



USING SQL QUERIES



PIZZA SALES ANALYSIS

QUESTIONS SOLVED

- Q1.-- total pizza ordered.
- Q2.-- calculate the total revenue generated by pizza sales
- Q3.-- identify the highest price pizza...
- Q4.-- identify the most common size pizza ordered
- Q5.-- list the top5 most ordered pizza types along with quantities.
- Q6.-- find the total quantity of pizzas sold based on category
- Q7.-- determine the orders on the basis of hours of the day
- Q8.-- Select category wise no. of pizzas....
- Q9.-- group the order of pizzas by date and calculate the average ordered of pizza per day
- Q10.-- determine the top 3 most ordered pizzas based on revenue..
- Q11.-- total sales of the pizza category wise based on percentage of over all sales...

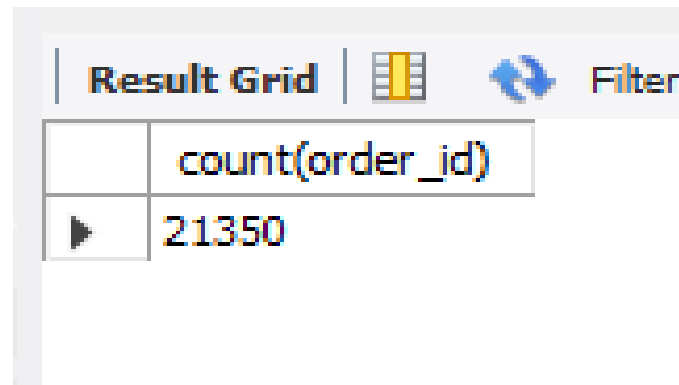
Q1.-- total pizza ordered.....

```
select count(order_id)
from orders
```

Result Grid				Filter
	count(order_id)			
▶	21350			

Q2.-- calculate the total revenue generated by pizza sales

```
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;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row with the value '21350'. The column header is 'count(order_id)'. There are icons for 'Filter' and a 'Refresh' button (circular arrow) above the grid.

	count(order_id)
▶	21350

Q3.-- identify the highest price pizza...

```
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;
```

	name	price
▶	The Greek Pizza	35.95

Q4.-- identify the most common size pizza ordered

SELECT

pizzas.size,count(order_details.order_details_id) as
order_count

FROM order_details

join pizzasON order_details.pizza_id = pizzas.pizza_id

group by pizzas.size

order by count(order_details_id) desc;

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

Q5.-- list the top5 most ordered pizza types along with quantities.

```
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;
```

Result Grid			Filter Rows:
	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	

Q6.-- find the total quantity of pizzas sold based on category

```
select
```

```
pizza_types.category, sum(order_details.quantity) as
```

```
total_orders
```

```
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;
```

Result Grid			Filter Rows:	
	category	total_orders		
▶	Classic	14888		
	Veggie	11649		
	Supreme	11987		
	Chicken	11050		

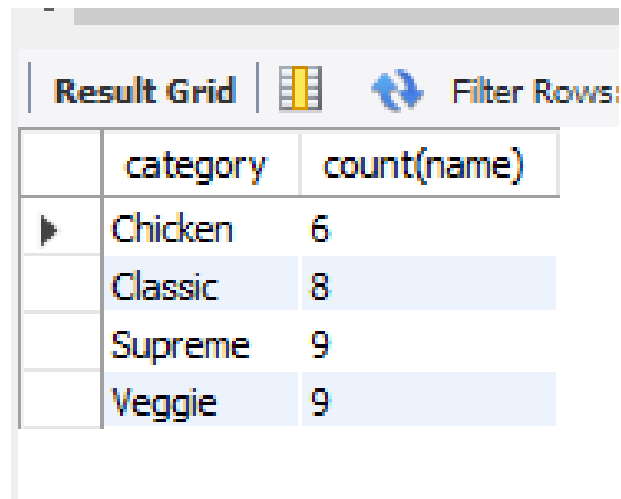
Q7.-- determine the orders on the basis of hours of the day

```
select hour(order_time), count(order_id)
from orders
group by hour(order_time)
```

Result Grid		
Filter Rows:		
	hour(order_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

Q8.– Select category wise no. of pizzas....

```
SELECT category , count(name)
from pizza_types
group by category;
```






The screenshot shows a database query result grid. At the top, there is a toolbar with the text 'Result Grid', a grid icon, a refresh icon, and the text 'Filter Rows:'. Below the toolbar is a table with two columns: 'category' and 'count(name)'. The table contains four rows of data: 'Chicken' with a count of 6, 'Classic' with a count of 8, 'Supreme' with a count of 9, and 'Veggie' with a count of 9. The first row is highlighted with a blue background.

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



Q9.-- group the order of pizzas by date and calculate the average ordered of pizza per day

```
select avg(quantity) as avg_order_per_day from (select orders.order_date,  
round(sum(order_details.quantity),0) as quantity  
from orders  
join  
order_detail on orders.order_id = order_details.order_id  
group by orders.order_date) as avg_orders;
```

Result Grid			 Filter Rows
	avg_order_per_day		
	138.4749		

Q10.-- determine the top 3 most ordered pizzas based on revenue..

```
select
pizza_types.name, round(sum(order_details.quantity *
pizzas.price),0) 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 3;
```

Result Grid |   Filter Rows:

	name	revenue
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

Q11.-- total sales of the pizza category wise
based on percentage of over all sales...

```
select  
pizza_types.category, round(sum(order_details.quantity *  
pizzas.price),0) as revenue  
from pizza_types  
join  
Pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id  
join  
order_detailson order_details.pizza_id=pizzas.pizza_id  
group by pizza_types.category  
order by revenue desc;
```

Result Grid			Filter Rows
	category	revenue	
▶	Classic	220053	
	Supreme	208197	
	Chicken	195920	
	Veggie	193690	

THANK YOU