s

**Set 8 - Assignments on Select Statements for Electronics Data**

**First Step 🡪** You need to copy-paste and execute in SSMS the entire script from the file whose name is **“Set 8\_Electronics\_Items\_Script\_File.sql”.** A new database will get created whose name will be **Electronics**. In that database some tables will get created with records after executing the script successfully.

**Write the Select Statements for the following:**

**Instructions:**

1. Hard coded values should not be used to get the answer set. The statement should have dynamic logic building for getting the answer set.
2. There can be multiple SQL techniques to get the same answer. If you can get the same answer by multiple ways, then you need to mention those all techniques as well.
3. Display the names of the sales persons who have sold 1 item in Aundh.

**select distinct SP.Salesman\_Name**

**from Transactions T join Sales\_Persons SP**

**on SP.Salesman\_Id = T.Salesman\_Id**

**join Area A on T.Area\_Id = A.Area\_Id**

**where A.Area\_Name = 'Aundh' and t.Qty\_Sold = 1**

1. Display the total of the billing value for selling of the product Onida TV done by Pushkar.

**select distinct IM.Item\_Cost\*sum(T.Qty\_Sold) over() as"Total Billing amount"**

**from Transactions T join Sales\_Persons SP**

**on t.Salesman\_Id = SP.Salesman\_Id**

**join Item\_Master IM on T.Item\_Id = IM.Item\_Id**

**where sp.Salesman\_Name = 'Pushkar'**

**and IM.Item\_Name = 'Onida TV'**

1. Display Salesman name, Area name, Quantity sold, total of quantity sold by each Salesman.

**select Distinct SP.Salesman\_Name, A.Area\_Name,sum(T.Qty\_Sold) over(partition by A.Ar;;;;;;; ea\_Id, SP.Salesman\_Name) as "Quantity", sum(T.Qty\_Sold) over(Partition by SP.Salesman\_Name) as "Total Quantity"**

**from Sales\_Persons SP join Transactions T**

**on SP.Salesman\_Id = T.Salesman\_Id**

**join Area A on A.Area\_Id = T.Area\_Id**

1. Display the Item names having the same reorder level.

**select IM.Item\_Name, SI.Reorder\_level**

**from Stock\_Items SI**

**join Item\_Master IM on SI.Item\_Id = IM.Item\_Id**

**where SI.Reorder\_level in (select Reorder\_level from Stock\_Items**

**GROUP BY Reorder\_level**

**HAVING count(Reorder\_level) >1)**

**order by SI.Reorder\_level**

1. Display the number of transactions done for the cheapest item.

**select count(t.Tr\_Id) as "Number of Transaction"**

**from Transactions T**

**Join Item\_Master IM on IM.Item\_Id = T.Item\_Id**

**where IM.Item\_Cost = (select min(Item\_Cost)**

**from Item\_Master)**

1. Display Item\_Id, Quantity\_In\_hand and running total of Quantity\_In\_hand of those records who have their running total above 200.

**with Running\_Total as(**

**select Item\_Id,Quantity\_In\_Hand, sum(Quantity\_In\_Hand) over(order by item\_ID) as "Running total"**

**from Stock\_Items)**

**select \* from Running\_Total**

**where [Running total] >200**