PIZZA SALES

Analyzed pizza sales data using SQL to identify trends, customer behavior, and best-selling products.

Retrieve the total number of orders placed.

```
11      -- 1.Retrieve the total number of orders placed.
12
13 •    select count(order_id) as total_orders from orders;
14
```

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

	revenue
▶	817860

Calculate the total revenue generated from pizza sales.

```
15      -- 2. Calculate the total revenue generated from pizza sales.
16
17  •   select round(sum(price*quantity)) as revenue from
18      order_details join pizzas
19      on order_details.pizza_id=pizzas.pizza_id;
20
```

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

	revenue
▶	817860

Identify the highest-priced pizza.

```
23 •    select pizza_types.name,pizzas.price from
24      pizza_types join pizzas
25      on pizza_types.pizza_type_id=pizzas.pizza_type_id
26      order by pizzas.price desc limit 1;
27
```

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

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```
28      -- 4.Identify the most common pizza size ordered.
29
30  •   select size,count(size) as size_count from pizzas
31      group by size order by size_count desc limit 1;
32
```

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

	size	size_count
▶	S	32

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

```
33      -- 5. Join the necessary tables to find the total quantity of each pizza category ordered.
34
35  •   select pizza_types.category,sum(order_details.quantity) as quantity
36      from pizza_types join pizzas
37      on pizza_types.pizza_type_id=pizzas.pizza_type_id
38      join order_details
39      on pizzas.pizza_id = order_details.pizza_id
40      group by pizza_types.category order by quantity;
```

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

category	quantity
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

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

```
45 •  select pizza_types.name,sum(order_details.quantity) as most_ord_type
46    from pizza_types join pizzas
47    on pizza_types.pizza_type_id = pizzas.pizza_type_id join order_details
48    on order_details.pizza_id = pizzas.pizza_id
49    group by pizza_types.name order by most_ord_type desc limit 5;
50
```

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

name	most_ord_type
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418

Determine the distribution of orders by hour of the day.

```
52      -- 7.Determine the distribution of orders by hour of the day.
53
54 •    select hour(time) as hours, count(order_id) as order_count
55      from orders group by hours order by order_count;
```

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

hours	order_count
9	1
10	8
23	28
22	663
21	1198
11	1231
15	1468
14	1472

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

```
59 •  select category,count(category) as category_count
60    from pizza_types group by category order by category_count;
```

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

	category	category_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
select avg(quantity_ordered) from  (select orders.date,sum(quantity) as quantity_ordered
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.date order by quantity_ordered desc) as avg_quant_ord;
```

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

avg(quantity_ordered)
138.4749

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

```
72 •  select pizza_types.name,sum(pizzas.price*order_details.quantity) as revenue
73    from pizza_types join pizzas
74    on pizza_types.pizza_type_id=pizzas.pizza_type_id
75    join order_details
76    on pizzas.pizza_id= order_details.pizza_id
77    group by pizza_types.name order by revenue desc limit 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

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

```
79      -- 11.Calculate the percentage contribution of each pizza type to total revenue.
80
81  •   select pizza_types.name,(sum(order_details.quantity*pizzas.price)/(select sum(order_details.quantity*pizzas.price) as
82      order_details join pizzas on order_details.pizza_id=pizzas.pizza_id))*100 as percent_contri
83
84      from pizza_types join pizzas
85      on pizza_types.pizza_type_id=pizzas.pizza_type_id
86      join order_details
87      on pizzas.pizza_id=order_details.pizza_id
88      group by pizza_types.name order by percent_contri desc;
```

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

name	percent_contri
The Thai Chicken Pizza	5.310719113863108
The Barbecue Chicken Pizza	5.2292565213327595
The California Chicken Pizza	5.063152308270878
The Classic Deluxe Pizza	4.668341484585331
The Spicy Italian Pizza	4.258827656394306
The Southwest Chicken Pizza	4.243482732773205
The Italian Supreme Pizza	4.093212524563375

Result Grid

Form Editor

Analyze the cumulative revenue generated over time.

```
90      -- 12.Analyze the cumulative revenue generated over time.
91
92  •   select date,sum(revenue) over(order by date) as cumulative_rev from
93
94      (select orders.date,sum(order_details.quantity*pizzas.price) as revenue
95      from order_details join pizzas
96      on order_details.pizza_id=pizzas.pizza_id
97      join orders on orders.order_id=order_details.order_id
98      group by orders.date) as a ;
99
```

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

date	cumulative_rev
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

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

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

select name, category,rank() over(partition by category order by revenue) as top_rev_pizzas
from
(select pizza_types.name,pizza_types.category,sum(pizzas.price*order_details.quantity) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.name,pizza_types.category) as a;
```

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

name	category	top_rev_pizzas
The Chicken Pesto Pizza	Chicken	1
The Chicken Alfredo Pizza	Chicken	2
The Southwest Chicken Pizza	Chicken	3
The California Chicken Pizza	Chicken	4
The Barbecue Chicken Pizza	Chicken	5
The Thai Chicken Pizza	Chicken	6
The Pepperoni, Mushroom, and Peppers Pizza	Classic	1

PIZZA SALES

Utilized WINDOW FUNCTIONS, SELECT, JOIN, WHERE, GROUP BY, and ORDER BY clauses to filter, aggregate, and sort data

Identified insights and trends from datasets, Defined and addressed business questions through data exploration.