



Assignment # 2

Presented By

M.Waqas (22I - 2469)

Abdullah Mansoor (22I - 8808)

To

Sir Bilal

Table of Contents

1 Assumptions.....	3
2 Even Queries	3
3 Odd Queries	23

1 Assumptions

Due to ambiguity in Query 67 a new attribute city was introduced in Customers Table.

2 Even Queries

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays several SELECT statements (Query 2, 4, 6, 8, 10, 12) against the Products table. The results pane shows the output of Query 10, which selects products with stock_level between 10 and 20. The results are as follows:

name	price
T-Shirt	20
Laptop	1300
Lipstick	16
Basketball	36
Board Game	50
Coffee	10
Gran Turismo 7	5
Novel	15
Pizza	40

Below the results, a message indicates "Query executed successfully." and shows the system status bar with "65°F Light rain" and the date "3/27/2024".

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays several SELECT statements (Query 2, 4, 6, 8, 10, 12) against the Products table. The results pane shows the output of Query 10, which selects products with stock_level between 10 and 20. The results are as follows:

product_id	name	description	price	stock_level	category_id
2	Laptop	Laptop with RTX 3080	1300	50	2
4	Basketball	Outdoor Basketball	36	40	4
5	Board Game	Family Board Game	50	25	5
6	Coffee	Cappuccino Coffee	10	30	6
10	Pizza	Extra Large Pizza	40	45	10

Below the results, a message indicates "Query executed successfully." and shows the system status bar with "65°F Light rain" and the date "3/27/2024".

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio' is open. The query pane contains several commented-out queries (Query 2, Query 4, Query 6, Query 8, Query 10, Query 12) and one active query:

```
-- SELECT * FROM Products;
```

The results pane displays a single row of data from the Products table:

product_id	name	description	price	stock_level	category_id
1	2	Laptop	Laptop with RTX 3060	1300	50

Query executed successfully.

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio' is open. The query pane contains several commented-out queries (Query 2, Query 4, Query 6, Query 8, Query 10, Query 12) and one active query:

```
-- SELECT * FROM Products;
```

The results pane displays a single row of data from the Products table:

product_id	name	description	price	stock_level	category_id
1	2	Laptop	Laptop with RTX 3060	1300	50

Query executed successfully.

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

-- Query 10
SELECT DISTINCT category_id FROM Products;

-- Query 12
SELECT * FROM Products WHERE stock_level >= 10 AND stock_level <= 20;

-- Results 10 rows

category_id
1
2
3
4
5
6
7
8
9
10

Query executed successfully.

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

-- Query 6
SELECT * FROM Products WHERE price BETWEEN 1000 AND 5000;
SELECT * FROM Products WHERE price >= 1000 AND price <= 5000;

-- Query 8
SELECT * FROM Products WHERE name LIKE '%Shoes%';

-- Query 10
SELECT DISTINCT category_id FROM Products;

-- Query 12
SELECT * FROM Products WHERE stock_level >= 10 AND stock_level <= 20;
SELECT * FROM Products WHERE stock_level BETWEEN 10 AND 20;

-- Query 14
SELECT * FROM Products
WHERE description LIKE '%embroidered%' AND price < 3000;

-- Results 0 rows

product_id	name	description	price	stock_level	category_id
------------	------	-------------	-------	-------------	-------------

Query executed successfully.

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
-- Query 6
SELECT * FROM Products WHERE price BETWEEN 1000 AND 5000;
SELECT * FROM Products WHERE price >= 1000 AND price <= 5000;

-- Query 8
SELECT * FROM Products WHERE name LIKE '%Shoes%';

-- Query 10
SELECT DISTINCT category_id FROM Products;

-- Query 12
SELECT * FROM Products WHERE stock_level >= 10 AND stock_level <= 20;
SELECT * FROM Products WHERE stock_level BETWEEN 10 AND 20;

-- Query 14
SELECT * FROM Products WHERE description LIKE '%embroidered%' AND price < 3000;
```

100 %

Results Messages

product_id	name	description	price	stock_level	category_id
------------	------	-------------	-------	-------------	-------------

Query executed successfully.

Ready 65°F Light rain Search Col 1 Ch 1 INS 9:49 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
-- Query 12
SELECT * FROM Products WHERE stock_level >= 10 AND stock_level <= 20;
SELECT * FROM Products WHERE stock_level BETWEEN 10 AND 20;

-- Query 14
SELECT * FROM Products WHERE description LIKE '%embroidered%' AND price < 3000;

-- Query 16
SELECT * FROM Products WHERE name NOT LIKE '%L%';

-- Query 18
SELECT * FROM Products WHERE stock_level < (SELECT MIN(stock_level) FROM Products);

-- Query 20
SELECT * FROM Products WHERE name LIKE '%Suit';
```

100 %

Results Messages

product_id	name	description	price	stock_level	category_id
1	T-Shirt	Cotton T-Shirt For Adults	20	100	1
2	Basketball	Outdoor Basketball	35	40	4
3	Board Game	Family Board Game	50	25	5
4	Coffee	Cappuccino Coffee	10	30	6
5	Pen	Blue Ink Pen	5	150	7
6	Gran Turismo 7	PSS Game	80	60	8
7	Novel	Adventure Novel	15	80	9
8	Pizza	Extra Large Pizza	40	45	10

Query executed successfully.

Ready 65°F Light rain Search Col 1 Ch 1 INS 9:50 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

-- Query 12
SELECT * FROM Products WHERE stock_level >= 10 AND stock_level <= 20;
SELECT * FROM Products WHERE stock_level BETWEEN 10 AND 20;

-- Query 14
SELECT * FROM Products WHERE description LIKE '%embroidered%' AND price < 3000;

-- Query 16
SELECT * FROM Products WHERE name NOT LIKE 'L%';

-- Query 18
SELECT * FROM Products WHERE stock_level < (SELECT MIN(stock_level) FROM Products);

-- Query 20
SELECT * FROM Products WHERE name LIKE '%Suit';

100% ▾

Results Messages

product_id	name	description	price	stock_level	category_id

Query executed successfully.

Ready 65°F Light rain Search Col 1 Ch 1 INS 9:50 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

-- Query 16
SELECT * FROM Products WHERE name NOT LIKE 'L%';

-- Query 18
SELECT * FROM Products WHERE stock_level < (SELECT MIN(stock_level) FROM Products);

-- Query 20
SELECT * FROM Products WHERE name LIKE '%Suit';

-- Query 22
SELECT AVG(price) AS average_price_of_all_products FROM Products;

-- Query 24
SELECT MIN(stock_level) AS min_stock_level FROM Products;

-- Query 26
SELECT (AVG(stock_level*1.0)) AS average_stock_level FROM Products;

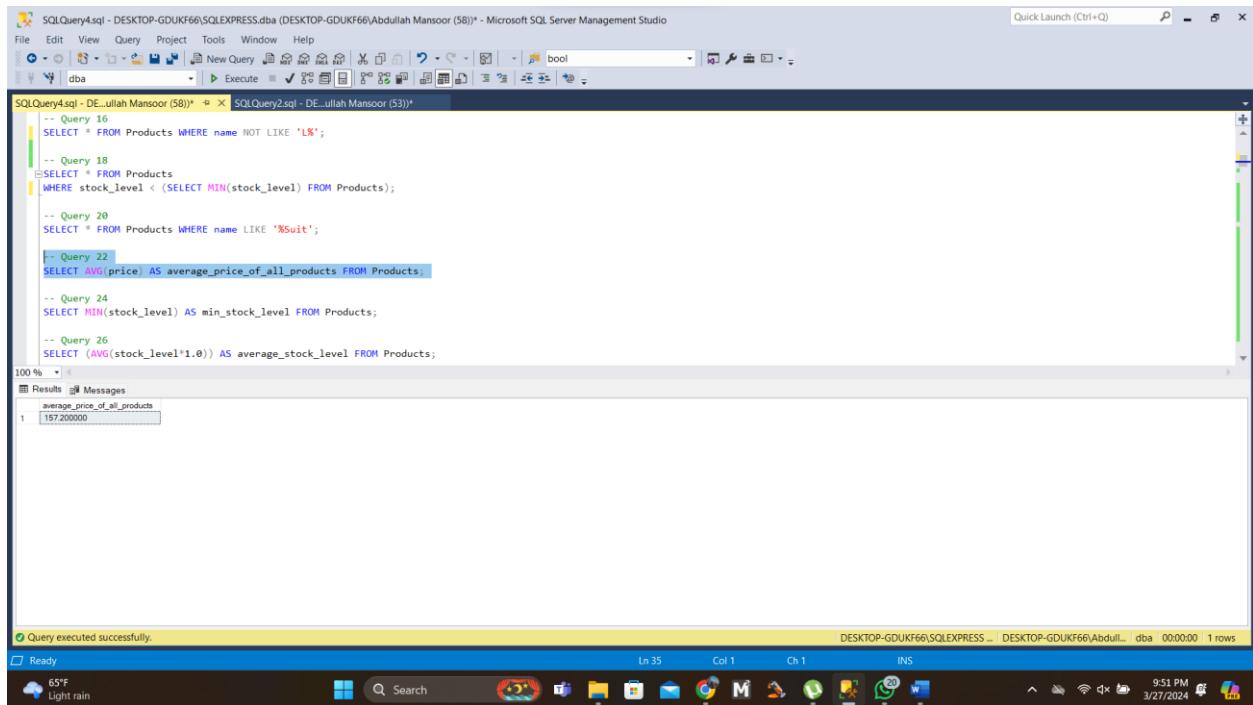
100% ▾

Results Messages

product_id	name	description	price	stock_level	category_id

Query executed successfully.

Ready 65°F Light rain Search Col 1 Ch 1 INS 9:50 PM 3/27/2024



```
-- Query 16
SELECT * FROM Products WHERE name NOT LIKE 'L%';

-- Query 18
ESELECT * FROM Products
WHERE stock_level < (SELECT MIN(stock_level) FROM Products);

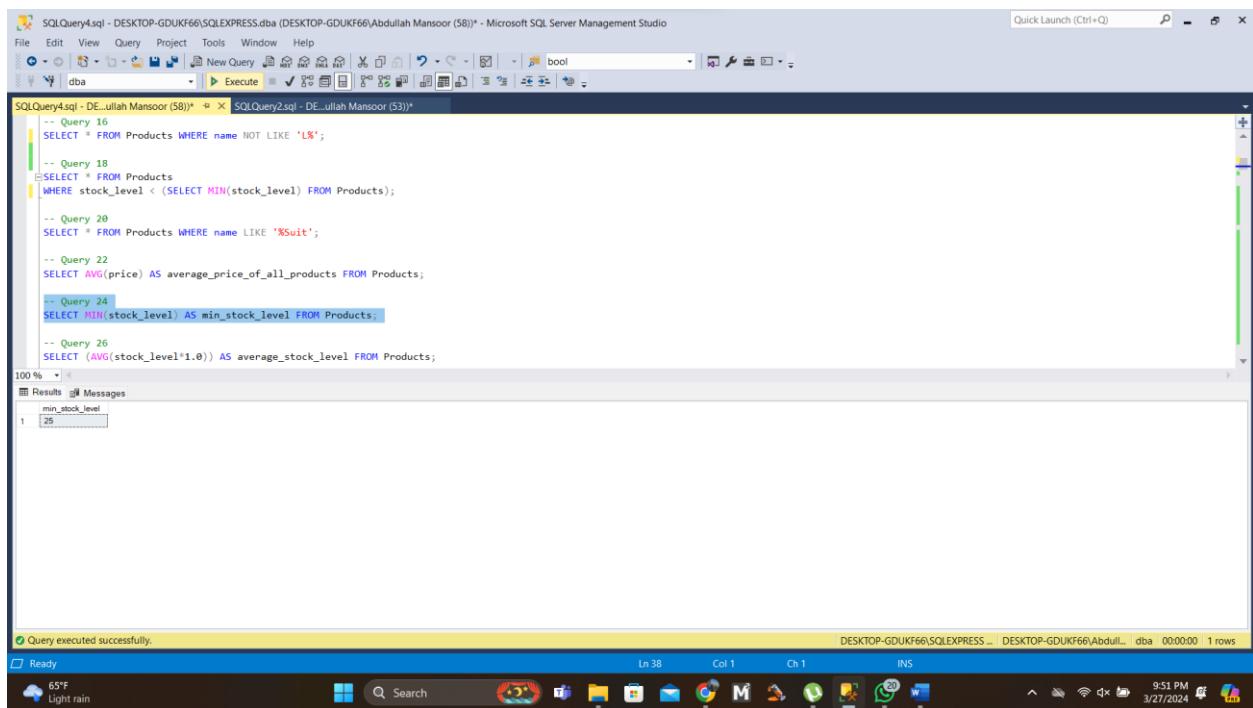
-- Query 20
SELECT * FROM Products WHERE name LIKE '%Suit';

-- Query 22
SELECT AVG(price) AS average_price_of_all_products FROM Products;

-- Query 24
SELECT MIN(stock_level) AS min_stock_level FROM Products;

-- Query 26
SELECT (AVG(stock_level*1.0)) AS average_stock_level FROM Products;
```

187.20000



```
-- Query 16
SELECT * FROM Products WHERE name NOT LIKE 'L%';

-- Query 18
ESELECT * FROM Products
WHERE stock_level < (SELECT MIN(stock_level) FROM Products);

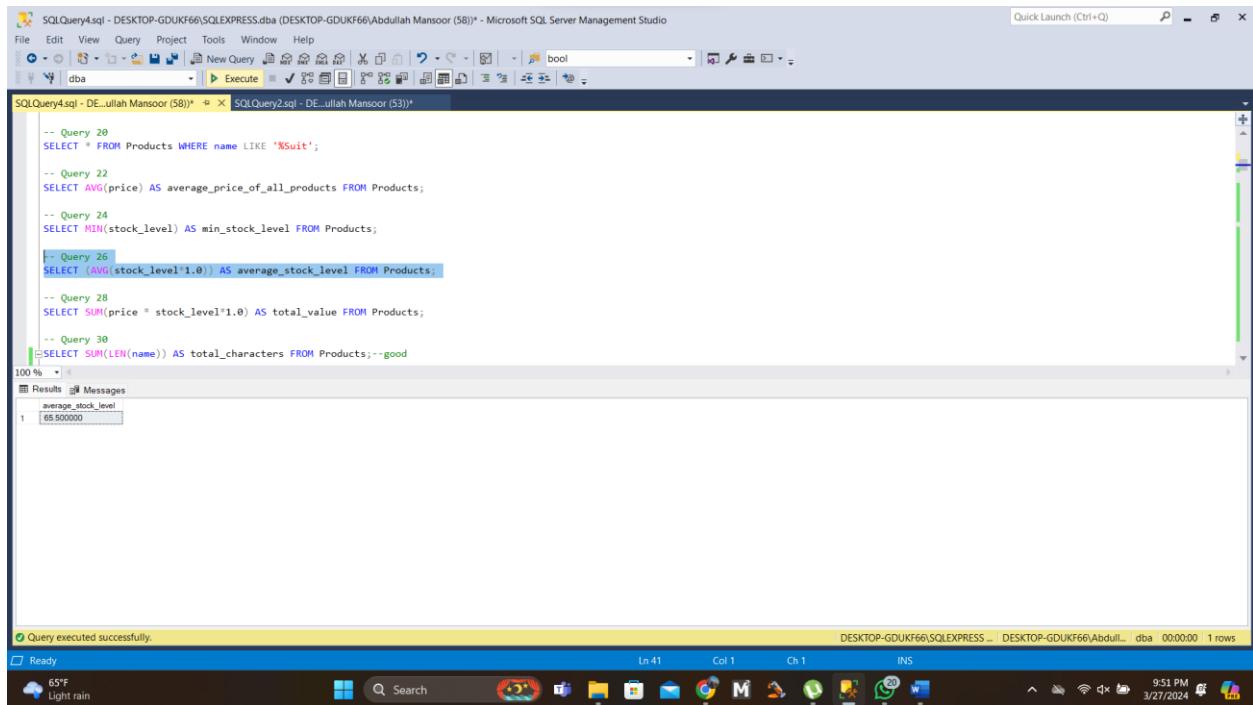
-- Query 20
SELECT * FROM Products WHERE name LIKE '%Suit';

-- Query 22
SELECT AVG(price) AS average_price_of_all_products FROM Products;

-- Query 24
SELECT MIN(stock_level) AS min_stock_level FROM Products;

-- Query 26
SELECT (AVG(stock_level*1.0)) AS average_stock_level FROM Products;
```

28



```
-- Query 20
SELECT * FROM Products WHERE name LIKE '%Suit';

-- Query 22
SELECT AVG(price) AS average_price_of_all_products FROM Products;

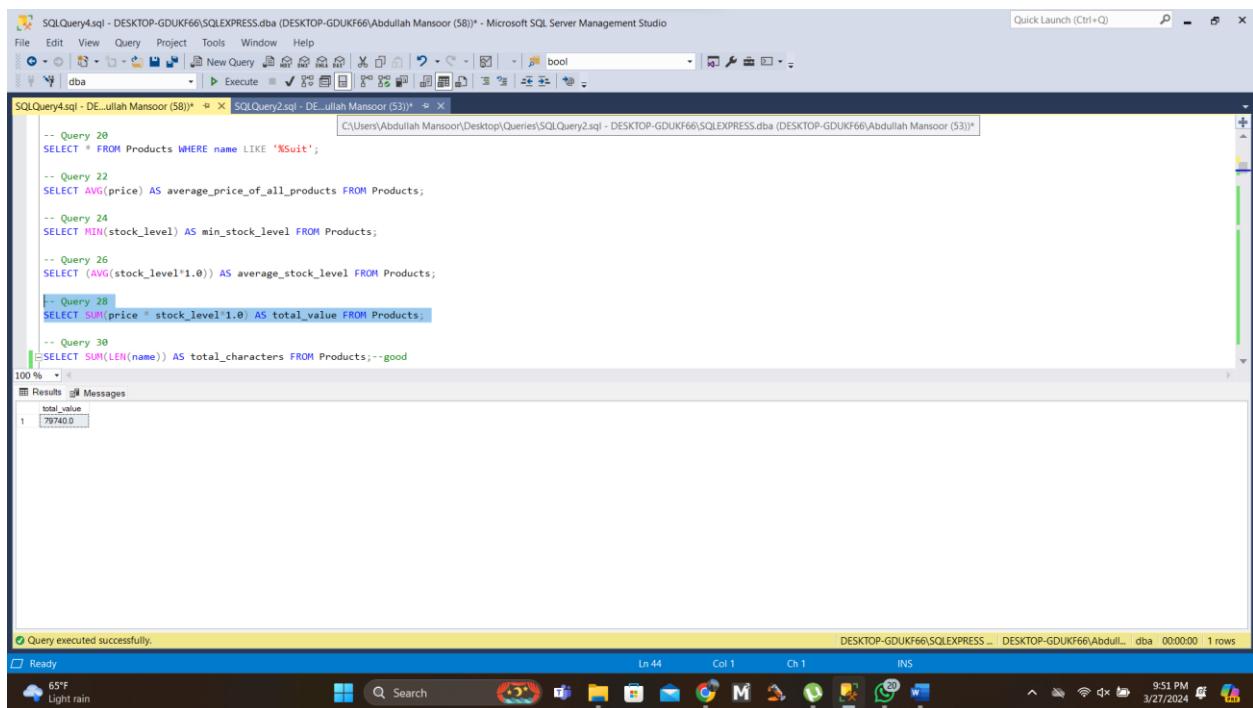
-- Query 24
SELECT MIN(stock_level) AS min_stock_level FROM Products;

-- Query 26
SELECT (AVG(stock_level*1.0)) AS average_stock_level FROM Products;

-- Query 28
SELECT SUM(price * stock_level*1.0) AS total_value FROM Products;

-- Query 30
SELECT SUM(LEN(name)) AS total_characters FROM Products;--good
```

average_stock_level
6850000



```
-- Query 20
SELECT * FROM Products WHERE name LIKE '%Suit';

-- Query 22
SELECT AVG(price) AS average_price_of_all_products FROM Products;

-- Query 24
SELECT MIN(stock_level) AS min_stock_level FROM Products;

-- Query 26
SELECT (AVG(stock_level*1.0)) AS average_stock_level FROM Products;

-- Query 28
SELECT SUM(price * stock_level*1.0) AS total_value FROM Products;

-- Query 30
SELECT SUM(LEN(name)) AS total_characters FROM Products;--good
```

total_value
797420

```
-- Query 22
SELECT AVG(price) AS average_price_of_all_products FROM Products;

-- Query 24
SELECT MIN(stock_level) AS min_stock_level FROM Products;

-- Query 26
SELECT (AVG(stock_level*1.0)) AS average_stock_level FROM Products;

-- Query 28
SELECT SUM(price * stock_level*1.0) AS total_value FROM Products;

-- Query 30
SELECT SUM(LEN(name)) AS total_characters FROM Products;

-- Query 32
SELECT * FROM Products
```

100 %

	total_characters
1	74

Query executed successfully.

```
-- Query 24
SELECT MIN(stock_level) AS min_stock_level FROM Products;

-- Query 26
SELECT (AVG(stock_level*1.0)) AS average_stock_level FROM Products;

-- Query 28
SELECT SUM(price * stock_level*1.0) AS total_value FROM Products;

-- Query 30
SELECT SUM(LEN(name)) AS total_characters FROM Products;

-- Query 32
SELECT * FROM Products
WHERE stock_level = (SELECT MIN(stock_level) FROM Products);

-- Query 34
```

100 %

product_id	name	description	price	stock_level	category_id
1	5	Board Game	Family Board Game	50	25

Query executed successfully.

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

-- Query 28
SELECT SUM(price * stock_level*1.0) AS total_value FROM Products;

-- Query 30
SELECT SUM(LEN(name)) AS total_characters FROM Products;

-- Query 32
SELECT * FROM Products
WHERE stock_level = (SELECT MIN(stock_level) FROM Products);

-- Query 34
SELECT SUM(price*1.0) AS total_price_of_all_products FROM Products;

-- Query 36
SELECT (MAX(price) - MIN(price)) AS price_difference_max_min FROM Products;

-- Query 38

100 %

Results Messages

total_price_of_all_products
1 1872.0

Query executed successfully.

Ready 65°F Light rain Search Col 1 Ch 1 INS 9:53 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

-- Query 28
SELECT SUM(price * stock_level*1.0) AS total_value FROM Products;

-- Query 30
SELECT SUM(LEN(name)) AS total_characters FROM Products;

-- Query 32
SELECT * FROM Products
WHERE stock_level = (SELECT MIN(stock_level) FROM Products);

-- Query 34
SELECT SUM(price*1.0) AS total_price_of_all_products FROM Products;

-- Query 36
SELECT (MAX(price) - MIN(price)) AS price_difference_max_min FROM Products;

-- Query 38

100 %

Results Messages

price_difference_max_min
1 1295

Query executed successfully.

Ready 65°F Light rain Search Col 1 Ch 1 INS 9:53 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```

-- WHERE stock_level = (SELECT MIN(stock_level) FROM Products);
-- Query 34
SELECT SUM(price*1.0) AS total_price_of_all_products FROM Products;

-- Query 36
SELECT (MAX(price) - MIN(price)) AS price_difference_max_min FROM Products;

-- Query 38
SELECT TOP 1 category_id, AVG(price) AS average_price
FROM Products
GROUP BY category_id
ORDER BY average_price DESC;

-- Query 40
SELECT category_id, SUM(price * stock_level) AS total_value
FROM Products
GROUP BY category_id;

```

100% ▾

Results Messages

category_id	average_price
2	1300.00000

Query executed successfully.

Ready 65°F Light rain Search Col 1 Ch 1 INS 9:54 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```

SELECT SUM(price*1.0) AS total_price_of_all_products FROM Products;

-- Query 36
SELECT (MAX(price) - MIN(price)) AS price_difference_max_min FROM Products;

-- Query 38
SELECT TOP 1 category_id, AVG(price) AS average_price
FROM Products
GROUP BY category_id
ORDER BY average_price DESC;

-- Query 40
SELECT category_id, SUM(price * stock_level) AS total_value
FROM Products
GROUP BY category_id;

-- Query 42
SELECT AVG(order_total) AS average_order_total--correct

```

100% ▾

Results Messages

category_id	total_value
1	2000
2	65000
3	1200
4	1440
5	1250
6	300
7	750
8	4800
9	1200
10	1800

Query executed successfully.

Ready 65°F Light rain Search Col 1 Ch 1 INS 9:54 PM 3/27/2024

```
-- Query 40
SELECT category_id, SUM(price * stock_level) AS total_value
FROM Products
GROUP BY category_id;

-- Query 42
SELECT AVG(order_total) AS average_order_total
FROM (
    SELECT o.order_id, SUM(p.price * oi.quantity) AS order_total
    FROM Orders o
    JOIN Order_Items oi ON o.order_id = oi.order_id
    JOIN Products p ON oi.product_id = p.product_id
    GROUP BY o.order_id
) AS order_totals;
```

Results

average_order_total
1885.900000

Query executed successfully.

```
-- Query 42
SELECT AVG(order_total) AS average_order_total
FROM (
    SELECT o.order_id, SUM(p.price * oi.quantity) AS order_total
    FROM Orders o
    JOIN Order_Items oi ON o.order_id = oi.order_id
    JOIN Products p ON oi.product_id = p.product_id
    GROUP BY o.order_id
) AS order_totals;

-- Query 44
SELECT product_id, COUNT(*) AS total_reviews FROM Reviews GROUP BY product_id;

-- Query 46
SELECT c.customer_id,
       AVG(p.price) AS average_price
FROM Customers c
JOIN Reviews r ON c.customer_id = r.customer_id
JOIN Products p ON r.product_id = p.product_id
GROUP BY c.customer_id;
```

Results

product_id	total_reviews
1	2
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Query executed successfully.

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
-- Query 44
SELECT product_id, COUNT(*) AS total_reviews FROM Reviews GROUP BY product_id;

-- Query 46
SELECT c.customer_id, 
       AVG(p.price) AS average_price
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
JOIN Order_Items oi ON o.order_id = oi.order_id
JOIN Products p ON oi.product_id = p.product_id
GROUP BY c.customer_id;

-- Query 48
SELECT customer_id, COUNT(*) AS total_carts FROM Carts GROUP BY customer_id;
```

100% ▾

Results [Messages]

customer_id	average_price
1	20.000000
2	1300.000000
3	16.000000
4	36.000000
5	50.000000
6	10.000000
7	5.000000
8	80.000000
9	15.000000
10	40.000000

Query executed successfully.

Matches: (

65°F Light rain

DESKTOP-GDUKF66\SQLEXPRESS ... DESKTOP-GDUKF66\Abdull... dba 00:00:00 10 rows

9:55 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
JOIN Order_Items oi ON o.order_id = oi.order_id
JOIN Products p ON oi.product_id = p.product_id
GROUP BY c.customer_id;

-- Query 48
SELECT customer_id, COUNT(*) AS total_carts FROM Carts GROUP BY customer_id;

-- Query 50

--SELECT TOP 1 c.customer_id, --Good
--      SUM(p.price * oi.quantity) AS total_spending
--FROM Customers c
--JOIN Orders o ON c.customer_id = o.customer_id
--JOIN Order_Items oi ON o.order_id = oi.order_id
```

100% ▾

Results [Messages]

customer_id	total_carts
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Query executed successfully.

Matches: (

65°F Light rain

DESKTOP-GDUKF66\SQLEXPRESS ... DESKTOP-GDUKF66\Abdull... dba 00:00:00 10 rows

9:55 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
-- Query 48
SELECT customer_id, COUNT(*) AS total_carts FROM Carts GROUP BY customer_id;

-- Query 50
SELECT TOP 1 c.customer_id,
       SUM(p.price * oi.quantity) AS total_spending
  FROM Customers c
  JOIN Orders o ON c.customer_id = o.customer_id
  JOIN Order_Items oi ON o.order_id = oi.order_id
  JOIN Products p ON oi.product_id = p.product_id
 GROUP BY c.customer_id
 ORDER BY total_spending DESC;

-- Query 52
SELECT customer_id,
       AVG(order_total) AS average_order_total
  FROM Orders
 GROUP BY customer_id
 ORDER BY average_order_total DESC;
```

Results

customer_id	total_spending
1	14300

Query executed successfully.

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
-- Query 52
SELECT customer_id,
       AVG(order_total) AS average_order_total
  FROM (
    SELECT o.customer_id, SUM(p.price * oi.quantity) AS order_total
      FROM Orders o
      JOIN Order_Items oi ON o.order_id = oi.order_id
      JOIN Products p ON oi.product_id = p.product_id
     GROUP BY o.order_id, o.customer_id
   ) AS customer_orders
  GROUP BY customer_id;

-- Query 54
SELECT MONTH(order_date) AS month_num, COUNT(*) AS total_orders
  FROM Orders
```

Results

customer_id	average_order_total
1	40.000000
2	14300.000000
3	240.000000
4	144.000000
5	450.000000
6	170.000000
7	30.000000
8	400.000000
9	45.000000
10	40.000000

Query executed successfully.

SQLQuery2.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```

GROUP BY o.order_id, o.customer_id
) AS customer_orders
GROUP BY customer_id;

-- Query 54
SELECT MONTH(order_date) AS month_num, COUNT(*) AS total_orders
FROM Orders
GROUP BY MONTH(order_date);

-- Query 56
SELECT category_id, COUNT(*) AS total_reviews FROM Reviews JOIN Products ON Reviews.product_id = Products.product_id GROUP BY category_id;

-- Query 58
SELECT YEAR(order_date) AS year_num, COUNT(*) AS total_orders
FROM Orders
GROUP BY YEAR(order_date);

```

100% ▾

Results [! Messages]

month_num	total_orders
1	2
2	3
	6

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS ... DESKTOP-GDUKF66\Abdull... dba 00:00:00 2 rows

Matches: (

65°F Light rain

Search Col 1 Ch 1 INS

9:57 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```

GROUP BY o.order_id, o.customer_id
) AS customer_orders
GROUP BY customer_id;

-- Query 54
SELECT MONTH(order_date) AS month_num, COUNT(*) AS total_orders
FROM Orders
GROUP BY MONTH(order_date);

-- Query 56
SELECT category_id, COUNT(*) AS total_reviews FROM Reviews JOIN Products ON Reviews.product_id = Products.product_id GROUP BY category_id;

-- Query 58
SELECT YEAR(order_date) AS year_num, COUNT(*) AS total_orders
FROM Orders
GROUP BY YEAR(order_date);

```

100% ▾

Results [! Messages]

category_id	total_reviews
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS ... DESKTOP-GDUKF66\Abdull... dba 00:00:00 10 rows

Matches: (

64°F Light rain

Search Col 1 Ch 1 INS

9:57 PM 3/27/2024

SQLQuery2.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
-- Query 54
--SELECT MONTH(order_date) AS month_num, COUNT(*) AS total_orders
--FROM Orders
--GROUP BY MONTH(order_date);

-- Query 56
SELECT category_id, COUNT(*) AS total_reviews FROM Reviews JOIN Products ON Reviews.product_id = Products.product_id GROUP BY category_id;

-- Query 58
SELECT YEAR(order_date) AS year_num, COUNT(*) AS total_orders
FROM Orders
GROUP BY YEAR(order_date);

-- Query 60
SELECT o.customer_id, --correct
       SUM(p.price * oi.quantity) AS total_value_purchased
FROM Orders o
JOIN Order_Items oi ON o.order_id = oi.order_id
JOIN Products p ON oi.product_id = p.product_id
GROUP BY o.customer_id;
```

100 %

Results Messages

year_num	total_orders
2024	10

Query executed successfully.

Matches: (

64°F Light rain

DESKTOP-GDUKF66\SQLEXPRESS ... DESKTOP-GDUKF66\Abdull... dba 00:00:00 1 rows

9:58 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
-- SELECT YEAR(order_date) AS year_num, COUNT(*) AS total_orders
-- FROM Orders
-- GROUP BY YEAR(order_date);

-- Query 60
SELECT o.customer_id, --correct
       SUM(p.price * oi.quantity) AS total_value_purchased
FROM Orders o
JOIN Order_Items oi ON o.order_id = oi.order_id
JOIN Products p ON oi.product_id = p.product_id
GROUP BY o.customer_id;

-- Query 62
SELECT category_id, AVG(price*1.0) AS average_price FROM Products GROUP BY category_id;

-- Query 64
```

100 %

Results Messages

customer_id	total_value_purchased
1	40
2	14300
3	240
4	144
5	450
6	170
7	30
8	400
9	45
10	40

Query executed successfully.

Matches: (

64°F Light rain

DESKTOP-GDUKF66\SQLEXPRESS ... DESKTOP-GDUKF66\Abdull... dba 00:00:00 10 rows

9:58 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
--SELECT YEAR(order_date) AS year_num, COUNT(*) AS total_orders
FROM Orders
GROUP BY YEAR(order_date);

-- Query 60
SELECT o.customer_id, --correct
       SUM(p.price * oi.quantity) AS total_value_purchased
FROM Orders o
JOIN Order_Items oi ON o.order_id = oi.order_id
JOIN Products p ON oi.product_id = p.product_id
GROUP BY o.customer_id;

-- Query 62
SELECT category_id, AVG(price*1.0) AS average_price FROM Products GROUP BY category_id;

-- Query 64
```

Results Messages

category_id	average_price
1	20.000000
2	1300.000000
3	16.000000
4	36.000000
5	50.000000
6	10.000000
7	5.000000
8	80.000000
9	15.000000
10	40.000000

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS .. DESKTOP-GDUKF66\Abdull.. dba 00:00:00 10 rows

64°F Light rain Search Col 1 Ch 1 INS 9:58 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```
-- Query 62
SELECT category_id, AVG(price*1.0) AS average_price FROM Products GROUP BY category_id;

-- Query 64
SELECT category_id, AVG(stock_level) AS average_stock_level
FROM
    Products
GROUP BY
    category_id
HAVING
    COUNT(*) > 10;

-- Query 66
SELECT customer_id,
       AVG(order_total) AS average_order_value
FROM (
    SELECT o.customer_id, SUM(p.price * oi.quantity) AS order_total
    FROM Orders o
    JOIN Order_Items oi ON o.order_id = oi.order_id
    JOIN Products p ON oi.product_id = p.product_id
)
```

Results Messages

category_id	average_stock_level
-------------	---------------------

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS .. DESKTOP-GDUKF66\Abdull.. dba 00:00:00 0 rows

64°F Light rain Search Col 1 Ch 1 INS 9:59 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute bool

SQLQuery4.sql - D-E..ullah Mansoor (58)* [SQLQuery2.sql - D-E..ullah Mansoor (53)]

```
HAVING COUNT(*) > 10;

-- Query 66
SELECT customer_id, AVG(order_total) AS average_order_value
FROM (
    SELECT o.customer_id, SUM(p.price * oi.quantity) AS order_total
    FROM Orders o
    JOIN Order_Items oi ON o.order_id = oi.order_id
    JOIN Products p ON oi.product_id = p.product_id
    GROUP BY o.order_id, o.customer_id
) AS customer_orders
GROUP BY customer_id;
```

100 %

Result Messages

customer_id	average_order_value
1	40.000000
2	14300.000000
3	240.000000
4	144.000000
5	450.000000
6	170.000000
7	30.000000
8	400.000000
9	45.000000
10	40.000000

Query executed successfully.

Ready Ln 157 Col 5 Ch 5 INS

64°F Light rain

Search

DESKTOP-GDUKF66\SQLEXPRESS ... DESKTOP-GDUKF66\Abdullah ... dba 00:00:00 10 rows

10:04 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Save All Open All Close All Refresh Back Forward Home

Execute Execute (F5) Stop (Esc) Break (Shift+F5) Run (F8) Run (Shift+F8) Run (Ctrl+F8) Run (Shift+Ctrl+F8) Run (Ctrl+Shift+F8) Run (Shift+Ctrl+Shift+F8)

bool

-- Query 68

```
SELECT
    customer_id,
    total_order_value
FROM
    (
        SELECT
            o.customer_id,
            SUM(p.price * oi.quantity) AS total_order_value
        FROM
            Orders o
        JOIN
            Order_Items oi ON o.order_id = oi.order_id
        JOIN
            Products p ON p.product_id = oi.product_id
        GROUP BY
            o.customer_id
    ) AS customer_orders
WHERE
```

100 %

Results Messages

	customer_id	total_order_value
1	2	14300

Query executed successfully.

Matches: (

LN 206 Col 1 Ch 1 INS

64°F Light rain

Search

10:04 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```

File Edit View Query Project Tools Window Help
New Query Execute
SQLQuery4.sql - DE..ullah Mansoor (58)* SQLQuery2.sql - DE..ullah Mansoor (53)*
Orders o
JOIN Order_Items oi ON o.order_id = oi.order_id
JOIN Products p ON p.product_id = oi.product_id
GROUP BY o.customer_id
) AS max_order_value
);

-- Query 70
SELECT TOP 1 product_id, AVG(rating) AS average_rating FROM Reviews GROUP BY product_id ORDER BY average_rating DESC;

-- Query 72
SELECT category_id, AVG(rating*1.0) AS average_rating FROM Reviews JOIN Products ON Reviews.product_id = Products.product_id GROUP BY category_id;
-- Query 74

```

100% ▾

Results Messages

product_id	average_rating
1	5

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS ... DESKTOP-GDUKF66\Abdull... dba 00:00:00 1 rows

Matches (

64°F Light rain Search Col 1 Ch 1 INS 10:05 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

```

File Edit View Query Project Tools Window Help
New Query Execute
SQLQuery4.sql - DE..ullah Mansoor (58)* SQLQuery2.sql - DE..ullah Mansoor (53)*
Orders o
JOIN Order_Items oi ON o.order_id = oi.order_id
JOIN Products p ON p.product_id = oi.product_id
GROUP BY o.customer_id
) AS max_order_value
);

-- Query 70
SELECT TOP 1 product_id, AVG(rating) AS average_rating FROM Reviews GROUP BY product_id ORDER BY average_rating DESC;
-- Query 72
SELECT category_id, AVG(rating*1.0) AS average_rating FROM Reviews JOIN Products ON Reviews.product_id = Products.product_id GROUP BY category_id;
-- Query 74

```

100% ▾

Results Messages

category_id	average_rating
1	5.000000
2	4.000000
3	3.000000
4	2.000000
5	1.000000
6	5.000000
7	4.000000
8	3.000000
9	2.000000
10	1.000000

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS ... DESKTOP-GDUKF66\Abdull... dba 00:00:00 10 rows

Matches (

64°F Light rain Search Col 1 Ch 1 INS 10:05 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query New Item Task List Object Explorer Solution Explorer Properties Task List

Execute

Quick Launch (Ctrl+Q)

SQLQuery4.sql - DE..ullah Mansoor (58)* SQLQuery2.sql - DE..ullah Mansoor (53)*

```
-- Query 74
SELECT
    o.customer_id,
    SUM(p.price * oi.quantity) AS Highest_Total_Value
FROM
    Orders o
JOIN
    Order_Items oi ON o.order_id = oi.order_id
JOIN
    Products p ON oi.product_id = p.product_id
GROUP BY
    o.customer_id
HAVING
    SUM(p.price * oi.quantity) > (
        SELECT
            MAX(total_order_value)
        FROM (
            SELECT
                customer_id,
                SUM(price * quantity) AS total_order_value
            FROM
                Order_Items
            GROUP BY
                customer_id
        ) AS subquery
    )
```

100% □

Results Messages

customer_id	Highest_Total_Value
2	14300

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS .. DESKTOP-GDUKF66\Abdull.. dba 00:00:00 1 rows

Matches (

64°F Light rain Search

Ln 243 Col 1 Ch 1 INS

10:05 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query New Item Task List Object Explorer Solution Explorer Properties Task List

Execute

Quick Launch (Ctrl+Q)

SQLQuery4.sql - DE..ullah Mansoor (58)* SQLQuery2.sql - DE..ullah Mansoor (53)*

```
-- Query 76
SELECT TOP 1
    c.name AS category_name,
    SUM(p.price * oi.quantity) AS total_order_value
FROM
    Order_Items oi
JOIN
    Products p ON oi.product_id = p.product_id
JOIN
    Categories c ON p.category_id = c.category_id
GROUP BY
    c.name
ORDER BY
    total_order_value DESC;
```

```
-- Query 78
SELECT
    category_name,
    total_order_value
FROM
    Order_Items
GROUP BY
    category_name
ORDER BY
    total_order_value DESC;
```

100% □

Results Messages

category_name	total_order_value
Electronics	14300

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS .. DESKTOP-GDUKF66\Abdull.. dba 00:00:00 1 rows

Matches (

64°F Light rain Search

Ln 258 Col 1 Ch 1 INS

10:06 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q) P

SQLQuery4.sql - DE..ullah Mansoor (58)* SQLQuery2.sql - DE..ullah Mansoor (53)*

```
ORDER BY
    total_order_value DESC;

-- Query 78
SELECT
    p.name AS product_name,
    SUM(oi.quantity) AS total_order_quantity
FROM
    Order_Items oi
JOIN
    Products p ON oi.product_id = p.product_id
GROUP BY
    p.name
HAVING
    SUM(oi.quantity) = (
        SELECT
            MAX(total_quantity)
    )
```

100% 1 Results Messages

product_name	total_order_quantity
Coffee	17

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS .. DESKTOP-GDUKF66\Abdull.. dba 00:00:00 1 rows

Ready 64°F Light rain Search Col 1 Ch 1 INS 10:07 PM 3/27/2024

SQLQuery4.sql - DESKTOP-GDUKF66\SQLEXPRESS.dba (DESKTOP-GDUKF66\Abdullah Mansoor (58)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q) P

SQLQuery4.sql - DE..ullah Mansoor (58)* SQLQuery2.sql - DE..ullah Mansoor (53)*

```
-- Query 80
SELECT
    c.name AS category_name,
    AVG(p.stock_level) AS average_stock_level
FROM
    Products p
JOIN
    Categories c ON p.category_id = c.category_id
GROUP BY
    c.name
HAVING
    AVG(stock_level) = (
        SELECT
            MAX(avg_stock_level)
        FROM
            (SELECT
                AVG(stock_level) AS avg_stock_level)
```

100% 1 Results Messages

category_name	average_stock_level
Stationery	150

Query executed successfully.

DESKTOP-GDUKF66\SQLEXPRESS .. DESKTOP-GDUKF66\Abdull.. dba 00:00:00 1 rows

Ready 64°F Light rain Search Col 1 Ch 1 INS 10:07 PM 3/27/2024

3 Odd Queries

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

New Query Execute

DB Assignment

Object Explorer

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (60))

-- Query 1
SELECT * FROM Products;

-- Query 3
SELECT * FROM Products WHERE price > 1000;

-- Query 5
SELECT * FROM Products ORDER BY price DESC;

-- Query 7
SELECT * FROM Products WHERE name LIKE 'P%';

-- Query 9
SELECT * FROM Products ORDER BY name ASC;

-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;

Results Messages

product_id	name	description	price	stock_level	category_id
1	T-Shirt	Cotton T-Shirt For Adults	20	100	1
2	Laptop	Laptop with RTX 3060	1300	50	2
3	Lipstick	Red Colour Lipstick	16	75	3
4	Basketball	Outdoor Basketball	36	40	4
5	Board Game	Fairytale Board Game	50	25	5
6	Coffee	Cappuccino Coffee	10	30	6
7	Pens	Blue Ink Pens	5	150	7
8	Gran Turismo 7	PSS Game	80	60	8
9	Novel	Adventure Novel	15	80	9
10	Pizza	Extra Large Pizza	40	45	10

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) | WIKI\wiki8 (60) | DB_Assignment | 00:00:00 | 10 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

New Query Execute

DB Assignment

Object Explorer

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (60))

-- Query 1
SELECT * FROM Products;

-- Query 3
SELECT * FROM Products WHERE price > 1000;

-- Query 5
SELECT * FROM Products ORDER BY price DESC;

-- Query 7
SELECT * FROM Products WHERE name LIKE 'P%';

-- Query 9
SELECT * FROM Products ORDER BY name ASC;

-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;

Results Messages

product_id	name	description	price	stock_level	category_id
2	Laptop	Laptop with RTX 3060	1300	50	2

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) | WIKI\wiki8 (60) | DB_Assignment | 00:00:00 | 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute customer_id

DB Assignment

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8 (61))

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

-- Query 1
SELECT * FROM Products;

-- Query 3
SELECT * FROM Products WHERE price > 1000;

-- Query 5
SELECT * FROM Products ORDER BY price DESC;

-- Query 7
SELECT * FROM Products WHERE name LIKE 'PS';

-- Query 9
SELECT * FROM Products ORDER BY name ASC;

-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;

Results Messages

product_id	name	description	price	stock_level	category_id
2	Laptop	Laptop with RTX 3060	1300	50	2
8	Gran Turismo 7	PlayStation 5 Game	80	60	8
5	Board Game	Family Board Game	50	25	5
10	Pizza	Extra Large Pizza	40	45	10
4	Basketball	Outdoor Basketball	36	40	4
1	T-Shirt	Cotton T-Shirt For Adults	20	100	1
3	Lipstick	Red Colour Lipstick	16	75	3
9	Novel	Adventure Novel	15	80	9
6	Coffee	Cappuccino Coffee	10	30	6
7	Pen	Blue Ink Pen	5	150	7

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 10 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute customer_id

DB Assignment

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8 (61))

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

-- Query 1
SELECT * FROM Products;

-- Query 3
SELECT * FROM Products WHERE price > 1000;

-- Query 5
SELECT * FROM Products ORDER BY price DESC;

-- Query 7
SELECT * FROM Products WHERE name LIKE 'PS';

-- Query 9
SELECT * FROM Products ORDER BY name ASC;

-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;

Results Messages

product_id	name	description	price	stock_level	category_id
7	Pen	Blue Ink Pen	5	150	7
10	Pizza	Extra Large Pizza	40	45	10

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 2 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

```
-- Query 1
SELECT * FROM Products;

-- Query 3
SELECT * FROM Products WHERE price > 1000;

-- Query 5
SELECT * FROM Products ORDER BY price DESC;

-- Query 7
SELECT * FROM Products WHERE name LIKE 'P%';

-- Query 9
SELECT * FROM Products ORDER BY name ASC;

-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;
```

Results Messages

product_id	name	description	price	stock_level	category_id	
1	4	Basketball	Outdoor Basketball	36	40	4
2	5	Board Game	Family Board Game	50	25	5
3	6	Coffee	Ground Coffee	10	50	6
4	8	Gran Turismo 7	PSS Game	80	60	8
5	2	Laptop	Laptop with RTX 3060	1300	50	2
6	3	Lipstick	Red Colour Lipstick	16	75	3
7	9	Novel	Adventure Novel	15	80	9
8	7	Pen	Blue Ink Pen	5	150	7
9	10	Pizza	Extra Large Pizza	40	45	10
10	1	T-Shirt	Cotton T-Shirt For Adults	20	100	1

Query executed successfully.

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

```
-- Query 1
SELECT * FROM Products;

-- Query 3
SELECT * FROM Products WHERE price > 1000;

-- Query 5
SELECT * FROM Products ORDER BY price DESC;

-- Query 7
SELECT * FROM Products WHERE name LIKE 'P%';

-- Query 9
SELECT * FROM Products ORDER BY name ASC;

-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;
```

Results Messages

product_id	name	description	price	stock_level	category_id	
1	2	Laptop	Laptop with RTX 3060	1300	50	2

Query executed successfully.

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

```
-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;

-- Query 13
SELECT * FROM Products WHERE name LIKE 'MS%' AND price > 5000;

-- Query 15
SELECT * FROM Products ORDER BY stock_level DESC;

-- Query 17
SELECT * FROM Products WHERE price > (SELECT AVG(price) FROM Products);

-- Query 19
SELECT * FROM Products WHERE category_id IN (1, 3, 4);

-- Query 21
SELECT COUNT(*) AS total_products FROM Products;
```

Results Messages

product_id	name	description	price	stock_level	category_id

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 0 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

```
-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;

-- Query 13
SELECT * FROM Products WHERE name LIKE 'MS%' AND price > 5000;

-- Query 15
SELECT * FROM Products ORDER BY stock_level DESC;

-- Query 17
SELECT * FROM Products WHERE price > (SELECT AVG(price) FROM Products);

-- Query 19
SELECT * FROM Products WHERE category_id IN (1, 3, 4);

-- Query 21
SELECT COUNT(*) AS total_products FROM Products;
```

Results Messages

product_id	name	description	price	stock_level	category_id
1	Blue Pen	Blue Ink Pen	5	100	7
2	T-Shirt	Cotton T-Shirt For Adults	20	100	1
3	Novel	Adventure Novel	15	80	9
4	Lipstick	Red Colour Lipstick	16	75	3
5	Gran Turismo 7	PSS Game	80	60	8
6	Laptop	Laptop with RTX 3060	1300	50	2
7	Pizza	Extra Large Pizza	40	45	10
8	Basketball	Outdoor Basketball	30	40	4
9	Coffee	Cappuccino Coffee	10	30	6
10	Board Game	Family Board Game	50	25	5

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 10 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

```
-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;

-- Query 13
SELECT * FROM Products WHERE name LIKE 'MS%' AND price > 5000;

-- Query 15
SELECT * FROM Products ORDER BY stock_level DESC;

-- Query 17
SELECT AVG(price*1.0) FROM Products;
SELECT * FROM Products WHERE price > (SELECT AVG(price) FROM Products);

-- Query 19
SELECT * FROM Products WHERE category_id IN (1, 3, 4);

-- Query 21
SELECT COUNT(*) AS total_products FROM Products;
```

Results Messages

product_id	name	description	price	stock_level	category_id
1	Laptop	Laptop with RTX 3060	1300	50	2

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

```
-- Query 11
SELECT * FROM Products WHERE price <= 3000 AND category_id = 2;

-- Query 13
SELECT * FROM Products WHERE name LIKE 'MS%' AND price > 5000;

-- Query 15
SELECT * FROM Products ORDER BY stock_level DESC;

-- Query 17
SELECT AVG(price*1.0) FROM Products;
SELECT * FROM Products WHERE price > (SELECT AVG(price) FROM Products);

-- Query 19
SELECT * FROM Products WHERE category_id IN (1, 3, 4);

-- Query 21
SELECT COUNT(*) AS total_products FROM Products;
```

Results Messages

product_id	name	description	price	stock_level	category_id
1	Colour T-Shirt	Colour T-Shirt For Adults	20	100	1
2	Lipstick	Red Colour Lipstick	18	75	3
3	Basketball	Outdoor Basketball	98	40	4

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 3 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8 (61))

Creation.sql - WIKI\...nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\...t (WIKI\wiki8 (60))

```
-- Query 21
SELECT COUNT(*) AS total_products FROM Products;

-- Query 23
SELECT MAX(price) AS max_price FROM Products;

-- Query 25
SELECT SUM(stock_level) AS total_stock_levels_of_all_products FROM Products;

-- Query 27
SELECT category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id;

-- Query 29
SELECT * FROM Products WHERE price = (SELECT MAX(price) FROM Products);

-- Query 31
SELECT category_id, AVG(price) AS average_price FROM Products GROUP BY category_id;
```

Results Messages

total_products
10

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Item(s) Saved

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8 (61))

Creation.sql - WIKI\...nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\...t (WIKI\wiki8 (60))

```
-- Query 21
SELECT COUNT(*) AS total_products FROM Products;

-- Query 23
SELECT MAX(price) AS max_price FROM Products;

-- Query 25
SELECT SUM(stock_level) AS total_stock_levels_of_all_products FROM Products;

-- Query 27
SELECT category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id;

-- Query 29
SELECT * FROM Products WHERE price = (SELECT MAX(price) FROM Products);

-- Query 31
SELECT category_id, AVG(price) AS average_price FROM Products GROUP BY category_id;
```

Results Messages

max_price
1300

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8 (60))

Databases Security Server Objects Replication Management XEvent Profiler

Creation.sql - WIKI\...nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI...t (WIKI\wiki8 (60))

```
-- Query 21
SELECT COUNT(*) AS total_products FROM Products;

-- Query 23
SELECT MAX(price) AS max_price FROM Products;

-- Query 25
SELECT SUM(stock_level) AS total_stock_levels_of_all_products FROM Products;

-- Query 27
SELECT category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id;

-- Query 29
SELECT * FROM Products WHERE price = (SELECT MAX(price) FROM Products);

-- Query 31
SELECT category_id, AVG(price) AS average_price FROM Products GROUP BY category_id;
```

100 %

Results Messages

total_stock_levels_of_all_products
665

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8 (60))

Databases Security Server Objects Replication Management XEvent Profiler

Creation.sql - WIKI\...nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI...t (WIKI\wiki8 (60))

```
-- Query 21
SELECT COUNT(*) AS total_products FROM Products;

-- Query 23
SELECT MAX(price) AS max_price FROM Products;

-- Query 25
SELECT SUM(stock_level) AS total_stock_levels_of_all_products FROM Products;

-- Query 27
SELECT category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id;

-- Query 29
SELECT * FROM Products WHERE price = (SELECT MAX(price) FROM Products);

-- Query 31
SELECT category_id, AVG(price) AS average_price FROM Products GROUP BY category_id;
```

100 %

Results Messages

category_id	total_products
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 10 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute

Object Explorer

Creation.sql - WIKI\nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\ (WIKI\wiki8 (60))

```
-- Query 21
SELECT COUNT(*) AS total_products FROM Products;

-- Query 23
SELECT MAX(price) AS max_price FROM Products;

-- Query 25
SELECT SUM(stock_level) AS total_stock_levels_of_all_products FROM Products;

-- Query 27
SELECT category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id;

-- Query 29
SELECT * FROM Products WHERE price <= (SELECT MAX(price) FROM Products);

-- Query 31
SELECT category_id, AVG(price) AS average_price FROM Products GROUP BY category_id;
```

Results Messages

product_id	name	description	price	stock_level	category_id	
1	2	Laptop	Laptop with RTX 3060	1300	50	2

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute

Object Explorer

Creation.sql - WIKI\nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\ (WIKI\wiki8 (60))

```
-- Query 31
SELECT * FROM Products WHERE price = (SELECT MAX(price) FROM Products);

-- Query 32
SELECT category_id, AVG(price)^1.0 AS average_price FROM Products GROUP BY category_id;

-- Query 33
SELECT COUNT(*) AS total_products_in_stock FROM Products WHERE stock_level > 0;

-- Query 35
SELECT TOP 1 category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id ORDER BY total_products DESC;

-- Query 37
SELECT * FROM Products WHERE stock_level = (SELECT MAX(stock_level) FROM Products);

-- Query 39
SELECT COUNT(*) AS total_products_greater_than_5000 FROM Products WHERE price > 5000;

-- Query 41
```

Results Messages

category_id	average_price
1	1300.000000
2	1300.000000
3	18.000000
4	38.000000
5	50.000000
6	10.000000
7	5.000000
8	80.000000
9	15.000000
10	40.000000

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 10 rows

Item(s) Saved.

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

```

SELECT * FROM Products WHERE price = (SELECT MAX(price) FROM Products);

-- Query 31
SELECT category_id, AVG(price*1.0) AS average_price FROM Products GROUP BY category_id;

-- Query 33
SELECT COUNT(*) AS total_products_in_stock FROM Products WHERE stock_level > 0;

-- Query 35
SELECT TOP 1 category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id ORDER BY total_products DESC;

-- Query 37
SELECT * FROM Products WHERE stock_level = (SELECT MAX(stock_level) FROM Products);

-- Query 39
SELECT COUNT(*) AS total_products_greater_than_5000 FROM Products WHERE price > 5000;

-- Query 41

```

Results

total_products_in_stock
10

Messages

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Item(s) Saved

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

```

SELECT * FROM Products WHERE price = (SELECT MAX(price) FROM Products);

-- Query 31
SELECT category_id, AVG(price*1.0) AS average_price FROM Products GROUP BY category_id;

-- Query 33
SELECT COUNT(*) AS total_products_in_stock FROM Products WHERE stock_level > 0;

-- Query 35
SELECT TOP 1 category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id ORDER BY total_products DESC;

-- Query 37
SELECT * FROM Products WHERE stock_level = (SELECT MAX(stock_level) FROM Products);

-- Query 39
SELECT COUNT(*) AS total_products_greater_than_5000 FROM Products WHERE price > 5000;

-- Query 41

```

Results

category_id	total_products
2	1

Messages

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Item(s) Saved

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

```

SELECT * FROM Products WHERE price = (SELECT MAX(price) FROM Products);

-- Query 31
SELECT category_id, AVG(price*1.0) AS average_price FROM Products GROUP BY category_id;

-- Query 33
SELECT COUNT(*) AS total_products_in_stock FROM Products WHERE stock_level > 0;

-- Query 35
SELECT TOP 1 category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id ORDER BY total_products DESC;

-- Query 37
SELECT * FROM Products WHERE stock_level <= (SELECT MAX(stock_level) FROM Products);

-- Query 39
SELECT COUNT(*) AS total_products_greater_than_5000 FROM Products WHERE price > 5000;

-- Query 41

```

Results Messages

product_id	name	description	price	stock_level	category_id	
1	7	Pen	Blue Ink Pen	5	150	7

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Item(s) Saved

SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (61))

File Edit View Query Project Tools Window Help

DB Assignment Execute

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Creation.sql - WIKI\nt (WIKI\wiki8 (61))

```

SELECT * FROM Products WHERE price = (SELECT MAX(price) FROM Products);

-- Query 31
SELECT category_id, AVG(price*1.0) AS average_price FROM Products GROUP BY category_id;

-- Query 33
SELECT COUNT(*) AS total_products_in_stock FROM Products WHERE stock_level > 0;

-- Query 35
SELECT TOP 1 category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id ORDER BY total_products DESC;

-- Query 37
SELECT * FROM Products WHERE stock_level <= (SELECT MAX(stock_level) FROM Products);

-- Query 39
SELECT COUNT(*) AS total_products_greater_than_5000 FROM Products WHERE price > 5000;

-- Query 41

```

Results Messages

total_products_greater_than_5000	
1	0

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Item(s) Saved

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute Quick Launch (Ctrl+Q)

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

customer_id

Insertion.sql - WIKI\nt (WIKI\wiki8 (63)) Creation.sql - WIKI\nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (60))

```
-- Query 41
SELECT COUNT(*) AS total_orders FROM Orders;

-- Query 43
SELECT TOP 1 customer_id, COUNT(*) AS total_orders FROM Orders GROUP BY customer_id ORDER BY total_orders DESC;

-- Query 45
SELECT category_id, SUM(price * quantity) AS total_revenue FROM Order_Items JOIN Products ON Order_Items.product_id = Products.product_id GROUP BY category_id;
SELECT p.category_id, SUM(p.price * oi.quantity) AS total_revenue FROM Order_Items oi JOIN Products p ON oi.product_id = p.product_id GROUP BY p.category_id;

-- Query 47
SELECT
    pr.promotion_id,
    pr.name AS promotion_name,
    SUM((p.price * (pr.discount_value))) AS total_discounted_value,
    SUM((p.price - (p.price * pr.discount_value))) AS new_discounted_price
FROM
    pr.promotion_id,
    pr.name AS promotion_name,
    SUM((p.price * (pr.discount_value))) AS total_discounted_value,
    SUM((p.price - (p.price * pr.discount_value))) AS new_discounted_price
FROM
```

Results Messages

total_orders
10

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute Quick Launch (Ctrl+Q)

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

customer_id

Insertion.sql - WIKI\nt (WIKI\wiki8 (63)) Creation.sql - WIKI\nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (60))

```
-- Query 41
SELECT COUNT(*) AS total_orders FROM Orders;

-- Query 43
SELECT TOP 1 customer_id, COUNT(*) AS total_orders FROM Orders GROUP BY customer_id ORDER BY total_orders DESC;

-- Query 45
SELECT category_id, SUM(price * quantity) AS total_revenue FROM Order_Items JOIN Products ON Order_Items.product_id = Products.product_id GROUP BY category_id;
SELECT p.category_id, SUM(p.price * oi.quantity) AS total_revenue FROM Order_Items oi JOIN Products p ON oi.product_id = p.product_id GROUP BY p.category_id;

-- Query 47
SELECT
    pr.promotion_id,
    pr.name AS promotion_name,
    SUM((p.price * (pr.discount_value))) AS total_discounted_value,
    SUM((p.price - (p.price * pr.discount_value))) AS new_discounted_price
FROM
    pr.promotion_id,
    pr.name AS promotion_name,
    SUM((p.price * (pr.discount_value))) AS total_discounted_value,
    SUM((p.price - (p.price * pr.discount_value))) AS new_discounted_price
FROM
```

Results Messages

customer_id	total_orders
1	1

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

```
-- Query 41
SELECT COUNT(*) AS total_orders FROM Orders;

-- Query 43
SELECT TOP 1 customer_id, COUNT(*) AS total_orders FROM Orders GROUP BY customer_id ORDER BY total_orders DESC;

-- Query 45
SELECT category_id, SUM(price * quantity) AS total_revenue FROM Order_Items JOIN Products ON Order_Items.product_id = Products.product_id GROUP BY category_id;
SELECT p.category_id, SUM(p.price * oi.quantity) AS total_revenue FROM Order_Items oi JOIN Products p ON oi.product_id = p.product_id GROUP BY p.category_id;

-- Query 47
SELECT
    pr.promotion_id,
    pr.name AS promotion_name,
    SUM((p.price * pr.discount_value)) AS total_discounted_value,
    SUM((p.price - (p.price * pr.discount_value))) AS new_discounted_price
FROM
    Promotions pr
    INNER JOIN Product_Promotions pp ON pr.promotion_id = pp.promotion_id
    INNER JOIN Products p ON pp.product_id = p.product_id
GROUP BY
    pr.promotion_id, pr.name;
```

Results

category_id	total_revenue
1	40
2	14300
3	240
4	144
5	450
6	170
7	30
8	400
9	45
10	40

Query executed successfully.

SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (63)) - Microsoft SQL Server Management Studio

```
-- Query 47
SELECT
    pr.promotion_id,
    pr.name AS promotion_name,
    SUM((p.price * pr.discount_value)) AS total_discounted_value,
    SUM((p.price - (p.price * pr.discount_value))) AS new_discounted_price
FROM
    Promotions pr
    INNER JOIN Product_Promotions pp ON pr.promotion_id = pp.promotion_id
    INNER JOIN Products p ON pp.product_id = p.product_id
GROUP BY
    pr.promotion_id, pr.name;
```

Results

promotion_id	promotion_name	total_discounted_value	new_discounted_price
1	Summer Sale	4.000	16.000
2	Summer Sale	130.000	1170.000
3	Winter Sale	4.800	11.200
4	Back to School Sale	5.400	30.600
5	Holiday Sale	12.500	37.500
6	New Year Sale	5.000	5.000
7	Valentine Sale	0.250	4.750
8	Eid Sale	32.000	48.000
9	Independence Day Sale	1.500	13.500
10	Black Friday Sale	30.000	10.000

Query executed successfully.

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute New Query customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Insertion.sql - WIKI..nt (WIKI\wiki8 (63)) Creation.sql - WIKI..nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI..t (WIKI\wiki8 (60))

```
-- Query 49
SELECT order_id, SUM(quantity) AS total_items FROM Order_Items GROUP BY order_id;

-- Query 51
SELECT customer_id, COUNT(*) AS total_reviews FROM Reviews GROUP BY customer_id;

-- Query 53
SELECT TOP 1 category_id, AVG(price) AS average_price FROM Products GROUP BY category_id ORDER BY average_price DESC;

-- Query 55
SELECT
    o.customer_id,
    c.first_name,
    c.last_name,
    SUM(oi.quantity * p.price) AS total_revenue
FROM
    Orders o
    JOIN Order_Items oi ON o.order_id = oi.order_id
    JOIN Products p ON oi.product_id = p.product_id
    JOIN Customers c ON o.customer_id = c.customer_id
GROUP BY
    o.customer_id,
    c.first_name,
    c.last_name;
```

Results Messages

order_id	total_items
1	2
2	11
3	15
4	4
5	9
6	17
7	6
8	5
9	3
10	1

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 10 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute New Query customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Insertion.sql - WIKI..nt (WIKI\wiki8 (63)) Creation.sql - WIKI..nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI..t (WIKI\wiki8 (60))

```
-- Query 49
SELECT order_id, SUM(quantity) AS total_items FROM Order_Items GROUP BY order_id;

-- Query 51
SELECT customer_id, COUNT(*) AS total_reviews FROM Reviews GROUP BY customer_id;

-- Query 53
SELECT TOP 1 category_id, AVG(price) AS average_price FROM Products GROUP BY category_id ORDER BY average_price DESC;

-- Query 55
SELECT
    o.customer_id,
    c.first_name,
    c.last_name,
    SUM(oi.quantity * p.price) AS total_revenue
FROM
    Orders o
    JOIN Order_Items oi ON o.order_id = oi.order_id
    JOIN Products p ON oi.product_id = p.product_id
    JOIN Customers c ON o.customer_id = c.customer_id
GROUP BY
    o.customer_id,
    c.first_name,
    c.last_name;
```

Results Messages

customer_id	total_reviews
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 10 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute New Query customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Insertion.sql - WIKI\nt (WIKI\wiki8 (63)) Creation.sql - WIKI\nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (60))

```
-- Query 49
SELECT order_id, SUM(quantity) AS total_items FROM Order_Items GROUP BY order_id;

-- Query 51
SELECT customer_id, COUNT(*) AS total_reviews FROM Reviews GROUP BY customer_id;

-- Query 53
SELECT TOP 1 category_id, AVG(price) AS average_price FROM Products GROUP BY category_id ORDER BY average_price DESC;
```

-- Query 55

```
SELECT
    o.customer_id,
    c.first_name,
    c.last_name,
    SUM(oi.quantity * p.price) AS total_revenue
FROM
    Orders o
    INNER JOIN
        Order_Items oi ON o.order_id = oi.order_id
    INNER JOIN
        Products p ON oi.product_id = p.product_id
    INNER JOIN
        Customers c ON o.customer_id = c.customer_id
GROUP BY
    o.customer_id, c.first_name, c.last_name;
```

Results Messages

category_id	average_price
1	1300.00000

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute New Query customer_id

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Insertion.sql - WIKI\nt (WIKI\wiki8 (63)) Creation.sql - WIKI\nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (60))

```
-- Query 55
SELECT
    o.customer_id,
    c.first_name,
    c.last_name,
    SUM(oi.quantity * p.price) AS total_revenue
FROM
    Orders o
    INNER JOIN
        Order_Items oi ON o.order_id = oi.order_id
    INNER JOIN
        Products p ON oi.product_id = p.product_id
    INNER JOIN
        Customers c ON o.customer_id = c.customer_id
GROUP BY
    o.customer_id, c.first_name, c.last_name;
```

Results Messages

customer_id	first_name	last_name	total_revenue
1	Muhammad	Siddiqui	40
2	Ahmed	Ali	1400
3	Fatima	Khan	240
4	Muhammad	Ahmed	144
5	Ayesha	Siddiqui	450
6	Hassan	Khan	170
7	Zainab	Malik	30
8	Fatima	Raza	400
9	Bilal	Khan	45
10	Sana	Ahmed	40

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 10 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute customer_id

DB Assignment

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Insertionsql - WIKI_nt (WIKI\wiki8 (63)) Creationsql - WIKI_nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI_t (WIKI\wiki8 (60))

```

INNER JOIN Order_Items oi ON o.order_id = oi.order_id
INNER JOIN Products p ON oi.product_id = p.product_id
INNER JOIN Customers c ON o.customer_id = c.customer_id
GROUP BY o.customer_id, c.first_name, c.last_name;

-- Query 57
-- SELECT TOP 1
    o.customer_id,
    c.first_name,
    c.last_name,
    SUM(oi.quantity * p.price) AS total_order_value

```

100 %

Results Messages

product_id	average_rating
1	5

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute customer_id

DB Assignment

Object Explorer

WIKI\SQLEXPRESS (SQL Server 16.0.1000 - WIKI\wiki8)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

Insertionsql - WIKI_nt (WIKI\wiki8 (63)) Creationsql - WIKI_nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI_t (WIKI\wiki8 (60))

```

-- Query 59
-- SELECT TOP 1
    o.customer_id,
    c.first_name,
    c.last_name,
    SUM(oi.quantity * p.price) AS total_order_value
FROM Orders o
INNER JOIN Order_Items oi ON o.order_id = oi.order_id
INNER JOIN Products p ON oi.product_id = p.product_id
INNER JOIN Customers c ON o.customer_id = c.customer_id
GROUP BY o.customer_id, c.first_name, c.last_name
ORDER BY total_order_value DESC;

```

100 %

Results Messages

customer_id	first_name	last_name	total_order_value
1	Z	Ahmed	14300

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute SQL Object Explorer

Insertionsql - WIKI..nt (WIKI\wiki8 (63)) Creationsql - WIKI..nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI..t (WIKI\wiki8 (60))

```

    GROUP BY
        o.customer_id = c.customer_id
    ORDER BY
        total_order_value DESC;

-- Query 61
SELECT category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id;

-- Query 63
SELECT TOP 1 category_id, SUM(price * stock_level) AS total_value FROM Products GROUP BY category_id ORDER BY total_value DESC;

-- Query 65
SELECT customer_id, COUNT(*) AS total_orders FROM Orders GROUP BY customer_id;

-- Query 67
SELECT
    c.city,
    c.state

```

Results Messages

category_id	total_products
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 10 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute SQL Object Explorer

Insertionsql - WIKI..nt (WIKI\wiki8 (63)) Creationsql - WIKI..nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI..t (WIKI\wiki8 (60))

```

    GROUP BY
        o.customer_id = c.customer_id
    ORDER BY
        total_order_value DESC;

-- Query 61
SELECT category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id;

-- Query 63
SELECT TOP 1 category_id, SUM(price * stock_level) AS total_value FROM Products GROUP BY category_id ORDER BY total_value DESC;

-- Query 65
SELECT customer_id, COUNT(*) AS total_orders FROM Orders GROUP BY customer_id;

-- Query 67
SELECT
    c.city,
    c.state

```

Results Messages

category_id	total_value
1	65000

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

```

-- Query 61
SELECT category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id;

-- Query 63
SELECT TOP 1 category_id, SUM(price * stock_level) AS total_value FROM Products GROUP BY category_id ORDER BY total_value DESC;

-- Query 65
SELECT customer_id, COUNT(*) AS total_orders FROM Orders GROUP BY customer_id;

-- Query 67
SELECT
    c.city,
    COUNT(o.order_id) AS total_orders
FROM
    Customers c
JOIN
    Orders o ON c.customer_id = o.customer_id
GROUP BY
    c.city;

```

Results

customer_id	total_orders
1	1
2	2
3	3
4	1
5	1
6	6
7	1
8	8
9	9
10	1

Query executed successfully.

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

```

-- Query 65
SELECT customer_id, COUNT(*) AS total_orders FROM Orders GROUP BY customer_id;

-- Query 67
SELECT
    c.city,
    SUM(p.stock_level) AS total_products_in_stock
FROM
    Customers c
JOIN
    Carts ct ON c.customer_id = ct.customer_id
JOIN
    Cart_Items ci ON ct.cart_id = ci.cart_id
JOIN
    Products p ON ci.product_id = p.product_id
GROUP BY
    c.city;

```

Results

city	total_products_in_stock
Islamabad	225
Karachi	245
Lahore	185

Query executed successfully.

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

```

-- Query 69
SELECT TOP 1 category_id, AVG(price) AS average_price FROM Products GROUP BY category_id ORDER BY average_price DESC;

-- Query 71
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    COUNT(o.product_id) AS total_products_purchased
FROM
    Customers c
LEFT JOIN
    Orders o ON c.customer_id = o.customer_id
GROUP BY
    c.customer_id, c.first_name, c.last_name
ORDER BY
    total_products_purchased DESC;

```

Results

category_id	average_price
2	1300.00000

Query executed successfully.

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

```

-- Query 71
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    COUNT(o.product_id) AS total_products_purchased
FROM
    Customers c
LEFT JOIN
    Orders o ON c.customer_id = o.customer_id
LEFT JOIN
    Order_Items oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_id, c.first_name, c.last_name
ORDER BY
    total_products_purchased DESC;

```

Results

customer_id	first_name	last_name	total_products_purchased
1	Muhammad	Khan	1
2	Ahmed	Ali	1
3	Fatima	Khan	1
4	Muhammad	Ahmed	1
5	Ayesha	Siddiqui	1
6	Hassan	Khan	1
7	Zainab	Malik	1
8	Fatima	Raza	1
9	Bilal	Khan	1
10	Sana	Ahmed	1

Query executed successfully.

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

```

FROM [Customers] c
LEFT JOIN [Orders] o ON c.customer_id = o.customer_id
LEFT JOIN [Order_Items] oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_id, c.first_name, c.last_name
ORDER BY
    total_products_purchased DESC;

-- Query 73
SELECT TOP 1 category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id ORDER BY total_products DESC;

-- Query 75
SELECT TOP 1 product_id, SUM(quantity) AS total_orders FROM Order_Items GROUP BY product_id ORDER BY total_orders DESC;

```

Results

category_id	total_products
1	1

Query executed successfully.

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

```

FROM [Customers] c
LEFT JOIN [Orders] o ON c.customer_id = o.customer_id
LEFT JOIN [Order_Items] oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_id, c.first_name, c.last_name
ORDER BY
    total_products_purchased DESC;

-- Query 73
SELECT TOP 1 category_id, COUNT(*) AS total_products FROM Products GROUP BY category_id ORDER BY total_products DESC;

-- Query 75
SELECT TOP 1 product_id, SUM(quantity) AS total_orders FROM Order_Items GROUP BY product_id ORDER BY total_orders DESC;

```

Results

product_id	total_orders
6	17

Query executed successfully.

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute New Query Object Explorer

Insertion.sql - WIKI\nt (WIKI\wiki8 (63)) Creation.sql - WIKI\nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (60))

```
-- SELECT TOP 1
    c.customer_id,
    c.first_name,
    c.last_name,
    AVG(oi.quantity) AS avg_quantity_per_order
FROM
    Customers c
LEFT JOIN
    Orders o ON c.customer_id = o.customer_id
LEFT JOIN
    Order_Items oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_id, c.first_name, c.last_name
ORDER BY
    avg_quantity_per_order DESC
```

Results Messages

customer_id	first_name	last_name	avg_quantity_per_order
1	Hassan	Khan	17

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready

SQLQuery3.sql - WIKI\SQLEXPRESS.DB_Assignment (WIKI\wiki8 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

DB Assignment Execute New Query Object Explorer

Insertion.sql - WIKI\nt (WIKI\wiki8 (63)) Creation.sql - WIKI\nt (WIKI\wiki8 (61)) SQLQuery3.sql - WIKI\nt (WIKI\wiki8 (60))

```
-- SELECT TOP 1
    c.customer_id,
    c.first_name,
    c.last_name,
    AVG(oi.quantity * p.price) AS avg_order_value
FROM
    Customers c
LEFT JOIN
    Orders o ON c.customer_id = o.customer_id
LEFT JOIN
    Order_Items oi ON o.order_id = oi.order_id
LEFT JOIN
    Products p ON oi.product_id = p.product_id
GROUP BY
    c.customer_id, c.first_name, c.last_name
ORDER BY
    avg_order_value DESC
```

Results Messages

customer_id	first_name	last_name	avg_order_value
1	Ahmed	Ali	14300.0000

Query executed successfully.

WIKI\SQLEXPRESS (16.0 RTM) WIKI\wiki8 (60) DB_Assignment 00:00:00 1 rows

Ready