WELCOME TO PIZZA SALES DATABASE ANALYSIS!

My name is Sakshi Rajesh Bhavsar
student of B.E Computer Science & Engineering
batch (2021-25)

I'm excited to share my project as presentation on analyzing pizza sales data. This project aims to uncover valuable insights from a database using SQL queries in MYSQL Workbench.

So, let's see what insights we can uncover from the pizza sales database!

bsakshi2019@gmail.com

PROJECT INDEX

bsakshi2019@gmail.com

INTRODUCTION

PIZZA SALES PROJECT

USING SQL QUERIES

This project aims to provide valuable insights into the dynamics of pizza sales by analyzing comprehensive data on orders, customers, and operations.

By leveraging SQL queries, we will uncover trends and other pizza sales.

bsakshi2019@gmail.com

PROJECT OBJECTIVES

- Identify top-selling pizza types and toppings
- Analyze customer purchasing behavior and demographics
- Optimize inventory management and logistics
- Improve marketing strategies and target high-value customers

bsakshi2019@gmail.com

DATA MODEL

Table:orders

order_id	int PK
order_date	date
order_time	time

Table:pizzas

pizza_id	text
pizza_type_id	text
size	text
price	double

bsakshi2019@gmail.com

DATA MODEL

Table:pizza_types

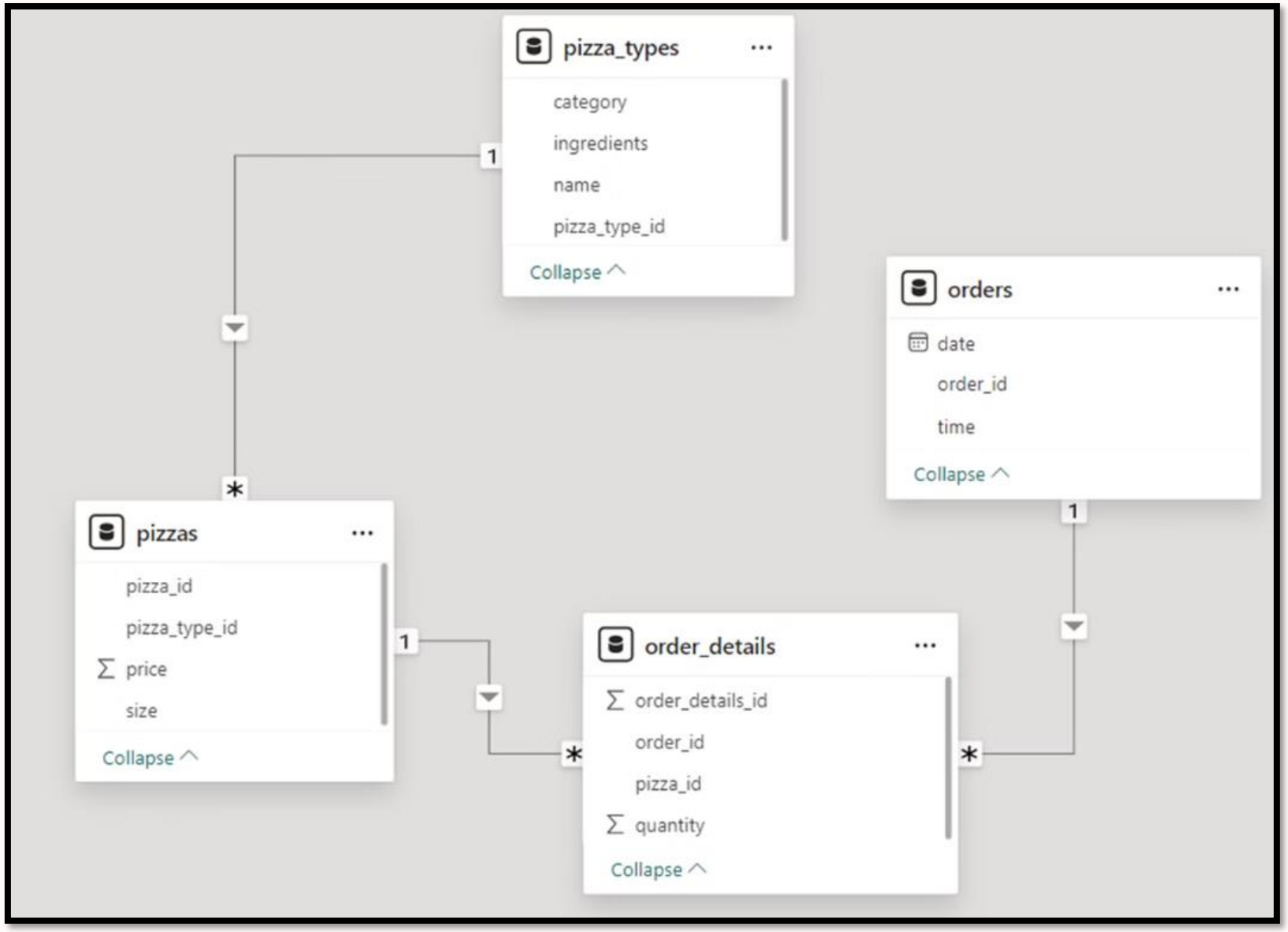
pizza_type_id	text
name	text
category	text
ingredients	text

Table:orders_details

order_details_id	int PK
order_id	int
pizza_id	text
quantity	int

bsakshi2019@gmail.com

ENTITY RELATIONSHIPS



bsakshi2019@gmail.com

SQL QUERIES FOR DATA EXPLORATION

1 **Top-Selling Pizzas**

Identify the most popular pizza types and toppings by analyzing order data.

2 **Customer Segmentation**

Classify customers based on factors like order frequency, average order value, and demographic data.

3 **Inventory Insights**

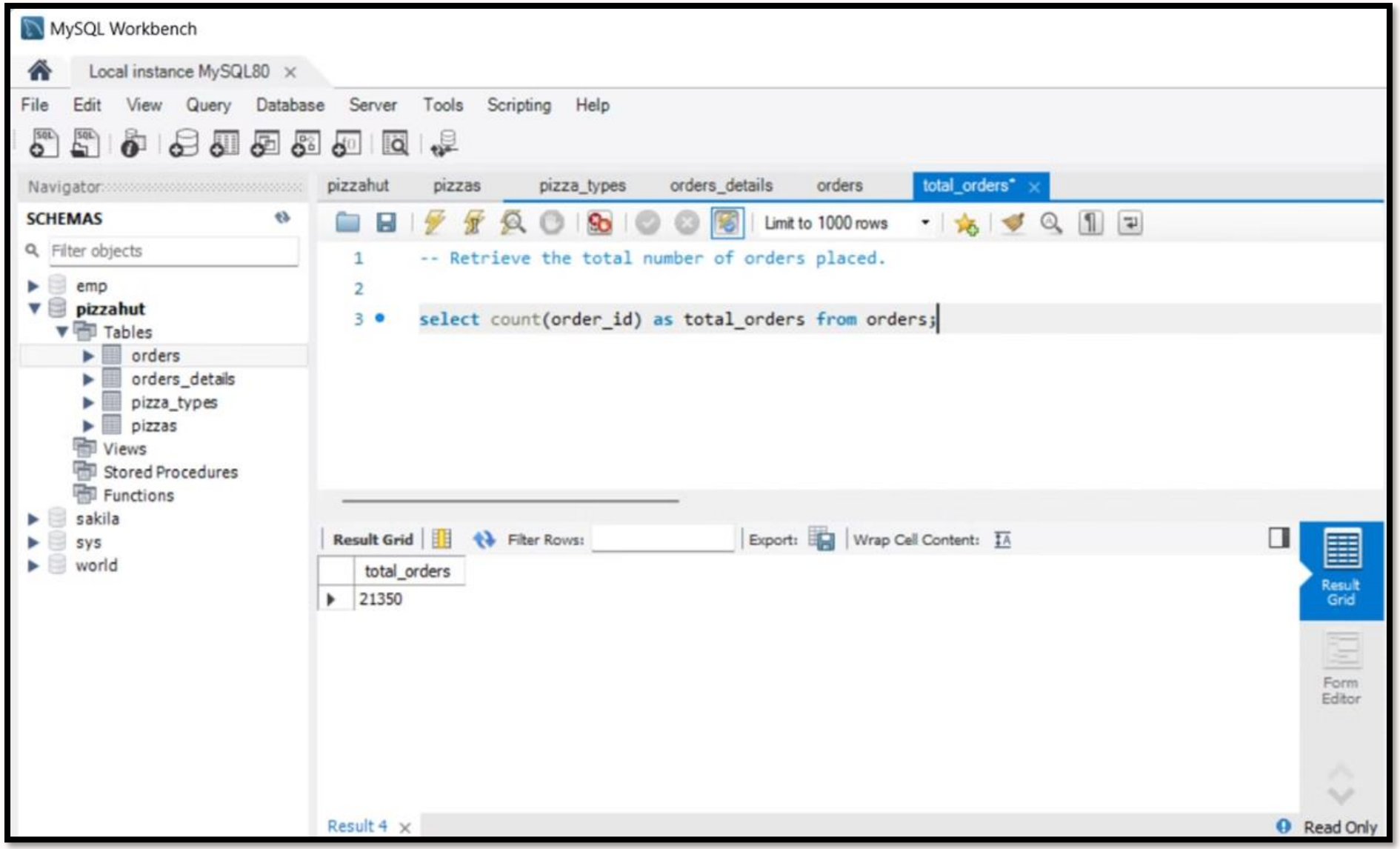
Monitor pizza quantity, order by hour, identify pizza sales, and revenue.

4 **Operational Efficiency**

Analyze order processing times, quantity, size, price and pizza category.

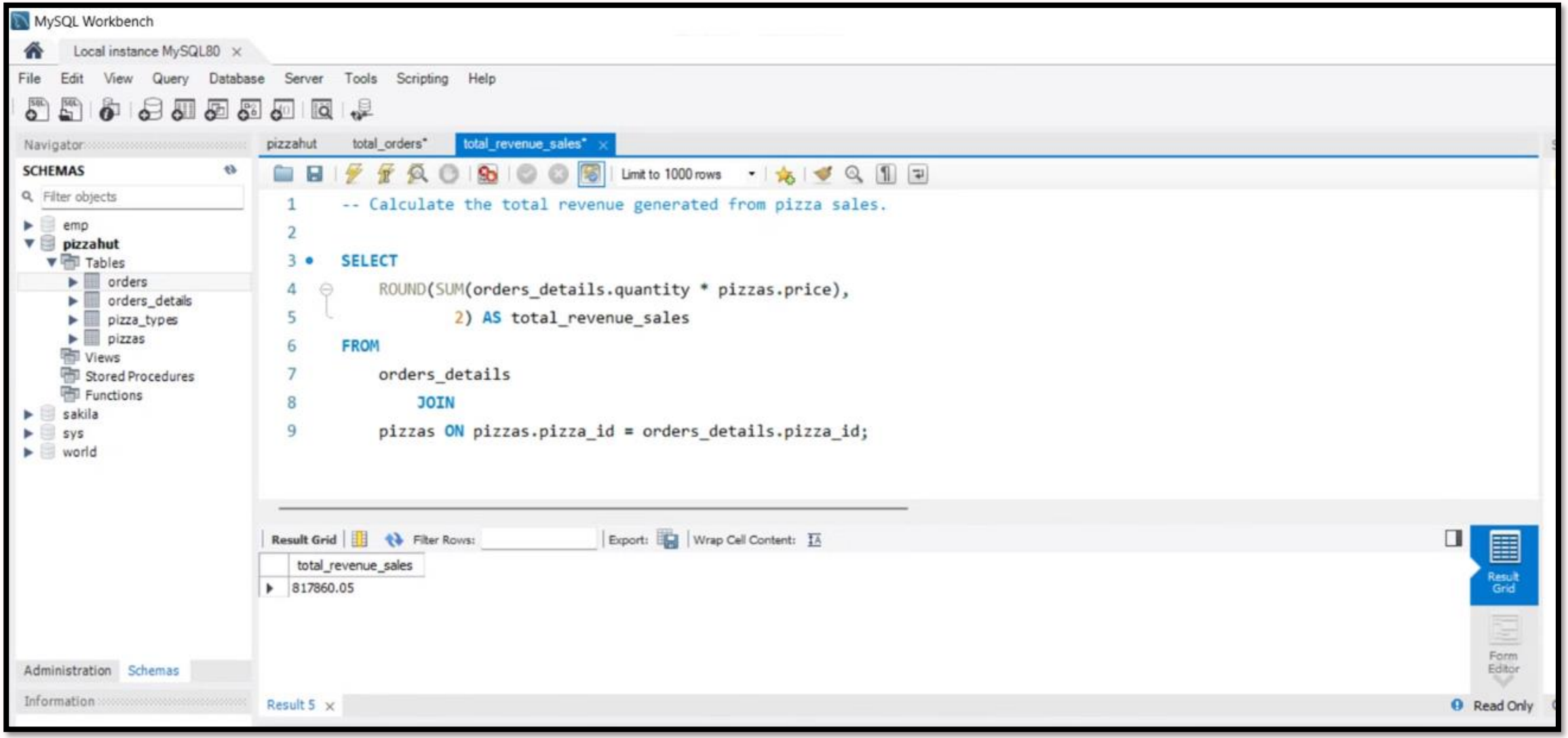
bsakshi2019@gmail.com

Query : Retrieve the total number of orders placed.



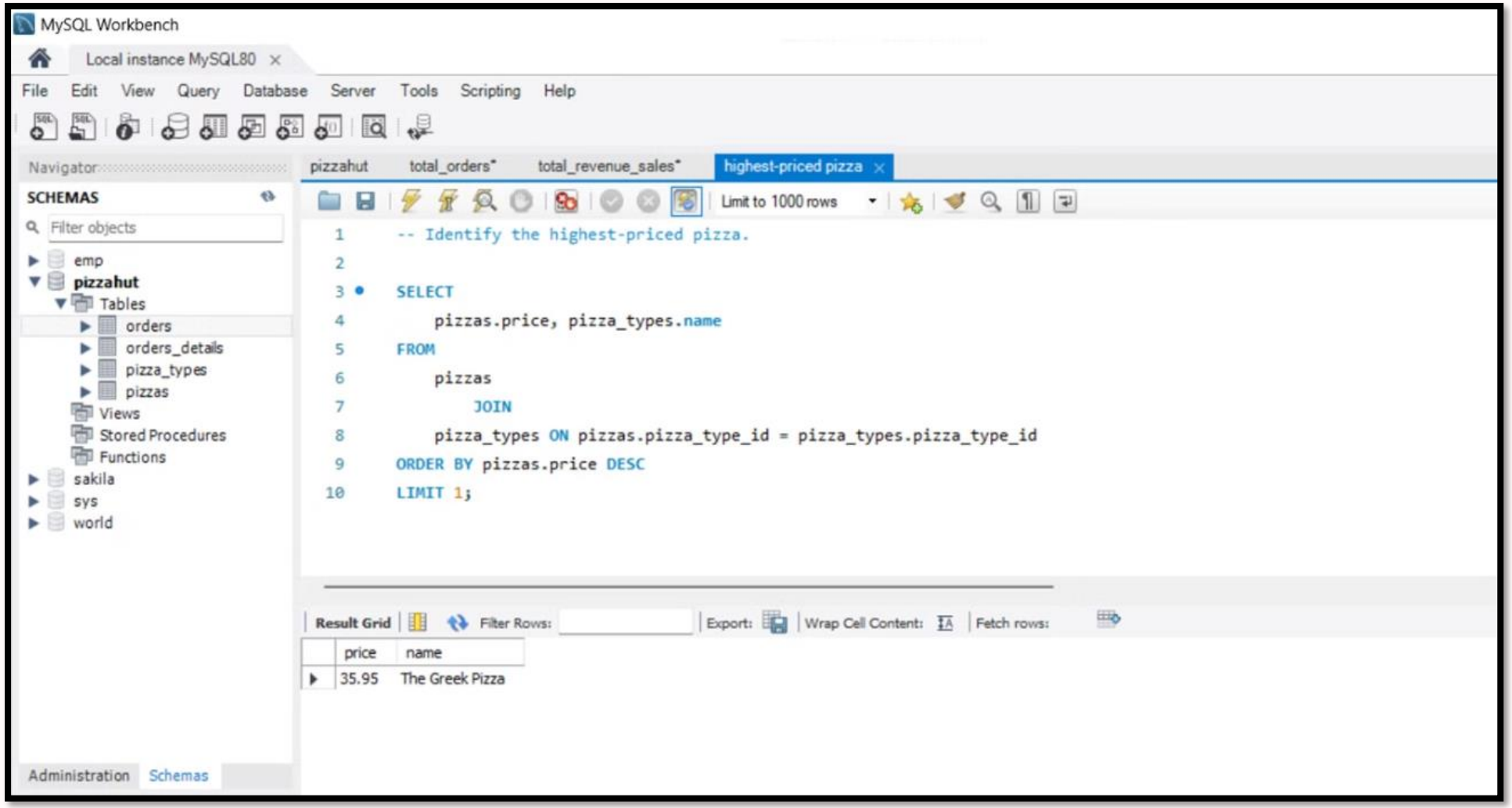
bsakshi2019@gmail.com

Query : Calculate the total revenue generated from pizza sales.



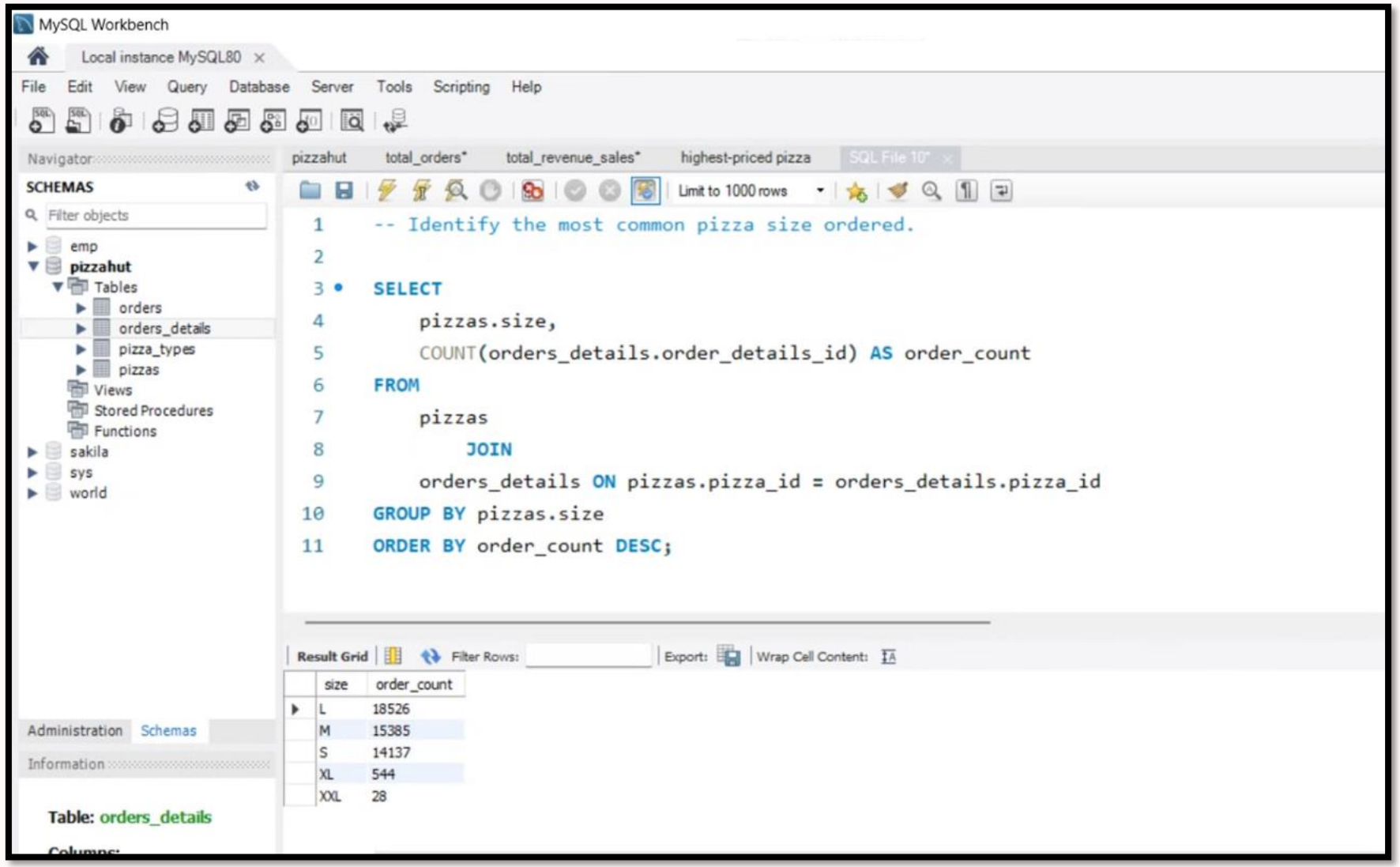
bsakshi2019@gmail.com

Query : Identify the highest-priced pizza



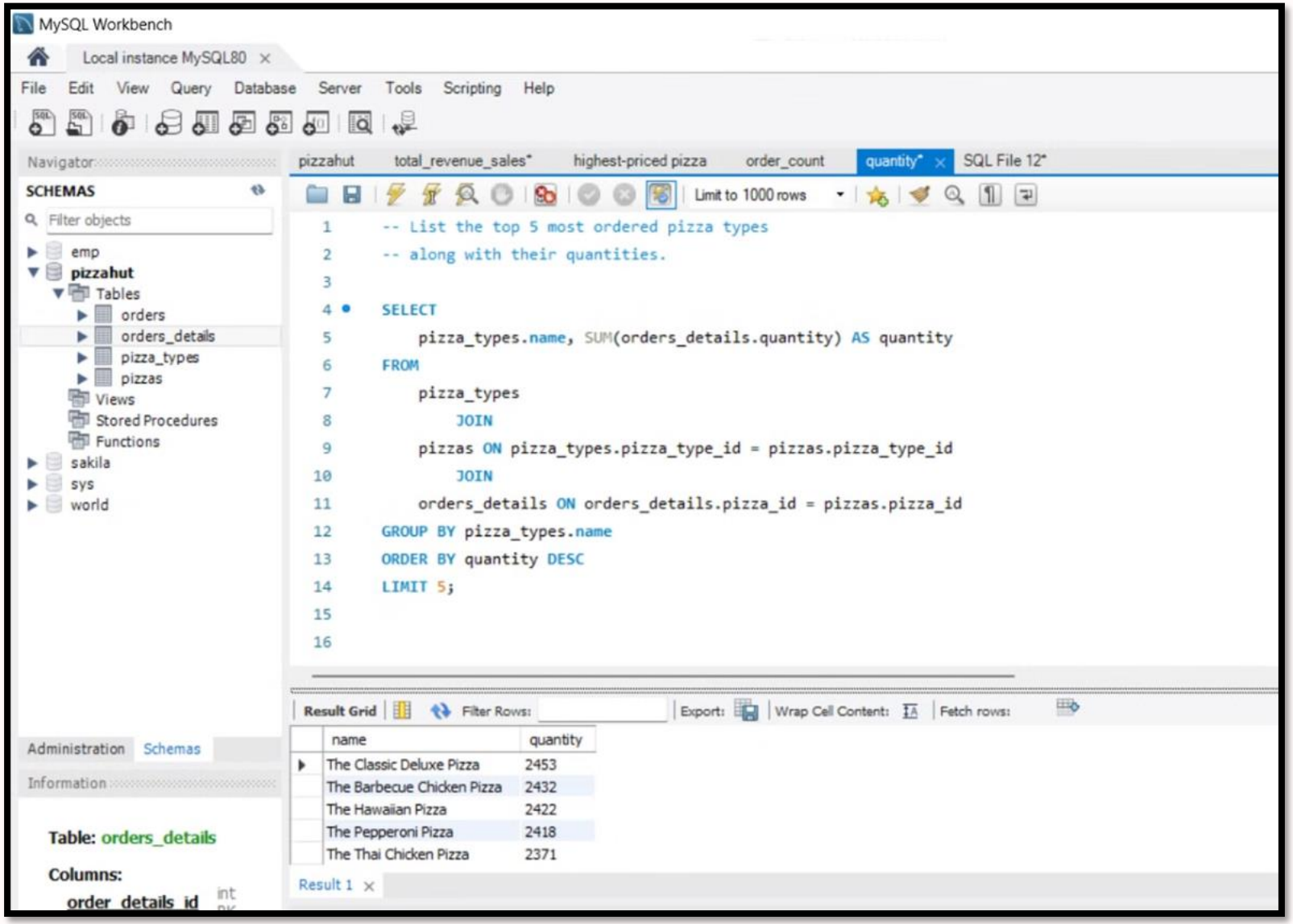
bsakshi2019@gmail.com

Query : Identify the most common pizza size ordered.



bsakshi2019@gmail.com

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



bsakshi2019@gmail.com

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

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- emp
- pizzahut
  - Tables
    - orders
    - orders_details
    - pizza_types
    - pizzas
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

Information

Table: orders_details

Columns:

pizzahut | total_revenue_sales* | highest-priced pizza | order_count | quantity* | category_quantity*

Limit to 1000 rows

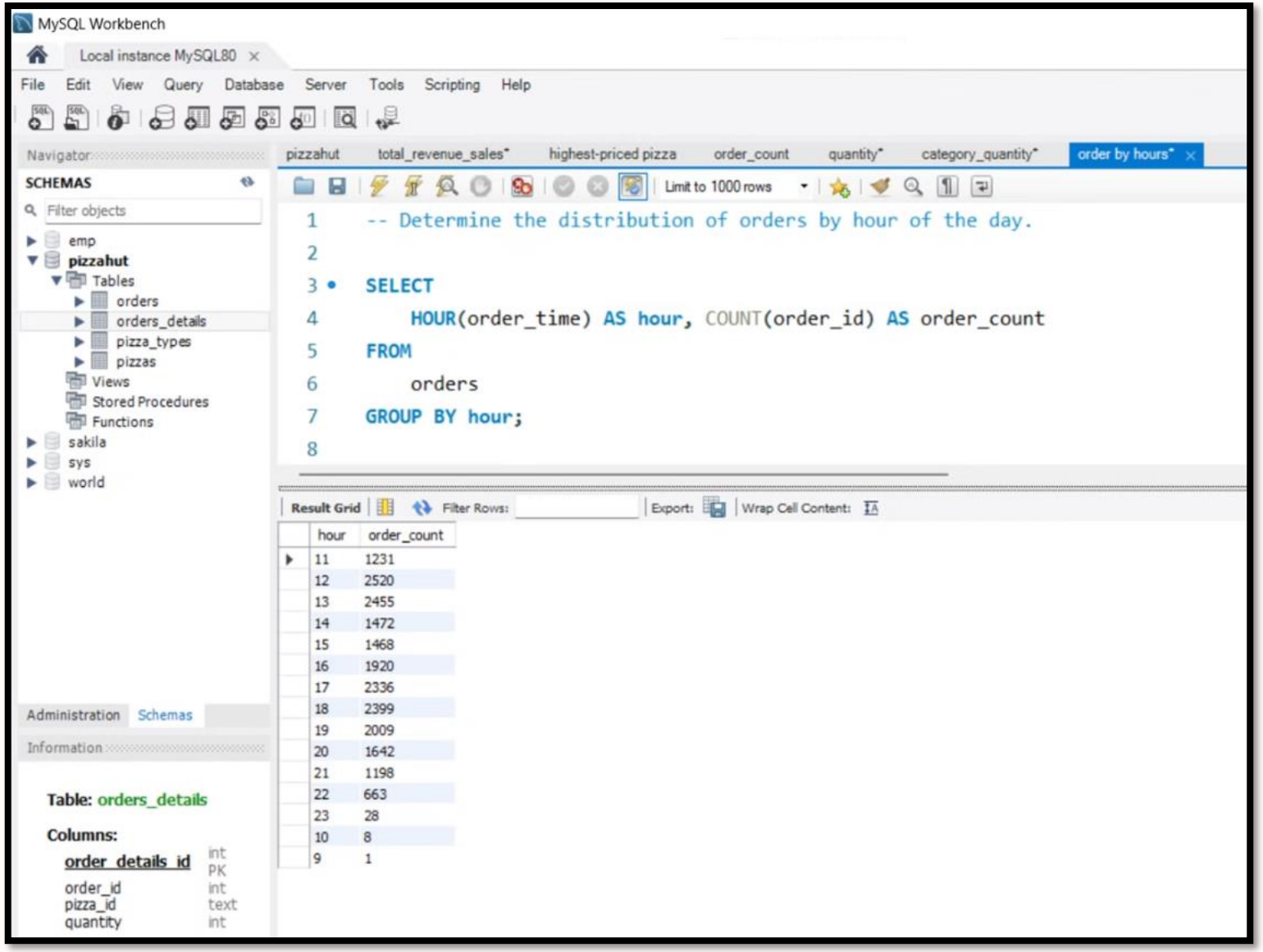
```
-- total quantity of each pizza category ordered.

SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS category_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY category_quantity DESC;
```

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

category	category_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

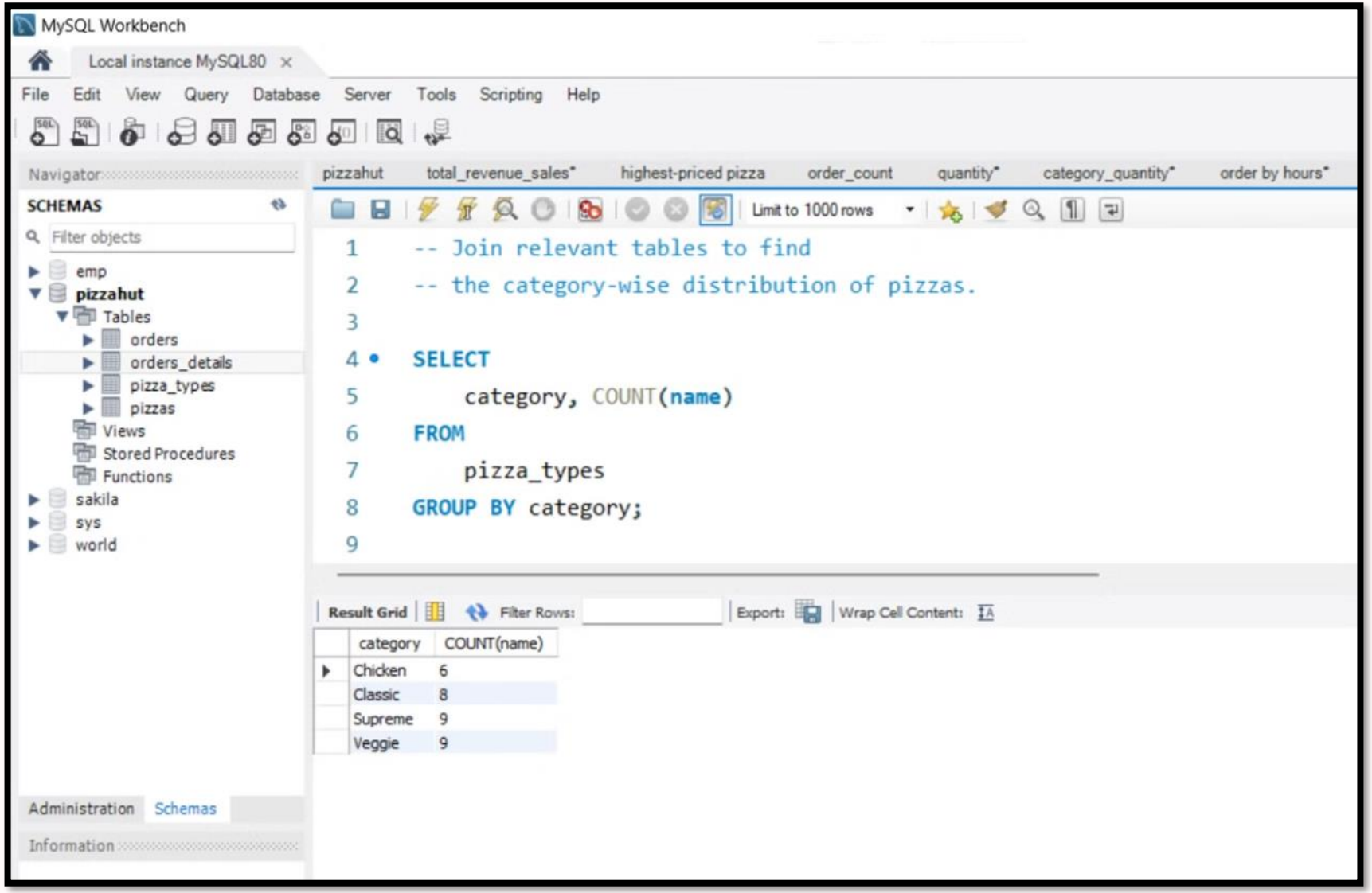
bsakshi2019@gmail.com

Query : Determine the distribution of orders by hour of the day.



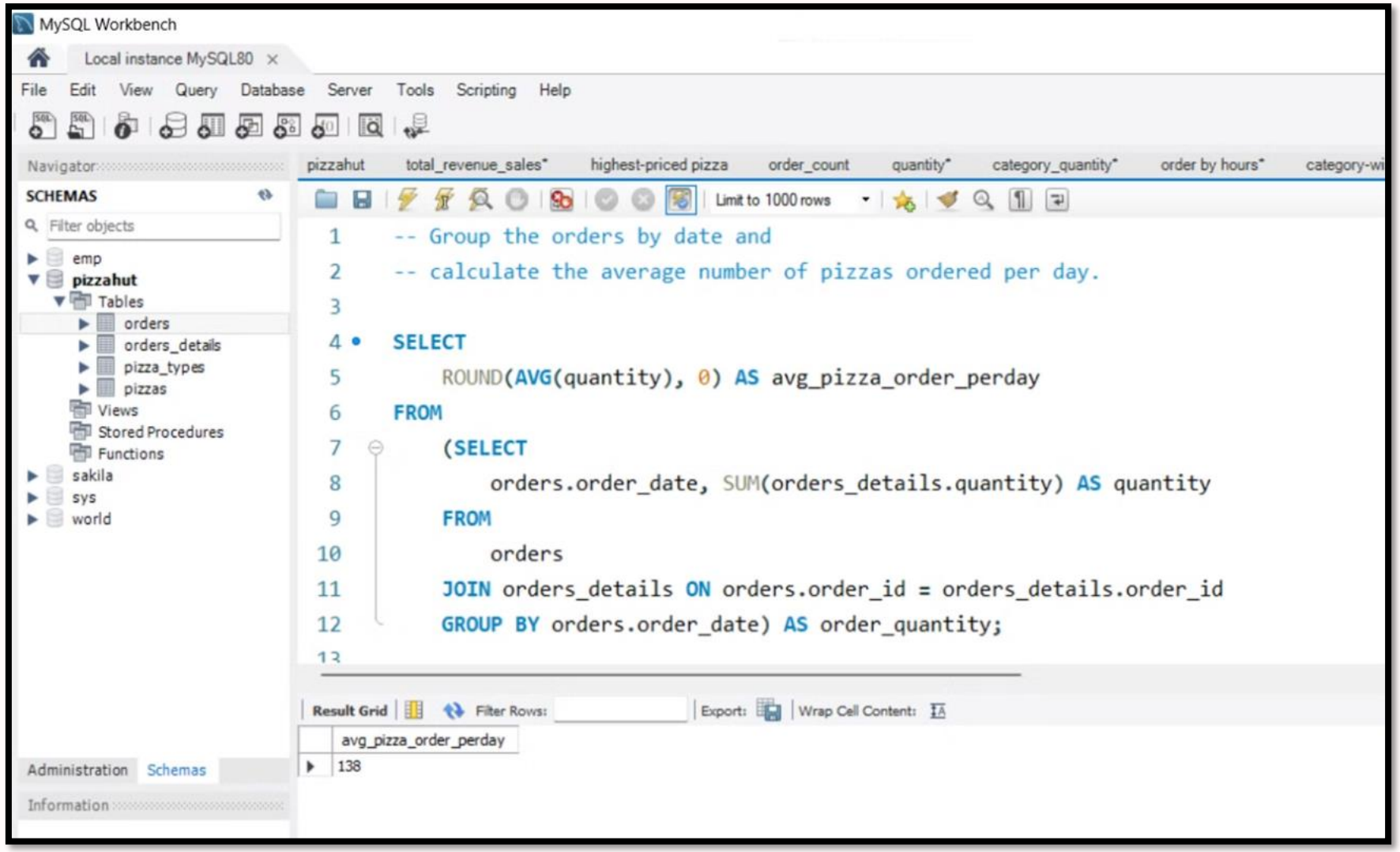
bsakshi2019@gmail.com

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



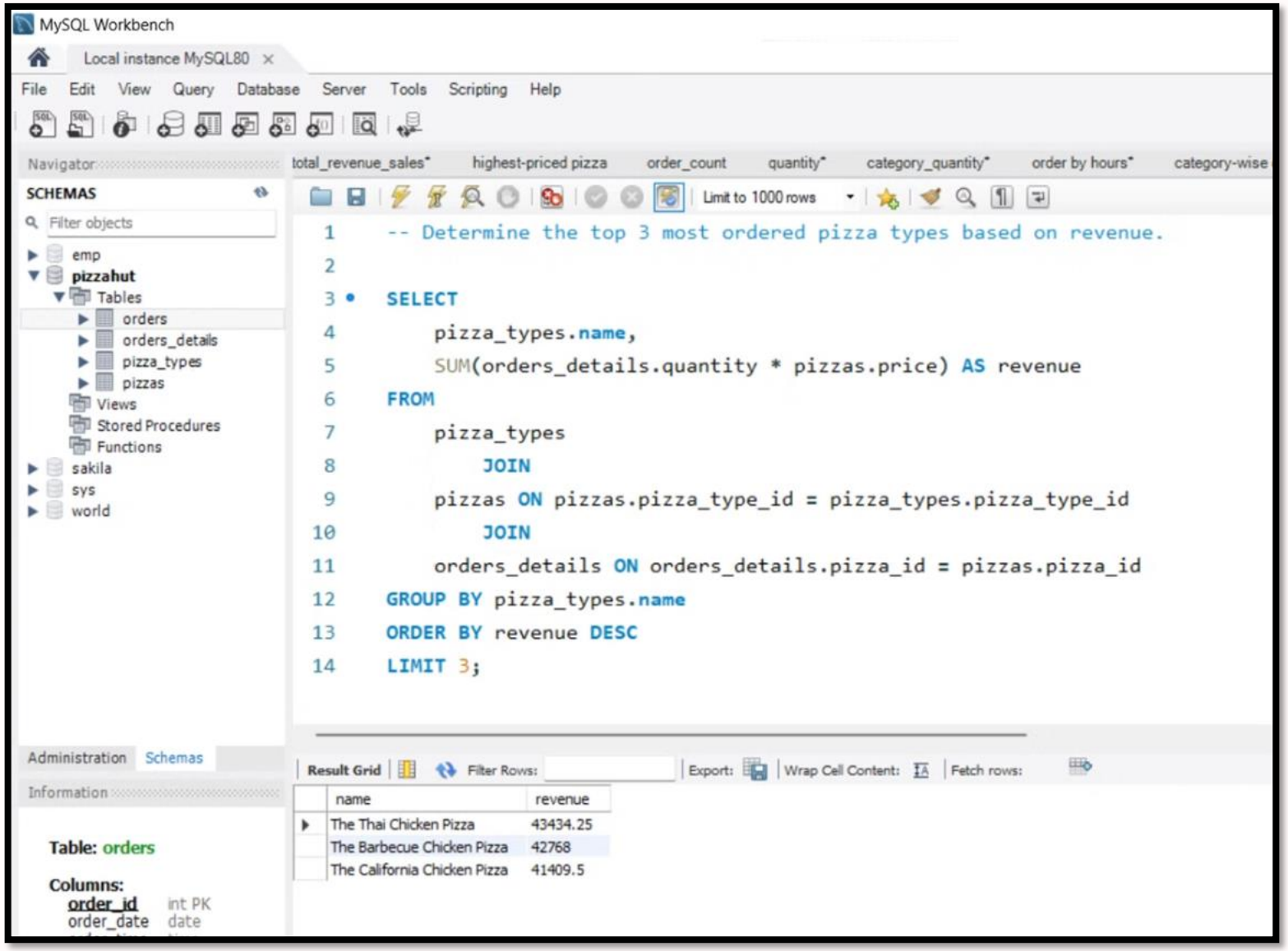
bsakshi2019@gmail.com

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



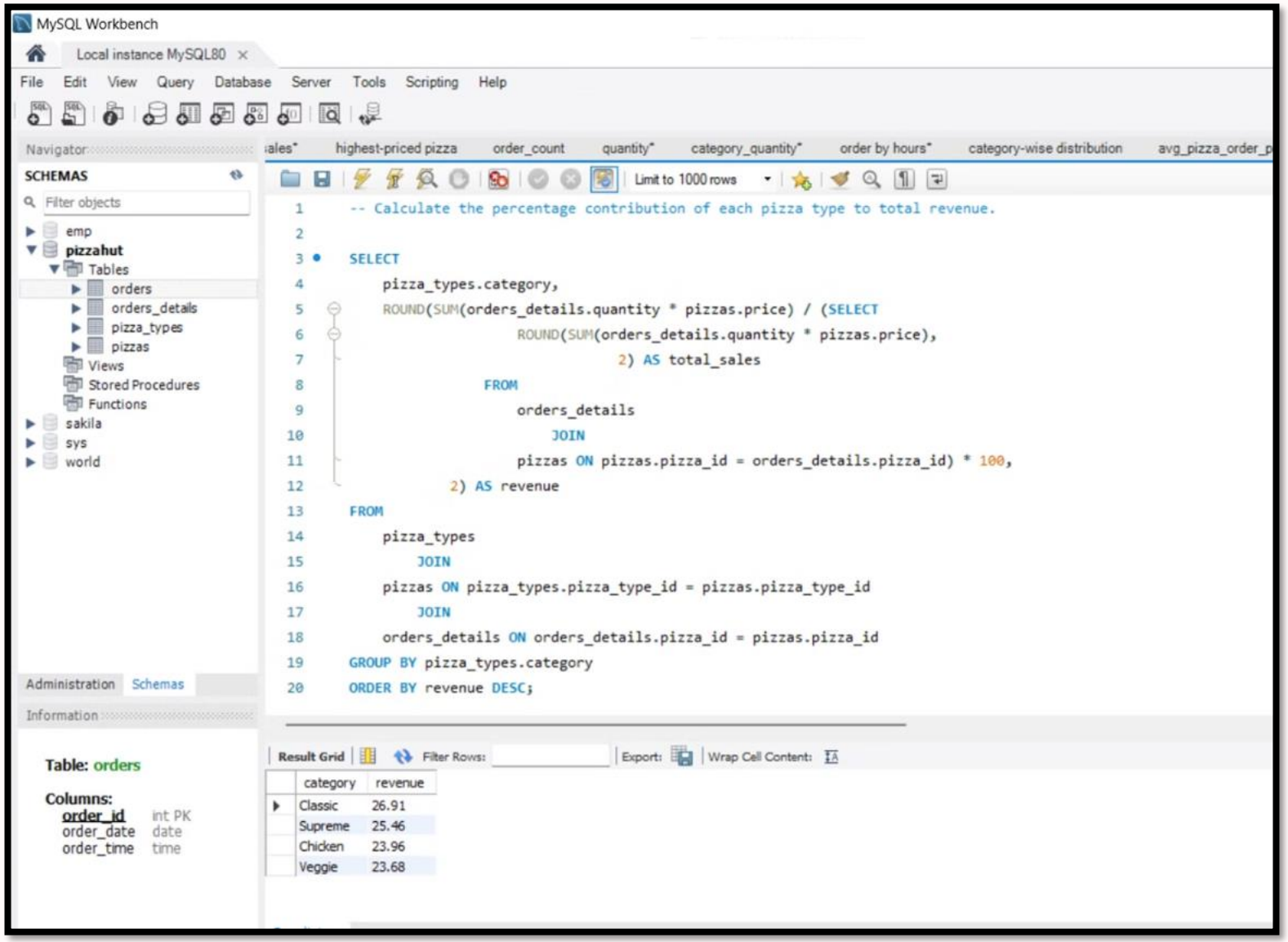
bsakshi2019@gmail.com

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



bsakshi2019@gmail.com

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



bsakshi2019@gmail.com

Query : 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

- emp
- pizzahut
  - Tables
    - orders
    - orders_details
    - pizza_types
    - pizzas
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

Information

Table: orders

Columns:

order_id	int PK
order_date	date
order_time	time

order_count | quantity* | category_quantity* | order by hours* | category-wise distribution | avg_pizza_order_perday | reve

Limit to 1000 rows

```
-- Analyze the cumulative revenue generated over time.

select order_date,
sum(revenue)
over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity*pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id=pizzas.pizza_id
join orders
on orders.order_id=orders_details.order_id
group by orders.order_date) as sales;
```

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

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

Result 1

bsakshi2019@gmail.com

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

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- emp
- pizzahut
  - Tables
    - orders
    - orders_details
    - pizza_types
    - pizzas
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

Information

Table: orders

Columns:

order_id	int PK
order_date	date
order_time	time

quantity* category_quantity* order by hours* category-wise distribution avg_pizza_order_perday revenue* percenta

Limit to 1000 rows

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

select name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category,pizza_types.name,
sum((orders_details.quantity)*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id=pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn<=3;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

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
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75

Result 1

bsakshi2019@gmail.com

RESULT - ANALYZING PIZZA SALES DATA

- Total number of orders placed (21350)
- Total revenue generated from pizza sales (817860.05)
- Identification of the highest-priced and most commonly ordered pizza types (pizza name- The Greek Pizza, Price- 35.95)
- The most common pizza size ordered (size of Large with order count of 18526)
- The most ordering pizza type quantities (name- The classis deluxe pizza with quantity of 2453)
- Average number of pizzas ordered per day (138)
- Determination of the top pizza types by revenue (The Thai chicken pizza of revenue- 4343.25)
- Evaluation of the percentage contribution of each pizza type to total revenue.
- Examination of the cumulative revenue over time.

bsakshi2019@gmail.com

CONCLUSION

By analyzing this data,
we have gain valuable insights to inform business decisions
and optimize our pizza operations.

bsakshi2019@gmail.com

Follow my GitHub for complete SQL project and it's used raw database:
https://github.com/bsakshi2019/Pizzahut-sales--sql-project.git

Reference:
pizza-sales---SQL/Questions.txt at main · Ayushi0214/pizza-sales---SQL · GitHub

bsakshi2019@gmail.com