
SQL PROJECT

INTRODUCTION

HELLO EVERYONE ,

THIS IS MY SQL PROJECT ON PIZZA SALES.

TODAY,I’LL BE SHARING INSIGHTS GATHERING FROM ANALYZING SALES DATA OF A PIZZA BUSINESS USING SQL QUERIES.

LET;’S DIVE INTO THE DETAILS!

BASIC QUERIES

RETRIEVE THE TOTAL NO. OF ORDER PLACED

```
-- Retrieve the total number of orders placed--

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

total_orders
14912

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```sql
#Calculate the total revenue generated from pizza sales.

select
(order_details.quantity * pizzas.price) as total_sales
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id;
```

total_sales
125853.99999999938

IDENTIFY HIGHEST PRICE PIZZA

```sql
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 COMMON PIZZA SIZE ORDERED

```
#identity the most common pizza size ordered
select quantity, count(order_details_id)
from order_details group by quantity;
```

quantity	count(order_details_id)
0	1
1	7367
2	127
3	5

```
select pizzas.size,count(order_details_id)
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size;
```

size	count(order_details_id)
L	2882
M	2313
S	2217
size	1
XL	82
XXL	5

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

```sql
    #List the top 5 most ordered pizza types along with their quantities
SELECT
pizza_types.name,
SUM(order_details.quantity) as 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 quantity
DESC
LIMIT 5;
```

name	quantity ▼ 1
The Pepperoni Pizza	416
The Barbecue Chicken Pizza	384
The California Chicken Pizza	379
The Hawaiian Pizza	356
The Classic Deluxe Pizza	345

INTERMEDIATE QUERIES

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

```sql
#JOIN THE NECCESSARY TABLES TO FIND THE TOTAL QUANTITY TO EACH PIZZA CATEGORY ORDERED
select pizza_types.category,sum(order_details.quantity) as 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
desc;
```

category	quantity
Veggie	1816
Supreme	1871
Classic	2271
Chicken	1678
category	0

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
#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)
23	20
22	457
21	835
20	1132
19	1389
18	1684
17	1654
16	1343
15	1032
14	1079
13	1680
12	1730
11	870
10	6
0	1

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

```
#JOIN RELEVENT TABLES TO FING THE CATEGORY-WISE DISTRIBUTION OF PIZZAS
select category,count(name) from pizza_types
group by category;
```

category	count(name)
category	1
Chicken	6
Classic	8
Supreme	9
Veggie	9

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NO OF PIZZAS ORDERED PER DAY

```
#GROUP THE ORDER BY DATE AND
#CALCULATE THE AVERAGE NO OF PIZZAS ORDERED PER DAY
select round(avg(quantity),0) as Average from
(select orders.order_date,sum(quantity) as quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as order_quantity;
```

Average
134

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```sql
#DETERMINE THE TOP3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.
select 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.name
order by revenue
DESC
limit 3;
```

name	revenue ▼ 1
The Barbecue Chicken Pizza	6828
The California Chicken Pizza	6540.25
The Thai Chicken Pizza	6274

ADVANCED QUERIES

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```sql
#CALCULATE THE PERCENTAGE 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 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;
```

category	revenue ▼ 1
Classic	26.65
Supreme	25.67
Veggie	24.07
Chicken	23.61
category	0.00

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

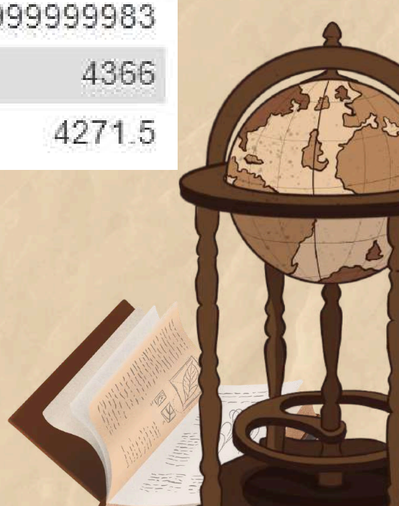
```
ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME
select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales
;
```

order_date	cum_revenue
0000-00-00	0
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
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006

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

```sql
#Determine the top3 most ordered pizza types based on revenue for each pizza category
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;
```

name	revenue
The California Chicken Pizza	6540.25
The Thai Chicken Pizza	6274
The Classic Deluxe Pizza	5382
The Pepperoni Pizza	5205.5
The Hawaiian Pizza	4700
The Italian Supreme Pizza	5278
The Sicilian Pizza	5080.75
The Spicy Italian Pizza	5018.25
The Four Cheese Pizza	4958.549999999983
The Five Cheese Pizza	4366
The Vegetables + Vegetables Pizza	4271.5




Thank You