



Home

About

Contact

**HELLO, I'M GODSONN BHATKAR
THIS IS MY SQL PROJECT ON
PIZZA SALES**



Home

About

Contact

THIS PROJECT SHOWS HOW I USED SQL TO FIND ANSWERS FROM PIZZA SALES DATA





RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

[About](#)[Contact](#)

2 • `select count(order_id) as total_order from orders`

Result Grid	
	count(order_id)
▶	21350



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
2 •   SELECT
3     ROUND(SUM(orders_details.quantity * pizzas.price),
4           2) AS total_sales
5
6   FROM
7     orders_details
8   JOIN
9     pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid	
	total_sales
▶	817860.05



IDENTIFY THE HIGHEST-PRICED PIZZA.

```
2  SELECT
3      pizza_types.name, pizzas.price
4  FROM
5      pizza_types
6          JOIN
7      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8  ORDER BY price DESC
9  LIMIT 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
2   SELECT
3       pizzas.size,
4       COUNT(orders_details.order_details_id) AS order_count
5   FROM
6       pizzas
7       JOIN
8           orders_details ON pizzas.pizza_id = orders_details.pizza_id
9   GROUP BY pizzas.size
10  ORDER BY order_count DESC;
```

Result Grid | Filter Rows

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



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

```
3   SELECT
4       pizza_types.name, SUM(orders_details.quantity) AS quantity
5   FROM
6       pizza_types
7       JOIN
8           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9       JOIN
10          orders_details ON orders_details.pizza_id = pizzas.pizza_id
11      GROUP BY pizza_types.name
12      ORDER BY quantity DESC
13      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



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

```
3   SELECT
4       pizza_types.category,
5       SUM(orders_details.quantity) AS quantity
6   FROM
7       pizza_types
8           JOIN
9       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10          JOIN
11      orders_details ON orders_details.pizza_id = pizzas.pizza_id
12  GROUP BY pizza_types.category
13  ORDER BY quantity DESC;
```

Result Grid | Filter Rows

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
2   SELECT  
3       HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
4   FROM  
5       orders  
6   GROUP BY HOUR(order_time);
```

Result Grid | Filter

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2222



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

```
2      -- category-wise distribution of pizzas.  
3 •  select category, count(name) from pizza_types  
4      group by category;
```

Result Grid | Filter Rows

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



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

```
2   SELECT
3       pizza_types.name,
4       SUM(orders_details.quantity * pizzas.price) AS revenue
5   FROM
6       pizza_types
7       JOIN
8           pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9       JOIN
10          orders_details ON orders_details.pizza_id = pizzas.pizza_id
11      GROUP BY pizza_types.name
12      ORDER BY revenue DESC
13      LIMIT 3;
```

Result Grid | Filter Rows:

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



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
3   select order_date,  
4       sum( revenue ) over(order by order_date) as cum_revenue  
5   from ( select orders.order_date,  
6           sum(orders_details.quantity * pizzas.price) as revenue  
7           from orders_details join pizzas on orders_details.pizza_id = pizzas.pizza_id join orders  
8           on orders.order_id = orders_details.order_id  
9       group by orders.order_date) as sales;
```

Result Grid | Filter Rows:

	order_date	cum_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