

# PIZZA-SALES PROBLEMS USING MYSQL WORKBENCH



# Let's start presentation

## SQL FOR DATA ANALYSIS

Here we will see the some pizza sales problem.....  
And solve it in MySQL workbench

My name is Gaurangi Chauhan. And I'm a computer science engg student at the University of Dr. APJ Abdul kalam University.

## **Basic:**

Retrieve the total number of orders placed.

Calculate the total revenue generated from pizza sales.

Identify the highest-priced pizza.

Identify the most common pizza size ordered.

List the top 5 most ordered pizza types along with their quantities.

## **Intermediate:**

Join the necessary tables to find the total quantity of each pizza category ordered.

Determine the distribution of orders by hour of the day.

Join relevant tables to find the category-wise distribution of pizzas.

Group the orders by date and calculate the average number of pizzas ordered per day.

Determine the top 3 most ordered pizza types based on revenue.

## **Advanced:**

Calculate the percentage contribution of each pizza type to total revenue.

Analyze the cumulative revenue generated over time.

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# BASIC

- Retrieve the total number of orders placed.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of databases, with 'pizzahut' selected. Under 'pizzahut', the 'Tables' folder is expanded, showing 'order\_details', 'orders', 'pizza\_types', and 'pizzas'. The 'order\_details' table is selected, and its columns are listed: 'order\_details\_id' (int PK), 'order\_id' (int), 'pizza\_id' (text), and 'quantity' (int). The main editor window shows a SQL query in 'SQL File 4\*':

```
-- Retrieve the total number of orders placed.  
  
select count(order_id) as total_orders from orders;
```

The query is executed, and the results are displayed in the 'Result Grid' at the bottom. The grid shows a single row with the value '21350' under the column 'total\_orders'.

total_orders
21350

The interface also includes a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help), a toolbar, and a status bar at the bottom indicating 'Formatted 1 statements.' and 'Read Only'.

- Calculate the total revenue generated from pizza sales.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

bankmanagementsystems

onlinebookstore

pizzahut

Tables

order\_details

orders

pizza\_types

pizzas

Columns

Indexes

Foreign Keys

Triggers

Administration Schemas

Information

Table: order\_details

Columns:

order\_details\_id int PK

order\_id int

pizza\_id int

quantity int

Object Info Session

Formatted 1 statements.

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* x SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\* SQL File 12\* SQL File

Limit to 1000 rows

```
1 -- Calculate the total revenue generated from pizza sales.
2
3 SELECT
4     ROUND(SUM(order_details.quantity * pizzas.price),
5           2) AS total_sales
6 FROM
7     order_details
8     JOIN
9     pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid

total_sales
817860.05

Result 1 x

Read Only

- Identify the highest-priced pizza.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

bankmanagementsystems  
onlinebookstore  
pizzahut

Tables

order\_details  
orders  
pizza\_types  
pizzas

Columns  
Indexes  
Foreign Keys  
Triggers

Administration Schemas

Information

Table: **order\_details**

Columns:

**order\_details\_id** int PK  
order\_id int  
pizza\_id text  
quantity int

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\* SQL File 12\* SQL File

Limit to 1000 rows

```
1 -- Identify the highest-priced pizza.  
2  
3 • SELECT  
4     pizza_types.name, pizzas.price  
5 FROM  
6     pizza_types  
7     JOIN  
8     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
9 ORDER BY pizzas.price DESC  
10 LIMIT 1;
```

Result Grid

name	price
The Greek Pizza	35.95

Result 1 x

Object Info Session

Formatted 1 statements.

Result Grid  
Form Editor  
Read Only



- Identify the most common pizza size ordered.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

bankmanagementsystems  
onlinebookstore  
pizzahut

Tables

order\_details  
orders  
pizza\_types  
pizzas

Columns  
Indexes  
Foreign Keys  
Triggers

Administration Schemas

Information

Table: order\_details

Columns:

order\_details\_id int PK  
order\_id int  
pizza\_id text  
quantity int

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* x SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\* SQL File 12\* SQL File

Limit to 1000 rows

```
19 ORDER BY order_count DESC ;
20
21
22 -- solution 2
23 SELECT
24     pizzas.size,
25     COUNT(order_details.order_details_id) AS order_count
26 FROM
27     pizzas
28     JOIN
29     order_details ON pizzas.pizza_id = order_details.pizza_id
30 GROUP BY pizzas.size
31 ORDER BY order_count DESC
32 LIMIT 3;
```

Result Grid

	size	order_count
▶	L	18526
	M	15385
	S	14137

Result 6 x

Object Info Session

Formatted 1 statements.

Result Grid  
Form Editor  
Read Only

- List the top 5 most ordered pizza types along with their quantities.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

bankmanagementsystems  
onlinebookstore  
pizzahut

Tables

order\_details  
orders  
pizza\_types  
pizzas

Columns  
Indexes  
Foreign Keys  
Triggers

Administration Schemas

Information

Table: **order\_details**

Columns:

**order\_details\_id** int PK  
order\_id int  
pizza\_id text  
quantity int

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\* SQL File 12\* SQL File

Limit to 1000 rows

```
1 -- List the top 5 most ordered pizza types along with their quantities.  
2  
3 • SELECT  
4     pizza_types.name, SUM(order_details.quantity) AS quantity  
5 FROM  
6     pizza_types  
7     JOIN  
8     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
9     JOIN  
10    order_details ON order_details.pizza_id = pizzas.pizza_id  
11 GROUP BY pizza_types.name  
12 ORDER BY quantity DESC  
13 LIMIT 5;
```

Result Grid

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Object Info Session

Result 4 x

Formatted 1 statements.

Read Only



# Intermediate

- Join the necessary tables to find the total quantity of each pizza category ordered.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'pizzahut' database selected, with tables 'order\_details', 'orders', 'pizza\_types', and 'pizzas' listed. Below this, the 'Table: order\_details' is shown with columns 'order\_id' (int PK), 'pizza\_id' (int), and 'quantity' (int). The main editor shows a SQL query in 'SQL File 9\*' that joins 'pizza\_types', 'pizzas', and 'order\_details' to find the total quantity for each pizza category. The query is as follows:

```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.
2
3
4  SELECT
5      pizza_types.category,
6      SUM(order_details.quantity) AS quantity
7  FROM
8      pizza_types
9      JOIN
10     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11     JOIN
12     order_details ON order_details.pizza_id = pizzas.pizza_id
13  GROUP BY pizza_types.category
14  ORDER BY quantity DESC;
```

Below the query editor, the 'Result Grid' shows the results of the query:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

The bottom of the interface shows 'Result 3' and a 'Read Only' status.

- Determine the distribution of orders by hour of the day.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

bankmanagementsystems  
onlinebookstore  
pizzahut

Tables

order\_details  
orders  
pizza\_types  
pizzas

Columns  
Indexes  
Foreign Keys  
Triggers

Administration Schemas

Information

Table: **order\_details**

Columns:

order\_details\_id int PK  
order\_id int  
pizza\_id text  
quantity int

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\* SQL File 12\* SQL File

Limit to 1000 rows

```
1 -- Determine the distribution of orders by hour of the day.  
2  
3 SELECT  
4     HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
5 FROM  
6     orders  
7 GROUP BY HOUR(order_time);
```

Result Grid

hour	order_count
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Object Info Session

Result 2

Formatted 1 statements.

Result Grid  
Form Editor  
Read Only

- Join relevant tables to find the category-wise distribution of pizzas.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

bankmanagementsystems  
onlinebookstore  
pizzahut

Tables

order\_details  
orders  
pizza\_types  
pizzas

Columns  
Indexes  
Foreign Keys  
Triggers

Administration Schemas

Information

Table: order\_details

Columns:

order\_details\_id int PK  
order\_id int  
pizza\_id text  
quantity int

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\* x SQL File 12\* SQL File

Limit to 1000 rows

```
1 -- Join relevant tables to find the category-wise distribution of pizzas.  
2  
3 • select category, count(name) from pizza_types  
4 group by category;
```

Result Grid

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Result 1 x

Object Info Session

Formatted 1 statements.

Result Grid  
Form Editor  
Read Only

- Group the orders by date and calculate the average number of pizzas ordered per day.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

bankmanagementsystems  
onlinebookstore  
pizzahut

Tables

order\_details  
orders  
pizza\_types  
pizzas

Columns  
Indexes  
Foreign Keys  
Triggers

Administration Schemas

Information

Table: order\_details

Columns:

order\_details\_id int PK  
order\_id int  
pizza\_id text  
quantity int

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\* SQL File 12\* x SQL File

Limit to 1000 rows

```
1 -- Group the orders by date and calculate the average number of pizzas ordered per day.
2
3
4
5 • SELECT
6     ROUND(AVG(quantity), 0) as avg_pizzas_ordered_per_day
7 FROM
8     (SELECT
9         orders.order_date, SUM(order_details.quantity) AS quantity
10      FROM
11         orders
12      JOIN order_details ON orders.order_id = order_details.order_id
13      GROUP BY orders.order_date) AS order_quantity;
```

Result Grid

avg_pizzas_ordered_per_day
138

Export: | Wrap Cell Content: |

Result 3 x

Read Only

Formatted 1 statements.



- Determine the top 3 most ordered pizza types based on revenue.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

bankmanagementsystems

onlinebookstore

pizzahut

Tables

order\_details

orders

pizza\_types

pizzas

Columns

Indexes

Foreign Keys

Triggers

Administration Schemas

Information

Table: order\_details

Columns:

order\_details\_id int PK

order\_id int

pizza\_id text

quantity int

SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\* SQL File 12\* SQL File 13\*

Limit to 1000 rows

```
1 -- Determine the top 3 most ordered pizza types based on revenue.
2
3 SELECT
4     pizza_types.name,
5     ROUND(SUM(order_details.quantity * pizzas.price),
6           0) AS revenue
7 FROM
8     pizza_types
9     JOIN
10    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
11    JOIN
12    order_details ON order_details.pizza_id = pizzas.pizza_id
13 GROUP BY pizza_types.name
14 ORDER BY revenue desc limit 3;
```

Result Grid

	name	revenue
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

Result 3

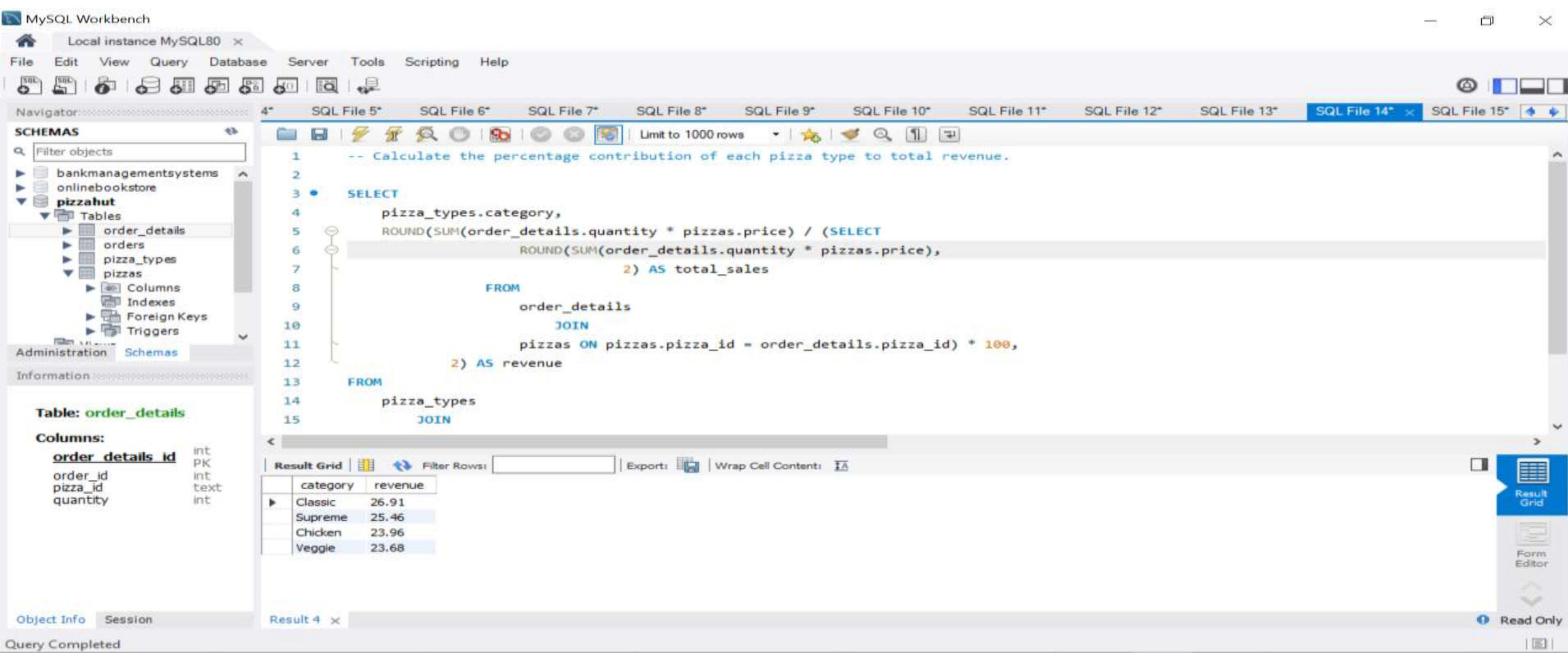
Formatted 1 statements.

Read Only



## Advanced:

- Calculate the percentage contribution of each pizza type to total revenue.



MySQL Workbench interface showing a SQL query to calculate the percentage contribution of each pizza type to total revenue.

The query is as follows:

```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2
3  SELECT
4      pizza_types.category,
5      ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
6          ROUND(SUM(order_details.quantity * pizzas.price),
7              2) AS total_sales
8          FROM
9              order_details
10             JOIN
11                 pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
12          2) AS revenue
13  FROM
14      pizza_types
15  JOIN
```

The result grid shows the following data:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Table: order\_details

Columns:

- order\_details\_id (int PK)
- order\_id (int)
- pizza\_id (text)
- quantity (int)

Query Completed

- Analyze the cumulative revenue generated over time.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

bankmanagementsystems  
onlinebookstore  
pizzahut

Tables

order\_details  
orders  
pizza\_types  
pizzas

Columns  
Indexes  
Foreign Keys  
Triggers

Administration Schemas

Information

Table: **order\_details**

Columns:

**order\_details\_id** int PK  
order\_id int  
pizza\_id text  
quantity int

SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\* SQL File 12\* SQL File 13\* SQL File 14\* SQL File 15\*

Limit to 1000 rows

```

1  -- Analyze the cumulative revenue generated over time.
2
3  select order_date, sum(revenue) over (order by order_date) as cum_revenue from (select orders.order_date,
4  sum(order_details.quantity*pizzas.price) as revenue
5  from order_details join pizzas
6  on order_details.pizza_id = pizzas.pizza_id
7  join orders
8  on orders.order_id = order_details.order_id
9  group by orders.order_date) as sales;
```

Result Grid

order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.500000000001
2015-01-16	36937.650000000001

Result 2

Query Completed

Read Only



**Determine the top 3 most ordered pizza types based on revenue for each pizza category**

**Try to solve this question**





**THANK YOU**