

Pizza store analysis



Q.Retrieve the total number of orders placed.

```
6 • select count(order_id) as total_count from orders;
```

The screenshot shows a MySQL Workbench interface. At the top, there is a status bar with the number '6' and a blue dot. Below it is a code editor window containing the SQL query: 'select count(order_id) as total_count from orders;'. The result grid below the code editor shows one row with the column 'total_count' and the value '21350'.

total_count
21350

Total number of order placed 21350

Calculate the total revenue generated from pizza sales.

SELECT

```
ROUND(SUM(od.quantity * p.price), 2)
```

FROM

```
order_details AS od
```

JOIN

```
pizzas AS p ON od.pizza_id = p.pizza_id;
```

	ROUND(SUM(od.quantity * p.price), 2)
▶	817860.05



Identify the highest-priced pizza.

```
select * from pizzas  
order by price desc  
limit 1 ;
```

Result Grid | Filter Rows: Export:

	pizza_id	pizza_type_id	size	price
▶	the_greek_xxL	the_greek	XXL	35.95



Identify the most common pizza size ordered.

```
SELECT  
    p.size, COUNT(od.order_id) AS total_oders  
FROM  
    pizzas AS p  
        JOIN  
    order_details AS od ON p.pizza_id = od.pizza_id  
GROUP BY p.size;
```

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



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

SELECT

```
pt.pizza_type_id,  
SUM(od.quantity) AS numberoforders,  
pt.name  
FROM  
pizzas AS p  
JOIN  
order_details AS od ON p.pizza_id = od.pizza_id  
JOIN  
pizza_types AS pt ON pt.pizza_type_id = p.pizza_type_id  
GROUP BY pt.name , pt.pizza_type_id  
ORDER BY numberoforders DESC
```

j

	pizza_type_id	numberoforders	name
▶	classic_dlx	2453	The Classic Deluxe Pizza
	bbq_ckn	2432	The Barbecue Chicken Pizza
	hawaiian	2422	The Hawaiian Pizza
	pepperoni	2418	The Pepperoni Pizza
	thai_ckn	2371	The Thai Chicken Pizza
	cali_ckn	2370	The California Chicken Pizza
	sicilian	1938	The Sicilian Pizza
	spicy_ital	1924	The Spicy Italian Pizza
	southw_ckn	1917	The Southwest Chicken Pizza
	big_meat	1914	The Big Meat Pizza
	four_cheese	1902	The Four Cheese Pizza
	ital_supr	1884	The Italian Supreme Pizza
	veggie_veg	1526	The Vegetables + Vegetable...



Determine the distribution of orders by hour of the day.

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

	hours	totalorders
>	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28

Join relevant tables to find the category-wise distribution of pizzas.

```
-- JOIN RELEVANT TABLES TO FIND THE  
SELECT  
    category, COUNT(name)  
FROM  
    pizza_project.pizza_types  
GROUP BY category
```

	category	COUNT(name)
>	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
select round(avg(quantity),0) from
(select o.date , sum(od.quantity) as quantity from orders as o
join order_details as od on o.order_id = od.order_id
group by o.date ) as orderavg;
```

Result Grid			 Filter Rows:	<input type="text"/>
	round(avg(quantity),0)			
▶	139			

Determine the top 5 most ordered pizza types based on revenue.

```
select pt.name , sum(od.quantity * p.price) as revenu from order_details as od
join pizzas as p on p.pizza_id = od.pizza_id
join pizza_types as pt on pt.pizza_type_id = p.pizza_type_id
group by pt.name
order by revenu desc limit 5
```

j

	name	revenu
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Spicy Italian Pizza	34831.25

Calculate the percentage contribution of each pizza type to total revenue.

```
SELECT pt.category ,  
    round(SUM(od.quantity * p.price) / (SELECT  
        round(SUM(od.quantity * p.price),2)  
    FROM  
        pizzas AS p  
        JOIN  
            order_details AS od ON p.pizza_id = od.pizza_id)*100,2) AS  
revenue FROM  
    pizzas AS p  
    JOIN  
        order_details AS od ON p.pizza_id = od.pizza_id  
        JOIN  
            pizza_types AS pt ON p.pizza_type_id = pt.pizza_type_id  
GROUP BY pt.category  
;
```

	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

Analyze the cumulative revenue generated over time.

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

	datebyDay	total	cum_revenue
▶	01-01-2023	2713.8500000000004	2713.8500000000004
	01-02-2023	3189.1999999999994	5903.049999999999
	01-03-2023	1598.55	7501.599999999999
	01-04-2023	2176.8500000000004	9678.45
	01-05-2023	2571.95	12250.400000000001
	01-06-2023	3067.7499999999995	15318.150000000001
	01-07-2023	2231.5	17549.65
	01-08-2023	2440.55	19990.2
	01-09-2023	2352.85	22343.05
	01-10-2023	3202.149999999996	25545.199999999997
	01-11-2023	1986.65	27531.85
	01-12-2023	2076.7	29608.55
	02-01-2023	2731.899999999996	32340.449999999997

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

```
select pt.category , pt.name , sum(od.quantity * p.price) as revenue from pizzas as p
join order_details as od  on od.pizza_id = p.pizza_id
join pizza_types as pt on pt.pizza_type_id = p.pizza_type_id
group by pt.category , pt.name
order by revenue desc
limit 3
:
```

Result Grid | Filter Rows: Export:

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