PIZZA SALES

USING SQL

In this project i have utilizes SQL Quires to solve questions that were related to Pizza Sales

BY SAURABH SALADE

Order_Details Table

Column Name	Data Type
order_details_id	int
order_id	int
pizza_id	string
quantity	int

Orders table

Column Name	Data Type
Order_id	int
order_date	date
order_time	time

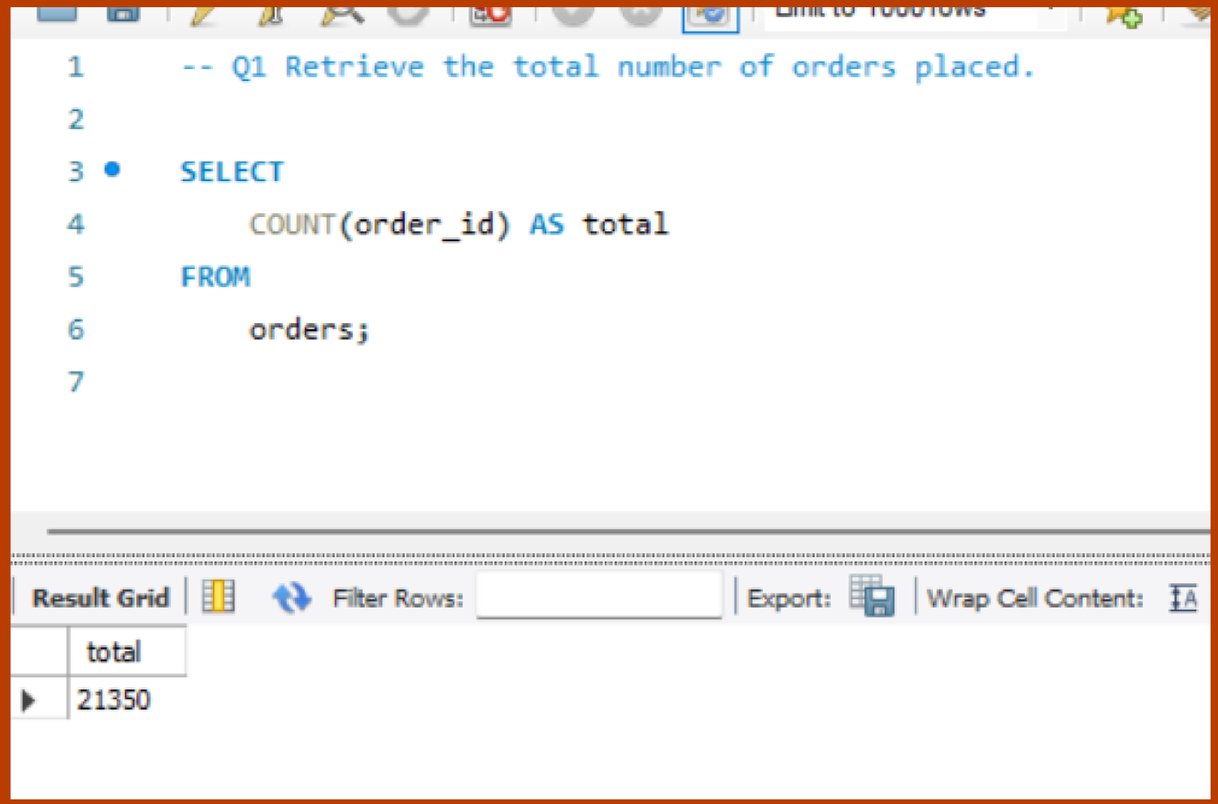
Pizza table

Column Name	Data Type
pizza_id	string
pizza_type_id	string
size	string
price	float

Pizza_types table

Column Name	Data Type
pizza_type_id	string
name	string
category	string
ingredients	string

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
-- Q2 Calculate the total revenue generated from pizza sales.

SELECT
    ROUND(SUM(quantity * price), 2) as total_revenue
FROM
    orders_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id;
```

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

total_revenue
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
-- Q Identify the highest-priced pizza.

SELECT
    pt.name, p.price
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY price DESC
LIMIT 1;

-- or using window function

with temp as
(select dense_rank() over(order by price desc ) as rn from pizza_types pt join  pizzas p on pt.pizza_type_id=p.pizza_type_id)
Select p.name from temp where rn=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.

```sql
-- Q4 Identify the most common pizza size ordered.
SELECT
    p.size, SUM(od.quantity) AS s
FROM
    orders o
        JOIN
    orders_details od ON o.order_id = od.order_id
        JOIN
    pizzas p ON p.pizza_id = od.pizza_id
GROUP BY p.size
ORDER BY s DESC
LIMIT 1;

-- or using window

with temp as
 (select p.size, sum(od.quantity) as s,  dense_rank () over(order by sum(quantity) desc) as rn
 from  orders o join orders_details od on o.order_id=od.order_id join pizzas p on p.pizza_id=od.pizza_id group by p.size)
 Select size, s from temp where rn=1;
```

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

size	s
L	18956

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
-- Q List the top 5 most ordered pizza types along with their quantities.
SELECT
    name, SUM(quantity) AS c
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    orders_details od ON od.pizza_id = p.pizza_id
GROUP BY name
ORDER BY c DESC
LIMIT 5;
-- using Window
with temp as
(select name, sum(quantity)as s, dense_rank() over(order by sum(quantity) desc)as rn
 from pizza_types pt join pizzas p on pt.pizza_type_id = p.pizza_type_id join orders_details od on od.pizza_id=p.pizza_id
 group by name) Select name, s from temp where rn between 1 and 5;
```

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

name	s
he Classic Deluxe Pizza	2453
he Barbecue Chicken Pizza	2432
he Hawaiian Pizza	2422
he Pepperoni Pizza	2418
he Thai Chicken Pizza	2371

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
-- Q Join the necessary tables to find the total quantity of each pizza category ordered.
SELECT
    category, SUM(quantity) AS s
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    orders_details od ON od.pizza_id = p.pizza_id
GROUP BY category
ORDER BY s;

 -- or using window

 select category, s  from
(select category, sum(quantity) as s, dense_rank() over(order by sum(quantity)) as rn from pizza_types pt join pizzas p on pt.pizza_type_id = p.pizza_type_id join orders_details
 group by category) as temp
```

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

category	s
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

Result Grid

Form Editor

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
-- Q Determine the distribution of orders by hour of the day.
SELECT
    COUNT(order_time), HOUR(order_time)
FROM
    orders
GROUP BY HOUR(order_time);
```

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

COUNT(order_time)	HOUR(order_time)
1231	11
2520	12
2455	13
1472	14
1468	15
1920	16
2336	17
2399	18
2009	19
1642	20
1198	21
663	22
28	23
8	10
1	9

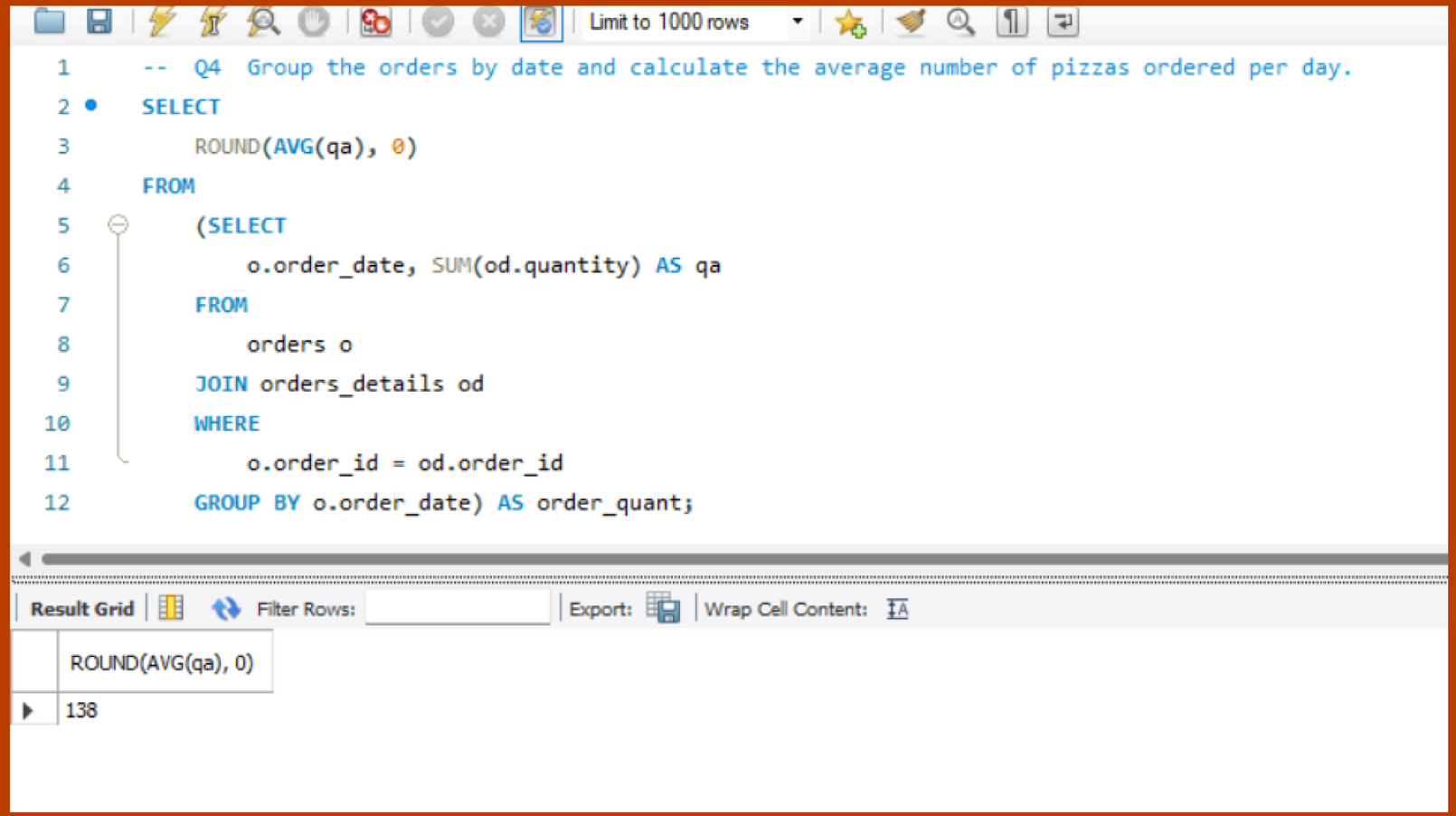
JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```sql
-- Q3 Join relevant tables to find the category-wise distribution of pizzas.
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

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

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

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



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

```sql
-- Q  Determine the top 3 most ordered pizza types based on revenue.
SELECT
    name, SUM(quantity * price) AS revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    orders_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.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
-- Q Calculate the percentage contribution of each pizza type to total revenue.

-- STEP 1, Calculate Total Revenue fir each pizza

With pizza_revenue as
(Select pt.category, SUM(o.quantity*p.price) as revenue
from pizza_types pt join pizzas p on pt.pizza_type_id=p.pizza_type_id
join orders_details o on o.pizza_id= p.pizza_id group by pt.category order by revenue desc),

-- STEP 2 calculate the overall revenue
total_revenue as(
select SUM(revenue) as total_revenue from pizza_revenue)

-- STEP 3 compute the % contribution

Select pr.category, pr.revenue,
(pr.revenue/tr.total_revenue)*100 As percent_contribution
from pizza_revenue pr,total_revenue tr
order by percent_contribution;
```

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

category	revenue	percent_contribution
Veggie	193690.45000000298	23.68259092738454
Chicken	195919.5	23.955137556847248
Supreme	208196.99999999822	25.45631126009858
Classic	220053.1000000001	26.905960255669626

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

```sql
-- Q2 Determine the top 3 most ordered pizza types based on revenue for each pizza category.
Select name, revenue, category from
(select category, name, revenue, dense_rank() over(partition by category order by revenue desc)
as rn from

(Select pt.category,pt.name, sum((od.quantity)*p.price) as revenue
from pizza_types pt join pizzas p on pt.pizza_type_id=p.pizza_type_id
join orders_details od on od.pizza_id=p.pizza_id
group by pt.category, pt.name) as x  ) as y  where rn<=3;
```

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

name	revenue	category
The Thai Chicken Pizza	43434.25	Chicken
The Barbecue Chicken Pizza	42768	Chicken
The California Chicken Pizza	41409.5	Chicken
The Classic Deluxe Pizza	38180.5	Classic
The Hawaiian Pizza	32273.25	Classic
The Pepperoni Pizza	30161.75	Classic
The Spicy Italian Pizza	34831.25	Supreme
The Italian Supreme Pizza	33476.75	Supreme
The Sicilian Pizza	30940.5	Supreme
The Four Cheese Pizza	32265.70000000065	Veggie
The Mexicana Pizza	26780.75	Veggie
The Five Cheese Pizza	26066.5	Veggie