

Superstores Summary Tables

Preface

- This database discusses the SuperStore sales data. The structure has 5 tables, namely cust_dimen (containing details about customer and their respective locations), prod_dimen (contains product category and their subcategories), orders_dimen (with order no, date, and priority), shipping_dimen (with ship date, order and shipping mode), and market_fact (orderwise customerwise marketwise order quantity, sales value, discount profit and shipping cost details).

Primary Keys and Foreign keys in the given database.

cust_dimen Table - Cust_id as Primary Key, foreign key is not available here

prod_dimen table - Prod_ID as Primary Key, , foreign key is not available here

orders_dimen Table -order_ID as a Primary Key

shipping_dimen Table-Order_ID as Foreign key ACT as relationship between Orders_dimen table and Shipping_dimen, shipping_id as Primary Key

market_fact table- Order_ID ,ship_ID,Prod_ID as Foreign key, Primary key is not available here.

Queries and Solutions

1. WRITE A QUERY TO DISPLAY THE CUSTOMER_NAME AND CUSTOMER_SEGMENT USING ALIAS NAME "CUSTOMER NAME", "CUSTOMER_SEGMENT" FROM TABLE CUST_DIMEN.

SELECT * FROM CUST_DIMEN;

**SELECT CUSTOMER_NAME"CUSTOMER NAME",CUSTOMER_SEGMENT"CUSTOMER_SEGMENT"
FROM CUST_DIMEN;**

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```
1 USE superstores;
2 /**Write a query to display the Customer_Name and Customer_Segment using alias name "Customer Name", "Customer Segment" from table Cust_dimen.**/
3 SELECT * FROM Cust_dimen;
4 SELECT CONCAT(Customer_Name,'',Customer_Segment) AS NAME FROM Cust_dimen;
5 SELECT Customer_name"Customer Name",Customer_Segment"Customer Segment" FROM Cust_dimen;
6
```

The results grid displays the following data:

Customer Name	Customer Segment
MUHAMMAD MACINTYRE	SMALL BUSINESS
BARRY FRENCH	CONSUMER
CLAY ROZENDAL	CORPORATE
CARLOS SOLTERO	CONSUMER
CARL JACKSON	CORPORATE
MARIA FERRER	CORPORATE

The output pane shows the following messages:

#	Time	Action	Message	Duration / Fetch
8	23:02:03	SELECT Customer_name, Customer_Segment WHERE (Customer_name = 'Customer Name',...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t...	0.000 sec
9	23:07:38	SELECT CONCAT(Customer_Name,'',Customer_Segment) AS CONCAT(Customer Name,'',C...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t...	0.000 sec
10	23:07:59	SELECT CONCAT(Customer_Name,'',Customer_Segment) AS Name FROM Cust_dimen LIM...	1000 row(s) returned	0.031 sec / 0.000 sec
11	23:14:38	SELECT CONCAT(Customer_Name,'',Customer_Segment) AS NAME FROM Cust_dimen LI...	1000 row(s) returned	0.016 sec / 0.015 sec
12	23:16:35	SELECT (Customer_name"Customer Name",Customer_Segment"Customer Segment") FROM...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t...	0.000 sec
13	23:17:12	SELECT Customer_name"Customer Name",Customer_Segment"Customer Segment" FROM ...	1000 row(s) returned	0.000 sec / 0.015 sec

2)WRITE A QUERY TO FIND ALL THE DETAILS OF THE CUSTOMER FROM THE TABLE CUST_DIMEN ORDER BY DESC.

SELECT * FROM CUST_DIMEN ORDER BY CUSTOMER_NAME DESC;

The screenshot shows a SQL IDE interface. On the left is a 'SCHEMAS' pane with a tree view containing 'hr', 'jmh', 'newschema', 'northwind', 'sakila', 'superstores', 'sys', and 'world'. The 'superstores' schema is selected, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main area is a 'Query' editor with a toolbar and a text area containing the following SQL script:

```
1 USE superstores;
2 /**Write a query to display the Customer_Name and Customer_Segment using alias name "Customer Name", "Customer Segment" from table Cust_dimen.**/
3 SELECT * FROM Cust_dimen;
4 SELECT CONCAT(Customer_Name,' ',Customer_Segment) AS NAME FROM Cust_dimen;
5 SELECT Customer_name"Customer Name",Customer_Segment"Customer Segment" FROM Cust_dimen;
6 /**Write a query to find all the details of the customer from the table cust_dimen order by desc.**/
7 SELECT * FROM Cust_dimen ORDER BY Customer_name DESC;
8
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has columns: Customer_Name, Province, Region, Customer_Segment, and Cust_id. The data is as follows:

Customer_Name	Province	Region	Customer_Segment	Cust_id
YOSEPH CARROLL	ALBERTA	WEST	CONSUMER	Cust_1798
YANA SORENGEN	YUKON	YUKON	CORPORATE	Cust_1519
YANA SORENGEN	NEWFOUNDLAND	ATLANTIC	CORPORATE	Cust_637
YANA SORENGEN	QUEBEC	QUEBEC	CORPORATE	Cust_851
YANA SORENGEN	BRITISH COLUMBIA	WEST	CORPORATE	Cust_1577
XYLONA PRICE	ONTARIO	ONTARIO	CORPORATE	Cust_1006

At the bottom is an 'Output' pane showing a log of actions. The log includes the following entries:

#	Time	Action	Message	Duration / Fetch
12	23:16:35	SELECT (Customer_name"Customer Name"/Customer_Segment"Customer Segment") FROM...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to...	0.000 sec
13	23:17:12	SELECT Customer_name"Customer Name"/Customer_Segment"Customer Segment" FROM ...	1000 row(s) returned	0.000 sec / 0.015 sec
14	23:20:33	SELECT * FROM Cust_dimen LIMIT 0,1000	1000 row(s) returned	0.000 sec / 0.016 sec
15	23:22:29	SELECT * FROM Cust_dimen ORDER BY Customer_name DESC LIMIT 0,1000	1000 row(s) returned	0.172 sec / 0.000 sec
16	23:22:55	SELECT ORDER BY Customer_name DESC * FROM Cust_dimen	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to...	0.000 sec
17	23:23:03	SELECT * FROM Cust_dimen ORDER BY Customer_name DESC LIMIT 0,1000	1000 row(s) returned	0.000 sec / 0.000 sec

3. WRITE A QUERY TO GET THE ORDER ID, ORDER DATE FROM TABLE ORDERS_DIMEN WHERE 'ORDER PRIORITY' IS HIGH.

SELECT * FROM ORDERS_DIMEN ORDER_ID WHERE ORDER_PRIORITY LIKE '%High';

The screenshot displays a SQL IDE interface. The left sidebar shows a 'SCHEMAS' tree with a search filter and a list of databases including 'hr', 'jmhjh', 'newschema', 'northwind', 'sakila', and 'superstores'. The 'superstores' database is selected, showing its tables, views, stored procedures, and functions. The main query editor contains a multi-line SQL script. The 'Result Grid' pane shows the results of the query, displaying columns 'Order_ID', 'Order_Date', 'Order_Priority', and 'Ord_id'. The 'Action Output' pane at the bottom shows the execution log with timestamps, actions, messages, and durations.

Query 1

```
4 • SELECT CONCAT(Customer_Name, '', Customer_Segment) AS NAME FROM Cust_dimen;
5 • SELECT Customer_name"Customer Name", Customer_Segment"Customer Segment" FROM Cust_dimen;
6 • /**Write a query to find all the details of the customer from the table cust_dimen order by desc.**/
7 • SELECT * FROM Cust_dimen ORDER BY Customer_name DESC;
8 • /**Write a query to get the Order ID, Order date from table orders_dimen where 'Order Priority' is high*/
9 • SELECT * FROM orders_dimen;
10 • SELECT * FROM orders_dimen Order_ID WHERE Order_Priority LIKE '%High';
11
```

Result Grid

Order_ID	Order_Date	Order_Priority	Ord_id
293	01-10-2012	HIGH	Ord_2
483	10-07-2011	HIGH	Ord_3
613	17-06-2011	HIGH	Ord_5
643	24-03-2011	HIGH	Ord_6
1540	04-08-2012	HIGH	Ord_17
1702	06-05-2011	HIGH	Ord_18

Action Output

#	Time	Action	Message	Duration / Fetch
34	00:12:31	SELECT * FROM orders_dimen Order_ID WHERE (Order_Priority, 'High') LIMIT 0, 1000	Error Code: 1241. Operand should contain 1 column(s)	0.047 sec
35	00:13:59	SELECT * FROM orders_dimen Order_ID , Order_date WHERE (Order_Priority, 'High') LIMIT 0, 1000	Error Code: 1146. Table 'superstores.order_date' doesn't exist	0.047 sec
36	00:15:53	SELECT * FROM orders_dimen Order_ID , Order_date WHERE Order_Priority LIKE '%High' LIMIT 0, 1000	Error Code: 1146. Table 'superstores.order_date' doesn't exist	0.016 sec
37	00:16:26	SELECT * FROM orders_dimen Order_ID WHERE Order_Priority LIKE '%High' LIMIT 0, 1000	1000 row(s) returned	0.141 sec / 0.062 sec
38	00:16:58	SELECT * FROM orders_dimen Order_ID, Order_date WHERE Order_Priority LIKE '%High' LIMIT 0, 1000	Error Code: 1146. Table 'superstores.order_date' doesn't exist	0.015 sec
39	00:17:13	SELECT * FROM orders_dimen Order_ID WHERE Order_Priority LIKE '%High' LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.000 sec

4. FIND THE TOTAL AND THE AVERAGE SALES (DISPLAY TOTAL_SALES AND AVG_SALES)

SELECT ROUND(SUM(SALES),2) AS 'TOTAL SALES', ROUND(AVG(SALES),2) AS 'AVERAGE SALES'

FROM MARKET_FACT;

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
9 • SELECT * FROM orders_dimen;  
10 • SELECT * FROM orders_dimen Order_ID WHERE Order_Priority LIKE '%High';  
11 • /**Find the total and the average sales (display total_sales and avg_sales)**/  
12 • SELECT * FROM market_fact;  
13 • SELECT Round(SUM(sales),2) AS 'TOTAL SALES', Round(AVG(sales),2) AS 'Average Sales'  
14 • from market_fact;  
15  
16
```

The 'Result Grid' shows the following data:

TOTAL SALES	AVERAGE SALES
14647187.9	1757.1

The 'Output' tab is also visible, showing the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
42	06:03:02	SELECT * FROM market_fact LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.015 sec
43	06:07:10	SELECT SUM(sales) AS 'TOTAL SALES', AVG(sales) AS 'Average Sales' from market_fact LI...	1 row(s) returned	0.015 sec / 0.000 sec
44	06:07:17	select round(sum(sales),2) as "TOTAL SALES", Round(avg(sales),2) as "AVERAGE SALES"...	1 row(s) returned	0.015 sec / 0.000 sec
45	06:07:38	SELECT SUM(sales) AS 'TOTAL SALES', AVG(sales) AS 'Average Sales' from market_fact LI...	1 row(s) returned	0.015 sec / 0.000 sec
46	06:09:01	SELECT Round(SUM(sales),2) AS 'TOTAL SALES', Round(AVG(sales),2) AS 'Average Sales'...	1 row(s) returned	0.015 sec / 0.000 sec
47	06:09:10	select round(sum(sales),2) as "TOTAL SALES", Round(avg(sales),2) as "AVERAGE SALES"...	1 row(s) returned	0.015 sec / 0.000 sec

5. Write a query to get the maximum and minimum sales from maket_fact table.

```
SELECT MAX(Sales) Maximum, MIN(Sales) Minimum  
  
FROM market_fact;
```

The screenshot shows a SQL IDE interface with a query editor, a results grid, and an output pane.

Query Editor:

```
97 /*6. */  
98 SELECT DISTINCT(Region), COUNT(*) AS no_of_cust FROM cust_dimen  
99 GROUP BY Region  
100 ORDER BY no_of_cust DESC;  
101  
102 SELECT MAX(Sales) Maximum, MIN(Sales) Minimum  
103 FROM market_fact;
```

Results Grid:

	Maximum	Minimum
▶	89061.05	2.24

Output Pane:

#	Time	Action	Message	Duration / Fetch
60	20:25:49	select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub...	17 row(s) returned	0.078 sec / 0.000 sec
61	20:30:02	Select c.Region as "Region", c.Customer_Name as "Customer Name", p.Product_Sub_Cate...	42 row(s) returned	0.047 sec / 0.000 sec
62	20:32:49	select Region, count(Cust_id) as "No. of Customers" from cust_dimen group by Region order ...	1 row(s) returned	0.016 sec / 0.000 sec
63	20:36:38	SELECT Prod_id "product id", Product_Sub_Category "product sub category" FROM prod_d...	8 row(s) returned	0.000 sec / 0.000 sec
64	20:37:55	SELECT DISTINCT(Region), COUNT(*) AS no_of_cust FROM cust_dimen GROUP BY Regi...	8 row(s) returned	0.016 sec / 0.000 sec
65	20:39:36	SELECT MAX(Sales) Maximum, MIN(Sales) Minimum FROM market_fact LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec

Query Completed

6. Display the number of customers in each region in decreasing order of no_of_customers. The result should contain columns Region, no_of_customers

```
SELECT DISTINCT(Region), COUNT(*) AS no_of_cust FROM cust_dimen
```

```
GROUP BY Region
```

```
ORDER BY no_of_cust DESC;
```

The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'SCHEMAS' tree with 'superstores' expanded. The central pane shows a query window with the following SQL code:

```
94 SELECT Prod_id "product id", Product_Sub_Category "product sub category"
95 FROM prod_dimen
96 WHERE Product_Category= "TECHNOLOGY" OR Product_Category= "FURNITURE";
97 /*6. */
98 SELECT DISTINCT(Region), COUNT(*) AS no_of_cust FROM cust_dimen
99 GROUP BY Region
100 ORDER BY no_of_cust DESC;
```

The 'Result Grid' shows the following data:

Region	no_of_cust
WEST	382
ATLANTIC	344
ONTARIO	337
PRARIE	313
QUEBEC	210
YUKON	130

The right pane shows the 'SQL Additions' tab with the 'SELECT' syntax help. The bottom pane shows the 'Output' tab with the 'Action Output' view, displaying the execution of the query and the number of rows returned.

7. Find the region having maximum customers (display the region name and max(no_of_customers))

select Region, count(Cust_id) as "No. of Customers"

from cust_dimen

group by Region

order by count(Cust_id) DESC

LIMIT 1;

The screenshot shows a SQL IDE interface with a query editor, a result grid, and an output pane. The query editor contains the following SQL code:

```
87 • Select *from market_fact;
88
89 • select Region, count(Cust_id) as "No. of Customers"
90   from cust_dimen
91  group by Region
92  order by count(Cust_id) DESC
93  LIMIT 1;
```

The result grid shows the following data:

Region	No. of Customers
WEST	382

The output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
57	20:22:54	Select 'from market_fact LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec
58	20:23:53	select Product_Category as "Product Category", Product_Sub_Category as "Product Sub Cat...	Error Code: 1054. Unknown column 'm.Prod_id' in 'on clause'	0.015 sec
59	20:24:07	select Product_Category as "Product Category", Product_Sub_Category as "Product Sub Cat...	Error Code: 1052. Column 'Prod_id' in on clause is ambiguous	0.047 sec
60	20:25:49	select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub...	17 row(s) returned	0.078 sec / 0.000 sec
61	20:30:02	Select c.Region as "Region", c.Customer_Name as "Customer Name", p.Product_Sub_Cate...	42 row(s) returned	0.047 sec / 0.000 sec
62	20:32:49	select Region, count(Cust_id) as "No. of Customers" from cust_dimen group by Region order ...	1 row(s) returned	0.016 sec / 0.000 sec

8. Find all the customers from Atlantic region who have ever purchased 'TABLES' and the number of tables purchased (display the customer name, no_of_tables purchased)

Select c.Region as "Region", c.Customer_Name as "Customer Name",

p.Product_Sub_Category as "Product Sub Category",

sum(m.Order_Quantity) as "Order Quantity"

from market_fact m

join cust_dimen c on m.Cust_id = c.Cust_id

join prod_dimen p on m.Prod_id = p.Prod_id

where c.Region = "ATLANTIC" and p.Product_Sub_Category = "TABLES"

group by c.Customer_Name

Order by sum(m.Order_Quantity) DESC;

The screenshot shows a SQL query editor with the following query:

```
79 sum(m.Order_Quantity) as "Order Quantity"
80 from market_fact m
81 join cust_dimen c on m.Cust_id = c.Cust_id
82 join prod_dimen p on m.Prod_id = p.Prod_id
83 where c.Region = "ATLANTIC" and p.Product_Sub_Category = "TABLES"
84 group by c.Customer_Name
85 Order by sum(m.Order_Quantity) DESC;
```

The query results are displayed in a table with the following columns: Region, Customer Name, Product Sub Category, and Order Quantity. The results are sorted by Order Quantity in descending order.

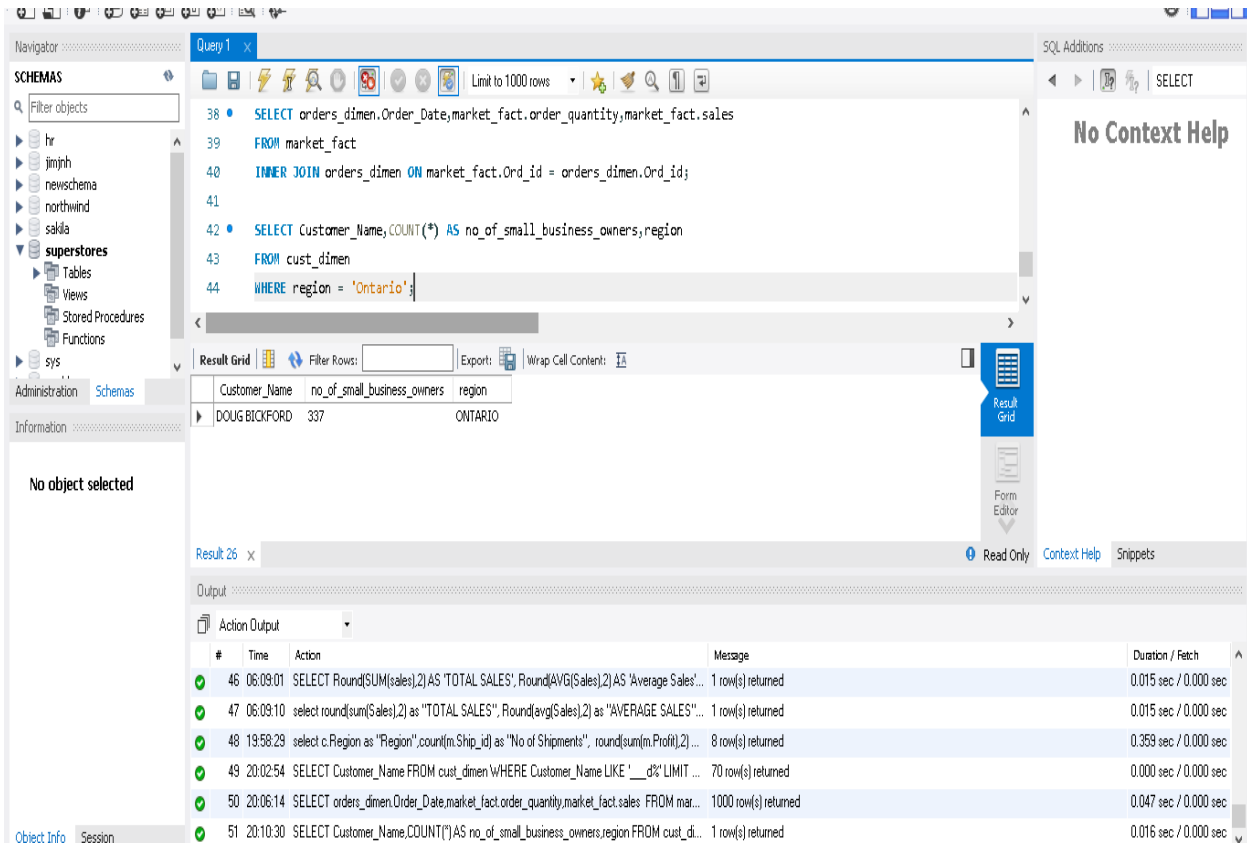
Region	Customer Name	Product Sub Category	Order Quantity
ATLANTIC	MUHAMMED YEDWAB	TABLES	59
ATLANTIC	BOBBY TRAFTON	TABLES	50
ATLANTIC	ALEXSANDRA GANNAWAY	TABLES	49
ATLANTIC	ELENI MCCRARY	TABLES	49
ATLANTIC	RICK DUSTON	TABLES	48

The bottom of the screenshot shows the Action Output window with the following messages:

#	Time	Action	Message	Duration / Fetch
56	20:22:12	select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub...	17 row(s) returned	0.235 sec / 0.000 sec
57	20:22:54	Select 'from market_fact LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec
58	20:23:53	select Product_Category as "Product Category", Product_Sub_Category as "Product Sub Cat...	Error Code: 1054. Unknown column 'm.Prod_id' in 'on clause'	0.015 sec
59	20:24:07	select Product_Category as "Product Category", Product_Sub_Category as "Product Sub Cat...	Error Code: 1052. Column 'Prod_id' in on clause is ambiguous	0.047 sec
60	20:25:49	select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub...	17 row(s) returned	0.078 sec / 0.000 sec
61	20:30:02	Select c.Region as "Region", c.Customer_Name as "Customer Name", p.Product_Sub_Cate...	42 row(s) returned	0.047 sec / 0.000 sec

9. Find all the customers from Ontario province who own Small Business. (display the customer name, no of small business owners)

select CUsTomer_name, Region, Customer_segment as no_of_small_Owner from cust_dimen
where Region = 'Ontario' and Customer_Segment = "small Business";



The screenshot shows a SQL IDE interface with a query editor, a schema navigator, and a results pane. The query executed is:

```

38 SELECT orders_dimen.Order_Date,market_fact.order_quantity,market_fact.sales
39 FROM market_fact
40 INNER JOIN orders_dimen ON market_fact.Ord_id = orders_dimen.Ord_id;
41
42 SELECT Customer_Name,COUNT(*) AS no_of_small_business_owners,region
43 FROM cust_dimen
44 WHERE region = 'Ontario';

```

The results pane displays a table with the following data:

Customer_Name	no_of_small_business_owners	region
DOUG BICKFORD	337	ONTARIO

The bottom pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
46	06:09:01	SELECT Round(SUM(sales),2) AS 'TOTAL SALES', Round(AVG(Sales),2) AS 'Average Sales'...	1 row(s) returned	0.015 sec / 0.000 sec
47	06:09:10	select round(sum(Sales),2) as "TOTAL SALES", Round(avg(Sales),2) as "AVERAGE SALES"...	1 row(s) returned	0.015 sec / 0.000 sec
48	19:58:29	select c.Region as "Region",count(m.Ship_id) as "No of Shipments", round(sum(m.Profit),2)...	8 row(s) returned	0.359 sec / 0.000 sec
49	20:02:54	SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '___d%' LIMIT ...	70 row(s) returned	0.000 sec / 0.000 sec
50	20:06:14	SELECT orders_dimen.Order_Date,market_fact.order_quantity,market_fact.sales FROM mar...	1000 row(s) returned	0.047 sec / 0.000 sec
51	20:10:30	SELECT Customer_Name,COUNT(*) AS no_of_small_business_owners,region FROM cust_di...	1 row(s) returned	0.016 sec / 0.000 sec

10. Find the number and id of products sold in decreasing order of products sold (display product id, no_of_products sold)

select Prod_id, sum(Order_Quantity) as "no_of_products sold"

from market_fact

group by Prod_id

order by sum(Order_Quantity) DESC;

The screenshot shows a SQL IDE interface with a query editor, a results grid, and an output window.

Query Editor: The query is as follows:

```
54 • SELECT Sales FROM market_fact ORDER BY Sales DESC LIMIT 2,1;
55
56
57 • select Prod_id, sum(Order_Quantity) as "no_of_products sold"
58 from market_fact
59 group by Prod_id
60 order by sum(Order_Quantity) DESC;
```

Results Grid: The results grid shows the following data:

Prod_id	no_of_products sold
Prod_6	30871
Prod_3	22992
Prod_4	22969
Prod_5	20648
Prod_8	19222

Output Window: The output window shows the execution of the query. The last row indicates that the query returned 17 rows.

#	Time	Action	Message	Duration / Fetch
50	20:06:14	SELECT orders_dimen.Order_Date,market_fact.order_quantity,market_fact.sales FROM mar...	1000 row(s) returned	0.047 sec / 0.000 sec
51	20:10:30	SELECT Customer_Name,COUNT(*) AS no_of_small_business_owners.region FROM cust_di...	1 row(s) returned	0.016 sec / 0.000 sec
52	20:12:32	select cust_dimen.customer_name,cust_dimen.region,cust_dimen.cust_id,market_fact.sales f...	1000 row(s) returned	0.234 sec / 0.031 sec
53	20:15:57	SELECT Sales FROM market_fact ORDER BY Sales DESC LIMIT 2,1	1 row(s) returned	0.031 sec / 0.000 sec
54	20:16:11	SELECT Sales FROM market_fact ORDER BY Sales DESC LIMIT 2,1	1 row(s) returned	0.016 sec / 0.000 sec
55	20:20:05	select Prod_id, sum(Order_Quantity) as "no_of_products sold" from market_fact group by Pro...	17 row(s) returned	0.079 sec / 0.000 sec

11. Display product Id and product sub category whose product category belongs to Furniture and Technology. The result should contain columns product id, product sub category.

SELECT Prod_id "product id", Product_Sub_Category "product sub category"

FROM prod_dimen

WHERE Product_Category= "TECHNOLOGY" OR Product_Category= "FURNITURE";

The screenshot shows a SQL IDE interface with a query editor, a results grid, and an output pane. The query editor contains the following SQL code:

```
90 from cust_dimen
91 group by Region
92 order by count(cust_id) DESC
93 LIMIT 1;
94 • SELECT Prod_id "product id", Product_Sub_Category "product sub category"
95 FROM prod_dimen
96 WHERE Product_Category= "TECHNOLOGY" OR Product_Category= "FURNITURE";
```

The results grid shows the following data:

product id	product sub category
Prod_4	TELEPHONES AND COMMUNICATION
Prod_5	OFFICE FURNISHINGS
Prod_8	COMPUTER PERIPHERALS
Prod_10	BOOKCASES
Prod_11	TABLES

The output pane shows the following messages:

#	Time	Action	Message	Duration / Fetch
58	20:23:53	select Product_Category as "Product Category", Product_Sub_Category as "Product Sub Cat...	Error Code: 1054. Unknown column 'm.Prod_id' in 'on clause'	0.015 sec
59	20:24:07	select Product_Category as "Product Category", Product_Sub_Category as "Product Sub Cat...	Error Code: 1052. Column 'Prod_id' in on clause is ambiguous	0.047 sec
60	20:25:49	select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub...	17 row(s) returned	0.078 sec / 0.000 sec
61	20:30:02	Select c.Region as "Region", c.Customer_Name as "Customer Name", p.Product_Sub_Cate...	42 row(s) returned	0.047 sec / 0.000 sec
62	20:32:49	select Region, count(Cust_id) as "No. of Customers" from cust_dimen group by Region order ...	1 row(s) returned	0.016 sec / 0.000 sec
63	20:36:38	SELECT Prod_id "product id", Product_Sub_Category "product sub category" FROM prod_di...	8 row(s) returned	0.000 sec / 0.000 sec

12. Display the product categories in descending order of profits (display the product category wise profits i.e. product_category, profits)?

SELECT MAX(Sales) Maximum, MIN(Sales) Minimum

FROM market_fact;

SELECT P.product_category, SUM(M.profit) "PROFIT"

FROM market_fact M

INNER JOIN prod_dimen P ON M.prod_id = P.prod_id

GROUP BY P.product_category

ORDER BY PROFIT DESC

Or

SELECT prod_dimen.product_category, Sum(market_fact.profit) from market_fact

JOIN prod_dimen ON market_fact.prod_id=prod_dimen.prod_id

group by prod_dimen.Product_category

order by profit Desc;

The screenshot shows a SQL IDE interface. The query editor contains the following SQL query:

```
SELECT MAX(Sales) Maximum, MIN(Sales) Minimum
FROM market_fact;

SELECT P.product_category, SUM(M.profit) "PROFIT"
FROM market_fact M
INNER JOIN prod_dimen P ON M.prod_id = P.prod_id
GROUP BY P.product_category
ORDER BY PROFIT DESC;
```

The results grid displays the following data:

product_category	PROFIT
TECHNOLOGY	806313.52000000014
OFFICE SUPPLIES	524956.76999999997
FURNITURE	75968.730000000004

The output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
61	20:30:02	Select c.Region as "Region", c.Customer_Name as "Customer Name", p.Product_Sub_Cate...	42 row(s) returned	0.047 sec / 0.000 sec
62	20:32:49	select Region, count(Cust_id) as "No. of Customers" from cust_dimen group by Region order ...	1 row(s) returned	0.016 sec / 0.000 sec
63	20:36:38	SELECT Prod_id "product id", Product_Sub_Category "product sub category" FROM prod_d...	8 row(s) returned	0.000 sec / 0.000 sec
64	20:37:55	SELECT DISTINCT(Region), COUNT(*) AS no_of_cust FROM cust_dimen GROUP BY Regi...	8 row(s) returned	0.016 sec / 0.000 sec
65	20:39:36	SELECT MAX(Sales) Maximum, MIN(Sales) Minimum FROM market_fact LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec
66	20:41:20	SELECT P.product_category, SUM(M.profit) "PROFIT" FROM market_fact M INNER JOIN pr...	3 row(s) returned	0.218 sec / 0.000 sec

13. Display the product category, product sub-category and the profit within each subcategory in three columns.

```
select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub  
Category",
```

```
round(sum(m.Profit), 2) as "Total Profits"
```

```
from market_fact m
```

```
join prod_dimen p on m.Prod_id = p.Prod_id
```

```
group by p.Product_Sub_Category
```

```
Order by p.Product_Category;
```

Or

```
SELECT prod_dimen.Product_category,prod_dimen.Product_sub_category, market_fact.profit  
from market_fact
```

```
JOIN prod_dimen ON market_fact.prod_id=prod_dimen.Prod_id;
```

The screenshot displays the SQL Developer interface. The 'Query' window contains the following SQL code:

```
68  
69 • select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub Category",  
70 round(sum(m.Profit), 2) as "Total Profits"  
71 from market_fact m  
72 join prod_dimen p on m.Prod_id = p.Prod_id  
73 group by p.Product_Sub_Category  
74 Order by p.Product_Category;
```

The 'Result Grid' shows the following data:

Product Category	Product Sub Category	Total Profits
FURNITURE	TABLES	-119468.18
FURNITURE	BOOKCASES	-33729.09
FURNITURE	OFFICE FURNISHINGS	100427.93
FURNITURE	CHAIRS & CHAIRMATS	122738.07
OFFICE SUPPLIES	SCISSORS, RULERS AND TRIMMERS	-7799.25

The 'Output' window shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
55	20:20:05	select Prod_id, sum(Order_Quantity) as "no_of_products sold" from market_fact group by Pro...	17 row(s) returned	0.079 sec / 0.000 sec
56	20:22:12	select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub...	17 row(s) returned	0.235 sec / 0.000 sec
57	20:22:54	Select "from market_fact LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec
58	20:23:53	select Product_Category as "Product Category", Product_Sub_Category as "Product Sub Cat...	Error Code: 1054. Unknown column 'm.Prod_id' in 'on clause'	0.015 sec
59	20:24:07	select Product_Category as "Product Category", Product_Sub_Category as "Product Sub Cat...	Error Code: 1052. Column 'Prod_id' in on clause is ambiguous	0.047 sec
60	20:25:49	select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub...	17 row(s) returned	0.078 sec / 0.000 sec

14. Display the order date, order quantity and the sales for the order.

SELECT orders_dimen.Order_Date,market_fact.order_quantity,market_fact.sales

FROM market_fact

INNER JOIN orders_dimen **ON** market_fact.Ord_id = orders_dimen.Ord_id;

The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'SCHEMAS' tree with 'superstores' expanded. The central pane shows a query window with the following SQL code:

```
36
37
38 • SELECT orders_dimen.Order_Date,market_fact.order_quantity,market_fact.sales
39 FROM market_fact
40 INNER JOIN orders_dimen ON market_fact.Ord_id = orders_dimen.Ord_id;
41
42
```

Below the query window, the 'Result Grid' shows the following data:

Order_Date	order_quantity	sales
28-05-2011	5	14.76
12-05-2009	12	41.97
12-05-2009	18	57.17
12-05-2009	11	81.25
12-05-2009	44	3202.25
12-05-2009	10	35.64

The bottom pane shows the 'Output' window with the following table:

#	Time	Action	Message	Duration / Fetch
45	06:07:38	SELECT SUM(sales) AS 'TOTAL SALES', AVG(Sales) AS 'Average Sales' from market_fact LI...	1 row(s) returned	0.016 sec / 0.000 sec
46	06:09:01	SELECT Round(SUM(sales),2) AS 'TOTAL SALES', Round(AVG(Sales),2) AS 'Average Sales'...	1 row(s) returned	0.015 sec / 0.000 sec
47	06:09:10	select round(sum(Sales),2) as "TOTAL SALES", Round(avg(Sales),2) as "AVERAGE SALES"...	1 row(s) returned	0.015 sec / 0.000 sec
48	19:58:29	select c.Region as "Region",count(m.Ship_id) as "No of Shipments", round(sum(m.Profit),2) ...	8 row(s) returned	0.359 sec / 0.000 sec
49	20:02:54	SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '___d%' LIMIT ...	70 row(s) returned	0.000 sec / 0.000 sec
50	20:06:14	SELECT orders_dimen.Order_Date,market_fact.order_quantity,market_fact.sales FROM mar...	1000 row(s) returned	0.047 sec / 0.000 sec

The status bar at the bottom indicates 'Query Completed'.

15. Display the names of the customers whose name contains the i) Second letter as 'R' ii) Fourth letter as 'D'

SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '_r%';

SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '__d%';

The screenshot shows a SQL IDE interface. On the left is a 'CHEMAS' schema browser with a tree view containing folders for 'hr', 'jimjh', 'newschema', 'northwind', 'sakila', 'superstores', 'Tables', 'Views', 'Stored Procedures', 'Functions', 'sys', 'administration', and 'Schemas'. The 'superstores' folder is expanded, showing 'Tables', 'Views', and 'Stored Procedures'. The main query editor displays a multi-line SQL query. The first part of the query is a subquery:
25 from market_fact m
26 join prod_dimen p on m.Prod_id = p.Prod_id
27 group by Product_Sub_Category
28 order by sum(m.Profit)
29 LIMIT 1)
30 group by c.Region
31 order by sum(m.Profit);
32
33
34 • SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '_r%';
35 • SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '__d%';
36
Below the query editor is a 'Result Grid' showing the results of the queries. The first query (line 34) has returned 10 rows of customer names. The second query (line 35) has returned 0 rows. The result grid has a header row with 'Customer_Name' and a data row with 'BRAD EASON'. The status bar at the bottom shows 'Object Info', 'Session', 'cust_dimen 24', and 'Read Only'.

```
25 from market_fact m
26 join prod_dimen p on m.Prod_id = p.Prod_id
27 group by Product_Sub_Category
28 order by sum(m.Profit)
29 LIMIT 1)
30 group by c.Region
31 order by sum(m.Profit);
32
33
34 • SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '_r%';
35 • SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '__d%';
36
```

Customer_Name
BRAD EASON
BRAD EASON
CHAD CUNNINGHAM
FRED WASSERMAN
CINDY SCHNELLING
CHAD CUNNINGHAM
TRUDY SCHMIDT
TRUDY SCHMIDT
CHAD SIEVERT
BRAD NORVELL
CHAD SIEVERT

16. Write a SQL query to to make a list with Cust_Id, Sales, Customer Name and their region where sales are between 1000 and 5000.

use superstores;

select cust_dimen.customer_name,cust_dimen.region,cust_dimen.cust_id,market_fact.sales

from market_fact

join cust_dimen on market_fact.cust_id = cust_dimen.cust_id

where sales between 1000 and 5000

GROUP BY cust_dimen.cust_id;

The screenshot shows a SQL IDE interface. On the left is a 'SCHEMAS' pane with a tree view containing 'hr', 'jimjh', 'newschema', 'northwind', 'sakila', 'superstores' (expanded), 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'superstores' schema is selected. The main query editor displays the following SQL code:

```
46 use superstores;
47 select cust_dimen.customer_name,cust_dimen.region,cust_dimen.cust_id,market_fact.sales
48 from market_fact
49 join
50 cust_dimen on market_fact.cust_id = cust_dimen.cust_id
51 where sales between 1000 and 5000
52 GROUP BY cust_dimen.cust_id;
```

Below the query editor is the 'Result Grid' showing 6 rows of data:

customer_name	region	cust_id	sales
AARON HAWKINS	QUEBEC	Cust_839	3364.248
AARON SMAYLING	PRARIE	Cust_371	3202.25
AARON SMAYLING	QUEBEC	Cust_931	1062.69
ADAM BELLAVANCE	ONTARIO	Cust_452	1974.66
ADAM HART	YUKON	Cust_1474	1480.91
AIMEE BDXBY	ONTARIO	Cust_565	1736.53

On the right side, there is a 'SQL Additions' pane with a 'SELECT' button and a 'No Context Help' message. Below the result grid is an 'Output' pane showing a log of actions and their durations:

#	Time	Action	Message	Duration / Fetch
47	06:09:10	select round(sum(Sales),2) as "TOTAL SALES", Round(avg(Sales),2) as "AVERAGE SALES"...	1 row(s) returned	0.015 sec / 0.000 sec
48	19:58:29	select c.Region as "Region",count(m.Ship_id) as "No of Shipments", round(sum(m.Profit),2)...	8 row(s) returned	0.359 sec / 0.000 sec
49	20:02:54	SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '___d%' LIMIT ...	70 row(s) returned	0.000 sec / 0.000 sec
50	20:06:14	SELECT orders_dimen.Order_Date,market_fact.order_quantity,market_fact.sales FROM mar...	1000 row(s) returned	0.047 sec / 0.000 sec
51	20:10:30	SELECT Customer_Name,COUNT(*) AS no_of_small_business_owners,region FROM cust_di...	1 row(s) returned	0.016 sec / 0.000 sec
52	20:12:32	select cust_dimen.customer_name,cust_dimen.region,cust_dimen.cust_id,market_fact.sales f...	1000 row(s) returned	0.234 sec / 0.031 sec

At the bottom left, there is a 'Query Completed' status bar.

17. Write a SQL query to find the 3rd highest sales.

SELECT Sales FROM market_fact ORDER BY Sales DESC LIMIT 2,1;

The screenshot shows a SQL IDE interface. On the left is a 'SCHEMAS' navigator with a tree view showing databases like 'hr', 'jiminh', 'newschema', 'northwind', 'sakila', 'superstores', 'sys', and 'sys'. The 'superstores' database is selected, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main query editor displays a SQL query:

```
48 from market_fact
49 join
50 cust_dimen on market_fact.cust_id = cust_dimen.cust_id
51 where sales between 1000 and 5000
52 GROUP BY cust_dimen.cust_id;
53
54 • SELECT Sales FROM market_fact ORDER BY Sales DESC LIMIT 2,1;
```

Below the query editor is a 'Result Grid' showing a single row with the value '41343.21'. On the right, there is a 'SQL Additions' panel with a 'SELECT' button and a 'No Context Help' message. At the bottom, the 'Output' window shows a list of actions and their results:

#	Time	Action	Message	Duration / Fetch
49	20:02:54	SELECT Customer_Name FROM cust_dimen WHERE Customer_Name LIKE '___d%' LIMIT ...	70 row(s) returned	0.000 sec / 0.000 sec
50	20:06:14	SELECT orders_dimen.Order_Date,market_fact.order_quantity,market_fact.sales FROM mar...	1000 row(s) returned	0.047 sec / 0.000 sec
51	20:10:30	SELECT Customer_Name,COUNT(*) AS no_of_small_business_owners,region FROM cust_di...	1 row(s) returned	0.016 sec / 0.000 sec
52	20:12:32	select cust_dimen.customer_name,cust_dimen.region,cust_dimen.cust_id,market_fact.sales f...	1000 row(s) returned	0.234 sec / 0.031 sec
53	20:15:57	SELECT Sales FROM market_fact ORDER BY Sales DESC LIMIT 2,1	1 row(s) returned	0.031 sec / 0.000 sec
54	20:16:11	SELECT Sales FROM market_fact ORDER BY Sales DESC LIMIT 2,1	1 row(s) returned	0.016 sec / 0.000 sec

The status bar at the bottom indicates 'Query Completed'.

18) Where is the least profitable product subcategory shipped the most? For the least profitable product sub-category, display the region-wise no_of_shipments and the profit made in each region in decreasing order of profits (i.e. region, no_of_shipments, profit_in_each_region) → Note: You can hardcode the name of the least profitable product subcategory

```
select c.Region as "Region",count(m.Ship_id) as "No of Shipments",  
       round(sum(m.Profit),2) as "Profit in each region"
```

```
from market_fact m
```

```
join cust_dimen c on m.Cust_id = c.Cust_id
```

```
join prod_dimen p on m.Prod_id = p.Prod_id
```

Where Product_Sub_Category = (

Select p.Product_Sub_Category

```
from market_fact m
```

```
join prod_dimen p on m.Prod_id = p.Prod_id
```

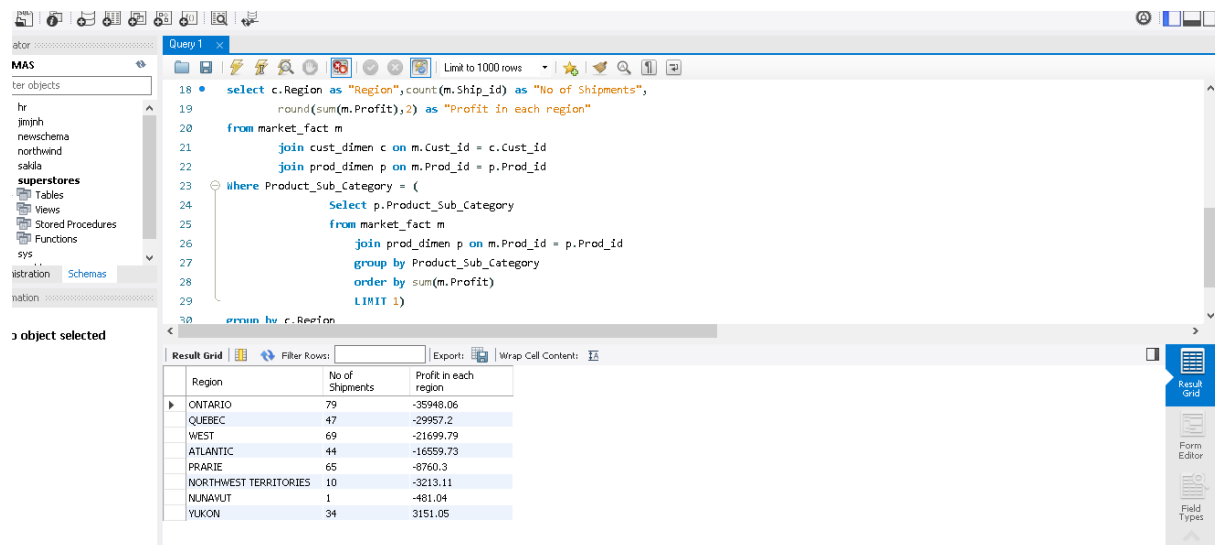
```
group by Product_Sub_Category
```

```
order by sum(m.Profit)
```

```
LIMIT 1)
```

```
group by c.Region
```

```
order by sum(m.Profit);
```



The screenshot shows a SQL IDE interface with a query editor on the left and a results grid on the right. The query in the editor is as follows:

```
18 select c.Region as "Region",count(m.Ship_id) as "No of Shipments",  
19 round(sum(m.Profit),2) as "Profit in each region"  
20 from market_fact m  
21 join cust_dimen c on m.Cust_id = c.Cust_id  
22 join prod_dimen p on m.Prod_id = p.Prod_id  
23 where Product_Sub_Category = (  
24 select p.Product_Sub_Category  
25 from market_fact m  
26 join prod_dimen p on m.Prod_id = p.Prod_id  
27 group by Product_Sub_Category  
28 order by sum(m.Profit)  
29 limit 1)  
30 group by c.Region  
31 order by sum(m.Profit);
```

The results grid on the right displays the following data:

Region	No of Shipments	Profit in each region
ONTARIO	79	-35948.06
QUEBEC	47	-29957.2
WEST	69	-21699.79
ATLANTIC	44	-16859.73
PRARIE	65	-8760.3
NORTHWEST TERRITORIES	10	-3213.11
NUNAVUT	1	-481.04
YUKON	34	3151.05

