HOMEWORK WEEK 1

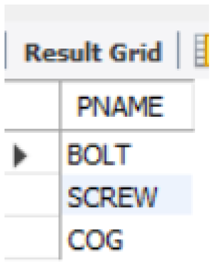
(handout for students)

TASK 1

USE PARTS DB TO WRITE THE FOLLOWING QUERIES

1. Find the name of each part where the weight is more than 14.

```
SELECT part.PNAME

FROM part

WHERE WEIGHT >14;
```

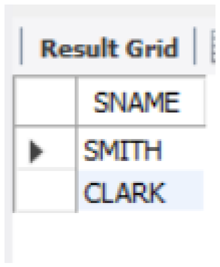


Result Grid

	PNAME
▶	BOLT
	SCREW
	COG

2. Find all **unique** supplier(s) where their status is equal to 20.

```
SELECT DISTINCT SNAME

FROM supplier

WHERE STATUS = 20;
```



Result Grid

	SNAME
▶	SMITH
	CLARK

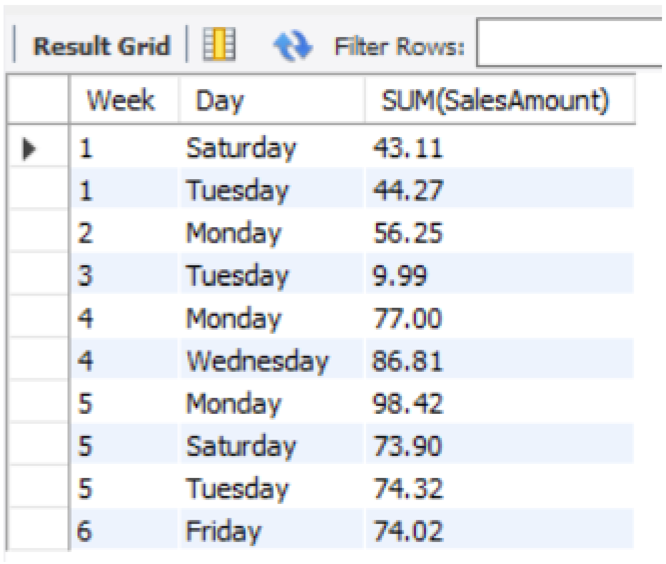
(*you must submit the code and results*)

TASK 2

USE SHOP SALES DB TO WRITE THE FOLLOWING QUERIES

1. Find out how many sales (amount) were recorded each week, per day (where available)
    - **This would look like:**
      Week 1, Tuesday, £x
      Week 1, Wednesday, £x
      Week 2, Monday, £x
      Week 2, Friday, £x

```
SELECT Week, Day, SUM(SalesAmount)

FROM sales1

GROUP BY Week, Day

ORDER BY Week;
```

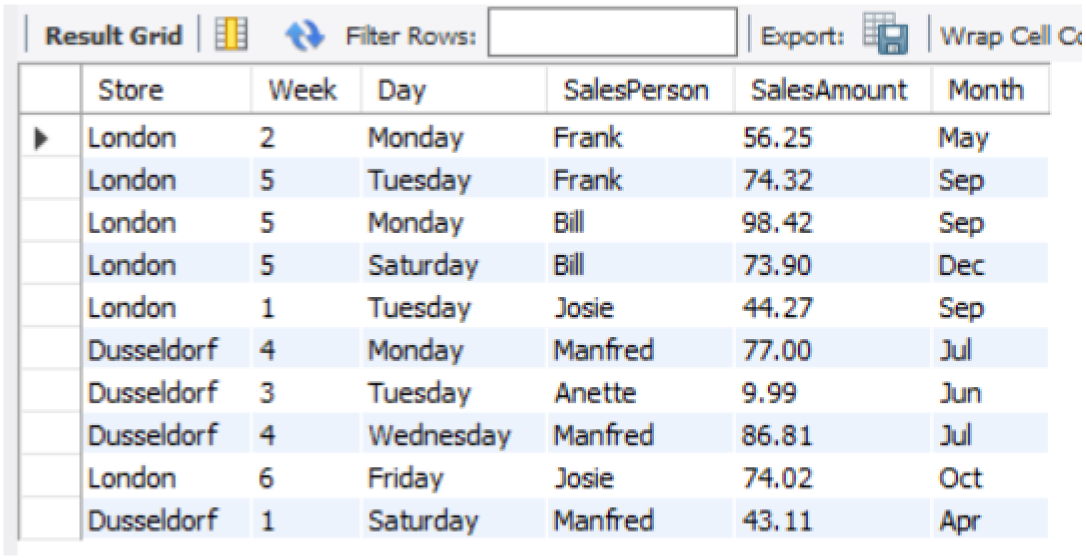


Result Grid | Filter Rows:

Week	Day	SUM(SalesAmount)
1	Saturday	43.11
1	Tuesday	44.27
2	Monday	56.25
3	Tuesday	9.99
4	Monday	77.00
4	Wednesday	86.81
5	Monday	98.42
5	Saturday	73.90
5	Tuesday	74.32
6	Friday	74.02

2. Change the name of salesperson Inga to be Annette instead, but only where Ignas Sales are <50.

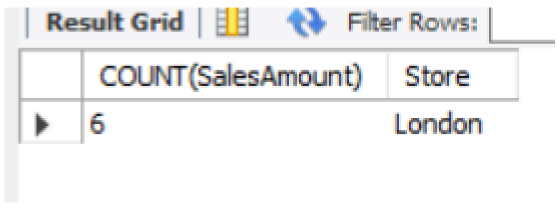
```
-- Disable SAFE UPDATES
SET SQL_SAFE_UPDATES = 0;
UPDATE sales1
SET SalesPerson = 'Anette'
WHERE SalesPerson='Inga' AND SalesAmount <50;

SELECT *
FROM sales1;

-- Re-enable SAFE UPDATES
SET SQL_SAFE_UPDATES = 1;
```



Store	Week	Day	SalesPerson	SalesAmount	Month
London	2	Monday	Frank	56.25	May
London	5	Tuesday	Frank	74.32	Sep
London	5	Monday	Bill	98.42	Sep
London	5	Saturday	Bill	73.90	Dec
London	1	Tuesday	Josie	44.27	Sep
Dusseldorf	4	Monday	Manfred	77.00	Jul
Dusseldorf	3	Tuesday	Anette	9.99	Jun
Dusseldorf	4	Wednesday	Manfred	86.81	Jul
London	6	Friday	Josie	74.02	Oct
Dusseldorf	1	Saturday	Manfred	43.11	Apr

3. Find out how many sales the London office logged
    - Note we're looking for quantity here - e.g. if London did 6 sales, then output would be 6)

```
SELECT COUNT(SalesAmount), Store

FROM sales1

GROUP BY Store

HAVING Store = 'London';
```

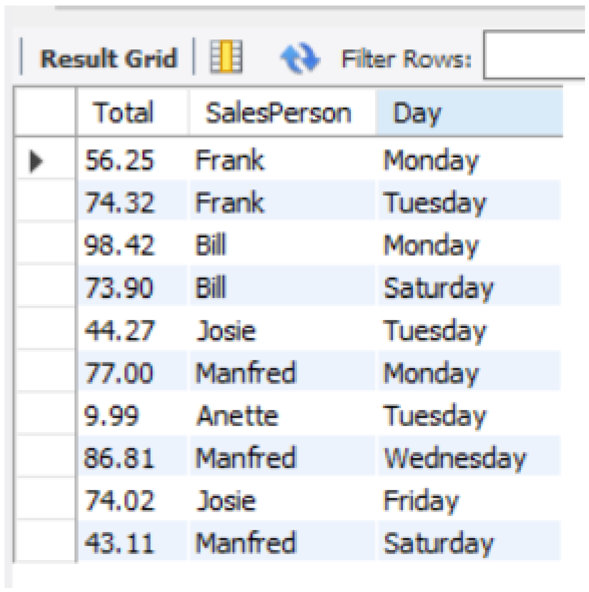


Result Grid | Filter Rows:

COUNT(SalesAmount)	Store
6	London

4. Find the total (sum) sales amount by each person by day

```
SELECT SUM(SalesAmount) AS Total, SalesPerson, Day

FROM sales1

GROUP BY SalesPerson, Day;
```

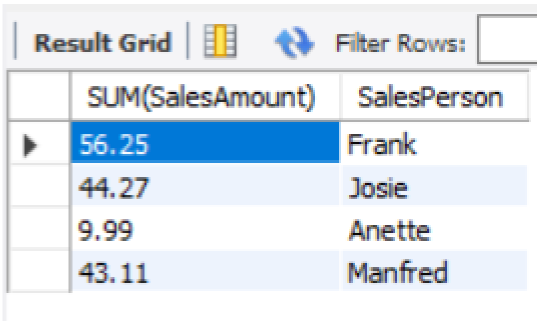


Result Grid | Filter Rows:

Total	SalesPerson	Day
56.25	Frank	Monday
74.32	Frank	Tuesday
98.42	Bill	Monday
73.90	Bill	Saturday
44.27	Josie	Tuesday
77.00	Manfred	Monday
9.99	Anette	Tuesday
86.81	Manfred	Wednesday
74.02	Josie	Friday
43.11	Manfred	Saturday

5. How much (sum) each person sold for between week 1 and week 3

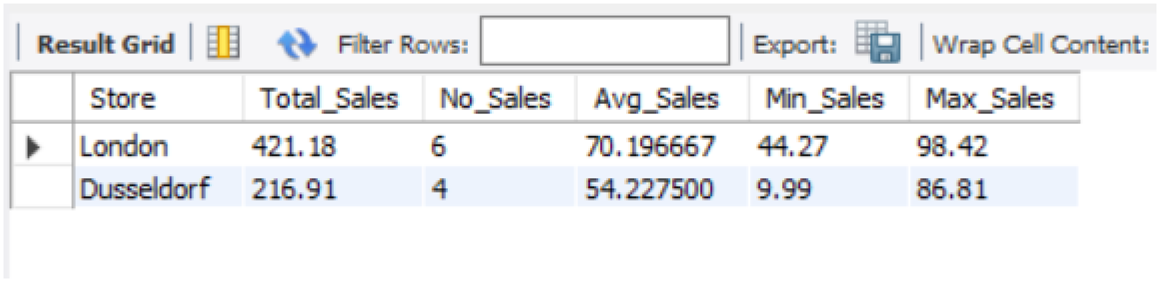
```
SELECT SUM(SalesAmount), SalesPerson

FROM sales1

WHERE Week >=1 AND Week <=3

GROUP BY SalesPerson;
```



SUM(SalesAmount)	SalesPerson
56.25	Frank
44.27	Josie
9.99	Anette
43.11	Manfred

6. For each store:
- The total of their sales;
- The number of sales;
- Their average sales;
- Their lowest sales amount;
- Their highest sales amount.

```
SELECT
Store,
  SUM(SalesAmount) AS Total_Sales,
  COUNT(SalesAmount) AS No_Sales,
  AVG (SalesAmount) AS Avg_Sales,
  MIN(SalesAmount) AS Min_Sales,
  MAX(SalesAmount) AS Max_Sales
FROM sales1
GROUP BY Store;
```



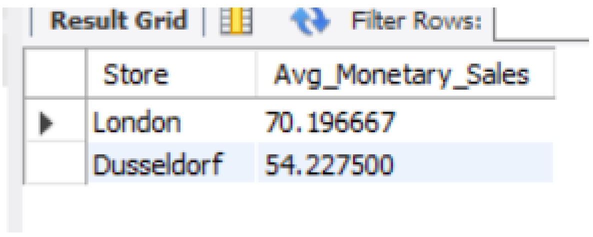
Store	Total_Sales	No_Sales	Avg_Sales	Min_Sales	Max_Sales
London	421.18	6	70.196667	44.27	98.42
Dusseldorf	216.91	4	54.227500	9.99	86.81

7. Find the average (AVG) monetary sales amount achieved by each store

SELECT Store, AVG(SalesAmount) AS Avg_Monetary_Sales

FROM sales1

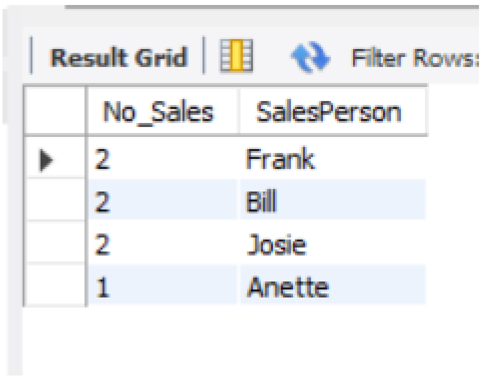
GROUP BY Store;



Store	Avg_Monetary_Sales
London	70.196667
Dusseldorf	54.227500

8. Count the number of sales by each person if they had less than 3 sales for the past period

```
SELECT COUNT(SalesAmount) AS No_Sales, SalesPerson

FROM sales1

GROUP BY SalesPerson

HAVING No_Sales <3;
```

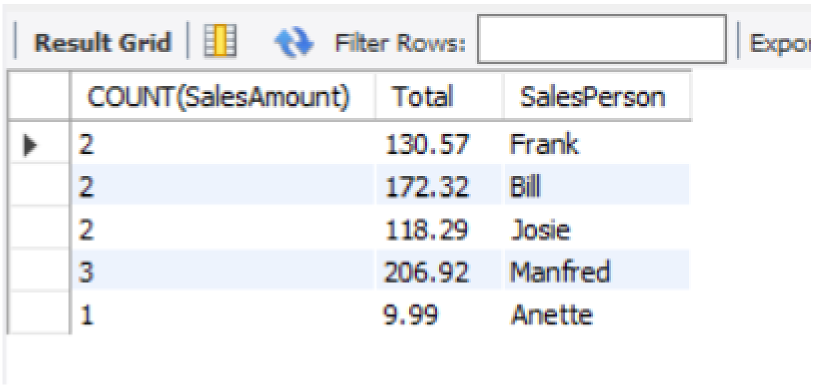


Result Grid | Filter Rows:

	No_Sales	SalesPerson
▶	2	Frank
	2	Bill
	2	Josie
	1	Anette

9. Find the number (count) of sales by each person, but only if they made less than or equal to £300 worth of sales for the past period

```
SELECT COUNT(SalesAmount), SUM(SalesAmount) AS Total,
  SalesPerson
FROM sales1
GROUP BY SalesPerson
HAVING Total <=300;
```



Result Grid | Filter Rows: | Expo

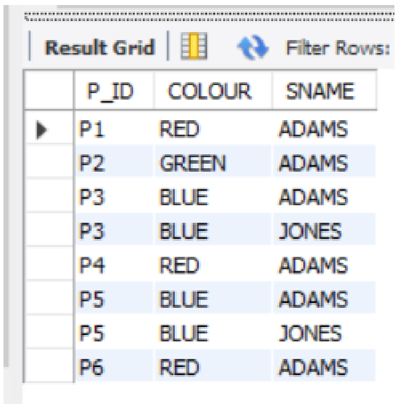
	COUNT(SalesAmount)	Total	SalesPerson
▶	2	130.57	Frank
	2	172.32	Bill
	2	118.29	Josie
	3	206.92	Manfred
	1	9.99	Anette

TASK 3

USE PARTS DB TO WRITE THE FOLLOWING QUERIES

1. Return the PartID, Colour and Supplier name, where the supplier's surname ends in an S, and the Supplier city is not London. Ensure the values are Unique.

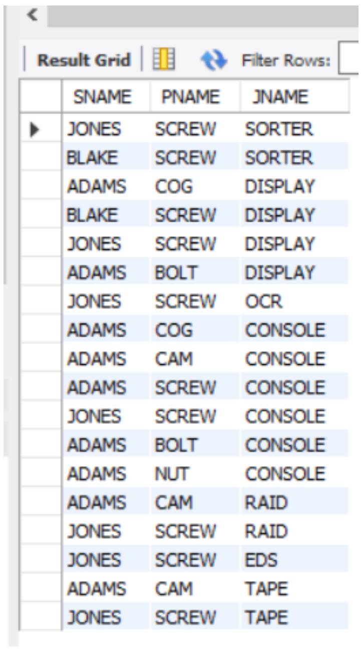
```
SELECT DISTINCT part.P_ID, part.COLOUR,
  supplier.SNAME

FROM part

JOIN supply

ON part.P_ID = supply.P_ID

JOIN supplier

ON supply.S_ID = supplier.S_ID

WHERE supplier.SNAME LIKE '%s'AND supplier.CITY !=
  'LONDON';
```



P_ID	COLOUR	SNAME
P1	RED	ADAMS
P2	GREEN	ADAMS
P3	BLUE	ADAMS
P3	BLUE	JONES
P4	RED	ADAMS
P5	BLUE	ADAMS
P5	BLUE	JONES
P6	RED	ADAMS

2. Return the supplier name, part name and project name for each case where the following conditions are true:
    i. The supplier supplies a project with a part;
    Ii. And where the supplier's city, project city and part city are the same.

```
SELECT DISTINCT supplier.SNAME, part.PNAME, project.JNAME

FROM supply

JOIN part

ON supply.P_ID = part.P_ID

JOIN supplier

ON supply.S_ID = supplier.S_ID

JOIN project

ON supply.J_ID = project.J_ID

WHERE part.CITY = supplier.CITY = project.CITY;
```



Result Grid | Filter Rows:

SNAME	PNAME	JNAME
JONES	SCREW	SORTER
BLAKE	SCREW	SORTER
ADAMS	COG	DISPLAY
BLAKE	SCREW	DISPLAY
JONES	SCREW	DISPLAY
ADAMS	BOLT	DISPLAY
JONES	SCREW	OCR
ADAMS	COG	CONSOLE
ADAMS	CAM	CONSOLE
ADAMS	SCREW	CONSOLE
JONES	SCREW	CONSOLE
ADAMS	BOLT	CONSOLE
ADAMS	NUT	CONSOLE
ADAMS	CAM	RAID
JONES	SCREW	RAID
JONES	SCREW	EDS
ADAMS	CAM	TAPE
JONES	SCREW	TAPE