



LARANA PIZZA

SQL PROJECT ON PIZZA SALES





LARANA PIZZA

HELLO!

In this project I have utilized SQL queries to solve questions that were related to pizza sales.



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

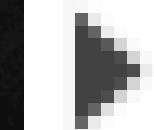
SELECT

COUNT(order_id) **AS** total_orders

FROM

orders;

total_orders



21350



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SELECT

```
ROUND(SUM(order_details.quantity * pizzas.price),  
2) AS total_revenue_generated
```

FROM

```
order_details
```

JOIN

```
pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

	total_orders
▶	21350



IDENTIFY THE HIGHEST-PRICED PIZZA.

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

	name	price
▶	The Greek Pizza	35.95



IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SELECT

size, COUNT(order_details_id) AS order_count

FROM

pizzas

INNER JOIN

order_details ON pizzas.pizza_id = order_details.pizza_id

GROUP BY size

	size	order_count
▶	L	13346



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

```
SELECT
    pizza_types.name, SUM(quantity) AS quantity_ordered
FROM
    order_details
    INNER JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
    INNER JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name
ORDER BY quantity_ordered DESC
LIMIT 5;
```

name	quantity_ordered
The Barbecue Chicken Pizza	1776
The Pepperoni Pizza	1745
The Classic Deluxe Pizza	1740
The Hawaiian Pizza	1708
The California Chicken Pizza	1705



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

SELECT

```
    pizza_types.category, SUM(quantity) AS quantity_ordered
FROM
    order_details
    INNER JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
    INNER JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY quantity ordered desc;
```

category	quantity_ordered
Classic	10679
Supreme	8613
Veggie	8463
Chicken	7918



DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

SELECT

HOUR(order_time) AS hours, COUNT(order_id) AS order_count

FROM

orders

GROUP BY hours

ORDER BY hours ASC;

hours	order_count
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663



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

SELECT

category, COUNT(pizza_type_id) AS total_pizzas

FROM

pizza_types

GROUP BY category;

category	total_pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9



GROUP THE ORDERS BY DATE AND
CALCULATE THE AVERAGE NUMBER OF PIZZAS
ORDERED PER DAY.

```
with order_quantity as
(
  select order_date,
         sum(quantity) as quantity_per_day
    from orders
   inner join order_details
      on orders.order_id = order_details.order_id
   group by order_date
)
select round(avg(quantity_per_day),0) as avg_pizzas_ordered
  from order_quantity;
```

avg_pizzas_ordered

138



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

SELECT

```
    pizza_types.name, (SUM(quantity * price)) AS revenue  
FROM  
    pizzas  
        INNER JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
        INNER JOIN  
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
GROUP BY pizza_types.name  
ORDER BY revenue DESC  
LIMIT 3;
```

name	revenue
The Barbecue Chicken Pizza	31180
The Thai Chicken Pizza	30661.75
The California Chicken Pizza	29786.75



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

```
SELECT
    pizza_types.category,
    ROUND((SUM(quantity * price) / (SELECT
                                            SUM(quantity * price)
                                        FROM
                                            order_details
                                        INNER JOIN
                                            pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100,
        2) AS percent_contribution
FROM
    order_details
    INNER JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
    INNER JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY category;
```

	category	percent_contribution
▶	Classic	26.84
	Veggie	23.91
	Supreme	25.45
	Chicken	23.81



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date,  
sum(revenue) over(order by order_date) as cum_revenue  
from  
(select order_date,  
sum(pizzas.price * order_details.quantity) as revenue  
from orders  
join order_details  
on orders.Order_id = order_details.order_id  
join pizzas  
on pizzas.pizza_id = order_details.pizza_id  
group by order_date  
) as sales;
```

order_date	cum_revenue
2015-01-01	2713.850000000000
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	23990.35000000000
2015-01-11	25862.65
2015-01-12	27781.7



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

```
with pizza_data as
(
  select category, name,
  dense_rank() over(partition by category order by revenue desc) as ranking
  from
  (
    select category, name,
    sum(price * quantity) as revenue
    from pizza_types
    inner join pizzas
    on pizza_types.pizza_type_id = pizzas.pizza_type_id
    inner join order_details
    on order_details.pizza_id = pizzas.pizza_id
    group by category, name
  ) as a
)
select category, name, ranking
from pizza_data
where ranking <=3;
```

category	name	ranking
Chicken	The Barbecue Chicken Pizza	1
Chicken	The Thai Chicken Pizza	2
Chicken	The California Chicken Pizza	3
Classic	The Classic Deluxe Pizza	1
Classic	The Hawaiian Pizza	2
Classic	The Pepperoni Pizza	3
Supreme	The Spicy Italian Pizza	1
Supreme	The Italian Supreme Pizza	2
Supreme	The Sicilian Pizza	3
Veggie	The Four Cheese Pizza	1
Veggie	The Five Cheese Pizza	2
Veggie	The Mexicana Pizza	3

PRESENTED BY:-
PARUL SAHU





LARANA PIZZA

THANK YOU!

