SQL PROJECT ON PIZZA SALES

HERE'S QUESTION

BASICS

1.Retrieve the total number of orders placed.
2.Calculate the total revenue generated from pizza sales.
3.Identify the highest-priced pizza.
4.Identify the most common pizza size ordered.
5.List the top 5 most ordered pizza types along with their quantities.

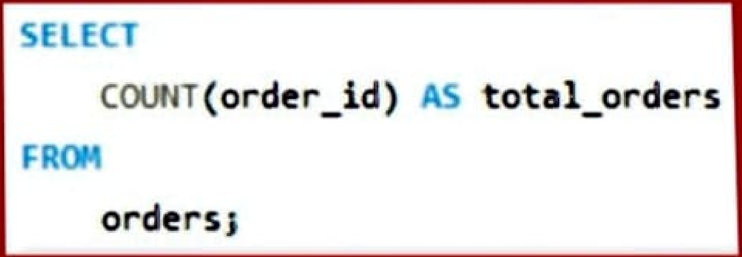
INTERMEDIATE

1.Join the necessary tables to find the total quantity of each pizza category ordered.
2.Determine the distribution of orders by hour of the day.
3.Join relevant tables to find the category-wise distribution of pizzas.
4.Group the orders by date and calculate the average number of pizzas ordered per day.
5.Determine the top 3 most ordered pizza types based on revenue.

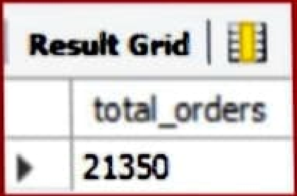
ADVANCED

1.Calculate the percentage contribution of each pizza type to total revenue.2.
Determine the top 3 most ordered pizza types based on revenue for each pizza category.

QUE-1.RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

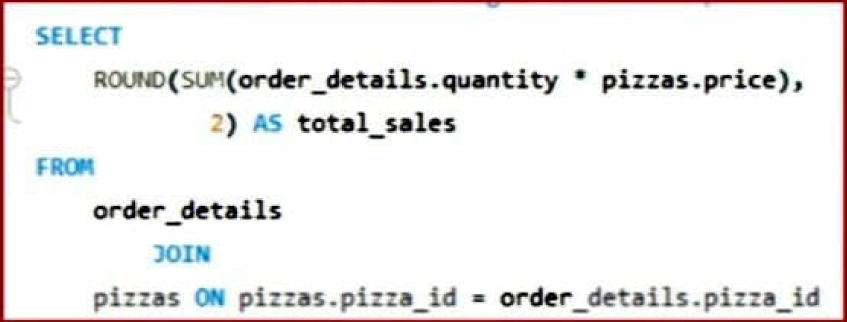


```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

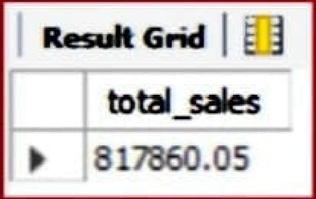


```
total_orders
21350
```

QUE-2.CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

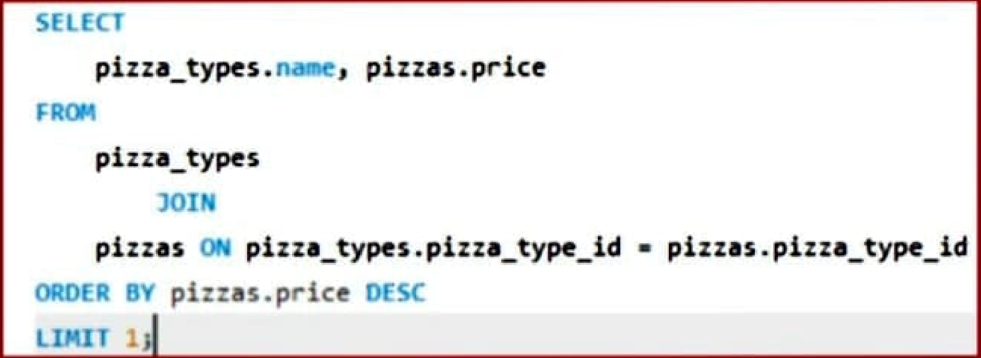


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

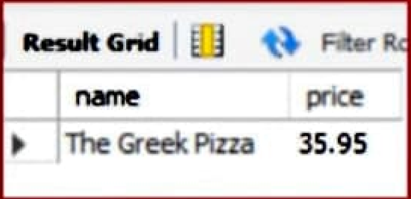


	total_sales
▶	817860.05

QUE-3.IDENTIFY THE HIGHEST-PRICED PIZZA.

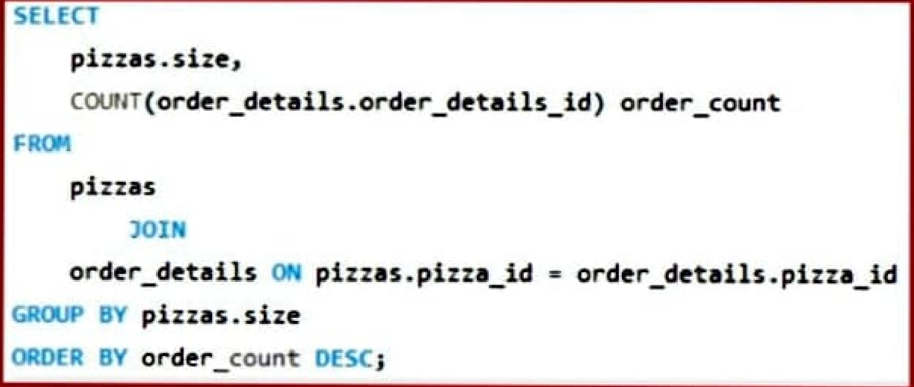


```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

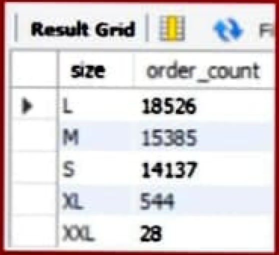


name	price
The Greek Pizza	35.95

QUE-4.IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

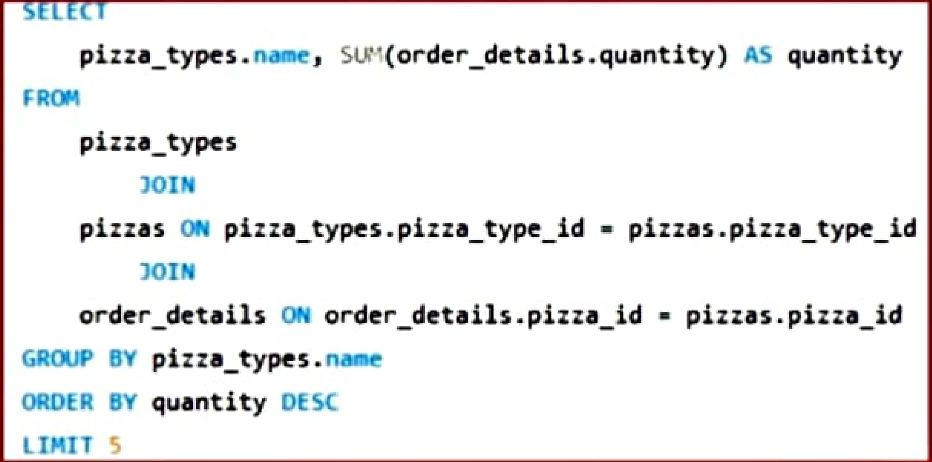


```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

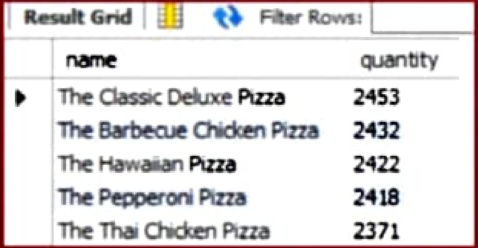


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

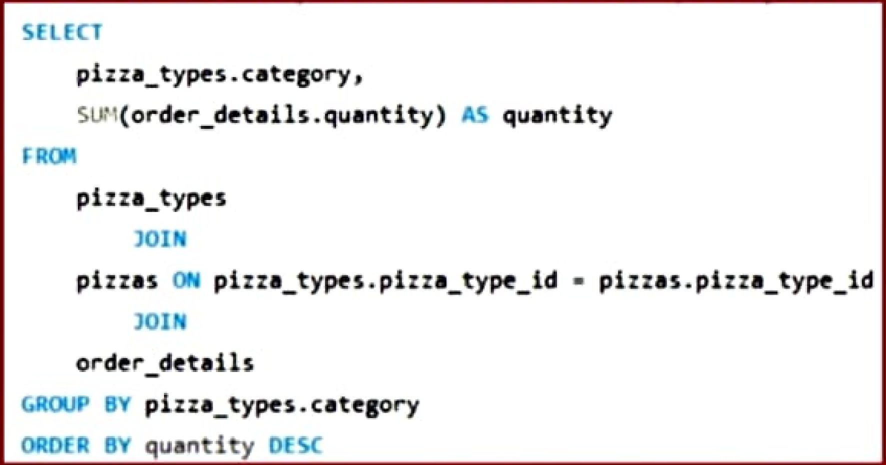


```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5
```

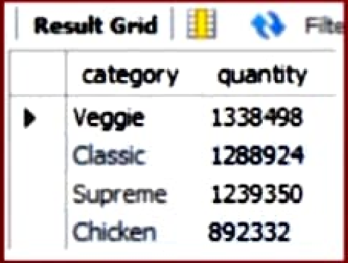


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

QUE-6.JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

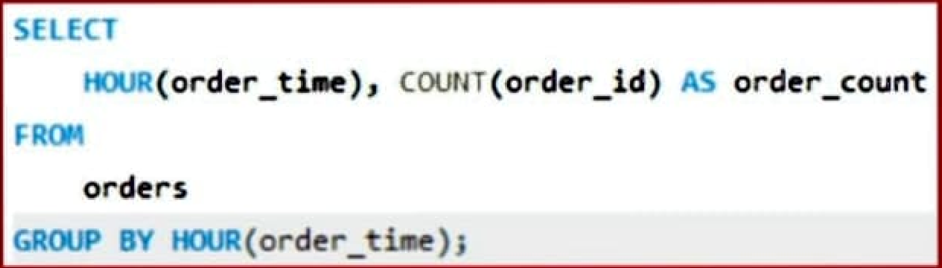


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details
GROUP BY pizza_types.category
ORDER BY quantity DESC
```

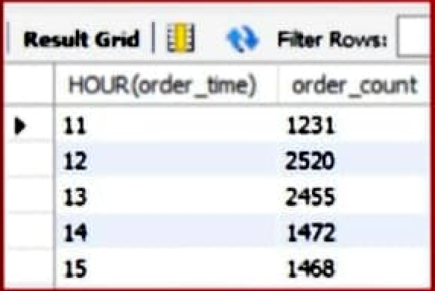


category	quantity
Veggie	1338498
Classic	1288924
Supreme	1239350
Chicken	892332

QUE-7.DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

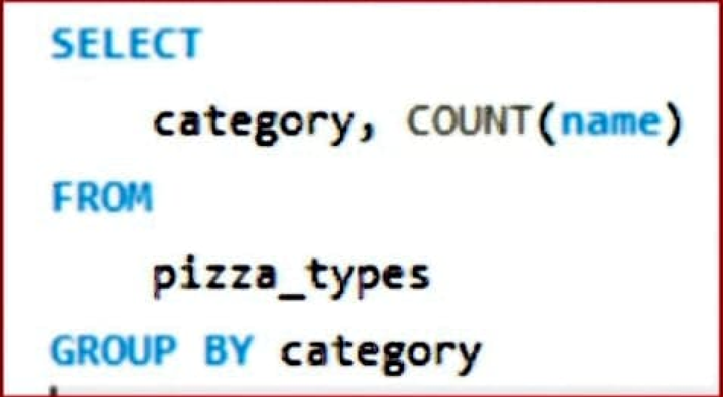


```
SELECT
    HOUR(order_time), COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

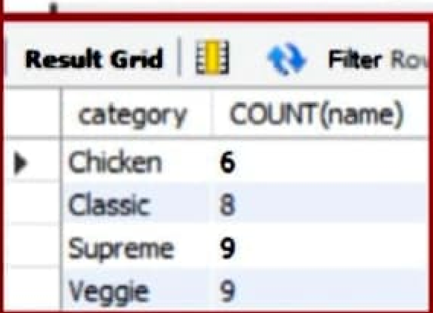


HOUR(order_time)	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

QUE-8.JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

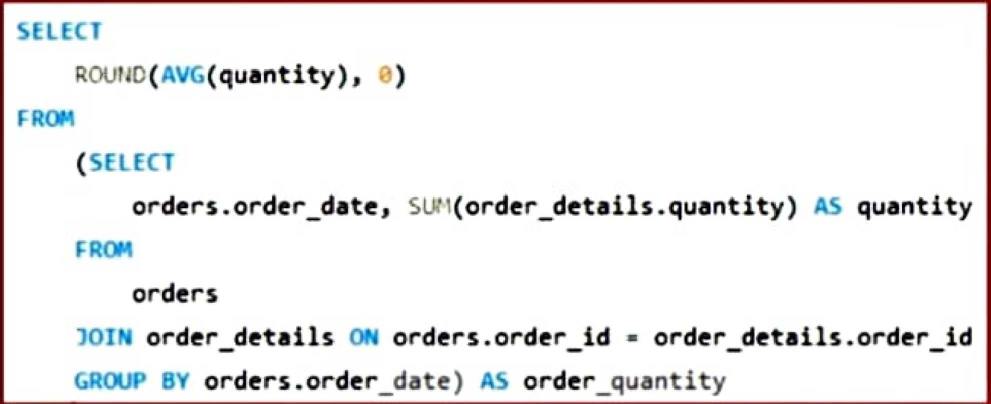


```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

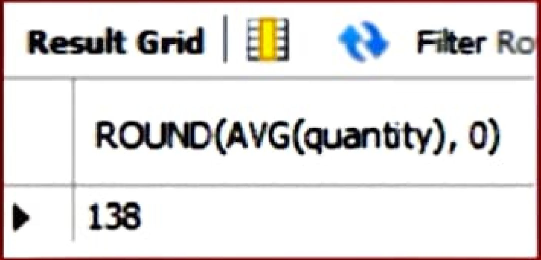


category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

QUE-9.GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

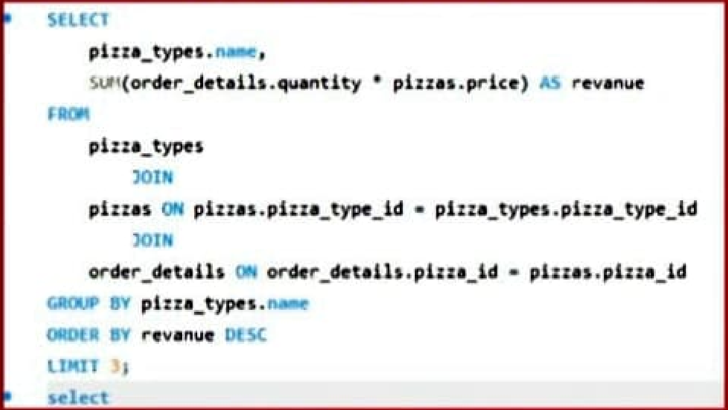


```sql
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity
```

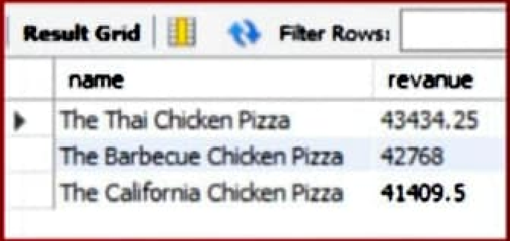


```
ROUND(AVG(quantity), 0)
138
```

QUE-10.DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

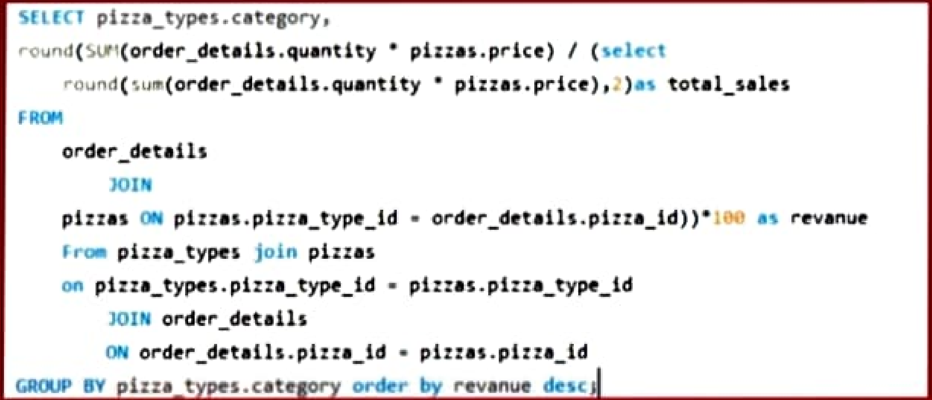


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revanue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revanue DESC
LIMIT 3;
select
```

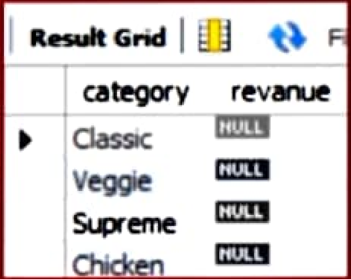


name	revanue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

QUE-11.CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

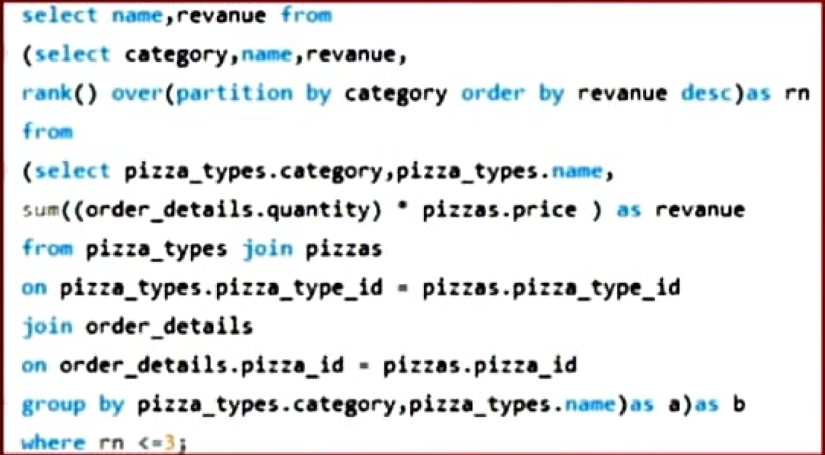


```
SELECT pizza_types.category,
round(SUM(order_details.quantity * pizzas.price) / (select
    round(sum(order_details.quantity * pizzas.price),2)as total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_type_id = order_details.pizza_id))*100 as revanue
    From pizza_types join 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 revanue desc;
```

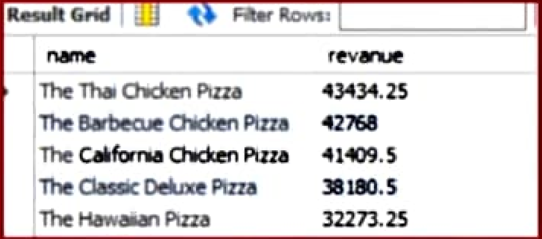


category	revanue
Classic	NULL
Veggie	NULL
Supreme	NULL
Chicken	NULL

QUE-12.DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.



```
select name,revanue from
(select category,name,revanue,
rank() over(partition by category order by revanue desc)as rn
from
(select pizza_types.category,pizza_types.name,
sum((order_details.quantity) * pizzas.price ) as revanue
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;
```



name	revanue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25

THANK YOU!