DOMINO'S PIZZA

SALES DATA ASSESSMENT

(Using SQL)

ANKITA BANSAL

OBJECTIVE

The dataset explores the Dominos Sales dataset and aims to drive relevant insights for the company. The data analyst has used SQL for extracting, manipulating and interpreting data to aid decision making.

Total number of orders placed

SQL Query

```
select count(order_id) as Total_Orders
from orders;
```

Output

	Total_Orders
▶	21350

Total revenue generated from pizza sales

SQL Query

```
SELECT ROUND(SUM(orders_details.Quantity * pizzas.price), 2)
AS Revenue_Generated
FROM orders_details
JOIN pizzas
ON pizzas.pizza_id = orders_details.pizza_id;
```

Output

	Revenue_Generated
▶	817860.05

Highest-priced Pizza

SQL Query

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

Output

	name	price
▶	The Greek Pizza	35.95

Most common pizza size ordered

SQL Query

```
select pizzas.size ,
count(orders_details.order_details_id) as order_count
from pizzas
join orders_details
on pizzas.pizza_id= orders_details.pizza_id
group by pizzas.size
order by order_count desc ;
```

Output

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

Top 5 most ordered pizza types along with their quantities

SQL Query

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

Output

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

Total quantity of each pizza category ordered

SQL Query

```sql
SELECT pizza_types.category,
SUM(orders_details.Quantity) AS Total_orders
FROM pizzas
JOIN orders_details ON pizzas.pizza_id = orders_details.pizza_id
JOIN pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY category
ORDER BY Total_orders ;
```

Output

category	Total_orders
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

Distribution of orders by hour of the day

SQL Query

```
SELECT HOUR(order_time) AS Hour, COUNT(order_id) AS Order_Count
FROM orders
GROUP BY HOUR(order_time);
```

Output

	Hour	Order_Count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009

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

SQL Query

```sql
SELECT ROUND(AVG(Quantity), 0) as Average_pizzas
FROM (SELECT orders.order_date, SUM(orders_details.Quantity) AS Quantity
      FROM orders
      JOIN orders_details ON orders.order_id = orders_details.order_id
      GROUP BY orders.order_date) AS Order_Quantity;
```

Output

	Average_pizzas
▶	138

Top 3 most ordered pizza types based on revenue

SQL Query

```sql
SELECT pizza_types.name,
SUM(pizzas.price * orders_details.Quantity) AS Total_Revenue
FROM pizzas
JOIN orders_details
ON orders_details.pizza_id = pizzas.pizza_id
JOIN pizza_types
ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.name
ORDER BY Total_Revenue DESC
LIMIT 3;
```

Output

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

Percentage contribution of each pizza type to total revenue

SQL Query

```
SELECT pizza_types.category, ROUND(SUM(pizzas.price * orders_details.Quantity) /
(SELECT ROUND(SUM(orders_details.Quantity * pizzas.price), 2) AS Total_Sales
FROM orders_details
JOIN pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100, 2) AS Total_Revenue
FROM pizzas
JOIN orders_details ON orders_details.pizza_id = pizzas.pizza_id
JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category
ORDER BY Total_Revenue DESC;
```

Output

	category	Total_Revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Cumulative Revenue generated over time

SQL Query

```
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 pizzas.pizza_id = orders_details.pizza_id
join orders
on orders.order_id = orders_details.order_id
group by orders.order_date) as Sales;
```

Output

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

Top 3 most ordered pizza types based on revenue for each pizza category

SQL Query

```
select name, category ,Revenue
from ( select name, category , Revenue ,
       rank() over (partition by category order by revenue desc ) as rn
       from ( select  pizza_types.name, pizza_types.category ,
              sum(pizzas.price * orders_details.Quantity) as Revenue
              from pizzas
              join pizza_types
               on pizza_types.pizza_type_id = pizzas.pizza_type_id
              join orders_details
               on pizzas.pizza_id = orders_details.pizza_id
              group by pizza_types.category, pizza_types.name) as ak ) as an
where rn in (1,2,3);
```

Output

name	category	Revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25

THANK YOU