

PIZZA SALES PROJECT USING SQL

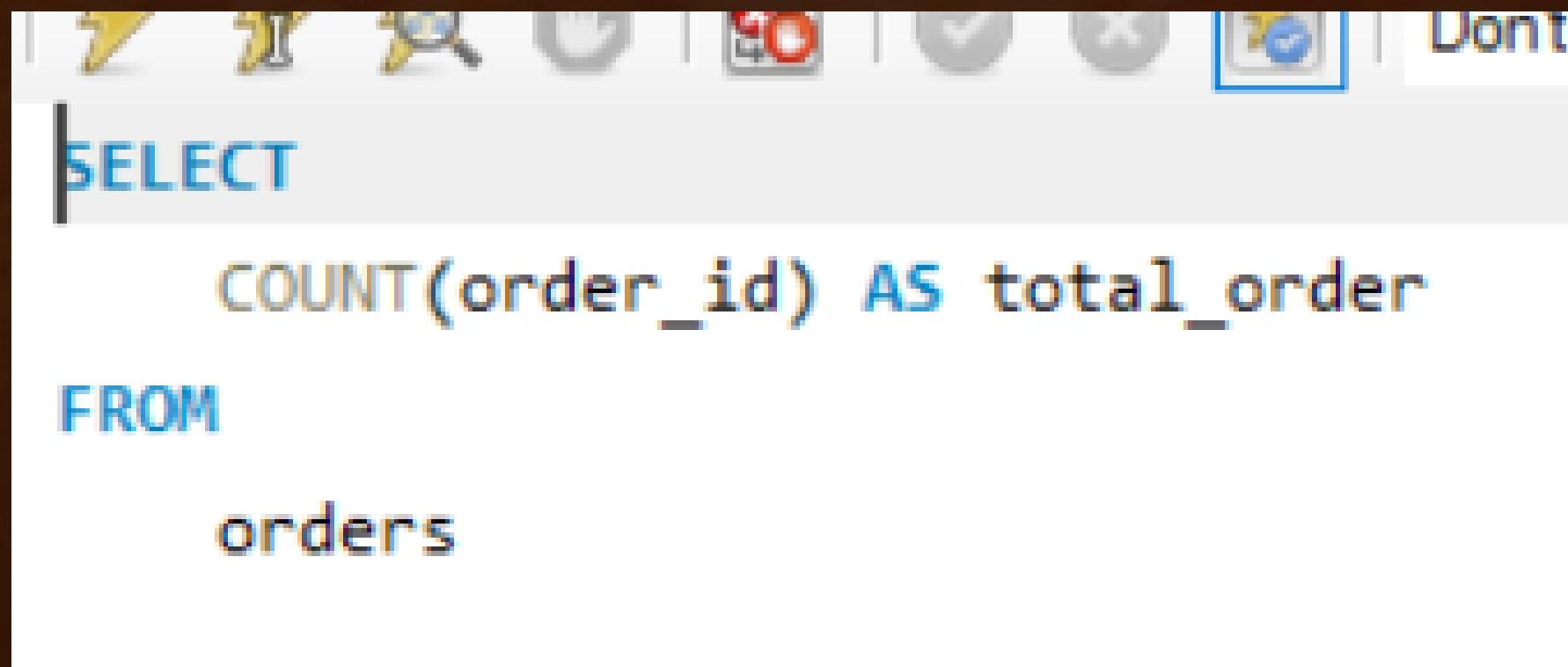


ABOUT OUR PIZZA SALES PROJECT

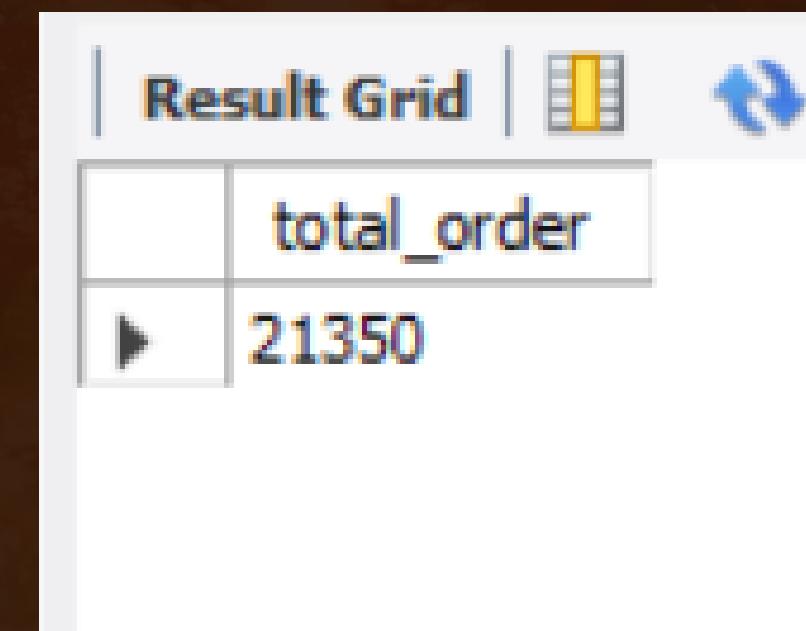
The Pizza Sales Analysis project aims to extract actionable business insights from historical sales data using SQL. The focus is on identifying trends, customer preferences, and performance metrics to support data-driven decision-making in a pizza business.



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



```
SELECT COUNT(order_id) AS total_order
FROM orders
```



	total_order
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(order_detail.quantity * pizzas.price),  
          0) AS total_revenue  
  
FROM  
    order_detail  
        JOIN  
    pizzas ON order_detail.pizza_id = pizzas.pizza_id
```

Result Grid	
	total_revenue
▶	817860

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



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

Result Grid | Filter Rows:

	name	revenue
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

IDENTIFY THE HIGHEST-PRICED 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 COMMON PIZZA SIZE ORDERED.

```
SELECT  
    COUNT(order_detail.order_details_id) as total_order, pizzas.size  
FROM  
    order_detail  
        JOIN  
    pizzas ON order_detail.pizza_id = pizzas.pizza_id  
GROUP BY pizzas.size  
LIMIT 1;
```

	total_order	size
▶	15385	M

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

```
SELECT
    SUM(order_detail.quantity) as quantity, pizza_types.name AS pizza_name
FROM
    order_detail
        JOIN
    pizzas ON order_detail.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_name order by quantity desc
LIMIT 5;
```

Result Grid		Filter Rows:
	quantity	pizza_name
▶	2453	The Classic Deluxe Pizza
	2432	The Barbecue Chicken Pizza
	2422	The Hawaiian Pizza
	2418	The Pepperoni Pizza
	2371	The Thai Chicken Pizza

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

```
SELECT
    pizza_types.category,
    SUM(order_detail.quantity) AS total_order
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

	category	total_order
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

-- Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(orders.order_time) AS hours, COUNT(orders.order_id)  
FROM  
    orders  
GROUP BY hours;
```

	hours	count(orders.order_id)
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

-- Join relevant tables to find the category-wise distribution of pizzas.

```
SELECT  
    COUNT(name), pizza_types.category  
FROM  
    pizza_types  
GROUP BY pizza_types.category;
```

Result Grid | Filter Rows:

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

-- Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(total_order_per_day), 0) AS avearage_order_per_day
FROM
    (SELECT
        orders.order_date AS order_dates,
        SUM(order_detail.quantity) AS total_order_per_day
    FROM
        orders
    JOIN order_detail ON orders.order_id = order_detail.order_id
    GROUP BY order_dates) AS order_quantity;
```

Result Grid	
	avearage_order_per_day
▶	138

Determine the top 3 most ordered pizza types based on revenue.

```
SELECT
    pizza_types.name AS pizza_name,
    ROUND(SUM(order_detail.quantity * pizzas.price),
          0) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

	pizza_name	revenue
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

Calculate the percentage contribution of each pizza type to total revenue.

```
SELECT
    pizza_types.name AS pizza_name,
    ROUND(SUM(order_detail.quantity * pizzas.price)/(SELECT
        ROUND(SUM(order_detail.quantity * pizzas.price),
        0) AS total_revenue
    )
FROM
    order_detail
    JOIN
    pizzas ON order_detail.pizza_id = pizzas.pizza_id)*100,
    2) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_name
ORDER BY revenue DESC;
```

Result Grid		Filter Rows:	Export:
	pizza_name	revenue	
	The Thai Chicken Pizza	5.31	
	The Barbecue Chicken Pizza	5.23	
	The California Chicken Pizza	5.06	
	The Classic Deluxe Pizza	4.67	
	The Spicy Italian Pizza	4.26	
	The Southwest Chicken Pizza	4.24	
	The Italian Supreme Pizza	4.09	
	The Hawaiian Pizza	3.95	
	The Four Cheese Pizza	3.95	
	The Sicilian Pizza	3.78	
▶	The Pepperoni Pizza	3.69	

-- Analyze the cumulative revenue generated over time.

```
select order_date,  
round(sum(revenue)over(order by order_date),0) as cum_revenue  
from  
  (select orders.order_date,sum(order_detail.quantity*pizzas.price) as revenue  
   from orders join  
   order_detail on orders.order_id = order_detail.order_id  
   join pizzas on pizzas.pizza_id = order_detail.pizza_id  
   group by orders.order_date)  
as sales;
```

Result Grid		
	order_date	cum_revenue
▶	2015-01-01 00:00:00	2714
	2015-01-02 00:00:00	5446
	2015-01-03 00:00:00	8108
	2015-01-04 00:00:00	9864
	2015-01-05 00:00:00	11930
	2015-01-06 00:00:00	14358
	2015-01-07 00:00:00	16561
	2015-01-08 00:00:00	19399
	2015-01-09 00:00:00	21526
	2015-01-10 00:00:00	23990
	2015-01-11 00:00:00	25863

-- Determine the top 3 most ordered pizza types based on revenue for each pizza category

```
select name,revenue,category 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_detail.quantity*pizzas.price)
as revenue
from pizzas join pizza_types on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_detail on order_detail.pizza_id=pizzas.pizza_id
group by pizza_types.category,pizza_types.name) as a)
b where rn<=3;
```

	name	revenue	category
▶	The Thai Chicken Pizza	43434.25	Chicken
	The Barbecue Chicken Pizza	42768	Chicken
	The California Chicken Pizza	41409.5	Chicken
	The Classic Deluxe Pizza	38180.5	Classic
	The Hawaiian Pizza	32273.25	Classic
	The Pepperoni Pizza	30161.75	Classic
	The Spicy Italian Pizza	34831.25	Supreme
	The Italian Supreme Pizza	33476.75	Supreme
	The Sicilian Pizza	30940.5	Supreme
	The Four Cheese Pizza	32265.70000000065	Veggie
	The Mexicana Pizza	26780.75	Veggie

-- Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT  
    ROUND(AVG(order_quantity.quantity), 0) AS order_pizza_per_day  
FROM  
(SELECT  
    orders.order_date, SUM(order_detail.quantity) AS quantity  
FROM  
    orders  
JOIN order_detail ON orders.order_id = order_detail.order_id  
GROUP BY orders.order_date) AS order_quantity;
```

Result Grid	
	order_pizza_per_day
▶	138

pizza sales project using sql

**THANK YOU
FOR ATTENTION**