Managing & Programming Database

Trishla Shah

Assignment 1
September 30, 2018.

Mohd Nawaz Hussain

A00428036

1(a) Get the names and locations of the suppliers who have shipped part with pno = 3.

```
69
70    /* Answer a */
71
72    SELECT
73       s.sname as "Supplier Name", s.city as " City"
74    FROM
75       s,
76       sp
77    WHERE
78       sp.pno = "3" AND s.sno = sp.sno;
79
```

100% 15:70

Result Grid | Filter Rows: Search | Expor

Supplier Name	City
sn1	London
sn2	Paris
sn3	London

Fig 1.1

1(b) Get the part numbers and names of parts that have been shipped by suppliers located in Paris with status at least 20.

```
81  /*Answer b*/
82
83  SELECT
84     p.pname as "Part Name" , p.pno as "Part Number"
85  FROM
86     p
87  WHERE
88     p.pno IN (SELECT
89           sp.pno
90        FROM
91           sp
92        WHERE
93           sp.sno IN (SELECT
94                 s.sno
95              FROM
96                 s
97              WHERE
98                 city = "Paris" AND status >= 20));
99
100 /*Answer c */
101
102 SELECT
```

100% 22:88

Result Grid Filter Rows: Search Export:

Part Name	Part Number
pn1	1
pn2	2
pn3	3
pn4	4

Fig 1.2

1(c) For each part, show the part number, name, and the number of suppliers who have supplied the part.

```
100  /*Answer c */
101
102  SELECT
103     p.pno AS "Parts Number",
104     p.pname AS "Part Name",
105     COUNT(s.sno) AS "No of Suppliers"
106  FROM
107     s,
108     p,
109     sp
110  WHERE
111     p.pno = sp.pno AND s.sno = sp.sno
112  GROUP BY p.pno;
113
114  /* Answer d */
115
```

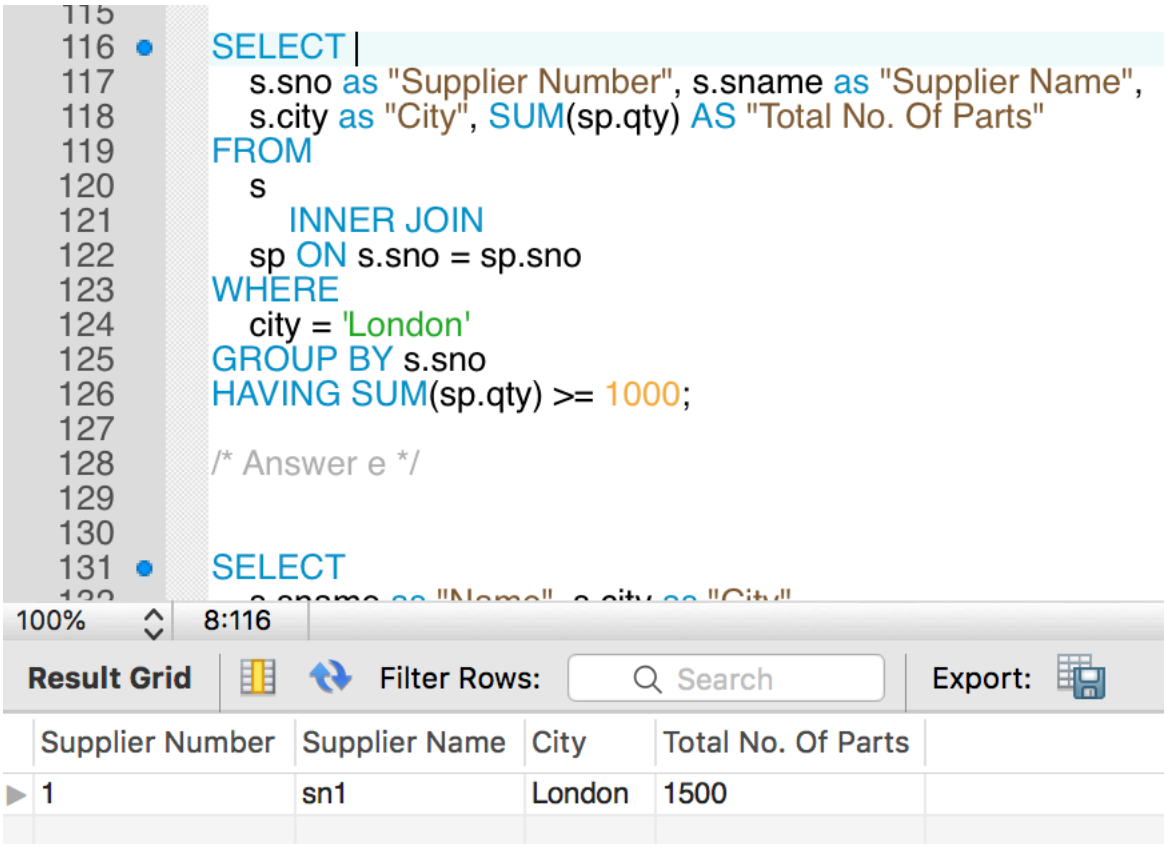
100% 29:103

Result Grid | Filter Rows: Search

Parts Number	Part Name	No of Suppliers
1	pn1	4
2	pn2	3
3	pn3	3
4	pn4	2
5	pn5	1

Fig 1.3

1 (d) For each London supplier who has shipped at least 1000 parts, show the name of the supplier and the total number of parts he/she has shipped.



```
115
116  SELECT
117     s.sno as "Supplier Number", s.sname as "Supplier Name",
118     s.city as "City", SUM(sp.qty) AS "Total No. Of Parts"
119  FROM
120     s
121        INNER JOIN
122     sp ON s.sno = sp.sno
123  WHERE
124     city = 'London'
125  GROUP BY s.sno
126  HAVING SUM(sp.qty) >= 1000;
127
128  /* Answer e */
129
130
131  SELECT
```

100% 8:116

Result Grid | Filter Rows: Search | Export:

Supplier Number	Supplier Name	City	Total No. Of Parts
1	sn1	London	1500

Fig 1.4

1 (e) Get the names and cities of the suppliers who have supplied all parts that weigh less than 4 grams.

```
131  SELECT
132     s.sname as "Name", s.city as "City"
133  FROM
134     s
135        INNER JOIN
136     sp ON s.sno = sp.sno
137        INNER JOIN
138     p ON sp.pno = p.pno
139  WHERE
140     p.weight < 4
141  GROUP BY sp.sno
142  HAVING COUNT(s.sno) = (SELECT
143        COUNT(p.pno)
144     FROM
145        p
146     WHERE
147        p.weight < 4);
148
```

100% 19:137

Result Grid | Filter Rows: Search

Name	City
sn1	London
sn2	Paris
sn3	London

Fig 1.5

Question 2(a)

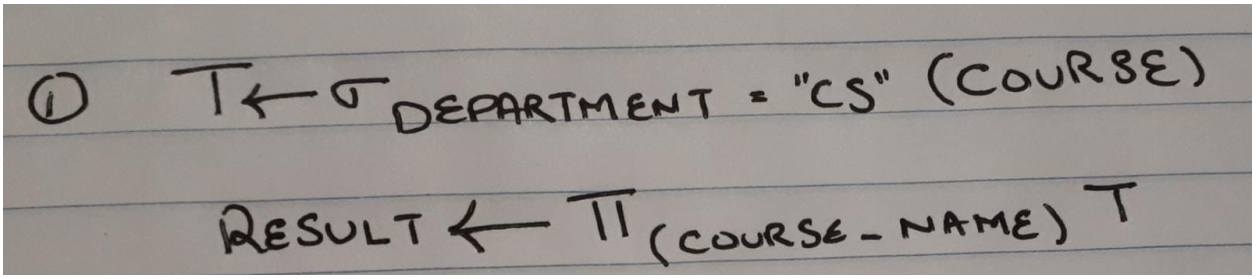


Fig 2.1

2(b)

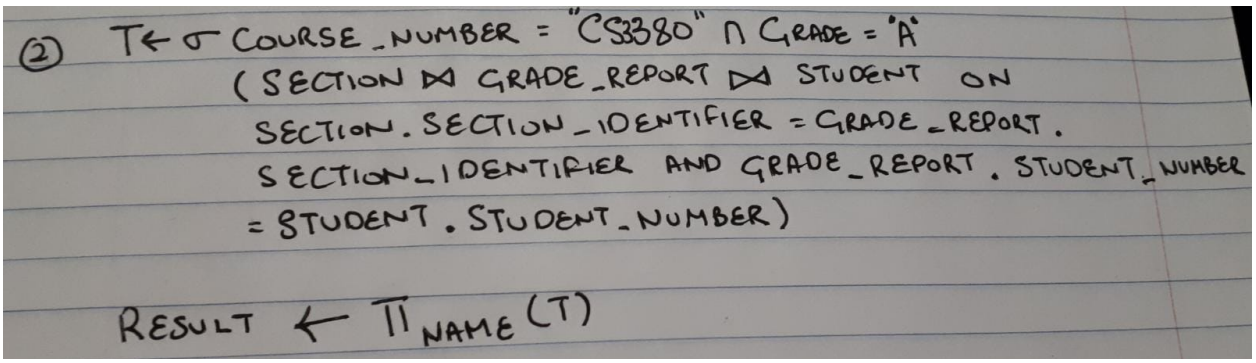
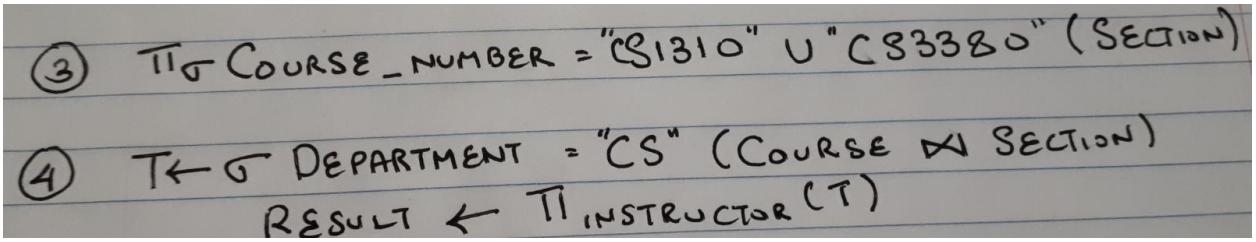


Fig 2.2

2(c) & (d)



Question 3(a)

```
1
2   /* Question 3a*/
3
4   SELECT
5      Country,
6      SUM(Gender = 'M') AS Male,
7      SUM(Gender = 'F') AS Female,
8      SUM(Gender = 'M') * 100 / COUNT(*) AS perMale
9   FROM
10     customer
11  GROUP BY country
12  ORDER BY perMale ASC;
13
14  /* Question 3b*/
```

100% 8:4

Result Grid | Filter Rows: Search | Export:

Country	Male	Female	perMale
ZA	1	3	25.0000
CA	7	8	46.6667
US	15	13	53.5714
AU	5	3	62.5000
DE	7	3	70.0000
TR	7	0	100.0000
IL	5	0	100.0000

Fig 3.1

Question 3(b)

```
15
16  SET sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
17
18  SELECT
19     p.product_id as "Product ID",
20     p.Product_Name as "Product Name",
21     SUM(o.quantity) AS "Total Sold"
22  FROM
23     product_dim p
24      Inner Join order_fact o
25  WHERE
26     o.product_id = p.Product_ID
27  GROUP BY p.product_id
28  ORDER BY SUM(o.quantity) DESC;
29
30
```

100% 8:18

Result Grid | Filter Rows: Search | Export:

Product ID	Product Name	Total Sold
230100700011	Hurricane 4	8
240700100001	Armour L	8
230100600016	Expedition Zero,Medium,Right,Charcoal	8
230100600030	Outback Sleeping Bag, Large,Left,Blue/Black	8
240400300035	Smasher Shorts	7
220101400387	N.d.gear Cap	7
240700200010	Bat - Home Run S	7
240500100017	A-team Sweat Round Neck, Small Logo	7
230100600031	Outback Sleeping Bag, Large,Right, Blue/Black	7
240100400129	Rollerskate Roller Skates Sq9 80-76mm/78a	6
240200100118	Hi-fly Intrepid Stand 8 Black	6
240800200035	Shine Black PRO	6
240200200039	Eagle Plain Cap	6

Result 2

Fig 3.2

Question 3(c)

```
32
33  SELECT
34     emp1.Employee_ID as "Employee ID",
35     emp1.Employee_Name as "Employee Name",
36     s.Job_Title as "Job Title",
37     s.Manager_ID as "Manager ID",
38     emp2.Employee_Name AS "Employee Name"
39  FROM
40     employee_addresses emp1,
41     employee_addresses emp2,
42     staff s
43  WHERE
44     emp2.employee_id = s.Manager_ID
45        AND emp1.Employee_ID = s.Employee_ID
46        AND s.Job_Title LIKE '%TEMP%'
47        OR s.Job_Title LIKE '%train%'
48  ORDER BY s.Employee_ID;
49
```

100% 38:46

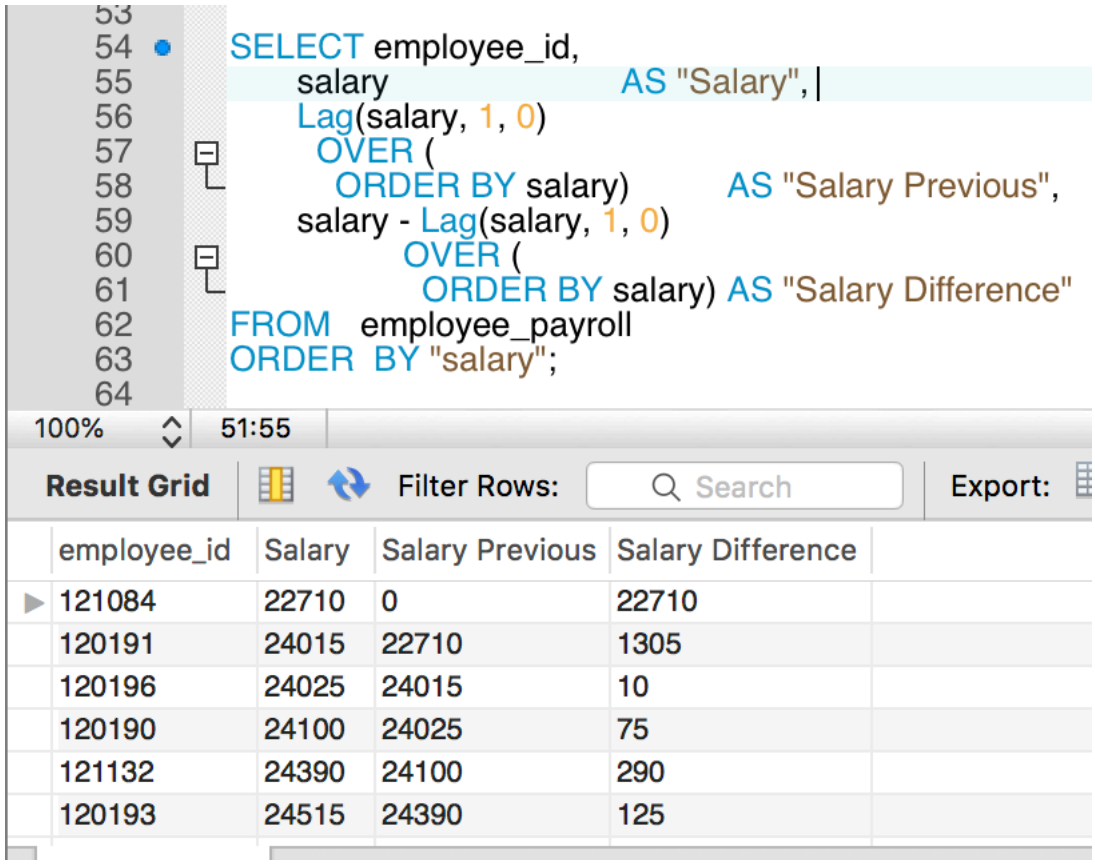
Result Grid | Filter Rows: Search | Export: | Fetch rows:

Employee ID	Employee Name	Job Title	Manager ID	Employee Name
120181	Cantatore, Lorian	Temp. Sales Rep.	120103	Dawes, Wilson
120182	Barreto, Geok-Seng	Temp. Sales Rep.	120103	Dawes, Wilson
120183	Blanton, Brig	Temp. Sales Rep.	120103	Dawes, Wilson
120184	Moore, Ari	Temp. Sales Rep.	120103	Dawes, Wilson
120185	Bahlman, Sharon	Temp. Sales Rep.	120103	Dawes, Wilson
120186	Quinby, Merryn	Temp. Sales Rep.	120103	Dawes, Wilson
120187	Catenacci, Reyne	Temp. Sales Rep.	120103	Dawes, Wilson
120188	Baran, Shanmuganathan	Temp. Sales Rep.	120103	Dawes, Wilson
120189	Lachlan, Mihailo	Temp. Sales Rep.	120103	Dawes, Wilson
121044	Abbott, Ray	Trainee	120103	Abbott, Ray
120145	Aisbitt, Sandy	Trainee	120103	Abbott, Ray
120761	Akinfolarin, Tameaka	Trainee	120103	Abbott, Ray

Result 3

Fig 3.3

Question 3(d)



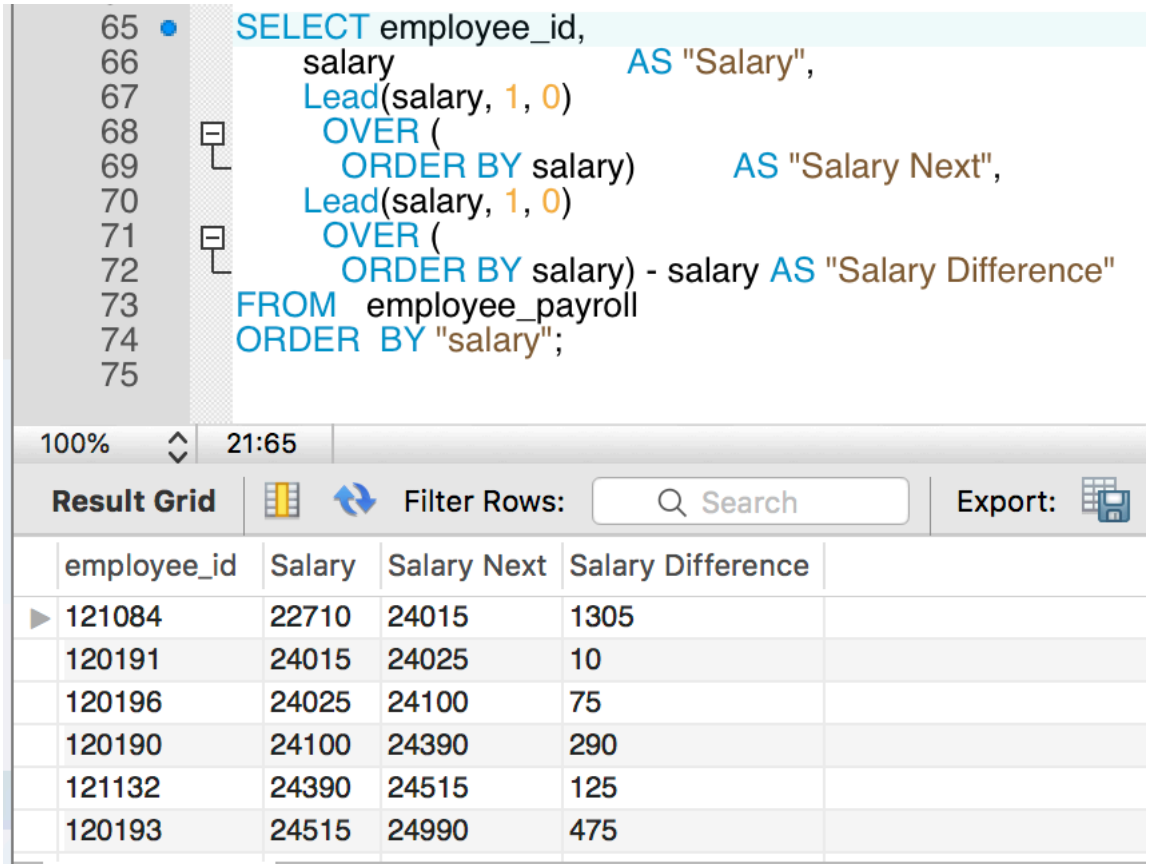
```
53
54  SELECT employee_id,
55         salary                    AS "Salary",
56         Lag(salary, 1, 0)
57           OVER (
58             ORDER BY salary)      AS "Salary Previous",
59         salary - Lag(salary, 1, 0)
60                  OVER (
61                    ORDER BY salary) AS "Salary Difference"
62  FROM   employee_payroll
63  ORDER  BY "salary";
64
```

100% 51:55

Result Grid | Filter Rows: Search | Export:

employee_id	Salary	Salary Previous	Salary Difference
121084	22710	0	22710
120191	24015	22710	1305
120196	24025	24015	10
120190	24100	24025	75
121132	24390	24100	290
120193	24515	24390	125

Fig 3.4



```
65  SELECT employee_id,
66         salary                    AS "Salary",
67         Lead(salary, 1, 0)
68           OVER (
69             ORDER BY salary)      AS "Salary Next",
70         Lead(salary, 1, 0)
71           OVER (
72             ORDER BY salary) - salary AS "Salary Difference"
73  FROM   employee_payroll
74  ORDER  BY "salary";
75
```

100% 21:65

Result Grid | Filter Rows: Search | Export:

employee_id	Salary	Salary Next	Salary Difference
121084	22710	24015	1305
120191	24015	24025	10
120196	24025	24100	75
120190	24100	24390	290
121132	24390	24515	125
120193	24515	24990	475

Fig 3.5

References

- https://stackoverflow.com/questions/36207042/error-code-1055-incompatible-with-sql-mode-only-full-group-by
- https://oracle-base.com/articles/misc/lag-lead-analytic-functions
- https://www.computing.dcu.ie/~mhughes/Lectures/Ch7Supp2Up.pdf
- https://www.youtube.com/watch?v=_ex8FpmMCeE&t=368s