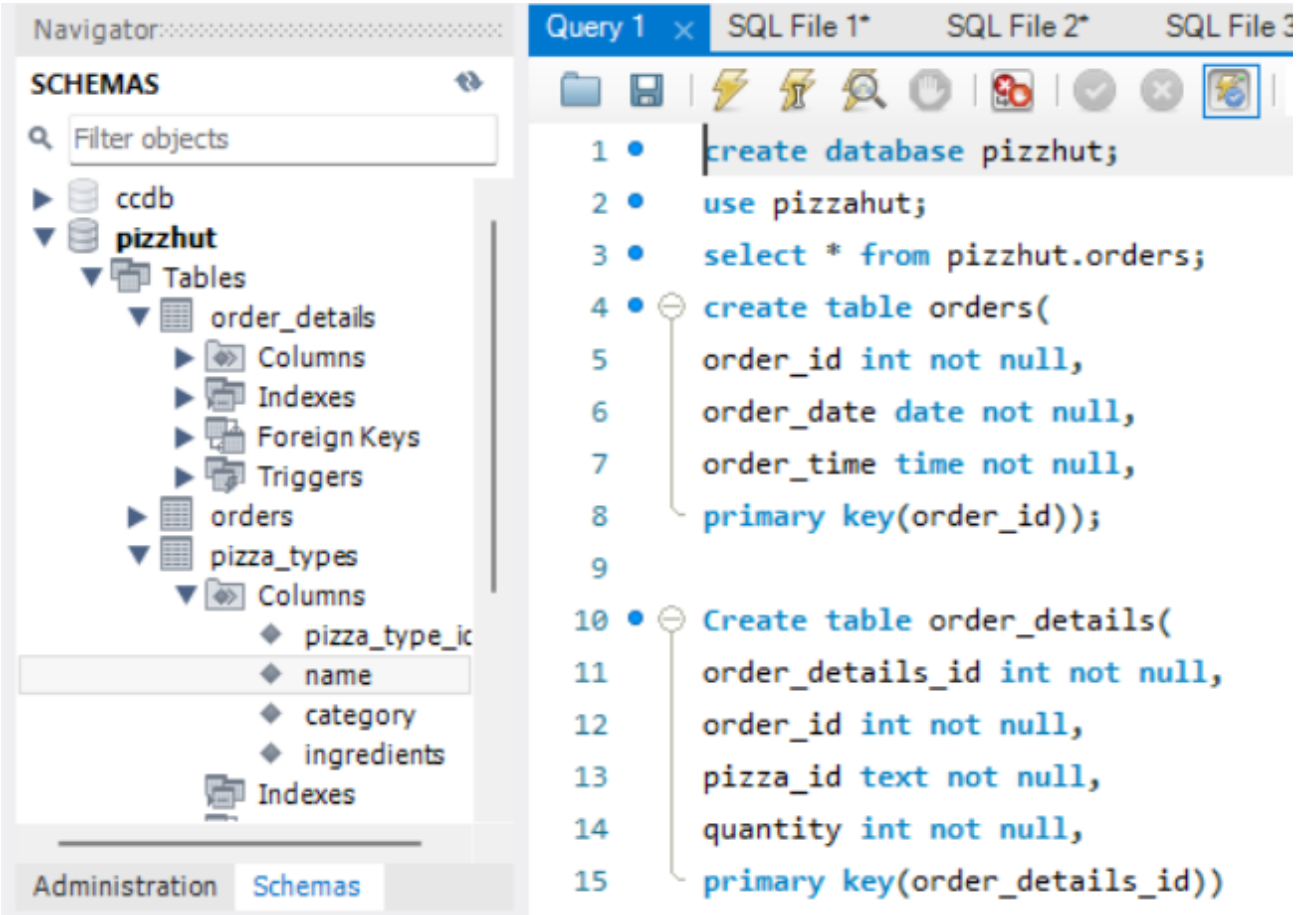
SQL PORTFOLIO PROJECT

The project includes a series of tasks that involve data extraction, transformation, and analysis using SQL queries.

Create Database and Tables



- Retrieve the total number of orders placed

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

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

count(*)
21350

-- Calculate the total revenue generated from pizza sales.

```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid | Filter

	revenue
▶	817860.05

-- Identify the highest-priced pizza.

```sql
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 | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

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
LIMIT 1;
```

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

size	order_count
L	18526

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

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

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

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

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

```
SELECT
    pizza_types.category,
    sum(order_details.quantity) AS quantity
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
order by quantity desc;
```

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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

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

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28

Result 4

Output

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

```
SELECT
    pizza_types.category, COUNT(pizza_types.name)
FROM
    pizza_types
GROUP BY pizza_types.category
```

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

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

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

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_order_perday
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;
```

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

avg_pizza_order_perday
138

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

```sql
select pizza_types.name ,sum(order_details.quantity*pizzas.price) as revenue
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 revenue desc limit 3;
```

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

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

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

```sql
WITH TotalRevenue AS (
    SELECT
        ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_sales
    FROM
        order_details
    JOIN
        pizzas ON pizzas.pizza_id = order_details.pizza_id
),
RevenuePerType AS (
    SELECT
        pizza_types.category,
        ROUND(SUM(order_details.quantity * pizzas.price), 2) AS type_revenue
    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
)
```

```sql
SELECT
    r.category,
    r.type_revenue,
    t.total_sales,
    ROUND((r.type_revenue / t.total_sales) * 100, 2) AS revenue_percentage
FROM
    RevenuePerType r
CROSS JOIN
    TotalRevenue t
ORDER BY
    revenue_percentage DESC;
```

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

category	type_revenue	total_sales	revenue_percentage
Classic	220053.1	817860.05	26.91
Supreme	208197	817860.05	25.46
Chicken	195919.5	817860.05	23.96
Veggie	193690.45	817860.05	23.68

Analyze the cumulative revenue generated over time.

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

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

order_date	cumm_revenue
2015-01-01	2713.850000000004
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

Result 1

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

```sql
with Revenueperpizza as
(select
        pizza_types.category,
        pizza_types.name,
        SUM(order_details.quantity * pizzas.price) AS revenue
    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,
        pizza_types.name)
,RankedPizza as
    (select category,name,revenue,rank()over(partition by category order by revenue desc) as rn
    from Revenueperpizza )

select category,name,revenue from
    RankedPizza
    where rn<=3
    order by category,rn;
```

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

category	name	revenue
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Result 5