



# **SQL PROJECT ON PIZZA SALES**



Name : Mittul Negi  
PEC Chandigarh

Using SQL queries to solve  
questions related to pizza sales

# total orders placed

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

Result Grid	
	COUNT(order_id)
▶	21350

# TOTAL REVENUE FROM PIZZA SALES

```
SELECT SUM(a.quantity * b.price)
FROM order_details AS a
JOIN pizzas AS b
ON a.pizza_id = b.pizza_id;
```

	Result Grid    Filter Rows:
	SUM(a.quantity * b.price)
▶	817860.049999993

# HIGHEST PRICED PIZZA

```
SELECT c.name, d.price  
FROM pizza_types AS c  
JOIN pizzas AS d  
ON c.pizza_type_id=d.pizza_type_id  
WHERE price = (SELECT MAX(price) FROM pizzas);
```

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95

# MOST COMMON PIZZA SIZE ORDERED

```
SELECT p.size, sum(a.quantity) as q
FROM order_details as a
inner join pizzas as p
ON a.pizza_id=p.pizza_id
group by size
order by q desc limit 1;
```

Result Grid		
	size	q
▶	L	18956

## LIST OF TOP 5 MOST ORDERD PIZZA TYPES WITH THEIR TYPES

```
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

## DETERMINE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT HOUR(time), COUNT(order_id)  
FROM orders  
GROUP BY HOUR(time);
```

	HOUR(time)	COUNT(order_id)
▶	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
	10	8
	9	1

GROUP THE ORDERS BY DATE AND CALCULATE THE AVG. NO. OF PIZZAS ORDERED  
PER DAY

```
③ SELECT AVG(daily_orders) FROM(  
    SELECT orders.date, SUM(order_details.quantity) AS daily_orders  
    FROM orders JOIN order_details  
    ON orders.order_id=order_details.order_id  
    GROUP BY orders.date) AS daily_avg;
```

	AVG(daily_orders)
▶	138.4749

## DETERMINE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```
SELECT pizza_types.name, SUM(order_details.quantity*pizzas.price) AS revenue  
FROM order_details JOIN pizzas  
ON order_details.pizza_id=pizzas.pizza_id  
JOIN pizza_types  
ON pizzas.pizza_type_id=pizza_types.pizza_type_id  
GROUP BY pizza_types.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

# CALCULATE %AGE CONTRIBUTION OF EACH PIZZA CATEGORY TO TOTAL REVENUE

```
SELECT pizza_types.category,  
CONCAT(ROUND(SUM(order_details.quantity*pizzas.price)/  
          (SELECT SUM(order_details.quantity*pizzas.price) 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

```
WITH cte AS (
  SELECT orders.date,
    SUM(order_details.quantity*pizzas.price) AS daily_rev
  FROM orders JOIN order_details
  ON orders.order_id=order_details.order_id
  JOIN pizzas
  ON pizzas.pizza_id=order_details.pizza_id
  GROUP BY orders.date)
SELECT date,SUM(daily_rev) OVER(ORDER BY date asc) AS cum_rev
FROM cte;
```

	date	cum_rev
▶	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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.35000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.30000000003
	2015-01-14	32358.70000000004
	2015-01-15	34343.5000000001
	2015-01-16	36937.6500000001
	2015-01-17	39001.7500000001
	2015-01-18	40978.60000000006
	2015-01-19	43365.75000000001
	2015-01-20	45753.5000000001

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

```
WITH cte2 AS(WITH cte1 AS (SELECT pizza_types.category,pizza_types.name,  
SUM(order_details.quantity*pizzas.price) AS revenue  
FROM order_details JOIN pizzas  
ON pizzas.pizza_id=order_details.pizza_id  
JOIN pizza_types  
ON pizzas.pizza_type_id=pizza_types.pizza_type_id  
GROUP BY category,name)  
SELECT category,name,revenue,  
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) as rnk  
FROM cte1)  
SELECT category,name,revenue,rnk  
FROM cte2  
WHERE rnk<=3
```

	category	name	revenue	rnk
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3
	Supreme	The Spicy Italian Pizza	34831.25	1
	Supreme	The Italian Supreme Pizza	33476.75	2
	Supreme	The Sicilian Pizza	30940.5	3
	Veggie	The Four Cheese Pizza	32265.70000000065	1
	Veggie	The Mexicana Pizza	26780.75	2
	Veggie	The Five Cheese Pizza	26066.5	3

THANK YOU