**HOMEWORK WEEK 1**

**TASK 1**

## USE PARTS DB TO WRITE THE FOLLOWING QUERIES

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

```
1  Use parts;
2
3  -- TASK 1.1
4  SELECT
5      p.pname, p.weight
6  FROM
7      part p
8  WHERE
9      p.weight > 14;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| pname | weight |
|---|---|
| BOLT | 17 |
| SCREW | 17 |
| COG | 19 |

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

```
12  Use parts;
13  -- TASK 1.2
14  SELECT DISTINCT
15      sr.sname, sr.status
16  FROM
17      supplier sr
18  WHERE
19      sr.status = 20;
```

Result Grid | Filter Rows:

| sname | status |
|---|---|
| SMITH | 20 |
| CLARK | 20 |

**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

```
1  USE shop;
2
3  -- TASK 2.1
4  SELECT
5      s.Week, s.Day, s.SalesAmount
6  FROM
7      sales1 s
8  ORDER BY
9      s.Week;
```

Result Grid | Filter Rows: | Export:

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

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

```
11  -- TASK 2.2
12  UPDATE
13      sales1 s
14  SET
15      s.SalesPerson = 'Anette'
16  WHERE
17      s.SalesPerson = 'Inga' AND s.SalesAmount < 50;
18
19  SELECT s.SalesPerson, s.SalesAmount from sales1 s
20
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| SalesPerson | SalesAmount |
|---|---|
| Frank | 56.25 |
| Frank | 74.32 |
| Bill | 98.42 |
| Bill | 73.90 |
| Josie | 44.27 |
| Manfred | 77.00 |
| Anette | 9.99 |
| Manfred | 86.81 |
| Josie | 74.02 |
| Manfred | 43.11 |

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)

```
21  USE shop;
22  -- TASK 2.3
23  SELECT
24      count(s.store) as sales_logged_in_London
25  FROM
26      sales1 s
27  WHERE
28      s.Store = 'London';
29
```

Result Grid | Filter Rows: | Export: | Wrap Cell

| sales_logged_in_London |
|---|
| 6 |

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

```
29  USE shop;
30  -- TASK 2.4
31  SELECT
32      s.SalesPerson, s.Day, sum(s.SalesAmount) as TotalSales
33  FROM
34      sales1 s
35  GROUP BY
36      s.SalesPerson, s.Day
37  ORDER BY
38      s.SalesPerson;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

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

```
39 •  USE shop;
40    -- TASK 2.5
41 •  SELECT
42        s.SalesPerson, sum(s.SalesAmount) as TotalSales
43    FROM
44        sales1 s
45    WHERE
46        s.Week BETWEEN 1 AND 3
47    GROUP BY
48        s.SalesPerson;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| SalesPerson | TotalSales |
|---|---|
| Frank | 56.25 |
| Josie | 44.27 |
| Anette | 9.99 |
| Manfred | 43.11 |

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

```
49 •  USE shop;
50    -- TASK 2.6
51 •  SELECT
52        s.Store, sum(s.SalesAmount) as TotalSales, count(s.SalesAmount) as SalesLoggedInStore,
53        avg(s.SalesAmount) as AverageSales, min(s.SalesAmount) as LowestSales, max(s.SalesAmount) as HigestSales
54    FROM
55        sales1 s
56    GROUP BY
57        s.Store;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| Store | TotalSales | SalesLoggedInStore | AverageSales | LowestSales | HigestSales |
|---|---|---|---|---|---|
| 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

```
58 •  USE shop;
59    -- TASK 2.7
60 •  SELECT
61        s.Store, avg(s.SalesAmount) as AverageSales
62    FROM
63        sales1 s
64    GROUP BY
65        s.Store;
66
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| Store | AverageSales |
|---|---|
| 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

```
66 •  USE shop;
67    -- TASK 2.8
68 •  SELECT
69        s.SalesPerson, count(s.SalesPerson) as NumberOfSales
70    FROM
71        sales1 s
72    GROUP BY s.SalesPerson
73    HAVING NumberOfSales < 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| SalesPerson | NumberOfSales |
|---|---|
| Frank | 2 |
| Bill | 2 |
| Josie | 2 |
| Anette | 1 |

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

```
74      -- TASK 2.9
75  •   USE shop;
76  •   SELECT
77          s.SalesPerson, count(s.SalesPerson) as NumberOfSales, sum(s.SalesAmount) as TotalSales
78      FROM
79          sales1 s
80      GROUP BY
81          s.SalesPerson
82      HAVING sum(s.SalesAmount) <= 300;
83
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| SalesPerson | NumberOfSales | TotalSales |
|---|---|---|
| Anette | 1 | 9.99 |
| Bill | 2 | 172.32 |
| Frank | 2 | 130.57 |
| Josie | 2 | 118.29 |
| Manfred | 3 | 206.92 |

**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.

```
1  •   Use Parts;
2
3      -- Task 3.1
4  •   SELECT distinct
5          part.P_ID, part.Colour, supplier.SNAME
6      FROM
7          part
8      INNER JOIN supply ON supply.P_ID = part.P_ID
9      INNER JOIN supplier ON supplier.S_ID = supply.S_ID
10     WHERE supplier.SNAME LIKE '%s' and supplier.CITY != 'London';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| 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.

```
13
14 •  Use Parts;
15    -- Task 3.2
16 •  SELECT supplier.SNAME, part.PNAME, project.JNAME
17    FROM part
18    INNER JOIN supply ON supply.P_ID = part.P_ID
19    INNER JOIN  supplier ON supplier.S_ID = supply.S_ID
20    INNER JOIN project ON project.J_ID = supply.J_ID
21    WHERE part.CITY = supplier.CITY AND supplier.CITY = project.CITY;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| SNAME | PNAME | JNAME |
|---|---|---|
| CLARK | COG | TAPE |