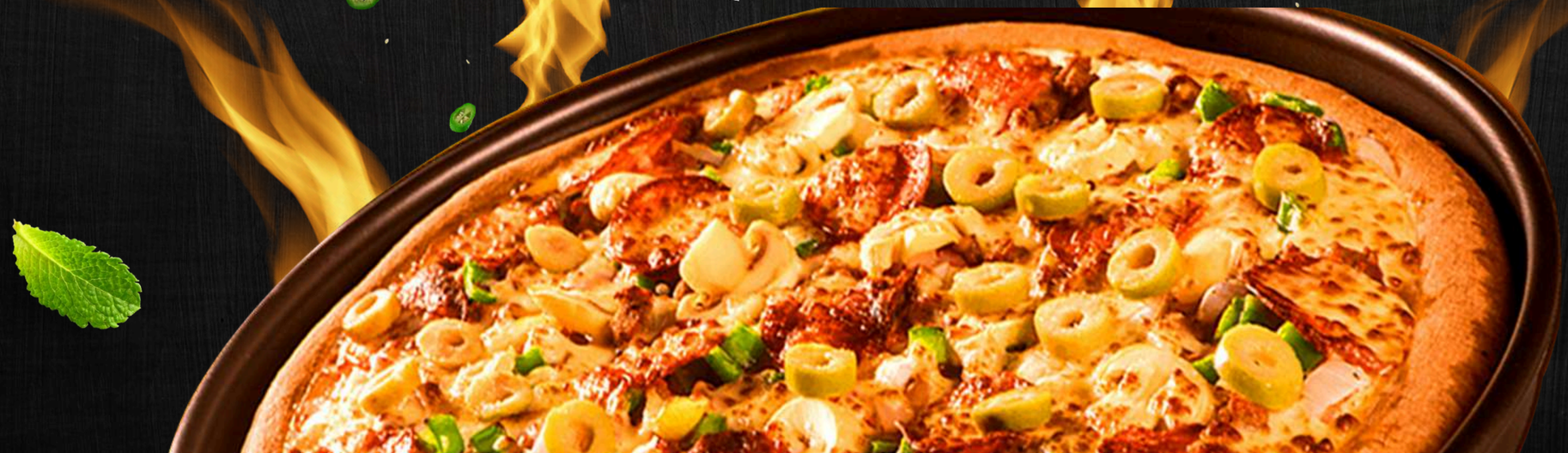
GAGAN TIW@RI

Home Menu About Us Contact

IN THIS PROJECT, I APPLIED SQL QUERIES TO EXTRACT, AGGREGATE, AND ANALYZE PIZZA SALES DATA IN ORDER TO ANSWER SPECIFIC BUSINESS-RELATED QUESTIONS



total_order
21350

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT
COUNT(ORDER_ID)
AS TOTAL_ORDER
FROM
ORDERS ;
```

GAGAN TIWARI

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
```

total_revenue
817860.05

IDENIFY THE HIGHEST-RICED PIZZA

name	price
The Greek Pizza	35.95

```
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;
```

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

```
SELECT  pizzas.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 pizzas.size
ORDER BY order_count DESC ;
```

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THERE QUNATITIES

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

```
SELECT
pizza_types.name, SUM(order_details.quantity) AS quantity
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.name
ORDER BY quantity DESC
LIMIT 5;
```

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

DETERMINE THE DISTRIBUTION OF ORDERS BY HOURS OF THE DAY ?

```
SELECT
  HOUR(ORDER_TIME), COUNT(ORDER_ID)
FROM
  ORDERS
GROUP BY HOUR(ORDER_TIME) ;
```

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

GROUP THE ORDER BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERD PER DAY

```
SELECT
ROUND(AVG(quantity), 0)
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;
```

ROUND(AVG(quantity), 0)
138

DETERMINE THE TOP 4 MOST ORDERED PIZZA TYPES 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 pizza_types.name order by revenue desc limit 4;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5

Advanced

CALCULATE THE % CONTRIBUTION OF EACH PIZZA TYPES 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;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

	order_date	cum_revenue
▶	2015-01-01	2713.850000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55

```
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 rever
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;
```

DETERMINE THE TOP 2 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

name	revenue
The Five Cheese Pizza	26066.5
The Mexicana Pizza	26780.75
The Pepperoni Pizza	30161.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Hawaiian Pizza	32273.25
The Italian Supreme Pizza	33476.75
The Spicy Italian Pizza	34831.25
The Classic Deluxe Pizza	38180.5
The California Chicken Pizza	41409.5
The Barbecue Chicken Pizza	42768
The Thai Chicken Pizza	43434.25

```
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;
```

THANK YOU FOR WATCHING

You'll work with practical datasets and mimic real professional workflows, making it a great hands-on resource for aspiring data analysts.