

Where Every Slice is a Taste of Perfection

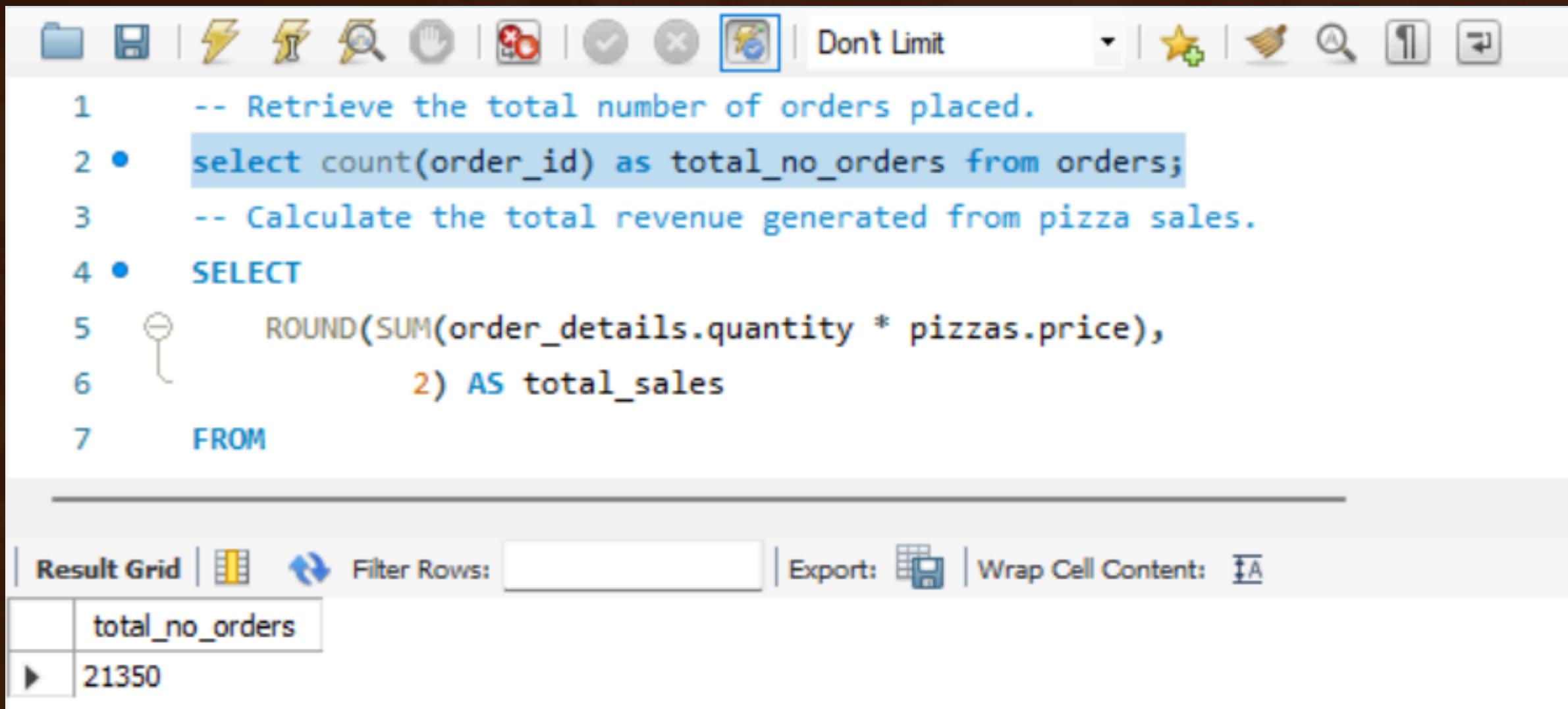
WELCOME TO PIZZA MANIA

ORDER
NOW

Start Your Slide



Retrieve the total number of orders placed.



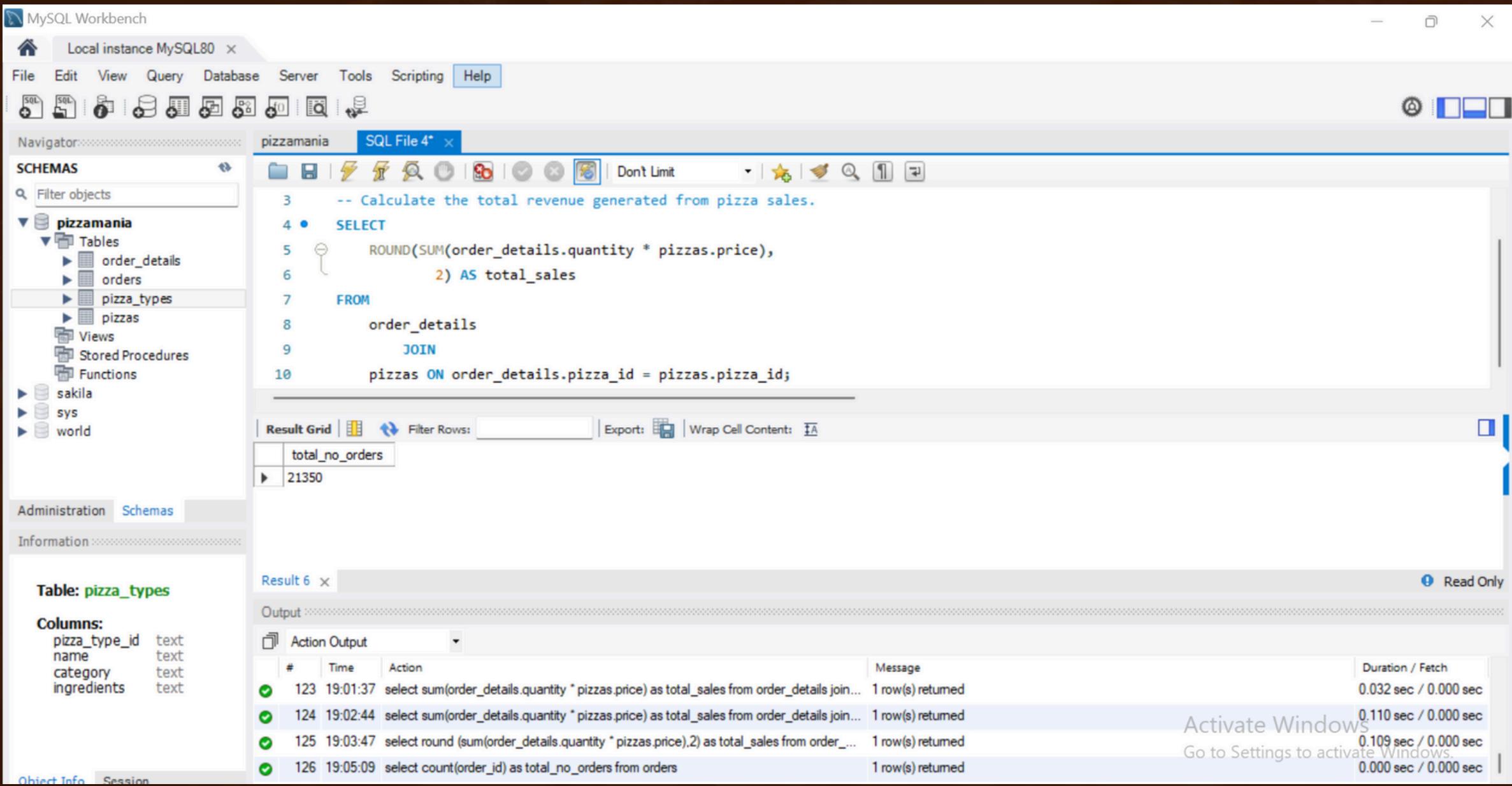
The screenshot shows a MySQL Workbench interface with a query editor and a result grid. The query retrieves the total number of orders placed and calculates the total revenue generated from pizza sales.

```
1 -- Retrieve the total number of orders placed.  
2 • select count(order_id) as total_no_orders from orders;  
3 -- Calculate the total revenue generated from pizza sales.  
4 • SELECT  
5     ROUND(SUM(order_details.quantity * pizzas.price),  
6           2) AS total_sales  
7 FROM
```

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

	total_no_orders
▶	21350

Calculate the total revenue generated from pizza sales.



The screenshot shows the MySQL Workbench interface with a SQL editor window titled "SQL File 4*". The code in the editor is:

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

The "Result Grid" pane below the editor shows a single row of results:

total_no_orders
21350

The "Result 6" pane at the bottom shows the execution history:

#	Time	Action	Message	Duration / Fetch
123	19:01:37	select sum(order_details.quantity * pizzas.price) as total_sales from order_details join...	1 row(s) returned	0.032 sec / 0.000 sec
124	19:02:44	select sum(order_details.quantity * pizzas.price) as total_sales from order_details join...	1 row(s) returned	0.110 sec / 0.000 sec
125	19:03:47	select round (sum(order_details.quantity * pizzas.price),2) as total_sales from order_...	1 row(s) returned	0.109 sec / 0.000 sec
126	19:05:09	select count(order_id) as total_no_orders from orders	1 row(s) returned	0.000 sec / 0.000 sec

IDENTIFY THE HIGHEST-PRICED PIZZA.

The screenshot shows the MySQL Workbench interface with a query window displaying the following SQL code:

```
13 •   SELECT
14     pizza_types.name, pizzas.price
15   FROM
16     pizza_types
17     JOIN
18       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
19   ORDER BY pizzas.price DESC
20   LIMIT 1;
```

The results grid shows one row:

name	price
The Greek Pizza	35.95

The bottom pane shows the execution history:

#	Time	Action	Message	Duration / Fetch
128	19:10:26	select price from pizzas	96 row(s) returned	0.000 sec / 0.000 sec
129	19:12:59	SELECT * FROM pizzamania.pizza_types	32 row(s) returned	0.000 sec / 0.000 sec
130	19:21:45	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	1 row(s) returned	0.000 sec / 0.000 sec
131	19:22:48	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	1 row(s) returned	0.000 sec / 0.000 sec

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

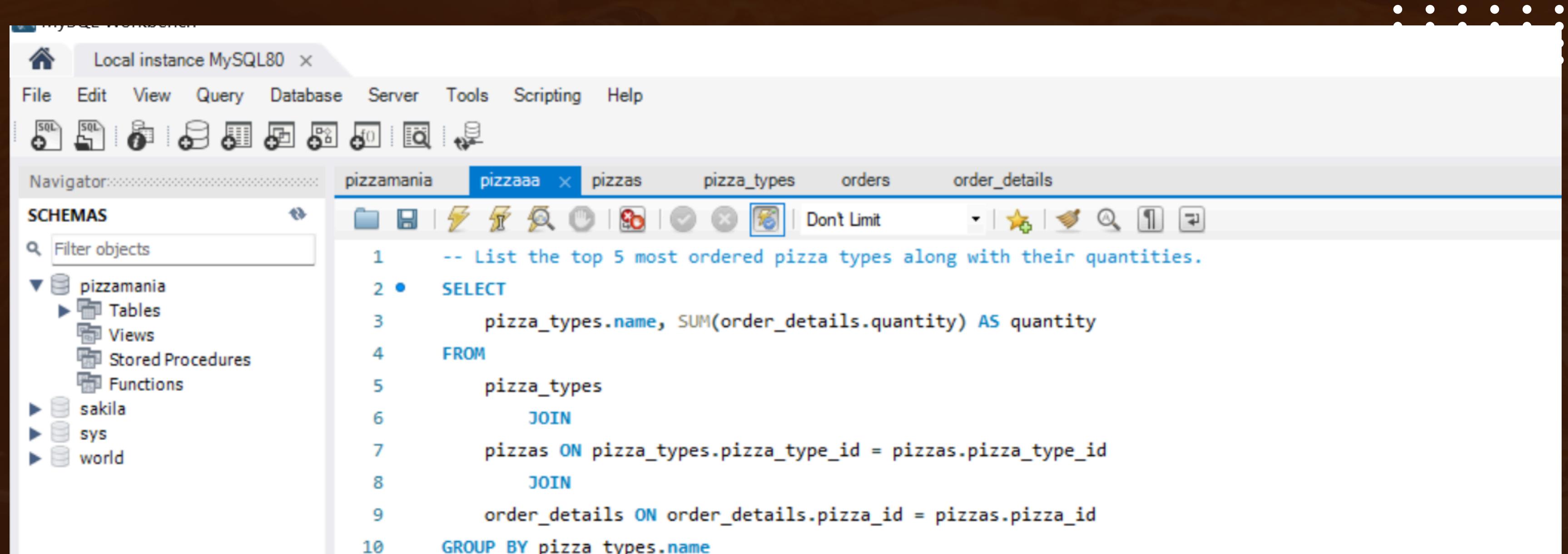
- Navigator:** Shows the schema **pizzamania** with tables **order_details**, **orders**, **pizza_types**, and **pizzas**.
- SQL File 4***: Contains the following SQL query:

```
-- Identify the most common pizza size ordered.  
select pizzas.size, count(order_details.order_details_id) as order_count  
from pizzas join order_details on pizzas.pizza_id = order_details.pizza_id  
group by pizzas.size order by order_count desc;
```
- Result Grid:** Displays the results of the query:

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28
- Result 12 x**: Shows the execution history:

#	Time	Action	Message	Duration / Fetch
134	19:29:33	SELECT * FROM pizzamania.order_details	48620 row(s) returned	0.000 sec / 0.094 sec
135	19:30:58	select pizza_id from order_details	48620 row(s) returned	0.000 sec / 0.062 sec
136	13:12:37	select pizzas.size, count(order_details.order_details_id) from pizzas join order_details... Error Code: 1054. Unknown column 'order_count' in 'order clause'	Error Code: 1054. Unknown column 'order_count' in 'order clause'	0.000 sec
137	13:13:32	select pizzas.size, count(order_details.order_details_id) as order_count from pizzas j... 5 row(s) returned	5 row(s) returned	0.234 sec / 0.000 sec

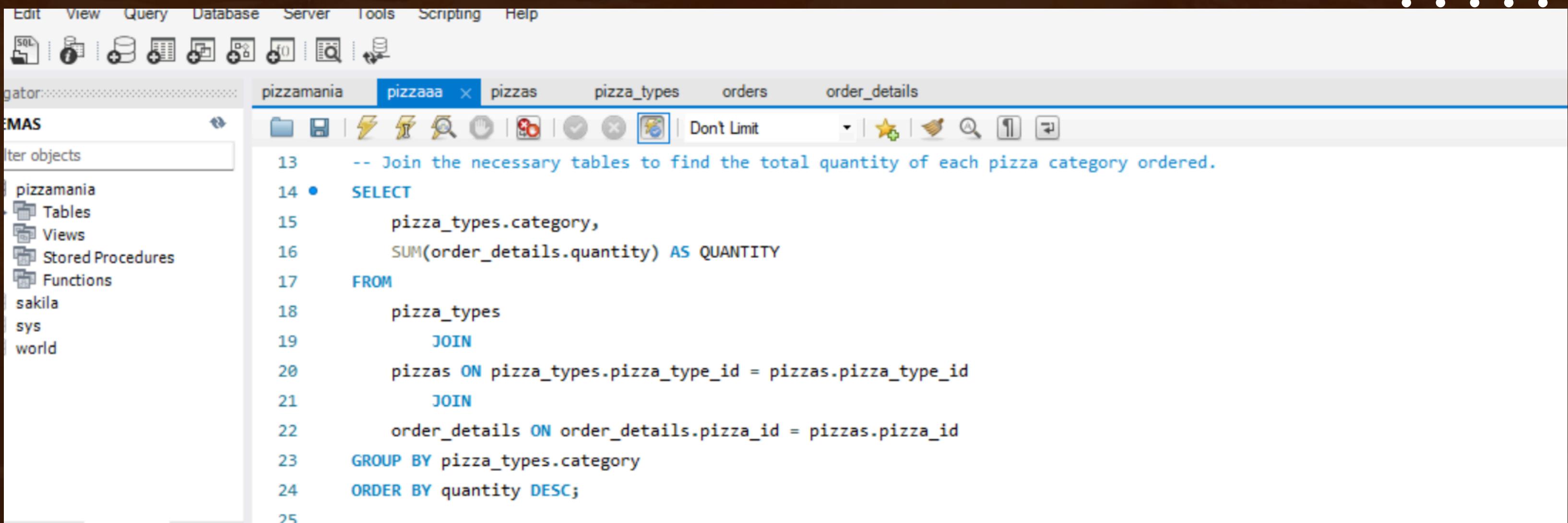
List the top 5 most ordered pizza types along with their quantities.



The screenshot shows the MySQL Workbench interface. The title bar says "Local instance MySQL80". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The left sidebar is titled "Navigator" and shows "SCHEMAS" with "pizzamania" expanded, displaying "Tables", "Views", "Stored Procedures", and "Functions". Other schemas listed are "sakila", "sys", and "world". The main area has tabs for "pizzamania", "pizzaaa", "pizzas", "pizza_types", "orders", and "order_details". The "pizzaaa" tab is active. The query editor contains the following SQL script:

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

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



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

- Toolbar:** Includes icons for SQL, Object Explorer, Database, Tools, Scripting, Help, and various file operations.
- Tab Bar:** Shows tabs for "pizzamania", "pizzaaa" (selected), "pizzas", "pizza_types", "orders", and "order_details".
- Object Explorer:** On the left, it lists databases (EMAS, sakila, sys, world) and tables (pizzamania, Tables, Views, Stored Procedures, Functions).
- Script Editor:** The main area contains a numbered SQL script:

```
13  -- Join the necessary tables to find the total quantity of each pizza category ordered.
14 •  SELECT
15      pizza_types.category,
16      SUM(order_details.quantity) AS QUANTITY
17  FROM
18      pizza_types
19      JOIN
20      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
21      JOIN
22      order_details ON order_details.pizza_id = pizzas.pizza_id
23  GROUP BY pizza_types.category
24  ORDER BY quantity DESC;
25
```



Determine the distribution of orders by hour of the day.

The screenshot shows the MySQL Workbench interface with a query editor window. The title bar reads "Local instance MySQL80". The menu bar includes "Edit", "View", "Query", "Database", "Server", "Tools", "Scripting", and "Help". The toolbar below the menu has icons for SQL, schema browser, and other database operations. The query editor window displays the following SQL code:

```
26 -- Determine the distribution of orders by hour of the day.
27 • select hour(order_time), count(order_id) from orders
28 group by hour(order_time);
29
```

The code is intended to select the hour of the order time and count the number of orders for each hour, grouped by hour.



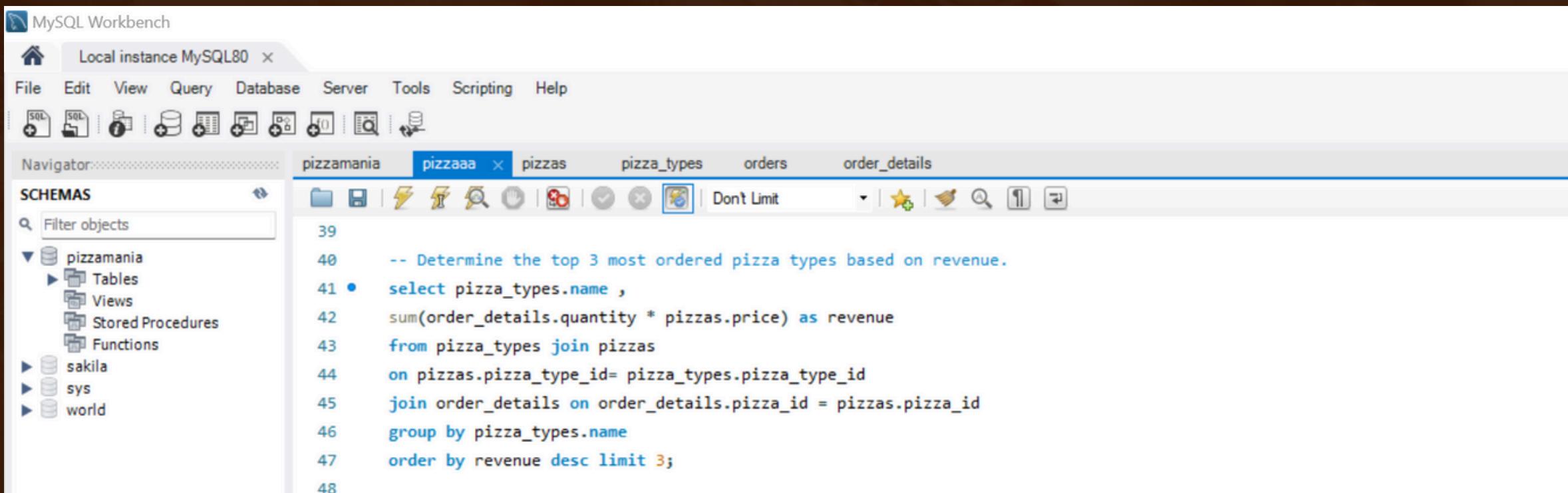
Join relevant tables to find the category-wise distribution of pizzas.

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

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
30 -- Join relevant tables to find the category-wise distribution of pizzas.
31 • select category, count(name) from pizza_types
32 group by category;
33
34 -- Group the orders by date and calculate the average number of pizzas ordered per day.
35 • select avg(quantity) from
36   (select orders.order_date, sum(order_details.quantity) as quantity
37   from orders join order_details on orders.order_id = order_details.order_id
38   group by orders.order_date) as order_quantity ;
39
```

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



The screenshot shows the MySQL Workbench interface with a query editor window. The window title is "Local instance MySQL80". The tabs at the top include "pizzamania", "pizzaaa" (which is selected), "pizzas", "pizza_types", "orders", and "order_details". The left sidebar shows the "SCHEMAS" tree with "pizzamania" expanded, showing "Tables", "Views", "Stored Procedures", and "Functions". Other schemas listed are "sakila", "sys", and "world". The main pane contains the following SQL code:

```
39
40  -- Determine the top 3 most ordered pizza types based on revenue.
41 • select pizza_types.name ,
42    sum(order_details.quantity * pizzas.price) as revenue
43    from pizza_types join pizzas
44      on pizzas.pizza_type_id= pizza_types.pizza_type_id
45    join order_details on order_details.pizza_id = pizzas.pizza_id
46    group by pizza_types.name
47    order by revenue desc limit 3;
48
```



Pizzamania Presentation

THANK YOU