



PIZZA SALES

09 June, 2024

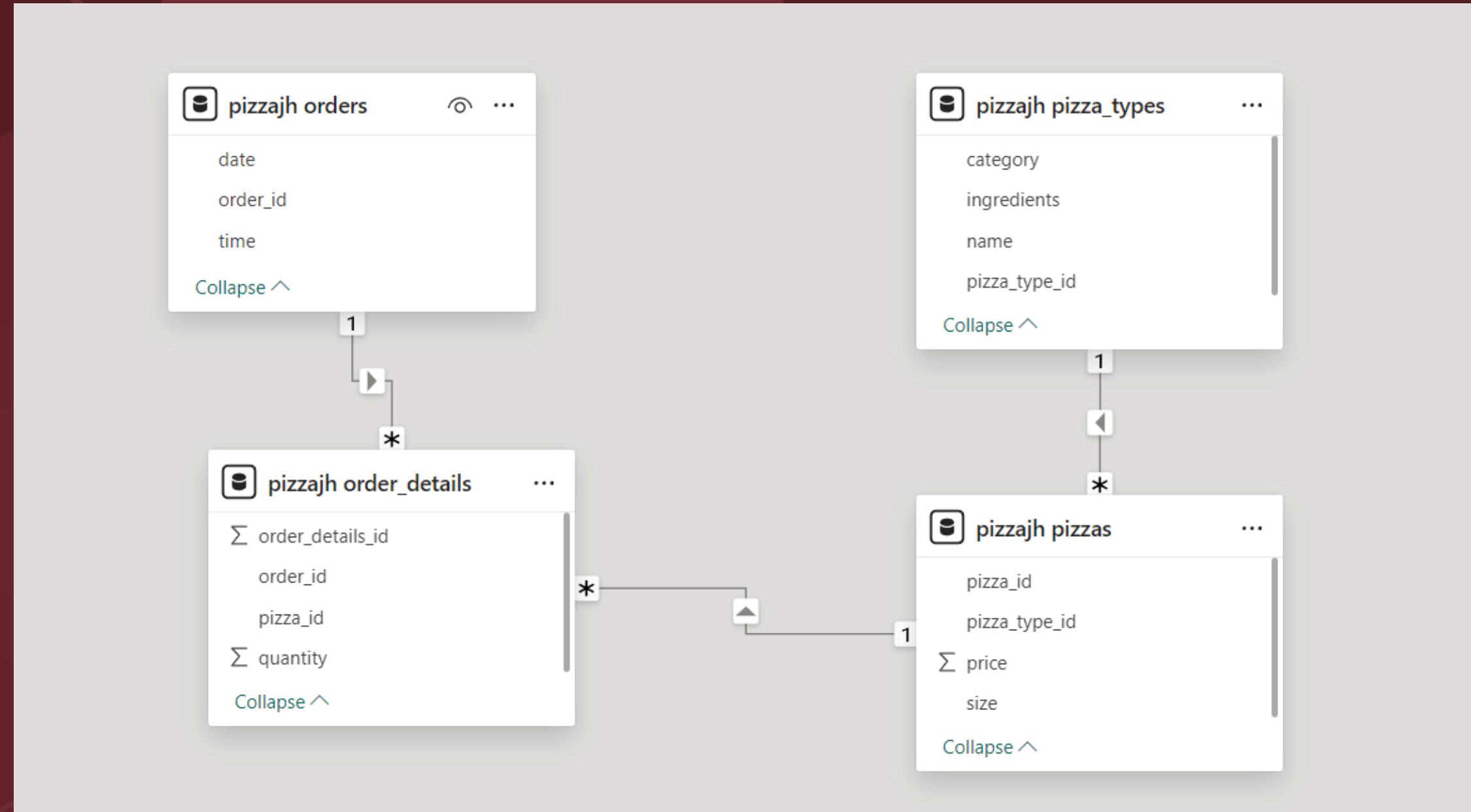


INTRODUCTION

Welcome to our Pizza Sales Report Presentation. In this project i have utilize SQL Query to solve question which are related to Pizza sales.



ER- DIAGRAM





1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT  
    COUNT(order_id) total_orders  
FROM  
    orders;
```

Result Grid	
	total_orders
▶	21350

2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue  
FROM  
    pizzas p  
    JOIN  
    order_details od ON p.pizza_id = od.pizza_id;
```

	total_revenue
▶	817860.05



3. IDENTIFY THE HIGHEST-PRICED PIZZA.

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



Result Grid		
	name	highest_price
▶	The Greek Pizza	35.95



4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
select p.size ,count(*) as c from order_details od
join pizzas p
on p.pizza_id = od.pizza_id
group by p.size
order by c desc limit 1;
```

Result Grid | Filter Rows:

	size	c
▶	L	18526





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

```
SELECT
    pt.name, SUM(od.quantity) total_orders
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY total_orders DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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



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

```
select pt.category,sum(od.quantity) total_qty from pizzas p
join pizza_types pt
on pt.pizza_type_id = p.pizza_type_id
join order_details od
on od.pizza_id = p.pizza_id
group by pt.category;
```

Result Grid | Filter Rows

	category	total_qty
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
select hour(time)as order_time ,count(order_id) as order_count from orders  
group by hour(time)  
order by order_count desc;
```

	order_time	order_count
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28
	10	8
	9	1

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

```
select category, count(name) from pizza_types  
group by category;
```

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



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

```
select avg(sum_qty) from (select o.date,sum(od.quantity) sum_qty from orders o  
join order_details od  
on o.order_id = od.order_id  
group by o.date) as data;
```

Result Grid | Filter R

	avg(sum_qty)
▶	138.4749



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

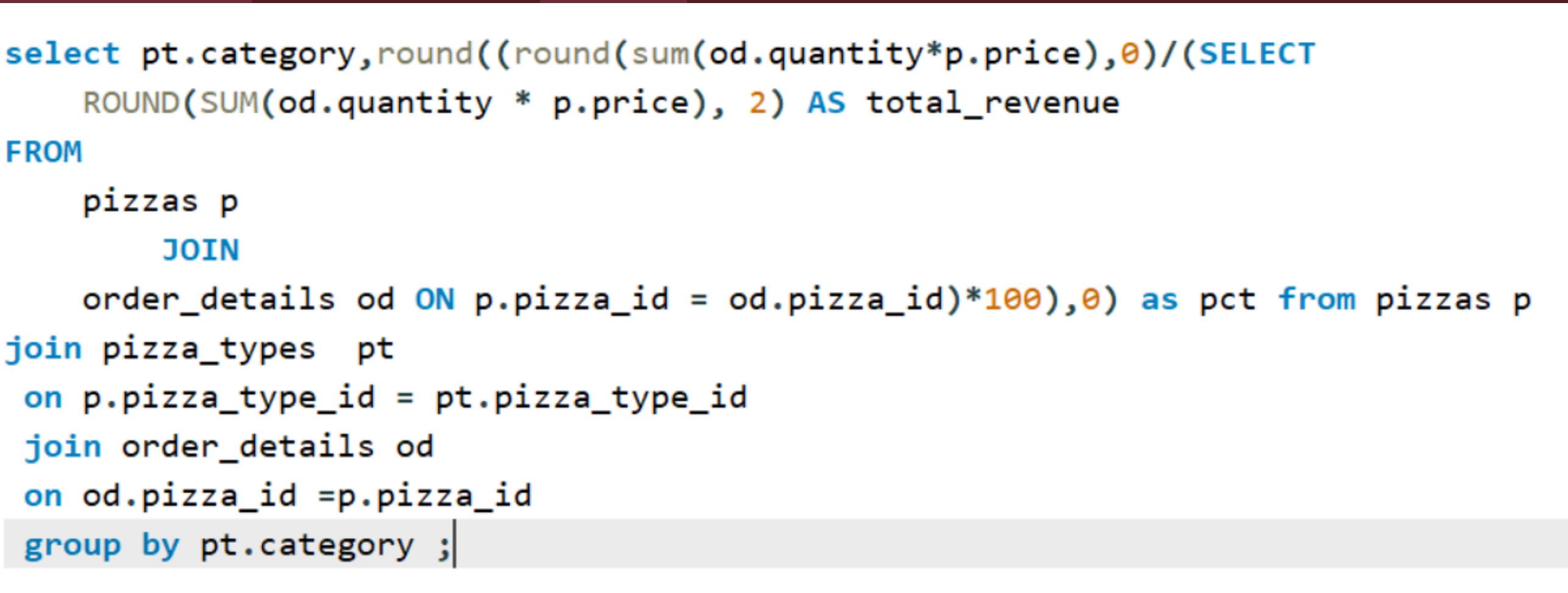
```
select pt.name,sum(od.quantity*p.price) revenue from pizzas p
join pizza_types pt
on p.pizza_type_id = pt.pizza_type_id
join order_details od
on od.pizza_id = p.pizza_id
group by pt.name
order by revenue desc
limit 3;
```

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



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

```
select pt.category,round((round(sum(od.quantity*p.price),0)/(SELECT  
    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue  
FROM  
    pizzas p  
    JOIN  
    order_details od ON p.pizza_id = od.pizza_id)*100),0) as pct from pizzas p  
join pizza_types pt  
on p.pizza_type_id = pt.pizza_type_id  
join order_details od  
on od.pizza_id =p.pizza_id  
group by pt.category ;
```



The screenshot shows a database result grid with the following data:

	category	pct
▶	Classic	27
	Veggie	24
	Supreme	25
	Chicken	24



12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select date ,sum(revenue) over(order by date) as cumulative
  from
(select o.date,round(sum(od.quantity*p.price),0) as revenue from pizzas p
join order_details od
on od.pizza_id = p.pizza_id
join orders o
on o.order_id = od.order_id
group by o.date)as sales;
```

Result Grid		
	date	cumulative
▶	2015-01-01	2714
	2015-01-02	5446
	2015-01-03	8108
	2015-01-04	9863
	2015-01-05	11929
	2015-01-06	14358
	2015-01-07	16560
	2015-01-08	19398
	2015-01-09	21525
	2015-01-10	23989
	2015-01-11	25861
	2015-01-12	27780
	2015-01-13	29830
	2015-01-14	32357
	2015-01-15	34342



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

```
select name,revenue,category from
  (select sb.name,sb.revenue ,sb.category,rank() over(partition by sb.category order by revenue desc) as rn from
   (select pt.category,pt.name,round(sum(od.quantity*p.price),0)as revenue from pizzas p
    join pizza_types pt
    on pt.pizza_type_id = p.pizza_type_id
    join order_details od
    on od.pizza_id = p.pizza_id
   group by pt.category,pt.name)sb)sd2
where rn<=3;
```

Result Grid | Filter Rows: | Exp

	name	revenue	category
▶	The Thai Chicken Pizza	43434	Chicken
	The Barbecue Chicken Pizza	42768	Chicken
	The California Chicken Pizza	41410	Chicken
	The Classic Deluxe Pizza	38180	Classic
	The Hawaiian Pizza	32273	Classic
	The Pepperoni Pizza	30162	Classic
	The Spicy Italian Pizza	34831	Supreme
	The Italian Supreme Pizza	33477	Supreme
	The Sicilian Pizza	30940	Supreme
	The Four Cheese Pizza	32266	Veggie
	The Mexicana Pizza	26781	Veggie
	The Five Cheese Pizza	26066	Veggie



THANK YOU

09 June, 2024