

SQL ASSIGNMENT

Task 1:- Understanding the Data

1. Describe the data in hand in your own words.

The database contains Sales details of a superstore.

- The structure has 5 tables,
- Cust_dimen (containing details about customer and their respective locations)
- Prod_dimen (containing product category and their subcategories)
- Orders_dimen (with order no, date, and priority)
- Shipping_dimen (with ship date, order and shipping mode),
- Market_fact (order wise customer wise marketwise order quantity,

These tables will give you information upon querying. These tables having dimensions and has related facts. Using these table we can derive various insights which helps in decisions making regarding product segment wise sales and profitability, shipping mode wise, region wise , profitability etc.

2. Identify and list the Primary Keys and Foreign Keys for this dataset provided to you(In case you don't find either primary or foreign key, then specially mention this in your answer)

ANSWER: primary key means we can identified each record uniquely in table and when we join two table then that primary key became foreign key for 2 nd table.

1) Table: cust_dimen:

- a. Primary key: cust_id
- b. Foreign key: No

2) Table: market_fact:

- a. Primary key: No
- c. Foreign_key: Ord_id, Prod_id, Ship_id and Cust_id

3) Table: prod_dimen:

- a. Primary key: prod_id
- b. Foreign key: No

4) Table:orders_dimen:

a. Primary key: Ord_id as Primary Key, although Order_ID is also there but it is advisable to use Ord_id as primary Key to ensure relationship consistency
But Order_ID will as foreign key in shipping_dimen will help retrieve the details.

b. Foreign key: ord_id

5) Table:shipping_dimen:

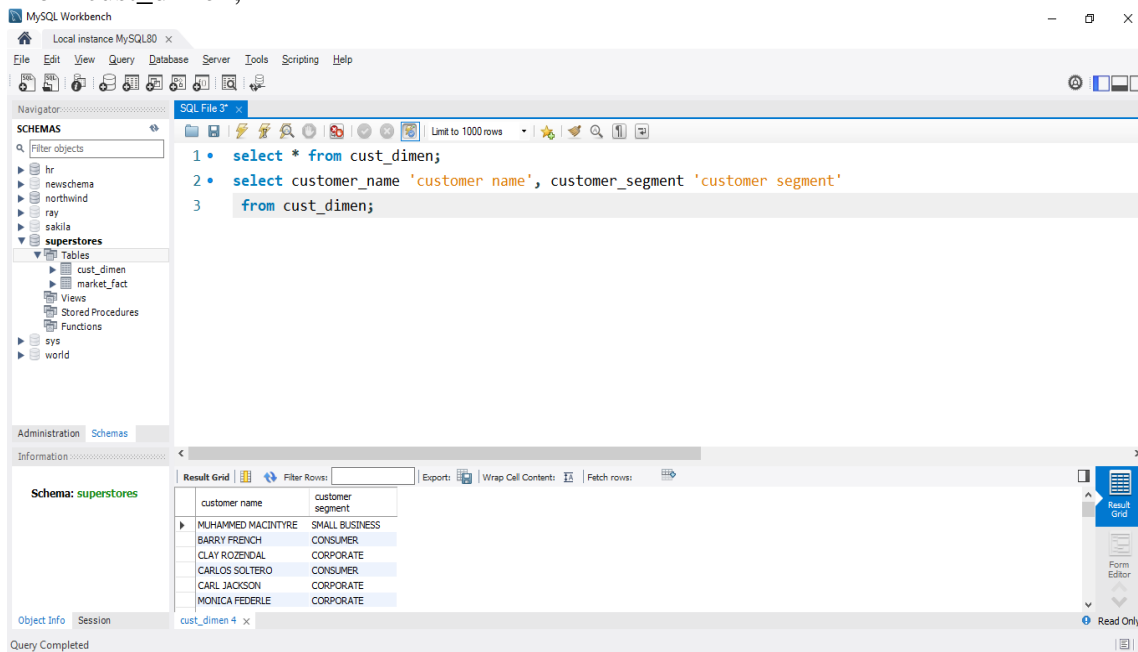
a. Primary key: Ship_id

b. Foreign key: Order_i

Task 2:- Basic & Advanced Analysis

1. Write a query to display the Customer_Name and Customer Segment using alias name "Customer Name", "Customer Segment" from table Cust_dimen.

```
select customer_name 'customer name', customer_segment 'customer segment'
from cust_dimen;
```



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 • select * from cust_dimen;
2 • select customer_name 'customer name', customer_segment 'customer segment'
3   from cust_dimen;
```

The Results tab is selected, displaying the following data:

customer name	customer segment
MUHAMMED MACINTYRE	SMALL BUSINESS
BARRY FRENCH	CONSUMER
CLAY ROZENDAL	CORPORATE
CARLOS SOLTERO	CONSUMER
CARL JACKSON	CORPORATE
MONICA FEDERLE	CORPORATE

The left sidebar shows the Schemas pane with the 'superstores' schema selected. The bottom status bar indicates 'Query Completed'.

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 the MySQL Workbench interface. The SQL Editor contains the query: `1 • select * from cust_dimen order by customer_name desc;`. The Results tab displays the output in a table format. The table has columns: Customer_Name, Province, Region, Customer_Segment, and Cust_id. The data is sorted by Customer_Name in descending order.

Customer_Name	Province	Region	Customer_Segment	Cust_id
YOSEPH CARROLL	ALBERTA	WEST	CONSUMER	Cust_1798
YANA SORENSEN	YUKON	YUKON	CORPORATE	Cust_1519
YANA SORENSEN	NEWFOUNDLAND	ATLANTIC	CORPORATE	Cust_637
YANA SORENSEN	QUEBEC	QUEBEC	CORPORATE	Cust_851
YANA SORENSEN	BRITISH COLUMBIA	WEST	CORPORATE	Cust_1577
XYLONA PRICE	ONTARIO	ONTARIO	CORPORATE	Cust_1006
WILLIAM BROWN	SASKACHEWAN	PRARIE	CORPORATE	Cust_1266

3. Write a query to get the Order ID, Order date from table orders_dimen where 'Order Priority' is high.

```
select Order_id, Order_date from orders_dimen where Order_Priority= 'high';
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `1 • select * from market_fact;`, `2 • select * from cust_dimen;`, `3 • select * from orders_dimen;`, and `4 • select Order_id, Order_date from orders_dimen where Order_Priority= 'high';`. The Results tab displays the output in a table format. The table has columns: Order_id and Order_date. The data is sorted by Order_id in ascending order.

Order_id	Order_date
293	01-10-2012
483	10-07-2011
613	17-06-2011
640	24-03-2011
1540	04-08-2012
1702	06-05-2011
1761	23-12-2010

4. Find the total and the average sales (display total_sales and avg_sales)

```
select sum(sales) as total_sales, avg(sales) as avg_sale from market_fact;
```

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 • select * from market_fact;  
2 • select sum(sales) as total_sales, avg(sales) as avg_sale  
3   from market_fact;
```

The Results panel at the bottom displays the output of the query in a table format:

total_sales	avg_sale
14647187.904000023	1757.1002763915576

The left sidebar shows the database schema, including the 'superstores' database and the 'market_fact' table.

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

```
select min(sales) as min_sales, max(sales) as max_sale  
from market_fact;
```

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 • select * from market_fact;  
2 • select min(sales) as min_sales, max(sales) as max_sale  
3   from market_fact;
```

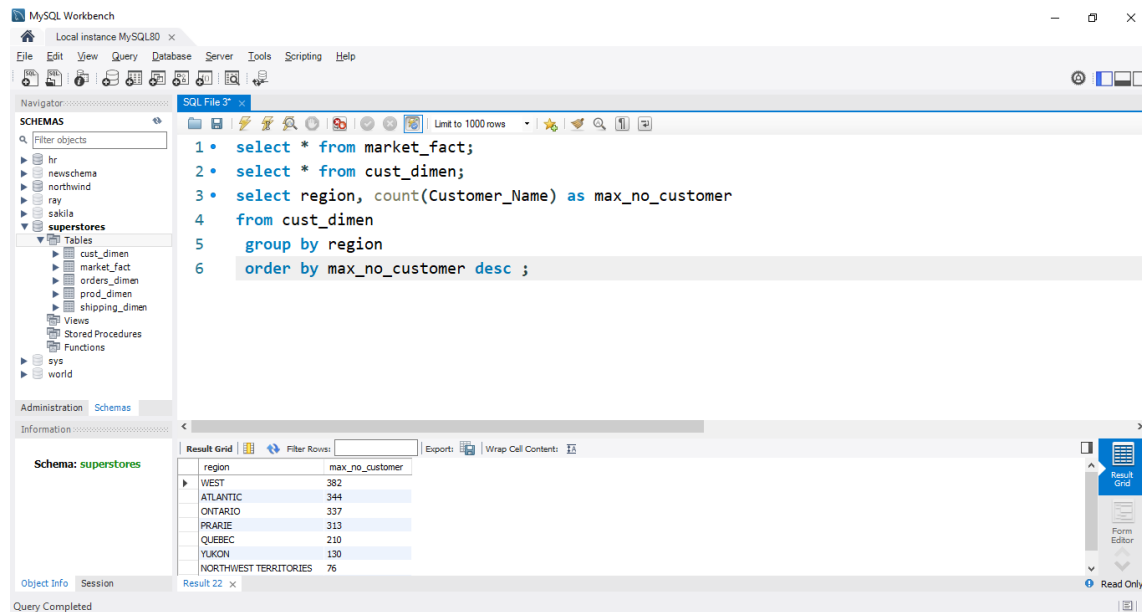
The Results panel at the bottom displays the output of the query in a table format:

min_sales	max_sale
2.24	89061.05

The left sidebar shows the database schema, including the 'superstores' database and the 'market_fact' table.

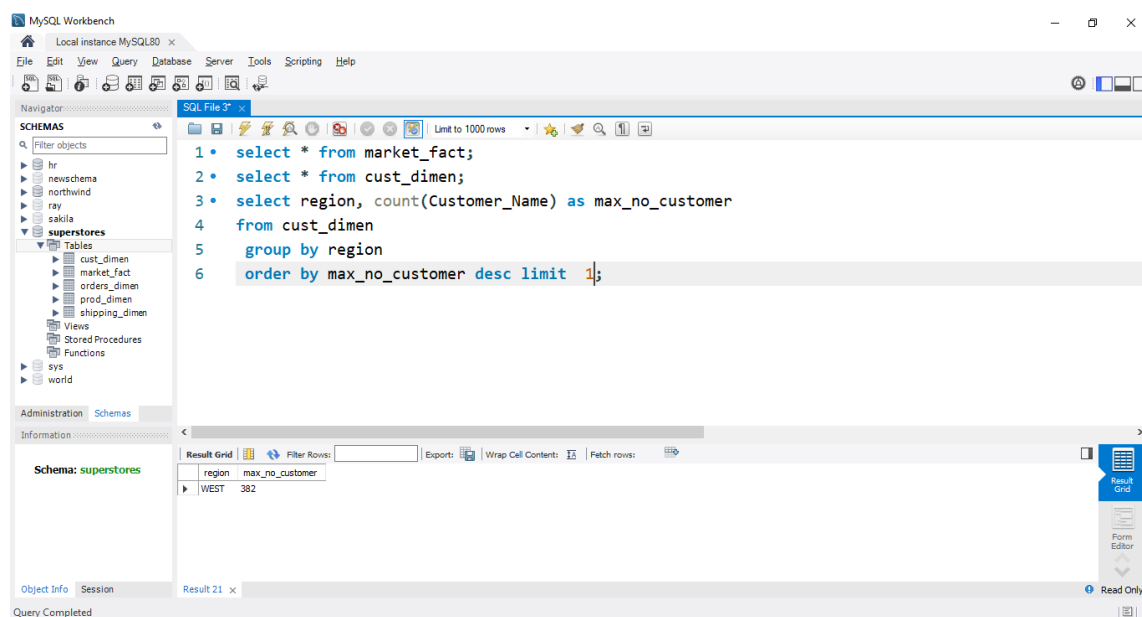
6. Display the number of customers in each region in decreasing order of no_of_customers. The result should contain columns Region, no_of_customers.

```
select region, count(Customer_Name) as max_no_customer  
from cust_dimen  
group by region  
order by max_no_customer desc ;
```



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

```
select region, count(Customer_Name) as max_no_customer  
from cust_dimen  
group by region  
order by max_no_customer desc limit 1;
```



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.Customer_name, count(m.Prod_id)
from cust_dimen c join
market_fact m
on m.Cust_id = c.Cust_id join
prod_dimen p
on p.Prod_id = m.Prod_id
where c.Region = 'ATLANTIC' and p.Product_Sub_Category = 'TABLES'
group by c.Customer_name;
```

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 select * from market_fact;
2 select * from cust_dimen;
3 select * from prod_dimen;
4 select c.Customer_name, count(m.Prod_id)
5   from cust_dimen c join
6       market_fact m
7     on m.Cust_id = c.Cust_id join
8       prod_dimen p
9     on p.Prod_id = m.Prod_id
10  where c.Region = 'ATLANTIC' and p.Product_Sub_Category = 'TABLES'
11  group by c.Customer_name;
```

The Results tab shows the following data:

Customer_name	count(m.Prod_id)
ALEXANDRA GANNAWAY	1
ANEMONE RATHNER	1
BARRY FRANZ	1
BECKY MARTIN	1
BEN PETERMAN	1
BOBBY TRAFION	1
BRADLEY TALBOTT	1

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

```
select Customer_name, count(*) as 'small business' from cust_dimen
where customer_segment='small business' and province = 'Ontario'
group by customer_name;
```

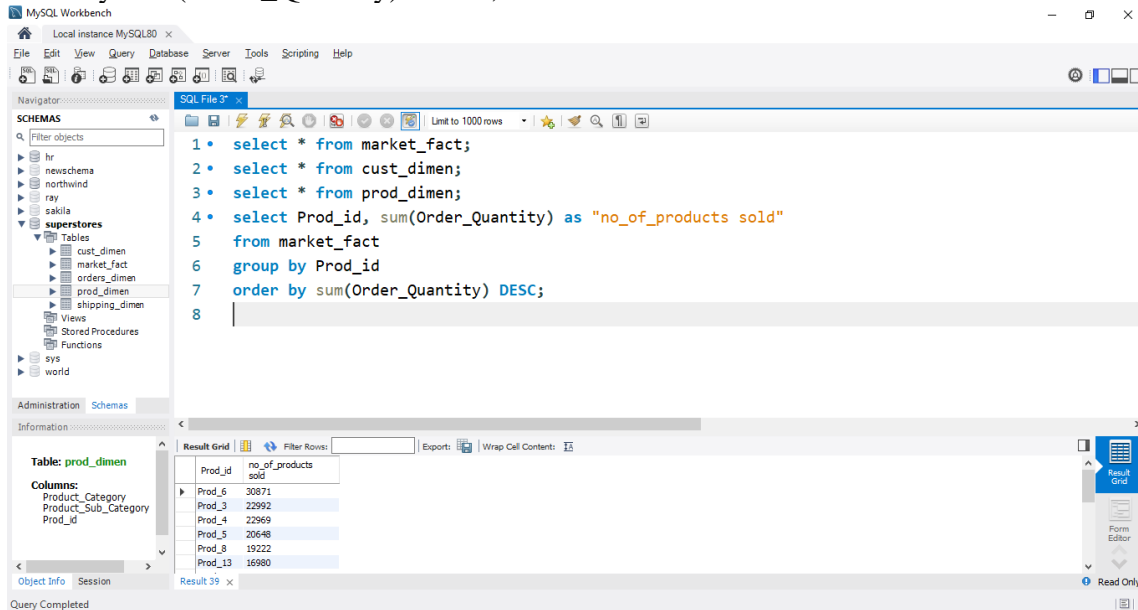
The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 select * from market_fact;
2 select * from cust_dimen;
3 select * from prod_dimen;
4 select Customer_name, count(*) as 'small business' from cust_dimen
5   where customer_segment='small business' and province = 'Ontario'
6   group by customer_name;
```

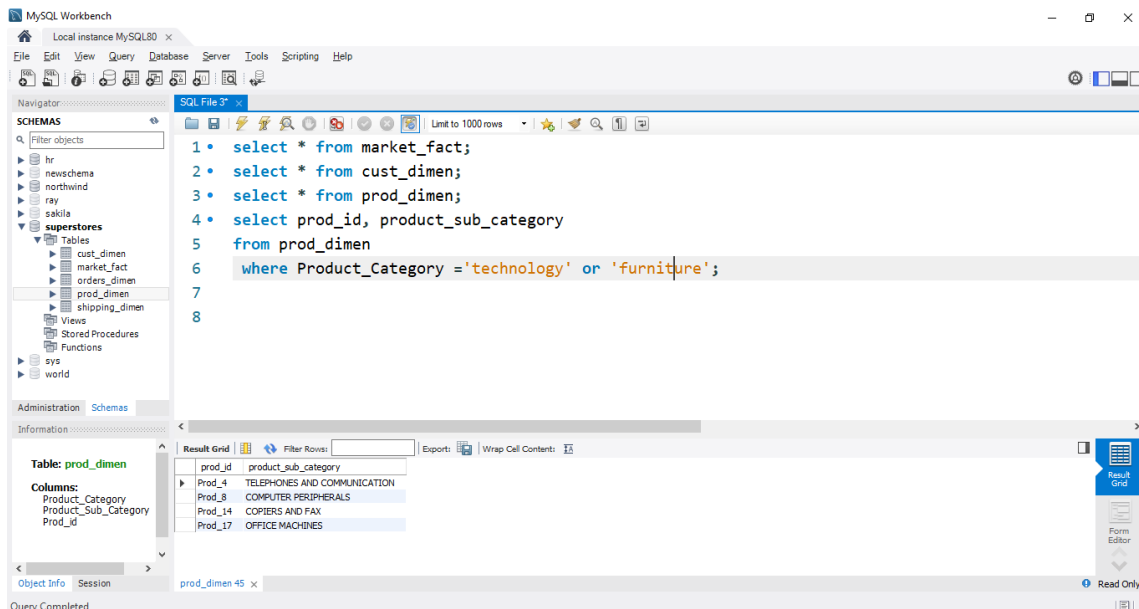
The Results tab shows the following data:

Customer_name	small business
CHRISTINA VANDERZANDEN	1
MEG O'CONNEL	1
CHRISTINE SUNDARESAM	1
DOUG O'CONNELL	1
CHRISTINE KARGATIS	1
CRAIG CARROLL	1

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;

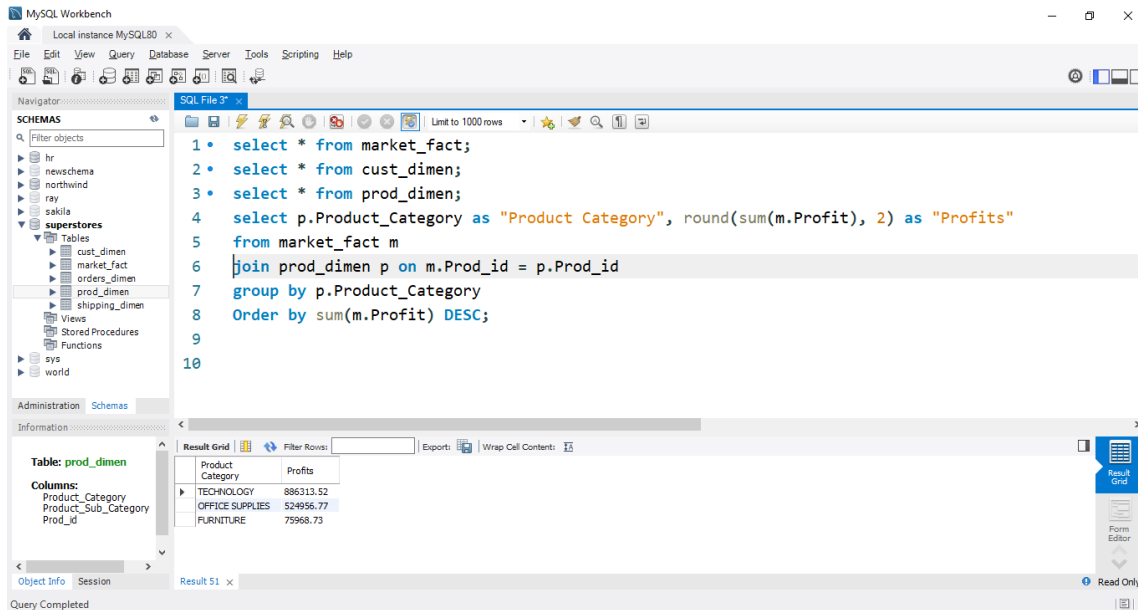


11. Display product Id and product sub category whose product category belongs to Furniture and Technology. The result should contain columns product id, product sub category.
select prod_id, product_sub_category from prod_dimen
where Product_Category = 'technology' or 'furniture';



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

```
select p.Product_Category as "Product Category", round(sum(m.Profit), 2) as "Profits"
from market_fact m
join prod_dimen p on m.Prod_id = p.Prod_id
group by p.Product_Category
Order by sum(m.Profit) DESC;
```



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 • select * from market_fact;
2 • select * from cust_dimen;
3 • select * from prod_dimen;
4 • select p.Product_Category as "Product Category", round(sum(m.Profit), 2) as "Profits"
5 • from market_fact m
6 • join prod_dimen p on m.Prod_id = p.Prod_id
7 • group by p.Product_Category
8 • Order by sum(m.Profit) DESC;
9
10
```

The left sidebar shows the Schemas pane with a tree view of the database structure. The 'prod_dimen' table is selected, and its columns (Product_Category, Product_Sub_Category, Prod_id) are listed in the Information pane.

The bottom pane shows the Result Grid with the following data:

Product Category	Profits
TECHNOLOGY	886313.52
OFFICE SUPPLIES	524956.77
FURNITURE	79968.73

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

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;
```

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 select * from market_fact;
2 select * from cust_dimen;
3 select * from prod_dimen;
4 select p.Product_Category as "Product Category", p.Product_Sub_Category as "Product Sub Category",
5 round(sum(m.Profit), 2) as "Total Profits"
6 from market_fact m
7 join prod_dimen p on m.Prod_id = p.Prod_id
8 group by p.Product_Sub_Category
9 Order by p.Product_Category;
```

The Results panel shows the output of the query, which is a table with three columns: Product Category, Product Sub Category, and Total Profits. The data is grouped by Product Sub Category.

Product Category	Product Sub Category	Total Profits
FURNITURE	TABLES	-113468.18
FURNITURE	BOOKCASES	-33729.09
FURNITURE	OFFICE FURNISHINGS	100427.93
FURNITURE	CHAIRS & CHAIRMATS	122738.07
OFFICE SUPPLIES	SCISSORS, RULERS AND TRIMMERS	-7799.25
OFFICE SUPPLIES	PENS & ART SUPPLIES	7564.78

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

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
7
8 select Order_Date, Order_Quantity, Sales from market_fact s, orders_dimen c
9 where s.Ord_id = c.Ord_id;
```

The Results panel shows the output of the query, which is a table with three columns: Order_Date, Order_Quantity, and Sales. The data is grouped by Order Date.

Order_Date	Order_Quantity	Sales
28-05-2011	5	14.76
30-10-2011	38	465.9
24-02-2011	27	305.05
25-12-2011	15	3364.248
25-12-2011	10	1410.93
15-08-2009	48	460.69
04-10-2010	30	443.46
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

15. Display the names of the customers whose name contains the

i) Second letter as 'R'

ii) Fourth letter as 'D'

```
8 • select Customer_Name from Cust_dimen
9   where Customer_Name like '_R%' and Customer_Name like '___D%';
10
11
12
```

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

Customer_Name
BRAD EASON
BRAD EASON
FRED WASSERMAN
TRUDY SCHMIDT
TRUDY SCHMIDT
BRAD NORVELL
BRAD NORVELL
BRAD EASON
FRED HOPKINS
FRED MCMATH
BRAD THOMAS
FRED MCMATH
BRADLEY DRUCKER

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

```
8 • select c.Cust_id,s.Sales,c.Customer_Name,c.Region from market_fact s,cust_dimen c
9   where s.Cust_id = c.Cust_id and Sales between 1000 and 5000;
10
11
12
```

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

Cust_id	Sales	Customer_Name	Region
Cust_3	1285.37	CLAY ROZENDAL	NUNAVUT
Cust_3	4965.7595	CLAY ROZENDAL	NUNAVUT
Cust_8	1815.49	NEOLA SCHNEIDER	NUNAVUT
Cust_9	1474.33	CARLOS DALY	NUNAVUT
Cust_12	4462.23	SYLVIA FOULSTON	NUNAVUT
Cust_13	2480.9205	JIM RADFORD	NUNAVUT
Cust_16	1210.0515	ANNIE CYPRUS	NUNAVUT
Cust_16	4253.009	ANNIE CYPRUS	NUNAVUT
Cust_19	1078.49	JACK GARZA	NUNAVUT
Cust_20	3554.46	JULIA WEST	NUNAVUT
Cust_24	3338.98	NICOLE HANSEN	NUNAVUT
Cust_25	1311.25	DOROTHY WARDLE	NUNAVUT

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

```
8 • select min(Sales) as `3rd highest salary`
9 FROM (
10 select Sales from market_fact order by Sales desc limit 3
11 ) as a;
```

Result Grid

	3rd highest salary
▶	41343.21

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.

```
9
10 • select Region, count(Ship_id) as no_of_shipment, sum(Profit) as profit_in_each_region from
11 cust_dimen c, market_fact s, prod_dimen p
12 where c.Cust_id = s.Cust_id and s.Prod_id = p.Prod_id
13 group by Region
14 order by profit_in_each_region asc;
```

Result Grid

	Region	no_of_shipment	profit_in_each_region
▶	NUNAVUT	55	1274.0500000000004
	YUKON	367	58211.529999999995
	NORTHWEST TERRITORIES	275	65085.620000000004
	QUEBEC	508	123393.229999999998
	ATLANTIC	725	196299.92
	WEST	1378	218579.520000000002
	PRARIE	1148	255930.37999999999
	ONTARIO	1225	272667.29999999997