



SQL
PROJECT
ON
PIZZA SALES





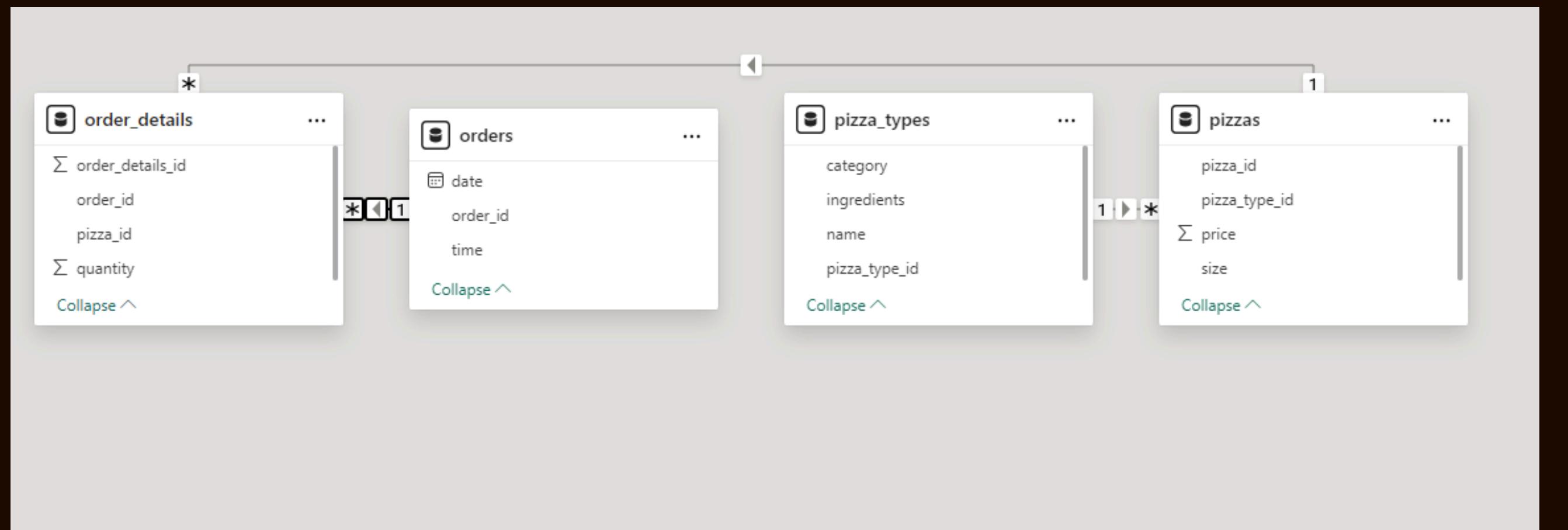
PIZZA SALES

ABOUT

Hello, I'm Neha Soni. I utilized SQL to analyze sales data, providing insights into customer behavior and sales trends. My focus was on optimizing queries and generating reports to support informed decision-making.



DATABASE SCHEMA OVERVIEW



Retrieve the total number of orders placed.

Select count(order_id) As Total_orders from
orders;

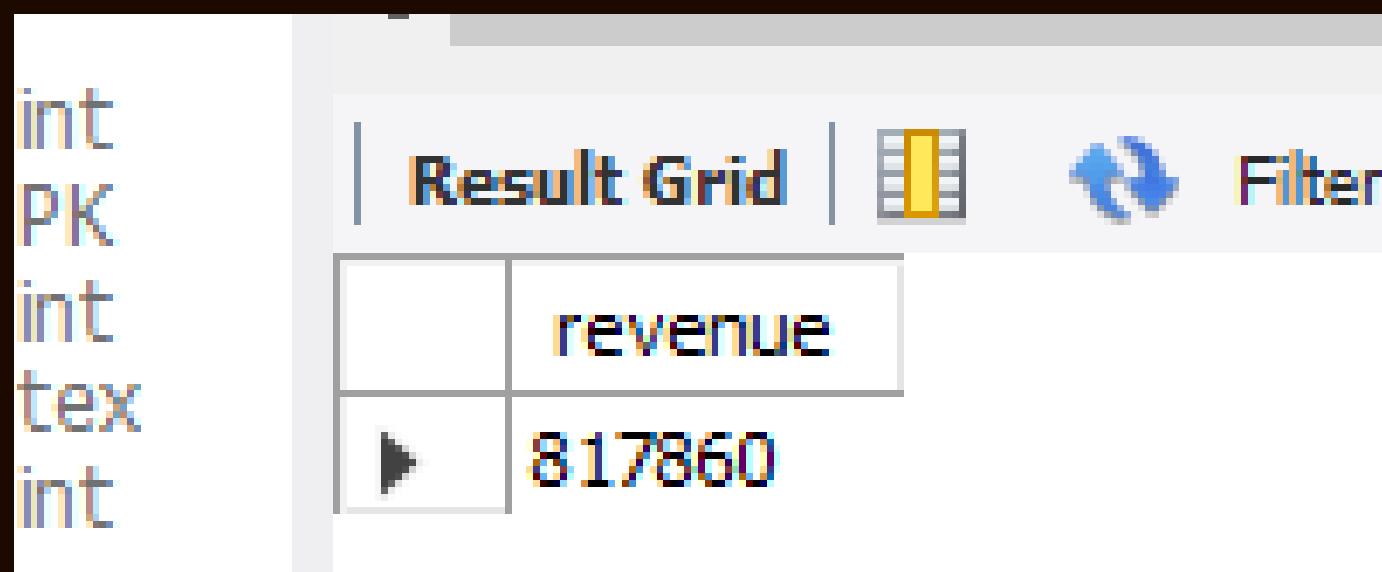


Result Grid	
	total_orders
▶	21350



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(quantity * Price), 0) AS revenue  
FROM  
    order_details  
JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```



	revenue
▶	817860

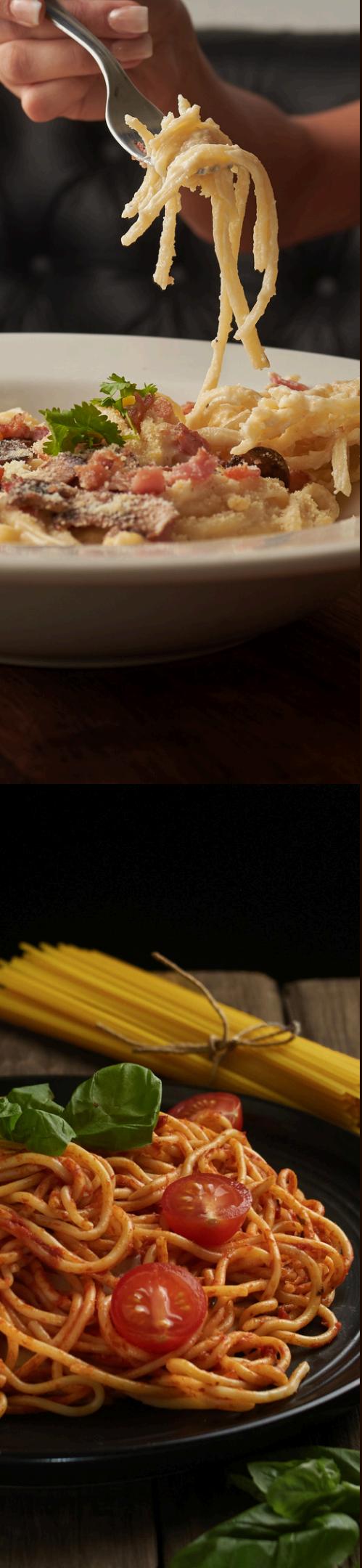
IDENTIFY THE HIGHEST PRICE PIZZA.

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

Screenshot of a MySQL Workbench result grid showing the query results:

	price	name
▶	35.95	The Greek Pizza





IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SELECT

size, COUNT(order_details.order_details_id) AS order_count

FROM

pizzas

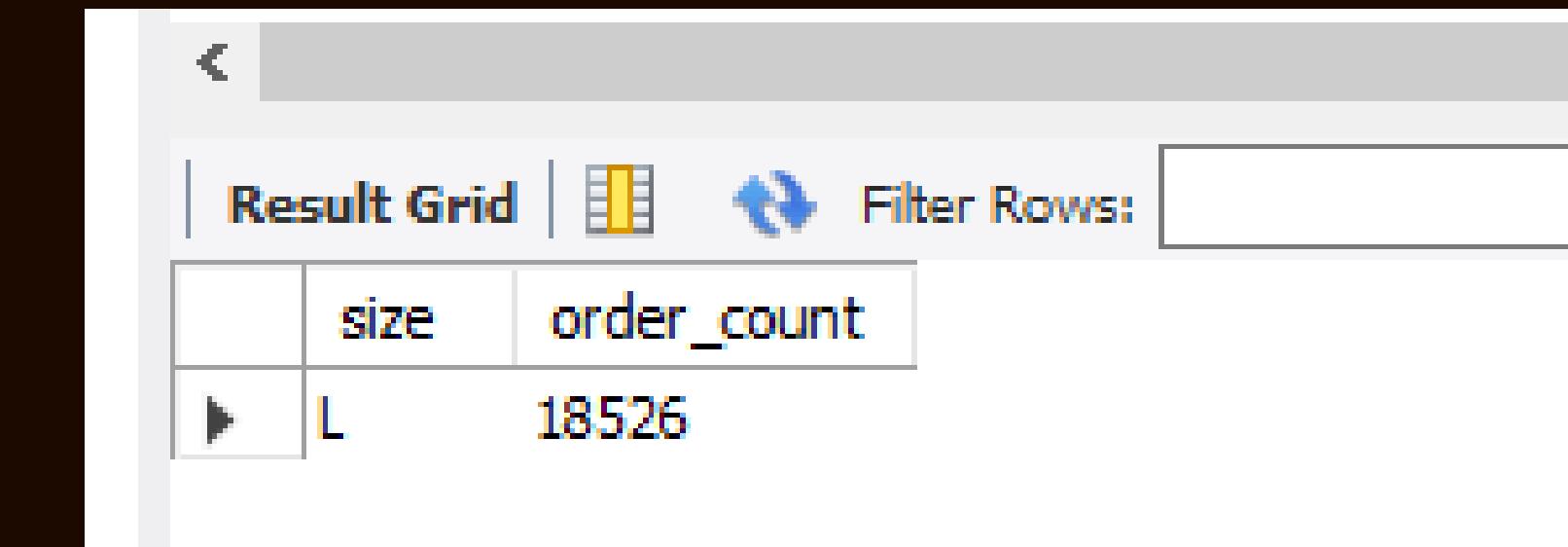
JOIN

order_details ON pizzas.pizza_id = order_details.pizza_id

GROUP BY size

ORDER BY size ASC

LIMIT 1;



	size	order_count
▶	L	18526

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

```
select pizza_types.name,sum(order_details.Quantity)as quantity from order_details  
join  
pizzas on order_details.pizza_id=pizzas.pizza_id  
join  
pizza_types on pizza_types.pizza_type_id=pizzas.pizza_type_id  
group by pizza_types.name order by Quantity desc limit 5;
```

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





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

```
select pizza_types.category,sum(order_details.Quantity)as quantity from  
order_details  
join  
pizzas on order_details.pizza_id=pizzas.pizza_id  
join  
pizza_types on pizza_types.pizza_type_id=pizzas.pizza_type_id  
group by pizza_types.category order by Quantity desc;
```

Result Grid | Filter Rows:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	hour	order_count
▶	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(name) as name_count
FROM
    pizza_types
GROUP BY category;
```

	category	name_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



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

```
select round(Avg(quantity),0) as avg_pizza_ordered_per_day from  
(select orders.order_date, sum(order_details.quantity) as quantity  
from orders  
join  
order_details on orders.order_id = order_details.order_id  
group by orders.order_date) as order_quantity;
```

	avg_pizza_ordered_per_day
▶	138





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

*
select pizza_types.name,round(sum(pizzas.price
order_details.quantity),0) as revenue from pizza_types
join pizzas on pizza_types.pizza_type_id =
pizzas.pizza_type_id
join order_details on pizzas.pizza_id = order_details.pizza_id group by
pizza_types.name order by revenue desc limit 3;



	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.category,concat(  
round(sum(pizzas.price * order_details.quantity)/  
(select round(sum(pizzas.price * order_details.quantity),0)  
as total_sales  
from pizzas  
join  
order_details 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 pizzas.pizza_id = order_details.pizza_id  
group by category order by revenue desc;
```

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

	category	revenue
▶	Classic	26.91%
	Supreme	25.46%
	Chicken	23.96%
	Veggie	23.68%



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;
```



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

Result Grid | Filter Rows:

	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

Result 30 ×



PIZZA SALES

THANK YOU

