



PIZZA SALES ANALYSIS



!

HELLO!

My name is Jyoti Kumari .In this project I have used SQL queries to solve problems related to pizza sales. It will help us to analyze the sales and put the data into increasing the profit margin.

Retrieve the total number of orders placed.

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

	total_orders
▶	21350

Total revenue generated by total 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 order_details.pizza_id = pizzas.pizza_id;
```

	total_revenue
▶	817860.05

Identify the highest priced pizza.

```
3  SELECT
4      pizza_types.name, pizzas.price
5  FROM
6      pizza_types
7      JOIN
8          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9  ORDER BY pizzas.price DESC
10 LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```
3  SELECT
4      pizzas.size, SUM(order_details.quantity) AS quantity_ordered
5  FROM
6      pizzas
7      JOIN
8          order_details ON pizzas.pizza_id = order_details.pizza_id
9  GROUP BY pizzas.size
10 ORDER BY quantity_ordered DESC
11 LIMIT 1;
```

	size	quantity_ordered
▶	L	18956

List the top 5 most ordered pizza type along with the quantity ordered.

```
3  SELECT
4      pizza_types.name, SUM(order_details.quantity) AS quantity
5  FROM
6      pizza_types
7          JOIN
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9          JOIN
10     order_details ON order_details.pizza_id = pizzas.pizza_id
11    GROUP BY pizza_types.name
12    ORDER BY quantity DESC
13    LIMIT 5;
```

	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

Determine distribution of orders by hours of day.

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

	hours	orders
▶	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
	10	8
	9	1

Group the orders by date and find out the average number of pizzas ordered per day

```
3  SELECT
4      ROUND(AVG(quantity), 0) AS avg_pizzas_ordered
5  FROM
6  (SELECT
7      orders.order_date, SUM(order_details.quantity) AS quantity
8  FROM
9      orders
10 JOIN order_details ON orders.order_id = order_details.order_id
11 GROUP BY orders.order_date) AS ordered_quantity;
```

	avg_pizzas_ordered
▶	138

	order_date	quantity
▶	2015-01-01	162
	2015-01-02	165
	2015-01-03	158
	2015-01-04	106
	2015-01-05	125
	2015-01-06	147
	2015-01-07	138
	2015-01-08	173
	2015-01-09	127
	2015-01-10	146
	2015-01-11	116
	2015-01-12	119
	2015-01-13	120
	2015-01-14	150
	2015-01-15	123
	2015-01-16	158
	2015-01-17	125
	2015-01-18	122
	2015-01-19	142
	2015-01-20	143
	2015-01-21	129

Determine the percentage contribution of each pizza category based on revenue.

```
3 • | SELECT
4     pizza_types.category,
5     (((SUM(order_details.quantity * pizzas.price)) / (SELECT
6         ROUND(SUM(order_details.quantity * pizzas.price),
7         2) AS total_revenue
8
9     FROM
10    order_details
11
12   JOIN
13     pizzas ON order_details.pizza_id = pizzas.pizza_id)) * 100) AS revenue
14
15   FROM
16     pizza_types
17
18   JOIN
19     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
20
21   JOIN
22     order_details ON order_details.pizza_id = pizzas.pizza_id
23
24 GROUP BY pizza_types.category;
```

	category	revenue
▶	Classic	26.90596025566967
	Veggie	23.682590927384577
	Supreme	25.45631126009862
	Chicken	23.955137556847287

Analyze the cumulative revenue generated over time.

```
3  select order_date ,  
4      sum(revenue) over(order by order_date) as cum_revenue  
5  from  
6  (select orders.order_date , 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 order_details.order_id = orders.order_id  
11  group by orders.order date ) as sales ;
```

order_date	cum_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
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000003
2015-01-14	32358.70000000004
2015-01-15	34343.5000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001

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

```
4 • select category , name , revenue from
5   (select category , name , revenue ,
6    rank() over(partition by category order by revenue desc) as rn
7    from (select pizza_types.category , pizza_types.name ,
8           sum(order_details.quantity * pizzas.price) as revenue
9           from pizza_types join pizzas
10          on pizza_types.pizza_type_id = pizzas.pizza_type_id
11         join order_details
12          on pizzas.pizza_id = order_details.pizza_id
13         group by pizza_types.category , pizza_types.name)as a) as b
14   where rn<=3;
```

	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	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	32265.7000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5