

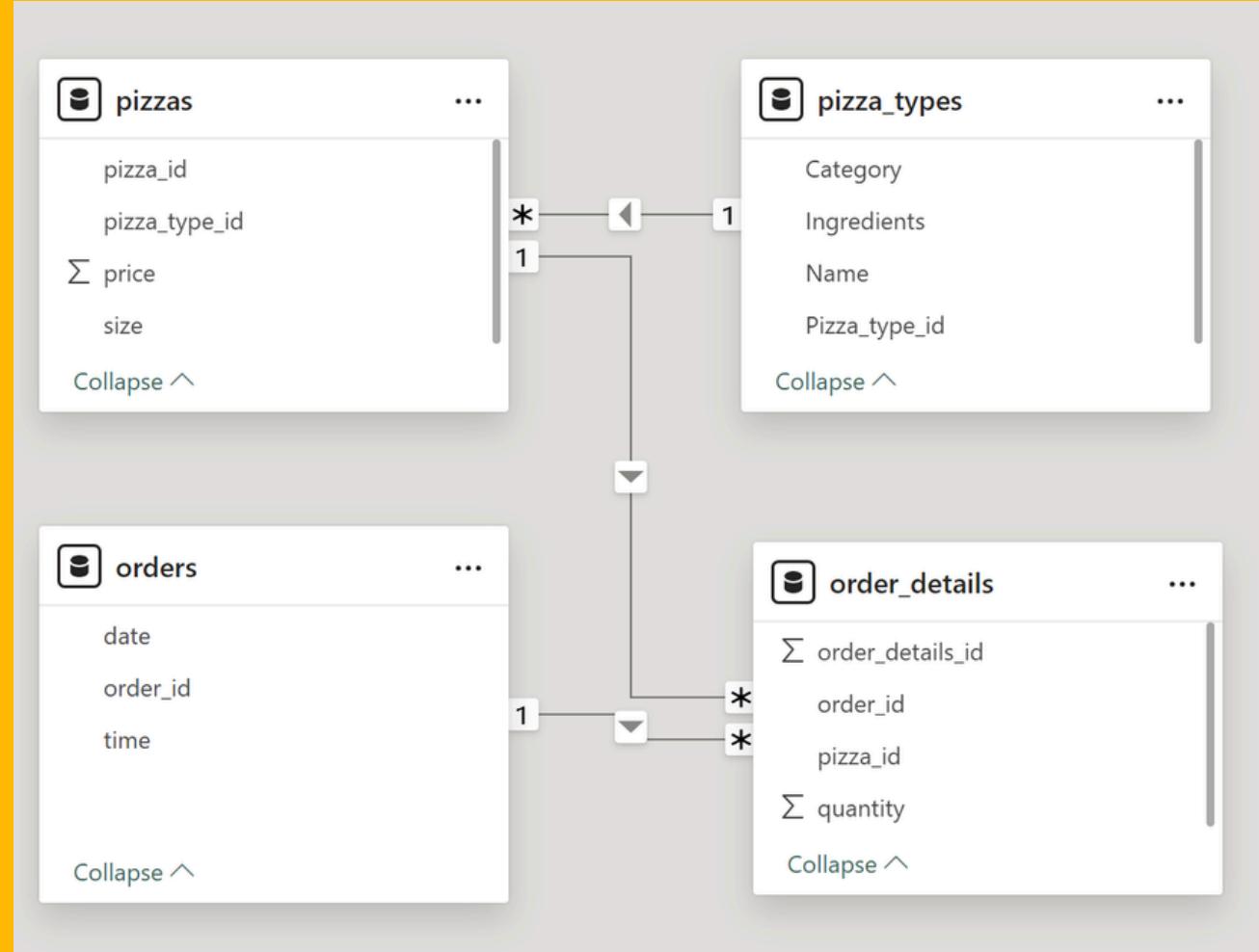


SQL PROJECT ON PIZZA SALES



HI, MY NAME IS ANIL, IN THIS PROJECT I HAVE UTILIZED SQL QUERIES TO SOLVE QUESTIONS THAT WERE RELATED TO PIZZA SALES

SCHEMA



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

SELECT

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

FROM

```
order_details
```

```
JOIN
```

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

Result Grid

	total_revenue
	817860.05

IDENTIFY THE HIGHEST PRICED PIZZA

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY 2 DESC
LIMIT 1;
```

Result Grid | Filter R

	name	price
▶	The Greek Pizza	35.95

LIST THE TOP 5 MOST ORDERED PIZZA TYPES WITH THEIR QUANTITIES

SELECT

```
    pizza_types.name, SUM(order_details.quantity) AS quantity
```

FROM

```
    pizzas
```

```
        JOIN
```

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

```
        JOIN
```

```
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

GROUP BY 1

ORDER BY 2 DESC

LIMIT 5;

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  
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
        JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY 1  
ORDER BY 2 DESC;
```

Result Grid |

	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 1
```

```
ORDER BY 2 DESC;
```

Result Grid | Filter

	hour	order_count
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472

FIND CATEGORY WISE DISTRIBUTION PF PIZZAS

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

Result Grid | Filter Rows:

	category	count(name)
>	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 daily_average  
FROM  
(SELECT  
    orders.order_date, SUM(order_details.quantity) AS quantity  
FROM  
    order_details  
JOIN orders ON orders.order_id = order_details.order_id  
GROUP BY 1) AS order_quantity;
```

Result Grid

	daily_average
▶	138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPE BASED ON REVENUE

SELECT

```
    pizza_types.name,  
    SUM(order_details.quantity * pizzas.price) AS revenue
```

FROM

```
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
        JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id
```

GROUP BY 1

ORDER BY 2 DESC

LIMIT 3;

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

CALCULATE 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;
```

Result Grid | Filter Rows:

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVERTIME

```
select order_date,  
sum(revenue) over(order by order_date) as cumulative_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	cumulative_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
	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

DETERMINE THE TOP 3 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;
```

Result Grid | Filter Rows:

	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Hawaiian Pizza	32273.25
6	The Pepperoni Pizza	30161.75
7	The Spicy Italian Pizza	34831.25
8	The Italian Supreme Pizza	33476.75