantity on order. Display information in reverse order by part number.

```
SELECT           pNo,SUM(qty) AS 'Total qty'
FROM            sp
GROUP BY        pNo
HAVING          SUM(qty) > 500
ORDER BY        pNo DESC;
```

9. For each part display the total, minimum, average, and maximum quantity of large individual orders as well as the number of such orders. A “larger order” is any tuple (sNo, pNo, qty) in SP that has a quantity larger than 150. Display information in reverse order by part number.

```
SELECT           pNo,SUM(qty) AS 'Total qty', MIN(qty), MAX(qty), AVG(qty)
FROM            sp
WHERE           qty IN (SELECT qty FROM sp WHERE qty > 150)
GROUP BY        pNo
ORDER BY        pNo DESC;
```

10. As above, but display information only for those parts for which the total quantity on order is greater than 500.

```
SELECT           pNo,SUM(qty) AS 'Total qty', MIN(qty), MAX(qty), AVG(qty)
FROM            sp
GROUP BY        pNo
HAVING          SUM(qty) > 500
ORDER BY        pNo DESC;
```

11. For each red part display the total, minimum, average, and maximum quantity of large individual orders, but only from suppliers from ‘London’, as well as the number of such orders. A “larger order” is any tuple (sNo, pNo, qty) in SP that has a quantity larger than 150. Display information in ascending order by the total quantity.

```
SELECT       sp.pNo, COUNT(sp.sNo) AS 'Total number of orders',SUM(qty) AS 'Total qty', MIN(qty),
             MAX(qty), AVG(qty)
FROM         sp JOIN p USING (pNo) JOIN s USING (sNo)
WHERE        s.sCity = 'London' AND p.pColor = 'Red' AND qty IN
             (SELECT qty FROM sp WHERE qty > 150)
GROUP BY     sp.pNo;
ORDER BY     SUM(qty) ASC;
```

2. Part 2 – Given a set of tables with their content (tuples/rows), show what would be retrieved if the given SQL queries were issued

Consider tables, of a Supplier-Parts DB, shown in Figure 2. Names of tables, columns and identification of primary and foreign keys are shown. *Questions in this part ask you* ***to show what would be displayed by the following SQL queries*** should they be executed against the tables having content (tuples/rows) shown in Figure 2. In each answer to a question, do not forget to display column headings in addition to the tuples/rows that would be shown.

Note for those who are familiar with the SAFE mode – assume that the SAFE mode is turned off (although it should not matter). If you feel that there is an error in the statement, e.g., instead of "FROM", it is spelled "FORM", please make and note a reasonable assumption (e.g., I assume that "FORM" should have been "FROM") and proceed. Also, *consider both single or double quotes to be correct for string literals.*

Recall that this assignment is not asking you to use any DBMS and hence no screenshots should be submitted. Write your answers directly in this doc.

1.

```
SELECT      sNo, SUM(qty)
FROM        sp
GROUP BY    sNo
ORDER BY    sNo DESC;
```

sNo	SUM(qty)
S5	200
S4	600
S3	500
S2	700
S1	1200

2.

```
SELECT      pNo, COUNT(pNo), SUM(qty), MIN(qty), AVG(qty), MAX(qty)
FROM        sp JOIN p USING (pNo)
GROUP BY    pNo
ORDER BY    pNo ASC;
```

pNo	COUNT(pNo)	SUM(qty)	MIN(qty)	AVG(qty)	MAX(qty)
P1	3	700	100	233.3333	300
P2	4	1000	200	250.0000	400
P3	3	1000	200	333.3333	400
P4	2	400	200	200.0000	200
P5	1	100	100	100.0000	100

3.

```
SELECT      pNo, COUNT(pNo), SUM(qty), MIN(qty), AVG(qty), MAX(qty)
FROM        sp JOIN p USING (pNo) JOIN s USING (sNo)
WHERE       sCity = 'London' AND pColor = 'Red' AND qty >150
GROUP BY    pNo
ORDER BY    SUM(qty);
```

pNo	COUNT(pNo)	SUM(qty)	MIN(qty)	AVG(qty)	MAX(qty)
P4	1	200	200	200.0000	200
P1	2	600	300	300.0000	300

S

sNo (PK)	**sNo (PK)**	**sNo (PK)**	**sNo (PK)**
…	…	…	…

P

pNo (PK)	**pName**	**pColor**	**pWeight**
…	…	…	…

SP

sNo (PK) (FK)	**pNo (PK) (FK)**	**qty**
…	…	…

Figure 1. Supplier-Parts DB Tables

****

S

sNo (PK)	**sName**	**sStatue**	**sCity**
S1	Smith	20	London
S2	Jones	10	London
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

P

pNo (PK)	**pName**	**pColor**	**pWeight**
P1	Nut	Red	12
P2	Bolt	Blue	17
P3	Screw	Green	17
P4	Screw	Red	14
P5	Cam	Blue	12

SP

sNo (PK) (FK)	**pNo (PK) (FK)**	**qty**
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S2	P1	300
S2	P2	400
S3	P1	100
S3	P2	200
S3	P3	200
S4	P2	200
S4	P3	400
S5	P4	200

Figure 2. Content of the Supplier-Parts DB Tables