PIZZA

SALES ANALYSIS

Objective

A purpose of this project is a examine the data set with sql and understand business of pizza sales growth by answering simple questions.

Q.1 Retrieve the total number of orders placed.

```
3 •  SELECT
4        COUNT(order_id) AS total_orders
5    FROM
6        orders;
```

Result Grid | Filter Rows: | Export:

	total_orders
▶	21350

Q.2 Calculate the total revenue generated from pizza sales.

```
3    SELECT
4        ROUND(SUM(order_details.quantity * pizzas.price),
5                2) AS total_revenue
6    FROM
7        order_details
8            JOIN
9        pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

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

	total_revenue
▶	817860.05

Q.3 Identify the highest-priced pizza.

```
3 •  select pizza_types.name, pizzas.price
4    from pizza_types join pizzas
5    on pizza_types.pizza_type_id = pizzas.pizza_type_id
6    order by pizzas.price desc
7    limit 1;
```

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

	name	price
▶	The Greek Pizza	35.95

Q.4 Identify the most common pizza size ordered.

```sql
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;
```

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

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

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

```sql
SELECT pizza_types.name,
SUM(order_details.quantity) AS total_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.name
ORDER BY total_quantity DESC
LIMIT 5;
```

name	total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q.6 Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
SELECT pizza_types.category,
SUM(order_details.quantity) AS total_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 total_quantity DESC;
```

Result Grid | Filter Rows:

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q.7 Determine the distribution of orders by hour of the day.

```
3 •    SELECT
4          HOUR(order_time) AS hour, COUNT(order_id) AS order_count
5      FROM
6          orders
7      GROUP BY HOUR(order_time);
```

Result Grid | Filter

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009

Result 1

Q.8 Join relevant tables to find the category-wise distribution of pizzas.

```
3 •    SELECT
4          category, COUNT(name)
5      FROM
6          pizza_types
7      GROUP BY category;
```

Result Grid | Filter Rows:

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

Q.9 Group the orders by date and calculate the average number of pizzas ordered per day.

```
4 •   select round(avg(quantity), 0) from
5     (select orders.order_date, sum(order_details.quantity) as quantity
6     from orders join order_details
7     on orders.order_id = order_details.order_id
8     group by orders.order_date) as order_quantity;
```

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

	round(avg(quantity), 0)
▶	138

Q.10 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 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;
```

Result Grid | Filter Rows:

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

Q.11 Calculate the percentage contribution of each pizza type to total revenue.

```sql
select pizza_types.category,
round(sum(order_details.quantity * pizzas.price) / (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)*100,2) 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
order by revenue desc;
```

Result Grid | Filter

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q.12 Analyze the cumulative revenue generated over time.

```
3   select order_date,
4   sum(revenue) over(order by order_date) as cum_revenue from
5   (select orders.order_date,
6   sum(order_details.quantity * pizzas.price) as revenue
7   from order_details join pizzas
8   on order_details.pizza_id = pizzas.pizza_id
9   join orders
10  on orders.order_id = order_details.order_id
11  group by orders.order_date) as sales;
```

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

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

```sql
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(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) as a) as b
where rn <= 3;
```

Result Grid | Filter Rows:

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
The Sicilian Pizza	30940.5

Result 1

Thank You