



# PIZZA SALES

---

# REPORT

Prepared by  
muhammad samed



# PREVIEW

Use sql to find multiple answers from pizza sales dataset.

Different variation of queries of sql on pizza sales dataset provide useful answers to questions are showcased in next slides. As a data analyst I am sure these answer of queries will be beneficial for pizza business

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the database schema with the 'orders' table selected. The central pane contains a SQL editor with the following code:

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

The results pane below shows the output of the query:

total_orders
21350

The status bar at the bottom indicates 'Result 3' and 'Read Only'.



# Calculate the total revenue generated from pizza sales.

The screenshot shows the SSMS interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for New Query, New Script, Open, Save, Print, and others.
- Navigator:** Shows the database structure under the schema **pizzahut**. It includes tables **order\_details** and **orders**, along with their columns and various indexes, foreign keys, and triggers.
- SQL Editor:** The main pane contains the following SQL query:

```
1  -- Calculate the total revenue generated from pizza sales.
2
3  • SELECT
4      ROUND(sum(order_details.quantity * pizzas.price),2) AS total_sales
5  FROM
6      order_details
7      JOIN
8          pizzas ON pizzas.pizza_id = order_details.pizza_id;
```
- Result Grid:** Below the editor, the results of the query are displayed in a grid:

total_sales
817860.05
- Information Panel:** On the left, it shows the **pizzas** table with its columns.
- Status Bar:** Shows "Result 2" and "Read Only".

[Back to Agenda Page](#)



# Identify the highest-priced pizza

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbars:** Standard MySQL Workbench toolbar.
- Navigator:** Shows the database structure under the schema "orders".
  - SCHEMAS:** pizza\_types (Columns: pizza\_type\_id, name, category, ingredients), pizzas (Columns: pizza\_id, pizza\_type\_id, size, price).
  - Tables:** orders, order\_details.
- SQL Editor:** SQL File 15\* contains the following query:

```
-- Identify the highest-priced pizza.  
SELECT pizza_types.name, pizzas.price  
FROM pizza_types  
JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY pizzas.price DESC  
LIMIT 1;
```
- Result Grid:** Shows the result of the query:

name	price
The Greek Pizza	35.95
- Information Panel:** Shows the definition of the "pizzas" table:

Table: pizzas

Columns:

pizza_id	text
pizza_type_id	text
size	text
price	double
- Object Info:** Shows the session information.
- Message Bar:** Error message: # 15 18:15:29 select pizza\_types.name pizzas.price from pizzas\_types join pizzas on pizzas\_types. Error Code: 1146. Table 'hizkiaut.pizzas\_types' doesn't exist.

Briefly elaborate on what you want to discuss.



# Identify the most common pizza size ordered

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local Instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** SQL, Schemas, Tables, Columns, Views, Functions, Procedures, Triggers, Events, Scripts, Results, Log, Help.
- Navigator:** SCHEMAS section shows `pizza_types` and `pizzas` tables.
- Query Editor:** SQL tab, SQL File 16\*, contains the following query:

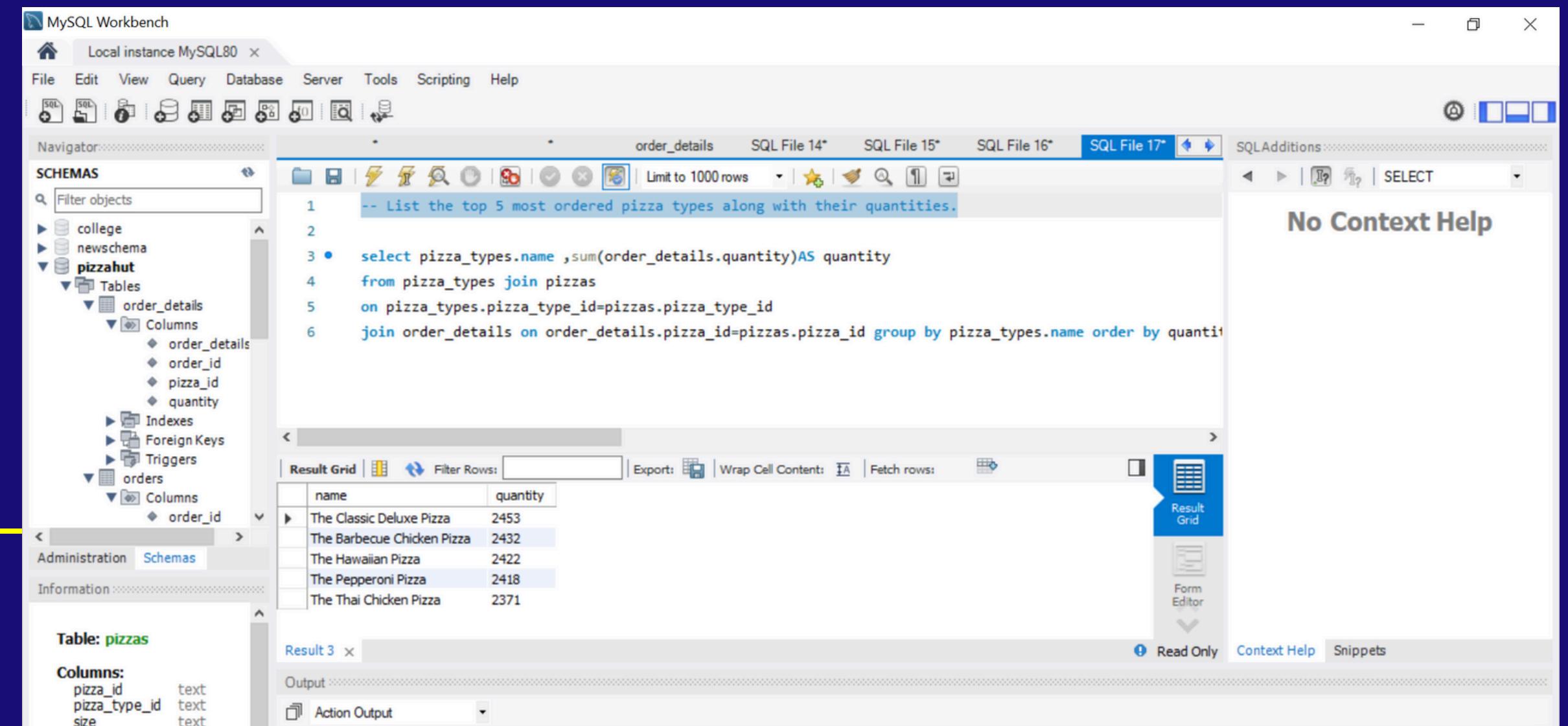
```
1 -- Identify the most common pizza size ordered.
2
3 • select pizzas.size,count(order_details.order_details_id)as order_count from pizzas join order_details
4 order by order_count desc limit 1;
```
- Result Grid:** Shows a single row:

size	order_count
L	18526
- Information Panel:** Table: `pizzas`, Columns: `pizza_id` text, `pizza_type_id` text, `size` text, `price` double.
- Action Output:** Shows three errors related to the count function:

#	Time	Action	Message	Duration / Fetch
21	18:45:04	select pizzas.size,count (order_details.order_details_id)as order_count from pizzas jo...	Error Code: 1630. FUNCTION pizzahut.count does not exist. Check the 'Function N...	0.000 sec
22	18:48:28	select pizzas.size,count (order_details.order_details_id)as order_count from pizzas jo...	Error Code: 1630. FUNCTION pizzahut.count does not exist. Check the 'Function N...	0.000 sec
23	18:50:26	select pizzas.size,count(order_details.order_details_id)as order_count from pizzas jo...	Error Code: 1054. Unknown column 'pizzas.pizzas_id' in 'on clause'	0.000 sec

[Back to Agenda Page](#)

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



The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **pizzahut** schema selected. Under **Tables**, **order\_details** and **orders** are expanded, showing their columns (**order\_id**, **order\_details**, **pizza\_id**, **quantity**).
- SQL Editor:** Contains the following SQL query:

```
-- List the top 5 most ordered pizza types along with their quantities.  
select pizza_types.name ,sum(order_details.quantity)AS quantity  
from pizza_types join pizzas  
on pizza_types.pizza_type_id=pizzas.pizza_type_id  
join order_details on order_details.pizza_id=pizzas.pizza_id group by pizza_types.name order by quantity
```
- Result Grid:** Displays the results of the query in a tabular format. The data is as follows:

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

[Back to Agenda Page](#)



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

The screenshot shows the MySQL Workbench interface with a query editor and results grid.

**Query Editor:**

```
1
2 -- Join the necessary tables to find the total quantity of each pizza category ordered.
3
4 • select pizza_types.category,
5   sum(order_details.quantity)as quantity
6   from pizza_types join pizzas
7   on pizza_types.pizza_type_id=pizzas.pizza_type_id
8   join order_details
9   on order_details.pizza_id =pizzas.pizza_id
10  group by pizza_types.category order by quantity desc;
```

**Result Grid:**

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	1050

**Output:**

#	Time	Action	Message	Duration / Fetch
1	00:00:22	select pizza_types.category, sum(order_details.quantity)as quantity from pizza_types j...	Error Code: 1052. Column 'pizza_type_id' in on clause is ambiguous	0.032 sec
2	00:02:13	select pizza_types.category, sum(order_details.quantity)as quantity from pizza_types j... 4 row(s) returned		0.563 sec / 0.000 sec

[Back to Agenda Page](#)



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

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema **pizzahut** selected, containing tables like **order\_details**, **orders**, **pizza\_types**, and **pizzas**.
- Query Editor:** Displays the SQL query:

```
-- Determine the distribution of orders by hour of the day.  
select hour(order_time)as hour, count(order_id) as order_count  
from orders  
group by (order_time);
```
- Result Grid:** Shows the output of the query:

hour	order_count
11	2
11	1
12	1
12	3
12	1
- Action Output:** Shows the execution history with the following entries:

#	Time	Action	Message	Duration / Fetch
1	00:00:22	select pizza_types.category, sum(order_details.quantity)as quantity from pizza_types j...	Error Code: 1052. Column 'pizza_type_id' in on clause is ambiguous	0.032 sec
2	00:02:13	select pizza_types.category, sum(order_details.quantity)as quantity from pizza_types j...	4 row(s) returned	0.563 sec / 0.000 sec
3	00:10:45	elect hour(order_time)as hour, count(order_id) as order_count from orders group by or...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corre...	0.000 sec
4	00:11:27	select hour(order_time)as hour, count(order_id) as order_count from orders group by (...	1000 row(s) returned	0.078 sec / 0.000 sec

[Back to Agenda Page](#)



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

The screenshot shows the MySQL Workbench interface. In the top-left corner, there's a note: "Join relevant tables to find the category-wise distribution of pizzas." This note is highlighted with a yellow background and has a small red error icon next to it. The interface includes a Navigator pane on the left showing schemas like 'college', 'newschema', and 'pizzahut' (which is expanded to show tables like 'order\_details', 'orders', 'pizza\_types', and 'pizzas'). A central Query Editor pane contains the following text:

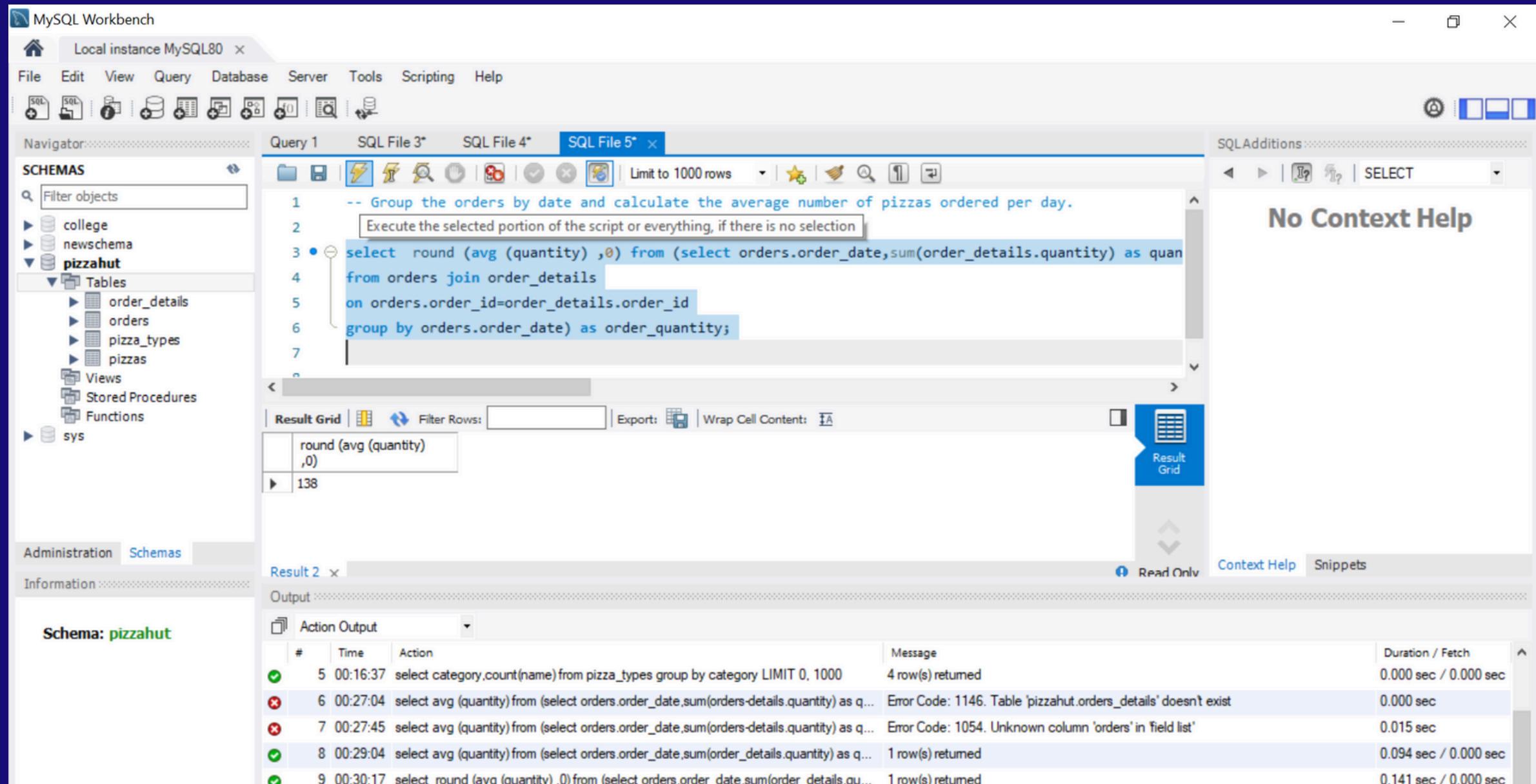
```
1 1 Join relevant tables to find the category-wise distribution of pizzas.  
2  
3
```

To the right of the Query Editor is a SQLAdditions panel titled "JOIN Syntax:" which provides information on MySQL JOIN syntax. Below the note in the Query Editor, there's a snippet of code:

```
>  
table_references:  
    escaped_table_reference [,  
  
    escaped_table_reference:  
        table_reference  
    | { OJ table_reference }
```

[Back to Agenda Page](#)

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



The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema **pizzahut** with tables **order\_details**, **orders**, **pizza\_types**, and **pizzas**.
- Query Editor (Query 1):** Contains the following SQL code:

```
-- Group the orders by date and calculate the average number of pizzas ordered per day.  
-- Execute the selected portion of the script or everything, if there is no selection  
select round (avg (quantity) ,0) from (select orders.order_date,sum(order_details.quantity) as quantity  
from orders join order_details  
on orders.order_id=order_details.order_id  
group by orders.order_date) as order_quantity;
```
- Result Grid:** Displays the result of the query:

round (avg (quantity) ,0)
138
- Output Window:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
5	00:16:37	select category.count(name) from pizza_types group by category LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
6	00:27:04	select avg (quantity) from (select orders.order_date,sum(orders-details.quantity) as q... Error Code: 1146. Table 'pizzahut.orders_details' doesn't exist	Error Code: 1146. Table 'pizzahut.orders_details' doesn't exist	0.000 sec
7	00:27:45	select avg (quantity) from (select orders.order_date,sum(orders-details.quantity) as q... Error Code: 1054. Unknown column 'orders' in 'Field list'	Error Code: 1054. Unknown column 'orders' in 'Field list'	0.015 sec
8	00:29:04	select avg (quantity) from (select orders.order_date,sum(order_details.quantity) as q... 1 row(s) returned	1 row(s) returned	0.094 sec / 0.000 sec
9	00:30:17	select round (avg (quantity) ,0) from (select orders.order_date,sum(order_details.quantity) as quantity from orders join order_details on orders.order_id=order_details.order_id group by orders.order_date) as order_quantity; 1 row(s) returned	1 row(s) returned	0.141 sec / 0.000 sec

PHONE

thanks for viewing File

---

Prepared by = muhammad sameed  
Junior data analyst  
[msameed001@gmail.com](mailto:msameed001@gmail.com)