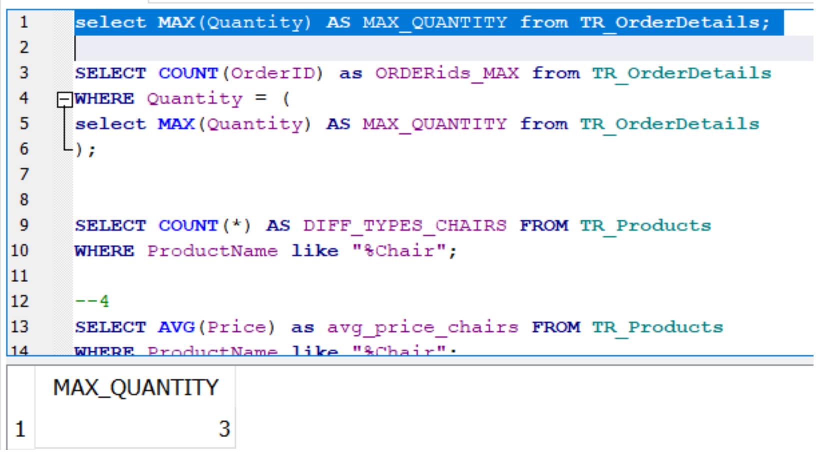
Part II – SQL (40 Marks)

Ashwath J

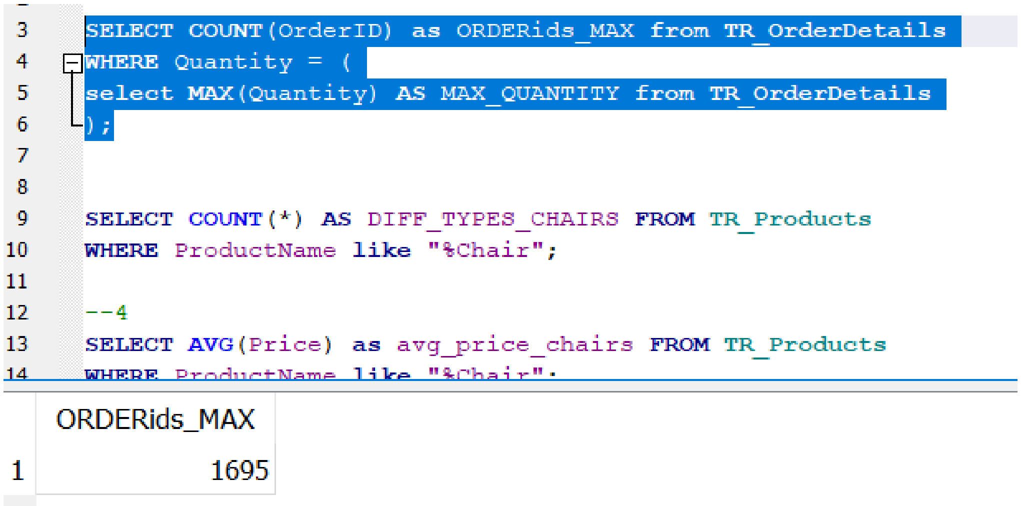
Problem Statement:
This is a Store wise Sales and inventory datasets for a retail store in the United States of America. The object of this project is to use the data and analyze the sales at various layers (such as product, store, city, state etc.) We shall also be answering a few key questions that shall help in pricing and product placement decisions.

Q1) What is the maximum quantity of any order ID in the data? Also, determine the number of orders placed which have this maximum quantity.(2 marks)



```
select MAX(Quantity) AS MAX_QUANTITY from TR_OrderDetails;

SELECT COUNT(OrderID) as ORDERids_MAX from TR_OrderDetails
WHERE Quantity = (
select MAX(Quantity) AS MAX_QUANTITY from TR_OrderDetails
);


SELECT COUNT(*) AS DIFF_TYPES_CHAIRS FROM TR_Products
WHERE ProductName like "%Chair";

--4
SELECT AVG(Price) as avg_price_chairs FROM TR_Products
WHERE ProductName like "%Chair";
```

```
   MAX_QUANTITY
1             3
```

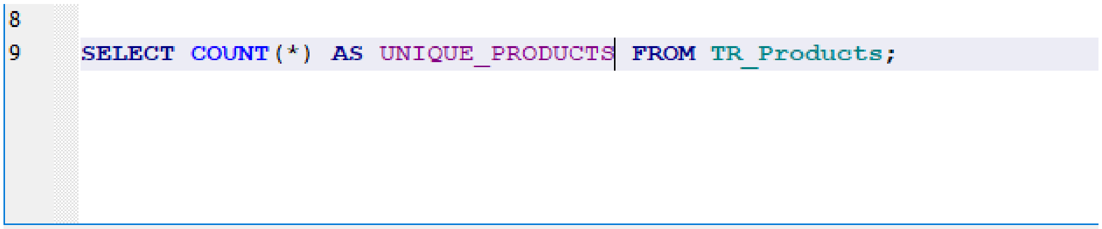


```
SELECT COUNT(OrderID) as ORDERids_MAX from TR_OrderDetails
WHERE Quantity = (
select MAX(Quantity) AS MAX_QUANTITY from TR_OrderDetails
);


SELECT COUNT(*) AS DIFF_TYPES_CHAIRS FROM TR_Products
WHERE ProductName like "%Chair";

--4
SELECT AVG(Price) as avg_price_chairs FROM TR_Products
WHERE ProductName like "%Chair";
```

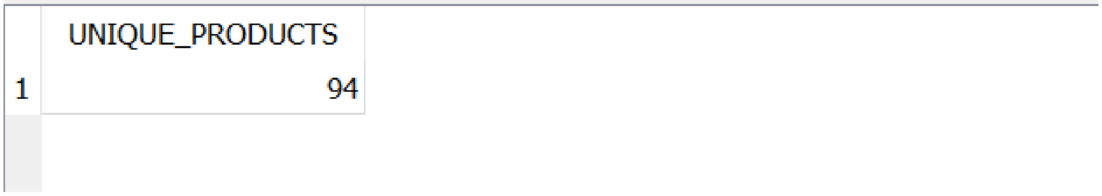
```
   ORDERids_MAX
1          1695
```

Part II – SQL (40 Marks)

Q2) Find the number of unique products that are sold. (2 marks)

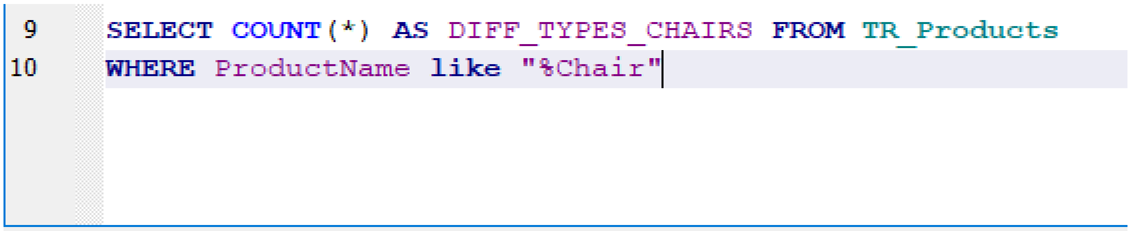


```
SELECT COUNT(*) AS UNIQUE_PRODUCTS FROM TR_Products;
```

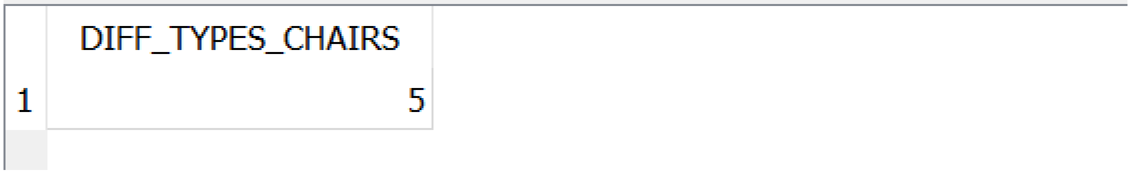


```
   UNIQUE_PRODUCTS
1               94
```

Q3) List the different types of “Chair” that are sold by using product table

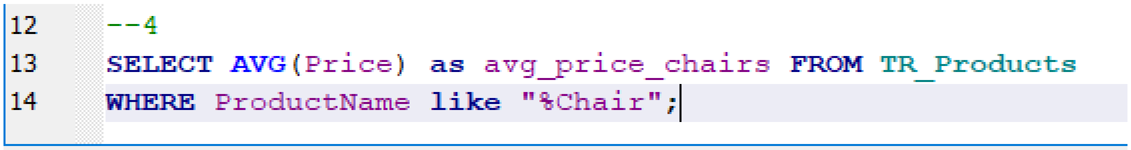


```
SELECT COUNT(*) AS DIFF_TYPES_CHAIRS FROM TR_Products
WHERE ProductName like "%Chair"
```

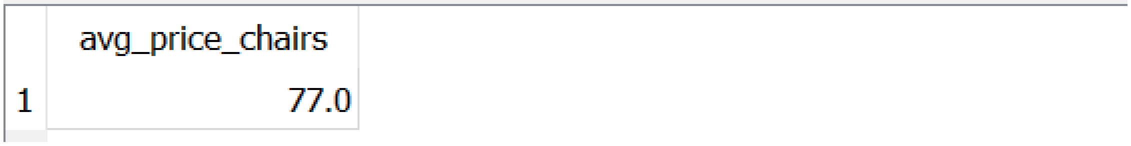


```
   DIFF_TYPES_CHAIRS
1                  5
```

Q4) What is the average price of each of these chair listed in the output of previous question?



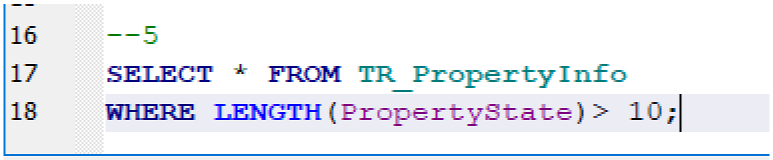
```
--4
SELECT AVG(Price) as avg_price_chairs FROM TR_Products
WHERE ProductName like "%Chair";
```



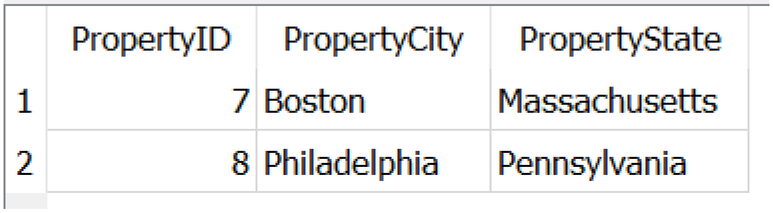
```
   avg_price_chairs
1              77.0
```

Part II – SQL (40 Marks)

Q5) Find the details of the Properties where the state names are more than 10 characters in length?

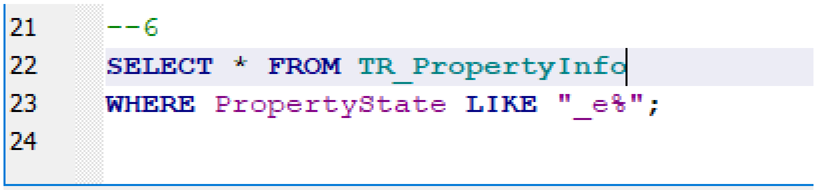


```
--5
SELECT * FROM TR_PropertyInfo
WHERE LENGTH(PropertyState)> 10;
```

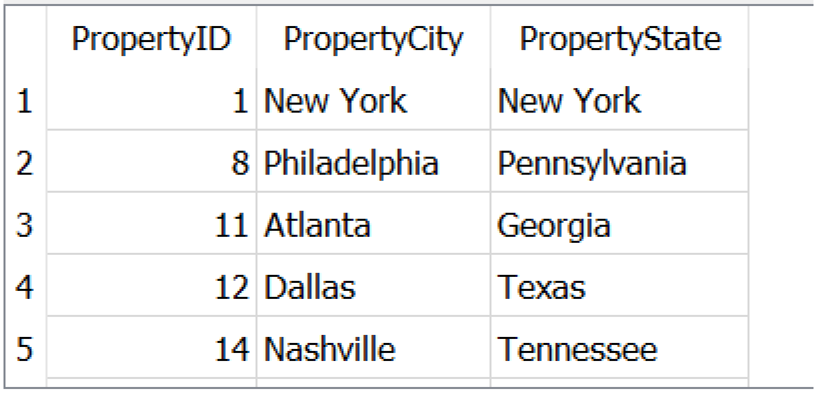


	PropertyID	PropertyCity	PropertyState
1	7	Boston	Massachusetts
2	8	Philadelphia	Pennsylvania

Q6) Find the details of the Properties where the second character of the city name is “e”.

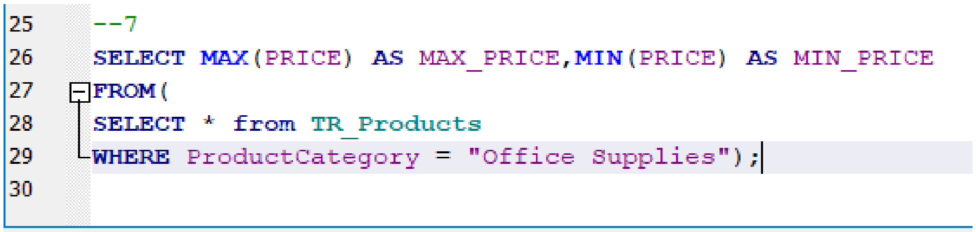


```
--6
SELECT * FROM TR_PropertyInfo
WHERE PropertyState LIKE "_e%";
```

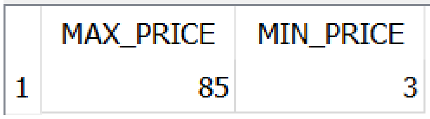


	PropertyID	PropertyCity	PropertyState
1	1	New York	New York
2	8	Philadelphia	Pennsylvania
3	11	Atlanta	Georgia
4	12	Dallas	Texas
5	14	Nashville	Tennessee

Q7) Find the minimum and maximum prices for products in the “Office Supplies” category



```
--7
SELECT MAX(PRICE) AS MAX_PRICE,MIN(PRICE) AS MIN_PRICE
FROM(
SELECT * from TR_Products
WHERE ProductCategory = "Office Supplies");
```



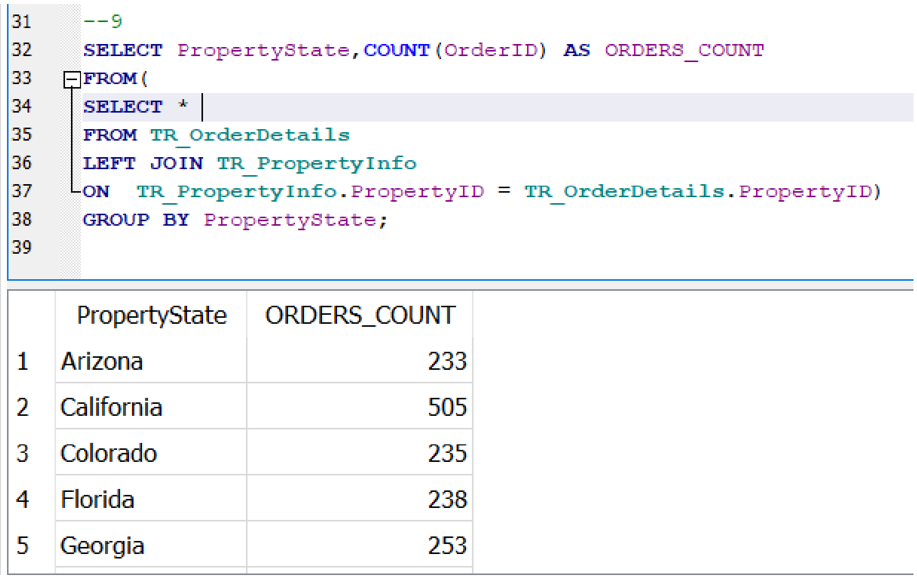
	MAX_PRICE	MIN_PRICE
1	85	3

Part II – SQL (40 Marks)

Q8) What is the purpose of using GROUP BY in SQL?

The **GROUP BY** Statement in SQL is used to arrange identical data into groups with the help of some functions. i.e if a particular column has the same values in different rows then it will arrange these rows in a group.

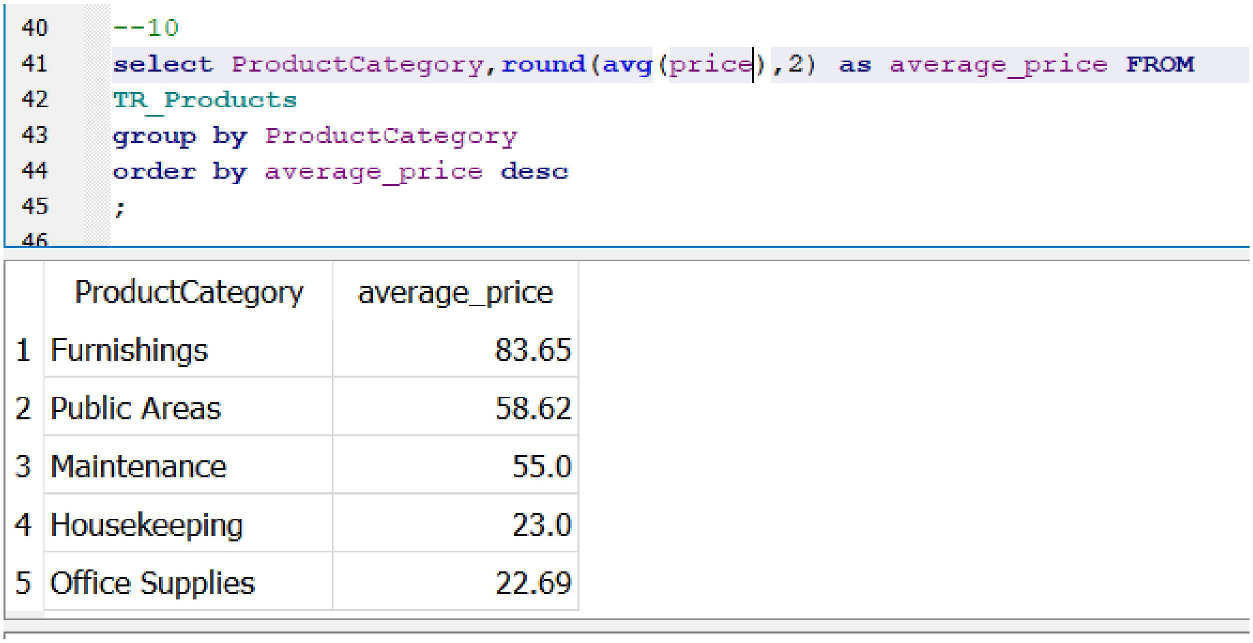
Q9) List the different states in which sales are made and count how many orders are there in each of the states?



```
--9
SELECT PropertyState,COUNT(OrderID) AS ORDERS_COUNT
FROM(
SELECT *
FROM TR_OrderDetails
LEFT JOIN TR_PropertyInfo
ON  TR_PropertyInfo.PropertyID = TR_OrderDetails.PropertyID)
GROUP BY PropertyState;
```

	PropertyState	ORDERS_COUNT
1	Arizona	233
2	California	505
3	Colorado	235
4	Florida	238
5	Georgia	253

Q10) Find the average price of items sold in each Product Category and sort it in a decreasing order.

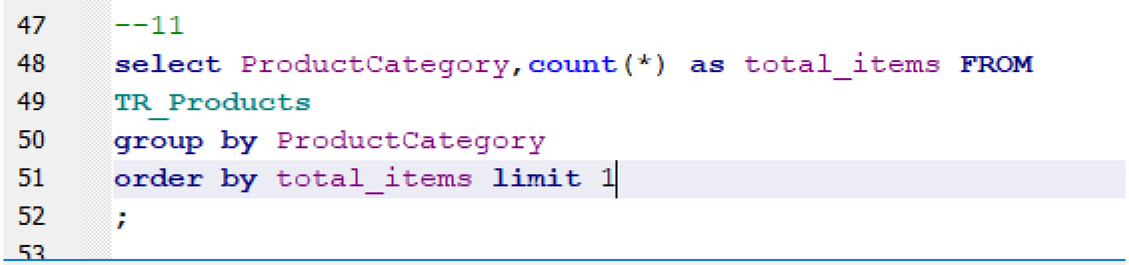


```
--10
select ProductCategory,round(avg(price),2) as average_price FROM
TR_Products
group by ProductCategory
order by average_price desc
;
```

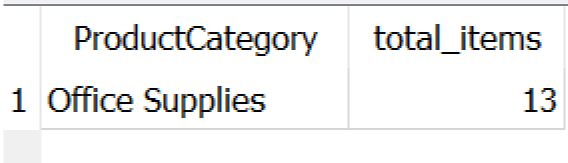
	ProductCategory	average_price
1	Furnishings	83.65
2	Public Areas	58.62
3	Maintenance	55.0
4	Housekeeping	23.0
5	Office Supplies	22.69

Part II – SQL (40 Marks)

Q11) Find the Product Category that sells the least number of products? Something for the management to focus on.



```
--11
select ProductCategory,count(*) as total_items FROM
TR_Products
group by ProductCategory
order by total_items limit 1
;
```

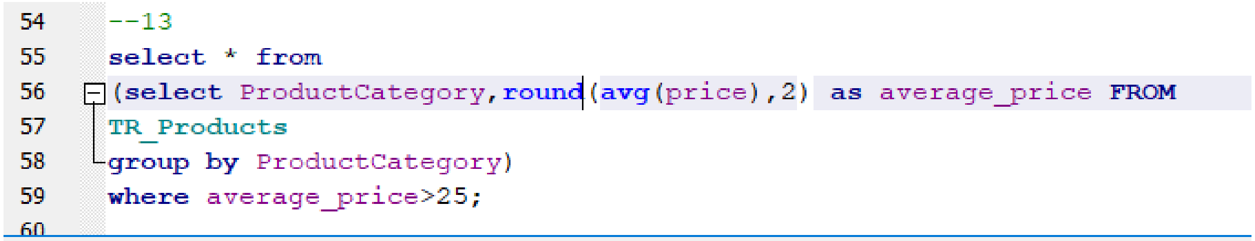


	ProductCategory	total_items
1	Office Supplies	13

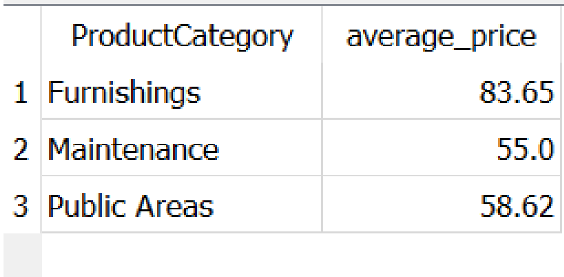
Q12) What is the difference between a WHERE v/s HAVING clause in SQL? (Hint: This is a theoretical question and needs to be explained with an clear example other than the application given in this project)

A **HAVING** clause is like a **WHERE** clause, but applies only to groups as a whole (that is, to the rows in the result set representing groups), whereas the **WHERE** clause applies to individual rows. A query can contain both a **WHERE** clause and a **HAVING** clause.

Q13) Select the Product categories where the average price is more than 25 (2 marks)



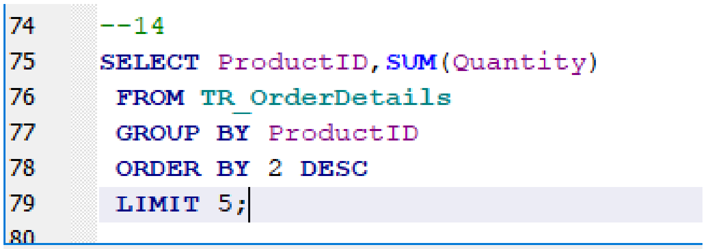
```
--13
select * from
(select ProductCategory,round(avg(price),2) as average_price FROM
TR_Products
group by ProductCategory)
where average_price>25;
```



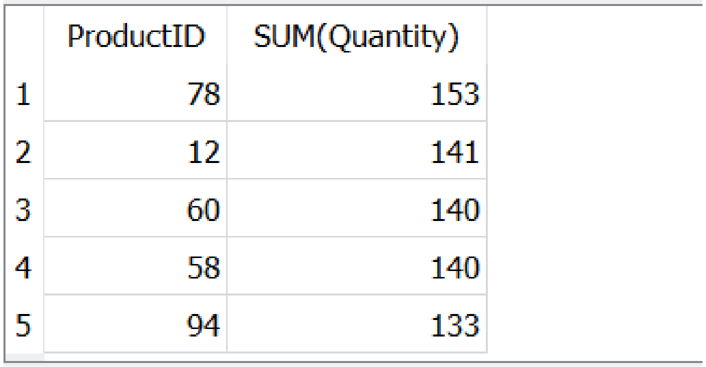
	ProductCategory	average_price
1	Furnishings	83.65
2	Maintenance	55.0
3	Public Areas	58.62

Part II – SQL (40 Marks)

Q14) Find the top 5 product IDs that sold the maximum quantities? (2 marks)

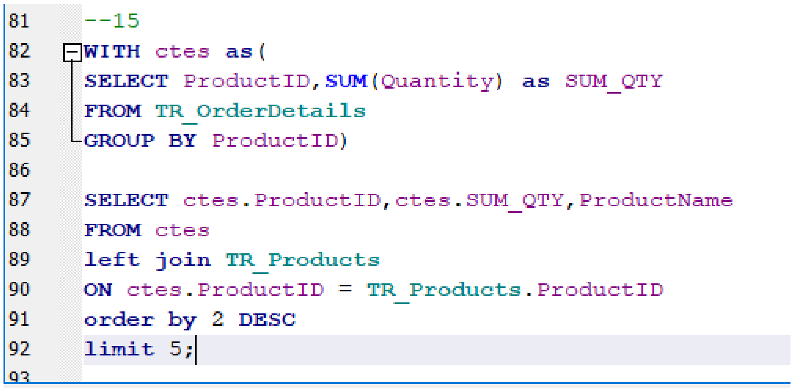


```
--14
SELECT ProductID,SUM(Quantity)
 FROM TR_OrderDetails
 GROUP BY ProductID
 ORDER BY 2 DESC
 LIMIT 5;
```

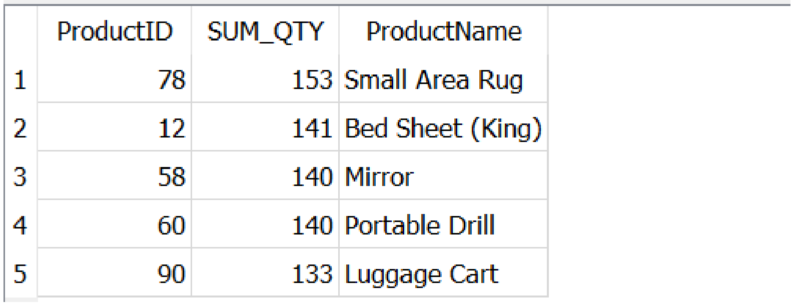


	ProductID	SUM(Quantity)
1	78	153
2	12	141
3	60	140
4	58	140
5	94	133

Q15) For the above question, print the product names instead of Product IDs. (2 marks)



```
--15
WITH ctes as(
SELECT ProductID,SUM(Quantity) as SUM_QTY
FROM TR_OrderDetails
GROUP BY ProductID)

SELECT ctes.ProductID,ctes.SUM_QTY,ProductName
FROM ctes
left join TR_Products
ON ctes.ProductID = TR_Products.ProductID
order by 2 DESC
limit 5;
```



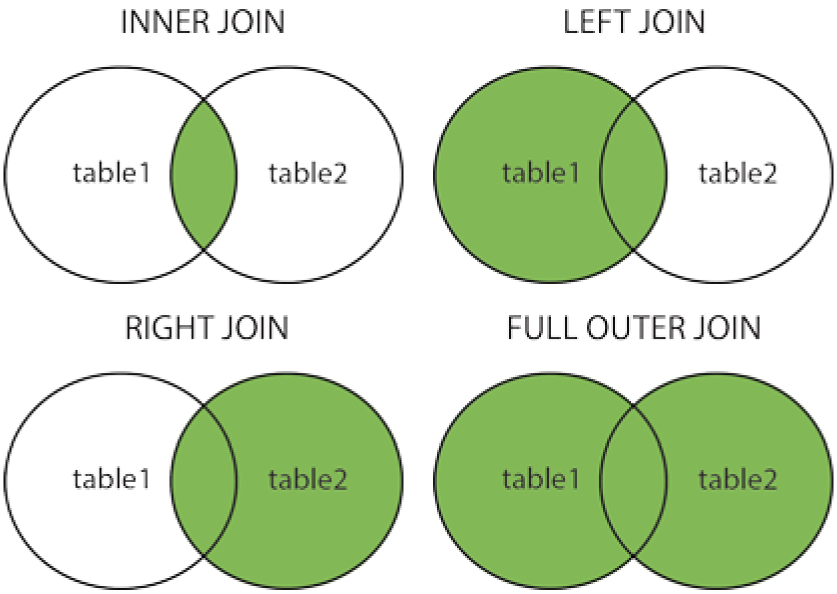
	ProductID	SUM_QTY	ProductName
1	78	153	Small Area Rug
2	12	141	Bed Sheet (King)
3	58	140	Mirror
4	60	140	Portable Drill
5	90	133	Luggage Cart

Part II – SQL (40 Marks)

Q16) Mention the different types of joins in SQL? Give simple examples of each. Also represent them using Venn diagrams (Hitnt: This is a theoretical question, the explanation needs to be in detail along with an example other than the one given in this project) (2 marks).

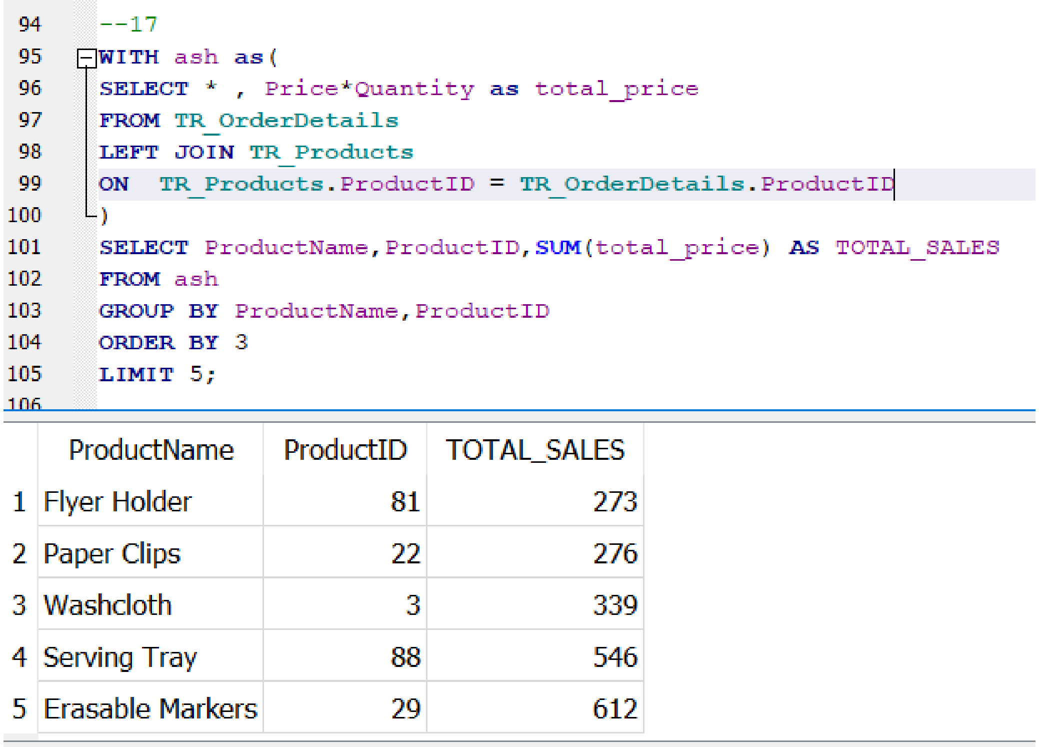
Here are the different types of the JOINs in SQL:

- `(INNER) JOIN`: Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN`: Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN`: Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN`: Returns all records when there is a match in either left or right table



Part II – SQL (40 Marks)

Q17) Determine the 5 products that give the overall minimum sales?

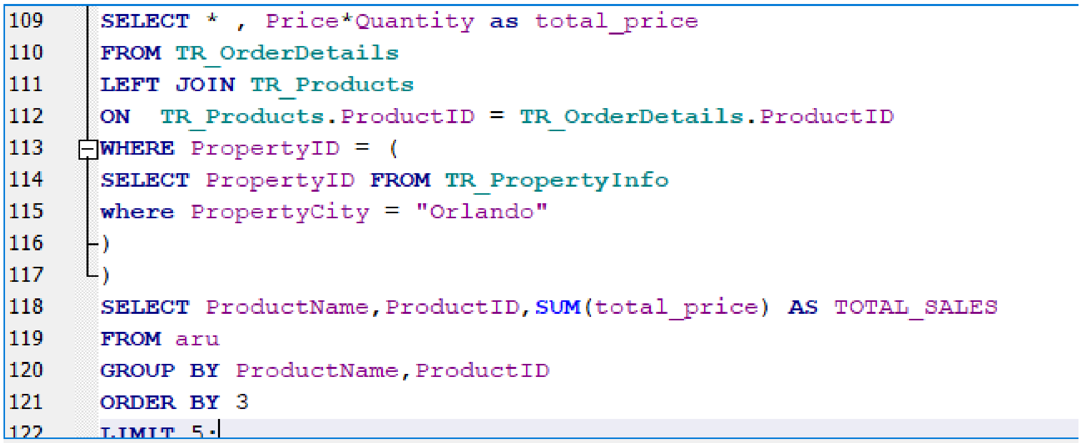


```sql
--17
WITH ash as(
SELECT * , Price*Quantity as total_price
FROM TR_OrderDetails
LEFT JOIN TR_Products
ON  TR_Products.ProductID = TR_OrderDetails.ProductID
)
SELECT ProductName,ProductID,SUM(total_price) AS TOTAL_SALES
FROM ash
GROUP BY ProductName,ProductID
ORDER BY 3
LIMIT 5;
```

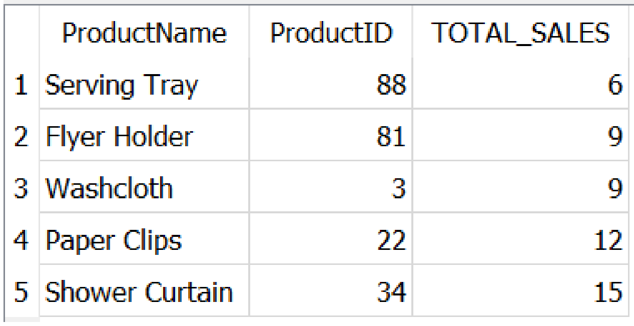
	ProductName	ProductID	TOTAL_SALES
1	Flyer Holder	81	273
2	Paper Clips	22	276
3	Washcloth	3	339
4	Serving Tray	88	546
5	Erasable Markers	29	612

18. Repeat the above query for the City of “Orlando”. (2 marks)

Part II – SQL (40 Marks)



```
SELECT * , Price*Quantity as total_price
FROM TR_OrderDetails
LEFT JOIN TR_Products
ON  TR_Products.ProductID = TR_OrderDetails.ProductID
WHERE PropertyID = (
SELECT PropertyID FROM TR_PropertyInfo
where PropertyCity = "Orlando"
)
)
SELECT ProductName,ProductID,SUM(total_price) AS TOTAL_SALES
FROM aru
GROUP BY ProductName,ProductID
ORDER BY 3
LIMIT 5;
```



	ProductName	ProductID	TOTAL_SALES
1	Serving Tray	88	6
2	Flyer Holder	81	9
3	Washcloth	3	9
4	Paper Clips	22	12
5	Shower Curtain	34	15

Q19) What is the difference between Drop, Truncate and Delete? Explain with examples.

DROP -

It is a Data Definition Language Command (DDL). It is used to drop the whole table. With the help of the "DROP" command we can drop (delete) the whole structure in one go i.e. it removes the named elements of the schema. By using this command the existence of the whole table is finished or lost.

TRUNCATE -

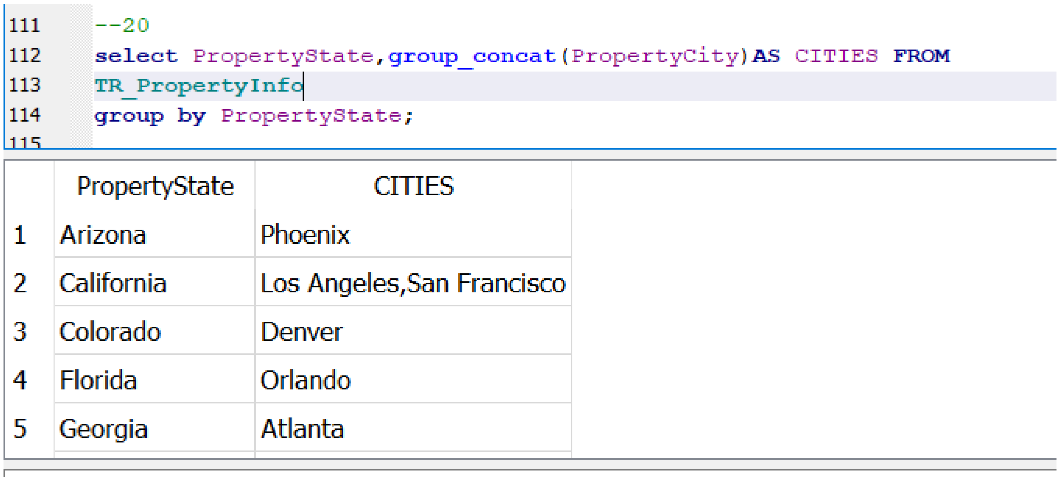
It is also a Data Definition Language Command (DDL). It is used to delete all the rows of a relation (table) in one go. With the help of the "TRUNCATE" command, we can't delete the single row as here WHERE clause is not used. By using this command the existence of all the rows of the table is lost. It is comparatively faster than the delete command as it deletes all the rows fastly.

DELETE -

Basically It is a data manipulation language command (DML).

It is used to delete one or more tuples of a table. With the help of the "DELETE" command, we can either delete all the rows in one go or can delete rows one by one. i.e., we can use it as per the requirement or the condition using the Where clause. It is comparatively slower than the TRUNCATE command. The TRUNCATE command does not remove the structure of the table.

Q20) Which are the cities that belong to the same states? (2 marks)



```
--20
select PropertyState,group_concat(PropertyCity)AS CITIES FROM
TR_PropertyInfo
group by PropertyState;
```

	PropertyState	CITIES
1	Arizona	Phoenix
2	California	Los Angeles,San Francisco
3	Colorado	Denver
4	Florida	Orlando
5	Georgia	Atlanta