

PIZZA SALES IN A MONTH



HELLO

Welcome to the Pizza Sales SQL Query
Project

I AM DEVKI NANDAN, IN THIS PRESENTATION, WE WILL EXPLORE THE DATA ANALYSIS OF PIZZA SALES OVER THE COURSE OF A MONTH USING SQL QUERIES. I HAVE USED FROM BASIC SUMMARIES TO MORE ADVANCED ANALYTICAL QUERIES TO EXTRACT VALUABLE INSIGHTS FROM SALES DATA.

DATABASES USED

	order_id	order_date	order_time
▶	1	2015-01-01	11:38:36
	2	2015-01-01	11:57:40
	3	2015-01-01	12:12:28
	4	2015-01-01	12:16:31
	5	2015-01-01	12:21:30

	order_details_id	order_id	pizza_id	quantity
▶	1	1	hawaiian_m	1
	2	2	classic_dlx_m	1
	3	2	five_cheese_l	1
	4	2	ital_supr_l	1
	5	2	mexicana_m	1

	pizza_id	pizza_type_id	size	price
▶	bbq_ckn_s	bbq_ckn	S	12.75
	bbq_ckn_m	bbq_ckn	M	16.75
	bbq_ckn_l	bbq_ckn	L	20.75
	cali_ckn_s	cali_ckn	S	12.75
	cali_ckn_m	cali_ckn	M	16.75

	pizza_type_id	name	category	ingredients
▶	bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Pepp...
	cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
	ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
	ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
	southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...

Questions tried to answer

BASIC:

- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.
- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.
- IDENTIFY THE HIGHEST-PRICED PIZZA.
- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.
- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

INTERMEDIATE:

- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
- JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
- 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.

ADVANCED:

- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.
- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

1

```
-- Retrieve total no of orders
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders
```

Result Grid	
	total_orders
▶	21350

2

```
-- Calculate total revenue generated by pizza sales
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
```

Result Grid	
	total_sales
▶	817860.05

3

```
-- calculate the highest price pizza
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

```
-- identify the most ordered size pizza

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

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

5

```
-- Retrieve top selling pizzas along with the quantity ordered

SELECT
    pizza_types.name, SUM(order_details.quantity) AS qty
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 qty DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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

6

```
-- find the total quantity of each pizza category ordered
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS cat_sales
FROM
    pizzas
        JOIN
    pizza_types 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.category
ORDER BY cat_sales DESC;
```

Result Grid | Filter Row

	category	cat_sales
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

```
-- determine the distribution of orders by hour of the day  
SELECT  
    HOUR(order_time), COUNT(order_id)  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	hour(order_time)	count(order_id)
▶	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

```
-- find category-wise distribution of pizzas

select
count(pizza_types.name) , pizza_types.category from pizza_types
group by pizza_types.category;
```

Result Grid | Filter Rows:

	count(pizza_types.name)	category
▶	6	Chicken
	8	Classic
	9	Supreme
	9	Veggie

9

```
-- Group by date and calculate avg orders
SELECT
    ROUND(AVG(sum), 0) as avg_orders_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS sum
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

	avg_orders_per_day
▶	138

10

```
-- determine the top 5 most ordered pizza based on revenue
|
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS sum
FROM
    pizzas
        JOIN
    pizza_types 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 sum DESC
LIMIT 5;
```

Result Grid | Filter Rows:

	name	sum
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Spicy Italian Pizza	34831.25

11

```
-- calculate the % contribution of each pizza type to total revenue

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 pizzas join pizza_types
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.category ;
```

Result Grid |

	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

```
-- analyse the cumulative revenue generated over time

select order_date,
       sum(revenue) over (order by order_date) as cum_revenue
  from
    (select
      orders.order_date,
      sum(order_details.quantity*pizzas.price) as revenue
     from order_details join pizzas
      on order_details.pizza_id=pizzas.pizza_id
    join orders
      on orders.order_id=order_details.order_id
   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
	2015-01-10	23990.350000000002
	2015-01-11	25862.65
	2015-01-12	27781.7

13

```
-- determine the top 3 pizza based on revenue for each category

select category, name , revenue from
(select category , name , revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select
pizza_types.name,
pizza_types.category,
sum(order_details.quantity *pizzas.price) as revenue
from pizzas join pizza_types
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.category, pizza_types.name) as a) as b
where rn<=3;
```

Result Grid | Filter Rows: Export:

	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	34834831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5

Thanks!