

MySQL Data Analysis Project:

PIZZA SALES INSIGHTS



HI, I'M ASIF SHAIKH



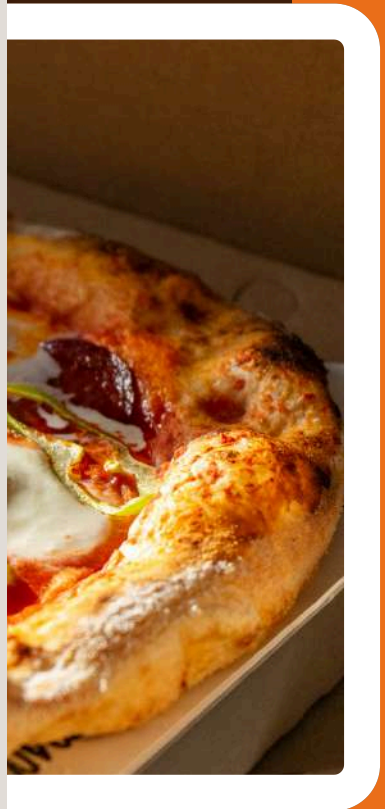
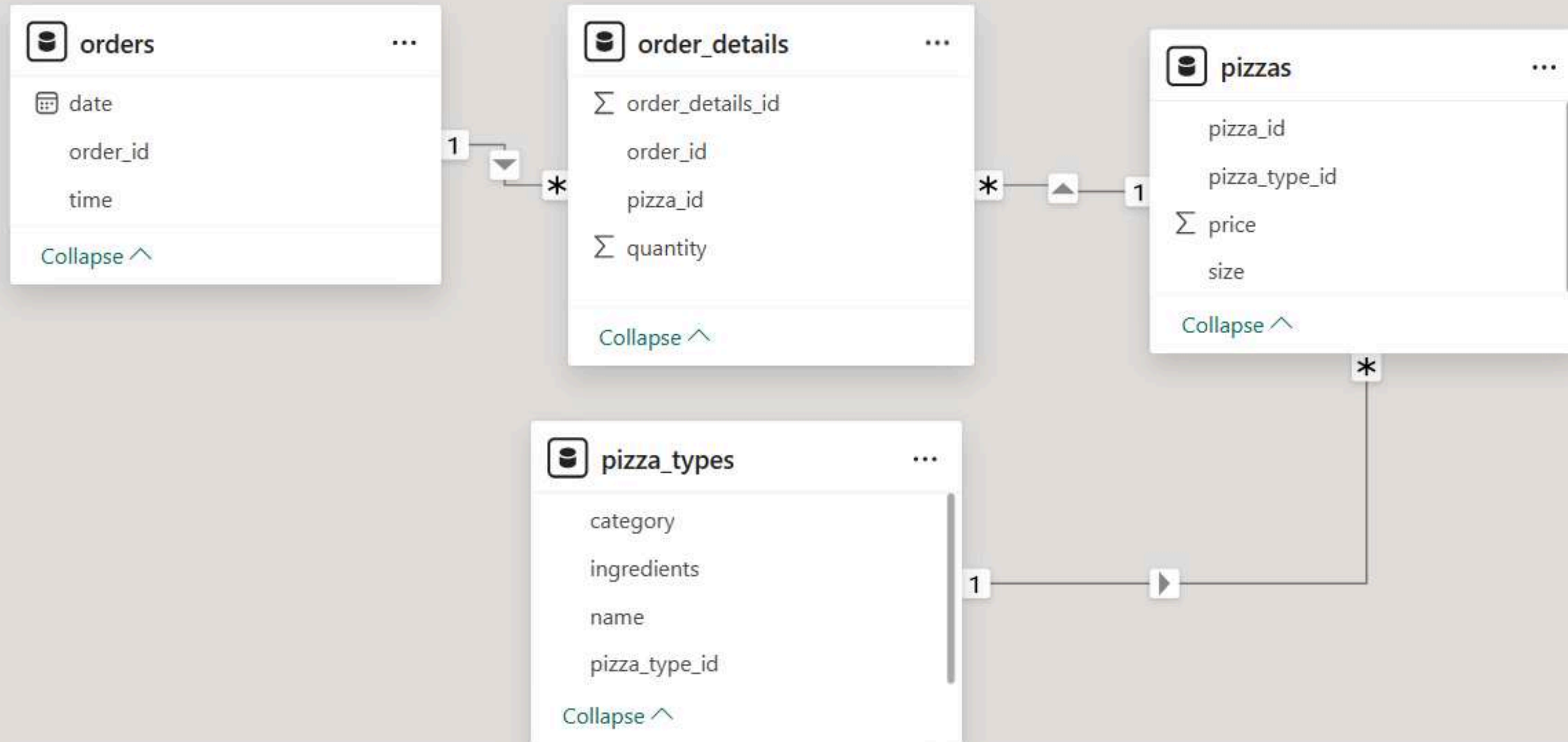
In this project, I utilized MySQL to explore, clean, and analyze a pizza sales dataset.

- Total revenue
- Peak order times
- Top-selling pizzas
- Revenue breakdown by type and category

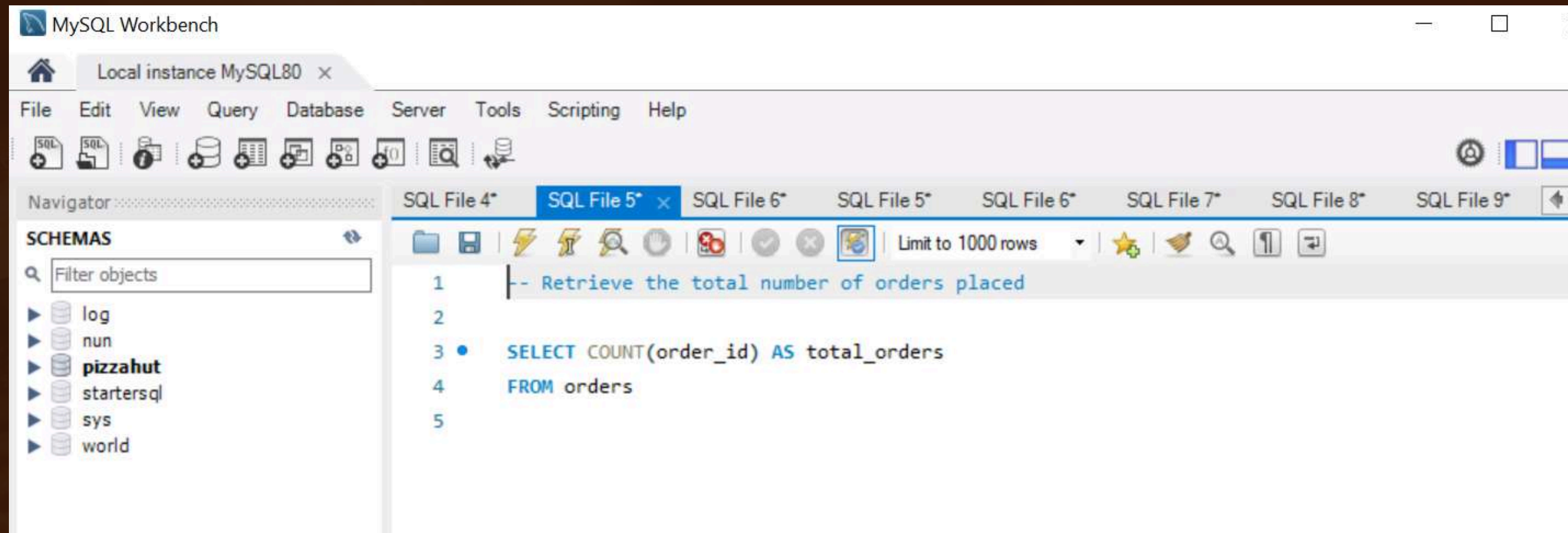
Dataset: Orders, Order Details, Pizzas, Pizza Types



SCHEMA



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES. :

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- log
- nun
- pizzahut**
- startersql
- sys
- world

SQL File 4* SQL File 5* SQL File 6* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9*

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;
```

Administration Schemas

Information

Table: **orders**

Columns:

- order_id** int PK
- order_date date
- order_time time

Result Grid

total_sales
817860.05

Result 3 x

Read Only

IDENTIFY THE HIGHEST-PRICED PIZZA.



The screenshot shows the MySQL Workbench interface for a local instance of MySQL 8.0. The 'Navigator' pane on the left displays the 'pizzahut' database schema. The 'SQL Editor' pane in the center contains the following SQL query:

```
-- Identify the highest price 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;
```

The 'Administration' pane at the bottom left shows the 'Table: orders' with columns: order_id (int PK), order_date (date), and order_time (time). The 'Result Grid' pane at the bottom right displays the query results:

name	price
The Greek Pizza	35.95

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

- log
- nun
- pizzahut
- startersql
- sys
- world

SQL File 4* SQL File 5* SQL File 6* SQL File 5* SQL File 6* x SQL File 7* SQL File 8* SQL File 9*

Limit to 1000 rows

```
1 -- Identify the most common pizza size ordered.
2
3 SELECT pizzas.size, COUNT(order_details.order_details_id) AS order_count
4 FROM pizzas JOIN order_details
5 ON pizzas.pizza_id = order_details.pizza_id
6 GROUP BY pizzas.size
7 ORDER BY order_count DESC
```

Administration Schemas

Information

Table: orders

Columns:

- order_id int PK
- order_date date
- order_time time

Result Grid

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

Result 1 x

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

- log
- nun
- pizzahut**
- startersql
- sys
- world

Administration Schemas Information

Table: **orders**

Columns:

- order_id** int PK
- order_date date
- order_time time

SQL File 4* SQL File 5* SQL File 6* SQL File 5* SQL File 6* SQL File 7* x SQL File 8* SQL File 9*

Limit to 1000 rows

```
1 -- List the Top 5 most ordered pizza types
2 -- along with their quantities.
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

Result 3 x Read Only

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 lists databases: log, nun, pizzahut, startersql, sys, and world. The 'Administration' tab is selected, showing the 'Schemas' section. Below it, the 'Information' section displays details for the 'orders' table, including columns: order_id (int PK), order_date (date), and order_time (time).

The main editor window shows a SQL query in 'SQL File 8*'. The query is as follows:

```
-- total quantity of each pizza cateogry order

SELECT pizza_types.category,
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.category
ORDER BY quantity DESC;
```

The query results are displayed in the 'Result Grid' at the bottom. The grid shows the following data:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

- log
- nun
- pizzahut**
- startersql
- sys
- world

Administration Schemas

Information

Table: **orders**

Columns:

- order_id int PK
- order_date date
- order_time time

SQL File 9* x SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*

Limit to 1000 rows

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

Result Grid

hours	order_count
20	1642
21	1198
22	663
23	28
10	8
9	1

Result 4 x Read Only

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' panel lists databases: log, nun, pizzahut, startersql, sys, and world. The 'pizzahut' database is selected. Below it, the 'Table: orders' is shown with columns: order_id (int PK), order_date (date), and order_time (time). The main editor displays a SQL query in 'SQL File 10':

```
1  -- Join the relevant tables to find the
2  -- category wise distribution of pizzas
3
4  •  SELECT category, COUNT(name)
5     FROM pizza_types
6     GROUP BY category
7
```

The query results are displayed in the 'Result Grid' at the bottom. The grid shows the following data:

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

The interface also shows a 'Navigator' panel with 'SQL File 10' selected, and a 'Result Grid' panel with a 'Filter Rows' input field and an 'Export' button. The status bar at the bottom indicates 'Result 1' and 'Read Only'.

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a list of databases: log, nun, pizzahut, startersql, sys, and world. The 'pizzahut' database is selected. Below the schemas, the 'Table: orders' is shown with columns: order_id (int PK), order_date (date), and order_time (time). The main editor window displays a SQL query in 'SQL File 11*'. The query is as follows:

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

The query is executed, and the result is displayed in the 'Result Grid' at the bottom. The result grid shows a single row with the value 138 for the column 'average_pizzas_ordered'.

average_pizzas_ordered
138

The bottom status bar indicates 'Result 5' and 'Read Only'.

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a list of databases: log, nun, pizzahut, startersql, sys, and world. The 'pizzahut' database is selected. Below the schemas, the 'Table: orders' is shown with columns: order_id (int PK), order_date (date), and order_time (time). The main editor window displays a SQL query in 'SQL File 12*'. The query is as follows:

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

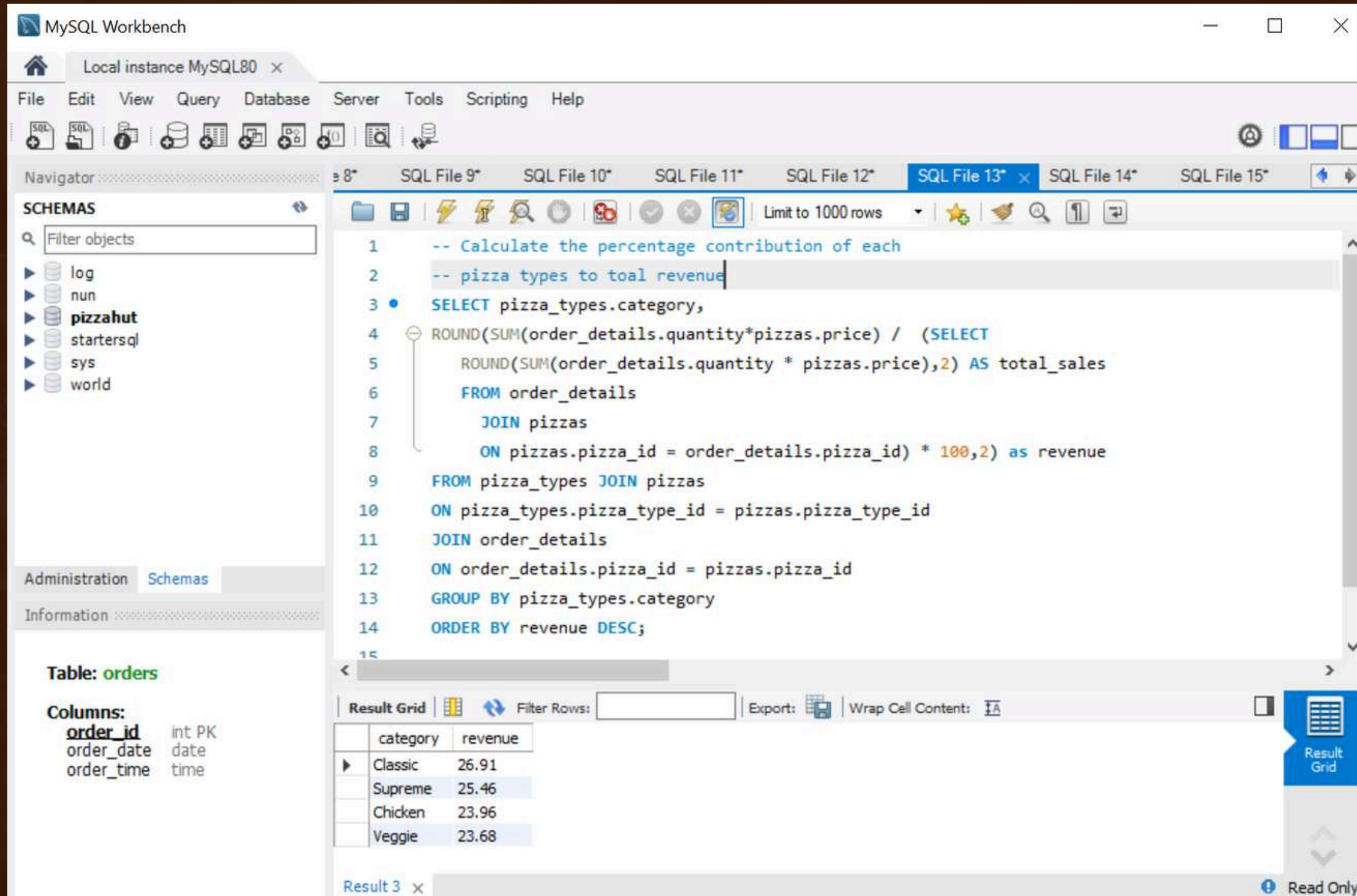
SELECT pizza_types.name,
SUM(order_details.quantity * pizzas.price) AS revenue
FROM pizza_types JOIN pizzas
ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN order_details
ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

The bottom panel shows the 'Result Grid' with the following data:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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 'Result 3' and 'Read Only'.

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA CATEGORY TO TOTAL REVENUE.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the Schemas pane with a list of databases: log, nun, pizzahut, startersql, sys, and world. The main editor window shows a SQL query in SQL File 13. The query is as follows:

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

The bottom right of the interface shows the Result Grid with the following data:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- log
- nun
- pizzahut**
- startersql
- sys
- world

Administration Schemas

Information

Table: **orders**

Columns:

- order_id** int PK
- order_date date
- order_time time

SQL File 14* x

```
1 -- Aalyze the cumulative revenue generated over time
2 SELECT order_date,
3 ROUND(SUM(revenue) OVER(ORDER BY order_date),2) AS cum_revenue
4 FROM
5 (SELECT orders.order_date,
6 SUM(order_details.quantity * pizzas.price) AS revenue
7 FROM order_details
8 JOIN pizzas
9 ON order_details.pizza_id = pizzas.pizza_id
10 JOIN orders
11 ON orders.order_id = order_details.order_id
12 GROUP BY order_date) AS sales;
13
```

Result Grid

	order_date	cum_revenue
▶	2015-01-01	2713.85
	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

Result 3 x

Read Only

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15* x

Limit to 1000 rows

```
1 -- Determine the top 3 most ordered pizzas types
2 -- based on revenue for each pizza category
3 • SELECT name, revenue FROM
4 (SELECT category, name, revenue,
5  RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rn
6  FROM
7  (SELECT pizza_types.category, pizza_types.name,
8   SUM((order_details.quantity) * pizzas.price) AS revenue
9   FROM pizza_types JOIN pizzas
10  ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11  JOIN order_details
12  ON order_details.pizza_id = pizzas.pizza_id
13  GROUP BY pizza_types.category, pizza_types.name) AS a) AS b
14 WHERE rn <=3;
```

Table: orders

Columns:

- order_id int PK
- order_date date
- order_time time

Result Grid

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25

Result 4 x

Read Only

**THANK YOU
FOR ATTENTION**

