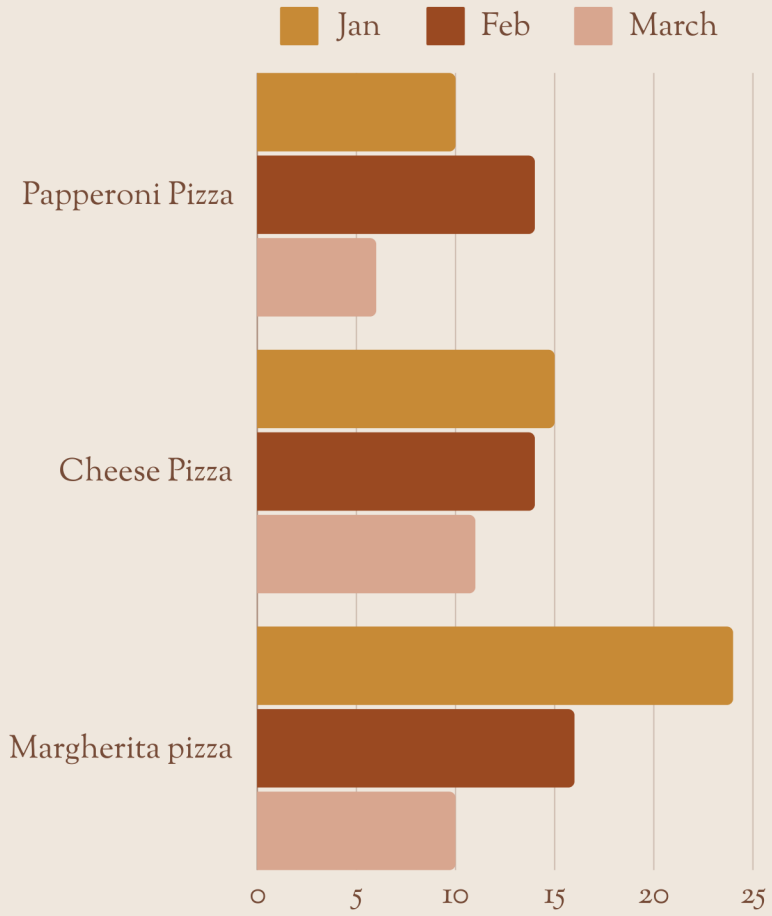
Ssameer
& Co.

PIZZA SALES

SQL QUERIES

Basic to advance



presented by
SSAMEER SKUMAR

INTRODUCTION

In my SQL project, I learned a lot about querying data. Took the dataset from kaggle about pizza sales of an area.There are total of 13 questions starting from basics to advanced level. I started with importing the csv file to SQL space to organise the data,then moved to the basics, like using the WHERE clause to filter data based on conditions, and using LIMIT and ORDER BY to control how the results are shown. Then, I moved on to more advanced stuff, like using GROUP BY to group data together and using functions to do calculations on groups of data. I also used the HAVING clause to filter data after grouping. After that, I learned about different ways to join tables together like INNER JOIN ,OUTER JOIN to combine data from different sources. But the most exciting part was when I learned about window functions. These let me do really cool stuff, like ranking rows and doing calculations across groups of rows. Overall, this project was a big learning experience for me, and it showed me how powerful SQL can be for analyzing data.

SCHEMA DIAGRAM(PIZZA SALES)

Order_details

Remote Field Name
order_details_id
order_id
pizza_id
quantity

Orders

Remote Field Name
order_id (orders.csv)
date
time

pizzas

Remote Field Name
pizza_id (pizzas.csv)
pizza_type_id
size
price

pizza_types

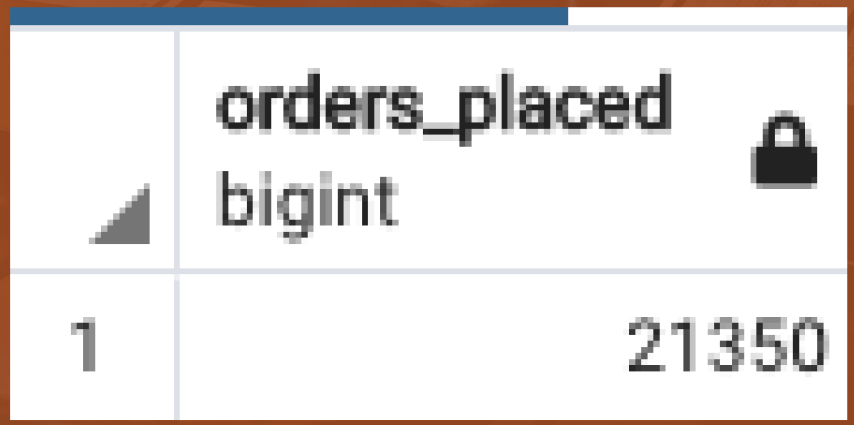
Remote Field Name
pizza_type_id (pizza_ty...
name
category
ingredients

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

SQL QUERY

```
select count(order_id) as orders_placed from orders
```

OUTPUT



	orders_placed bigint
1	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SQL QUERY

```
select sum(price*quantity) as total_revenue from
(select quantity,(select price from pizzas p where p.pizza_id = od.pizza_id)
from order_details od) a
```

OUTPUT

total_revenue
numeric
817860.05

FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

SQL QUERY

```
select category, count(*) as pizza_distribution
from pizza_types
group by category
```

OUTPUT

	category character varying	pizza_distribution bigint
1	Supreme	9
2	Chicken	6
3	Classic	8
4	Veggie	9

IDENTIFY THE HIGHEST-PRICED PIZZA.

SQL QUERY

```
select pizza_type_id,max(price) as price
from pizzas
group by pizza_type_id
order by max(price)desc
limit 1
```

OUTPUT

	pizza_type_id character varying (100)	price numeric
1	the_greek	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SQL QUERY

```
select size,count(quantity)
from
(select pizza_id,(select size from pizzas p
                  where od.pizza_id = p.pizza_id),quantity
                from order_details od) a
group by size
order by sum(quantity) desc
limit 1
```

OUTPUT

	size character varying (100)	count bigint
1	L	18526

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

SQL QUERY

```
select pt.pizza_type_id, sum(quantity) quantity
from order_details od inner join pizzas p
on od.pizza_id = p.pizza_id inner join pizza_types pt
on pt.pizza_type_id = p.pizza_type_id
group by pt.pizza_type_id
order by sum(quantity) desc
limit 5
```

OUTPUT

	pizza_type_id character varying	quantity bigint
1	classic_dlx	2453
2	bbq_ckn	2432
3	hawaiian	2422
4	pepperoni	2418
5	thai_ckn	2371

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

SQL QUERY

```sql
select pt.category as pizza_category,sum(quantity) as total_quantity_ordered
from order_details od inner join pizzas p
on od.pizza_id = p.pizza_id
inner join pizza_types pt
on p.pizza_type_id = pt.pizza_type_id
group by pt.category
order by sum(quantity) desc
```

OUTPUT

	pizza_category character varying	total_quantity_ordered bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

SQL QUERY

```sql
select round(avg(quantity_ordered_perday )) as "average_pizza_order/day"
from (
    select date,sum(quantity) quantity_ordered_perday
    from order_details od inner join orders o
    on od.order_id = o.order_id
    group by date
    order by 1
    ) a
```

OUTPUT

	average_pizza_order/day numeric
1	138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

SQL QUERY

```
select name,sum(price*quantity) as revenue
from order_details od inner join pizzas p
on od.pizza_id = p.pizza_id inner join pizza_types pt
on p.pizza_type_id = pt.pizza_type_id
group by name
order by sum(price*quantity) desc
limit 3
```

OUTPUT

	name character varying	revenue numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

SQL QUERY

```
SELECT CATEGORY AS PIZZA_TYPE,
    ROUND((SUM(PRICE * QUANTITY)) /
                                (SELECT SUM(PRICE * QUANTITY)
                                    FROM
                                        (SELECT QUANTITY,
                                                (SELECT PRICE
                                                    FROM PIZZAS P
                                                    WHERE P.PIZZA_ID = OD.PIZZA_ID)
                                            FROM ORDER_DETAILS OD) A) * 100,2) AS PERCENTAGE_CONTRIBUTION
FROM ORDER_DETAILS OD
INNER JOIN PIZZAS P ON OD.PIZZA_ID = P.PIZZA_ID
INNER JOIN PIZZA_TYPES PT ON P.PIZZA_TYPE_ID = PT.PIZZA_TYPE_ID
GROUP BY CATEGORY
ORDER BY PERCENTAGE_CONTRIBUTION DESC
```

OUTPUT

	pizza_type character varying	percentage_contribution numeric
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

SQL QUERY

```
select date, sum(revenue) over(order by date) as cum_revenue
from
(select date,sum(price*quantity) as revenue
from order_details od inner join pizzas p
on od.pizza_id = p.pizza_id inner join orders o
on od.order_id = o.order_id
group by date) a
```

OUTPUT

	date date	cum_revenue numeric
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4
10	2015-01-10	23990.35
11	2015-01-11	25862.65
12	2015-01-12	27781.7
13	2015-01-13	29831.3

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

SQL QUERY

```sql
SELECT CATEGORY,PIZZA_TYPE_ID,REVENUE
FROM
    (SELECT CATEGORY,PIZZA_TYPE_ID,REVENUE,
            RANK() OVER (PARTITION BY CATEGORY ORDER BY REVENUE DESC) AS RANKING
        FROM
            (SELECT PT.CATEGORY,
                    PT.PIZZA_TYPE_ID,
                    SUM(OD.QUANTITY * P.PRICE) AS REVENUE
                FROM ORDER_DETAILS OD
                INNER JOIN PIZZAS P ON OD.PIZZA_ID = P.PIZZA_ID
                INNER JOIN PIZZA_TYPES PT ON P.PIZZA_TYPE_ID = PT.PIZZA_TYPE_ID
                GROUP BY PT.CATEGORY,
                    PT.PIZZA_TYPE_ID
                ORDER BY 1) A) B
WHERE RANKING <= 3
```

OUTPUT

category character varying	pizza_type_id character varying	revenue numeric
Chicken	thai_ckn	43434.25
Chicken	bbq_ckn	42768
Chicken	cali_ckn	41409.5
Classic	classic_dlx	38180.5
Classic	hawaiian	32273.25
Classic	pepperoni	30161.75
Supreme	spicy_ital	34831.25
Supreme	ital_supr	33476.75
Supreme	sicilian	30940.5
Veggie	four_cheese	32265.7
Veggie	mexicana	26780.75
Veggie	five_cheese	26066.5