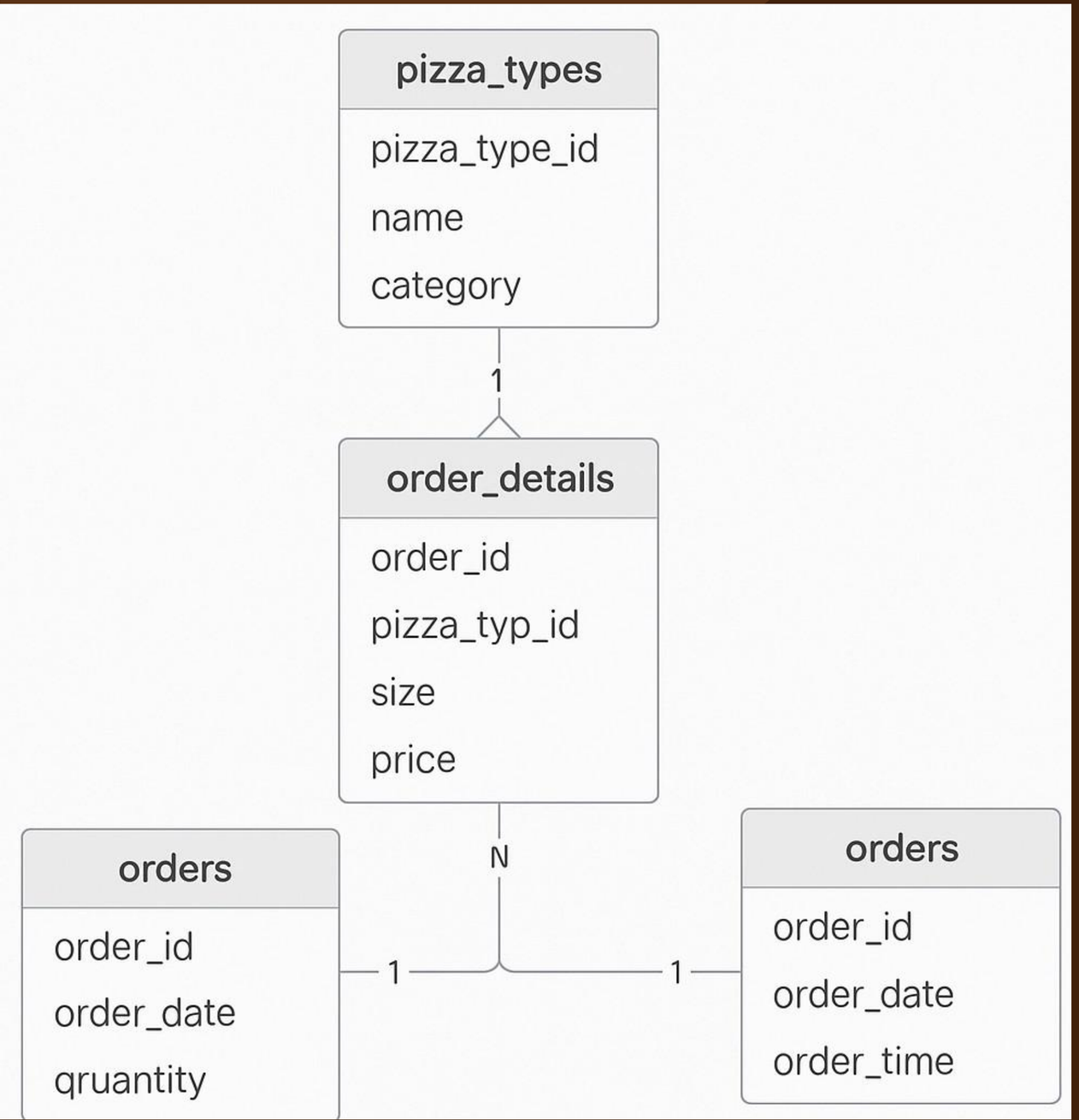
PIZZA SALES ANALYSIS

YUM PIZZA



TABLES SCHEMA

This schema represents a relational database for a pizza ordering system, comprising four core tables:

orders: Contains information about each order, including the order ID,date, and time.

order_details: Acts as a junction table connecting orders to the pizzas ordered. It tracks quantity and links to both the orders and pizzas tables.

pizzas: Represents individual pizza offerings with attributes such as size and price, and a foreign key to pizza_types.

pizza_types: Describes the general category and name of the pizza, such as "Veggie" "Meat Lovers".

QUESTIONS COVERED

BASIC:

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

IDENTIFY THE HIGHEST-PRICED PIZZA.

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

INTERMEDIATE:

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

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

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

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

ADVANCED:

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

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

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

Retrieve the total number of orders placed.

```sql
select count(order_id) total_orders
from orders;
```

Result:

	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

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

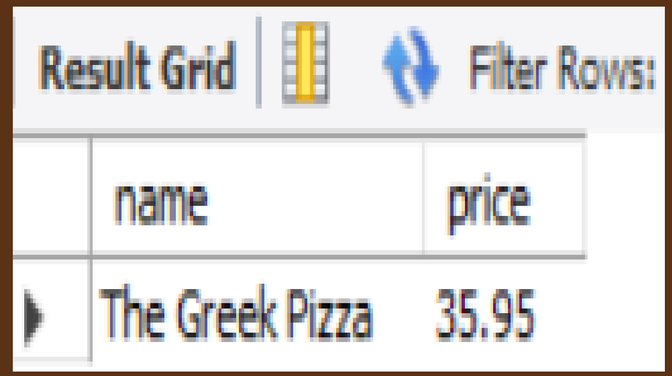
Result:

	total_revenue
▶	817860.05

Identify the highest-priced pizza.

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

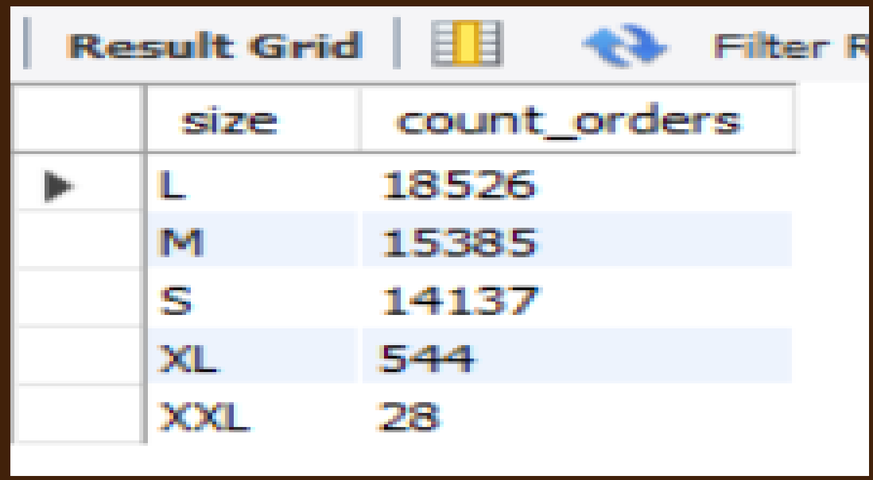
Result:



Identify the most common pizza size ordered.

```sql
select p.size, count(od.quantity) as count_orders
from pizzas p
join order_details od
on p.pizza_id = od.pizza_id
group by p.size
order by count_orders desc;
```

Result:



Result Grid | Filter R

size	count_orders
L	18526
M	15385
S	14137
XL	544
XXL	28

List the top 5 most ordered pizza types along with their quantities.

```
select pt.name,count(od.quantity) as total_quantity
from order_details od
join pizzas p
on p.pizza_id = od.pizza_id
join pizza_types pt
on pt.pizza_type_id = p.pizza_type_id
group by pt.name
order by  total_quantity desc limit 5;
```

Result:

Result Grid | Filter Rows:

name	total_quantity
The Classic Deluxe Pizza	2416
The Barbecue Chicken Pizza	2372
The Hawaiian Pizza	2370
The Pepperoni Pizza	2369
The Thai Chicken Pizza	2315

Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
select pt.Category,count(od.quantity) as total_quantity
from order_details od
join pizzas p
on p.pizza_id = od.pizza_id
join pizza_types pt
on pt.pizza_type_id = p.pizza_type_id
group by pt.category;
```

Result

Result Grid | Filter Rows:

Category	total_quantity
Classic	14579
Veggie	11449
Supreme	11777
Chicken	10815

5

Determine the distribution of orders by hour of the day.

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

Result:

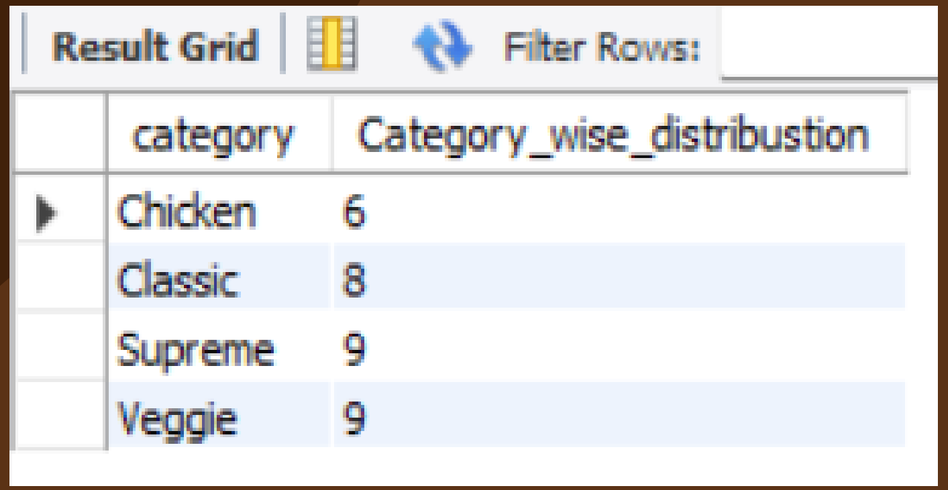
Result Grid | Filter Rows:

order_hours	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.

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

Result:



Result Grid | Filter Rows:

category	Category_wise_distribustion
Chicken	6
Classic	8
Supreme	9
Veggie	9

t

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

```
with pizza_order_per_day as(
select o.order_date,sum(od.quantity) as quantity_total from
orders o
join order_details od
on od.order_id = o.order_id
group by o.order_date)
select Round(Avg(quantity_total)) as avg_pizza_order_per_day from Pizza_order_per_day;
```

Result:

Result Grid | Filter Rows:

avg_pizza_order_per_day
138

Determine the top 3 most ordered pizza based on revenue.

```sql
select pt.name, round(Sum(price*quantity),0) as revenue
from pizza_types pt
join pizzas p
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;
```

Result:

Result Grid | Filter Rows:

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.

```sql
SELECT
category,
Round(SUM(quantity * price),0) AS revenue,
concat(ROUND(100.0 * SUM(quantity * price)/
 SUM(SUM(quantity * price)) OVER (), 2),"%") AS percent_contribution
FROM pizza_types pt
join pizzas p
on p.pizza_type_id = pt.pizza_type_id
join order_details od
on od.pizza_id = p.pizza_id
group by category
order by revenue desc;
```

Result:

Result Grid | Filter Rows:

category	revenue	percent_contribution
Classic	220053	26.91%
Supreme	208197	25.46%
Chicken	195920	23.96%
Veggie	193690	23.68%

Analyze the cumulative revenue generated over time.

```
select order_date,revenue, round(sum(revenue) over (order by order_date),2) as cum_revenue from
(select order_date, round(sum(price*quantity),2) as revenue
from orders o
join order_details od
on od.order_id = o.order_id
join pizzas p
on p.pizza_id = od.pizza_id
group by order_date) as sales;
```

Result:

Result Grid | Filter Rows:

order_date	revenue	cum_revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2	16560.7
2015-01-08	2838.35	19399.05
2015-01-09	2127.35	21526.4
2015-01-10	2463.95	23990.35
2015-01-11	1872.3	25862.65

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

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

Result:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	revenue	rn
The Chicken Alfredo Pizza	16900.25	2
The Southwest Chicken Pizza	34705.75	3
The Pepperoni, Mushroom, and Peppers Pizza	18834.5	1
The Big Meat Pizza	22968	2
The Napolitana Pizza	24087	3
The Brie Carre Pizza	11588.4999999999	1
The Spinach Supreme Pizza	15277.75	2
The Calabrese Pizza	15934.25	3
The Green Garden Pizza	13955.75	1
The Mediterranean Pizza	15360.5	2
The Spinach Pesto Pizza	15596	3