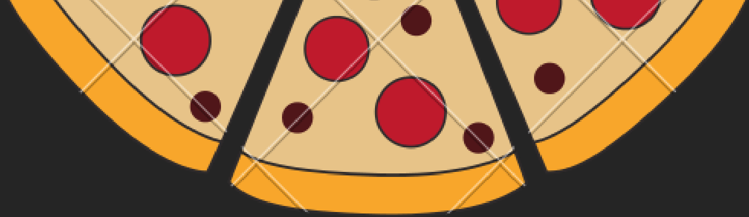
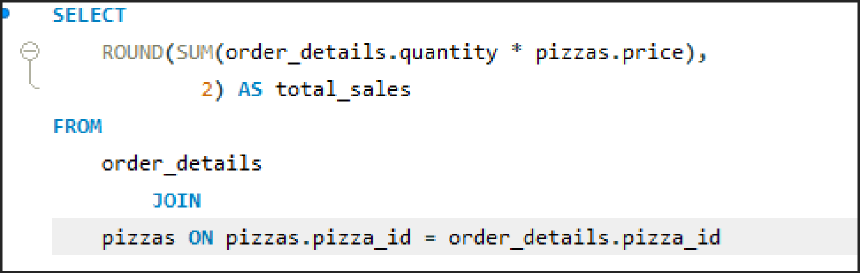
hello
lilesh mohane

PROJECT TITLE: ANALYZING PIZZA SALES USING SQL QUERIES

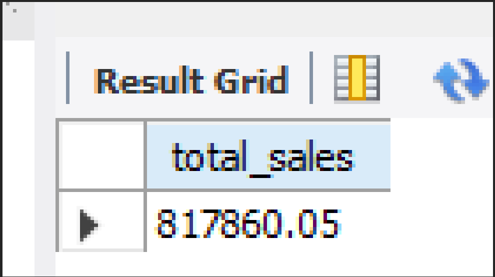
IN THIS PROJECT SOLVE SQL QUERIES QUESTIONS RELATED TO PIZZA SALES DATA



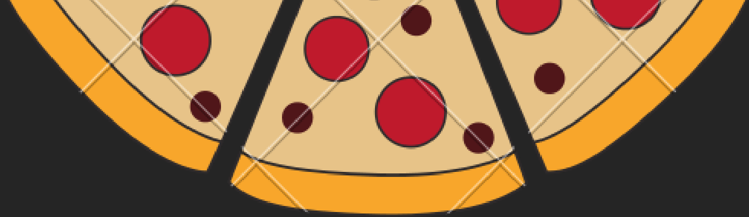
Q 1) CALCULATE TOTAL REVENUE GENERATED FROM PIZZA SALES



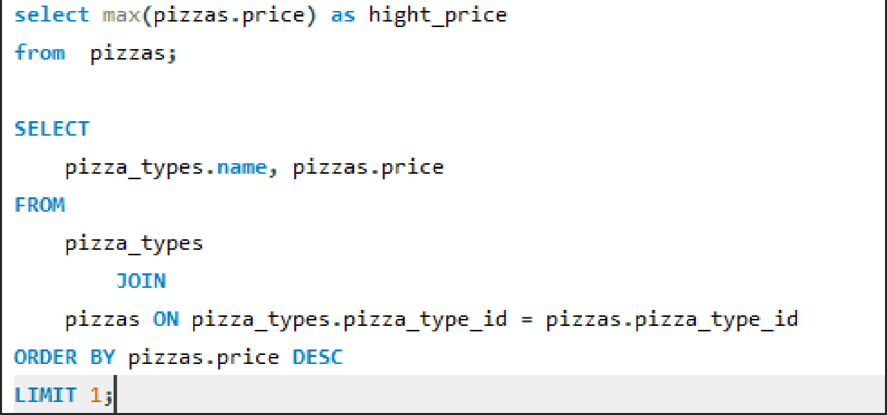
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```



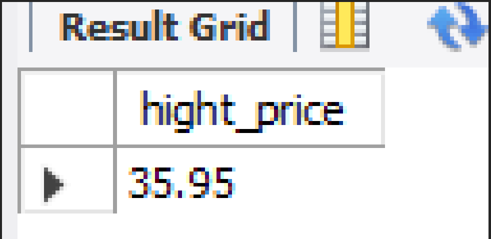
```
total_sales
817860.05
```



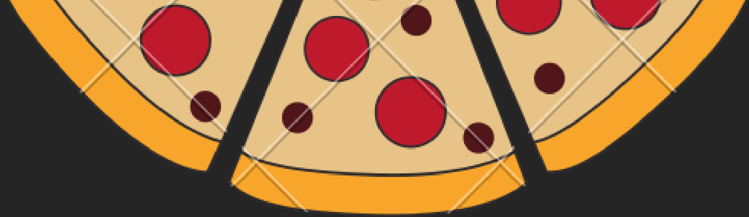
Q 2) IDENTIFY HIGHEST PRICE OF PIZZA



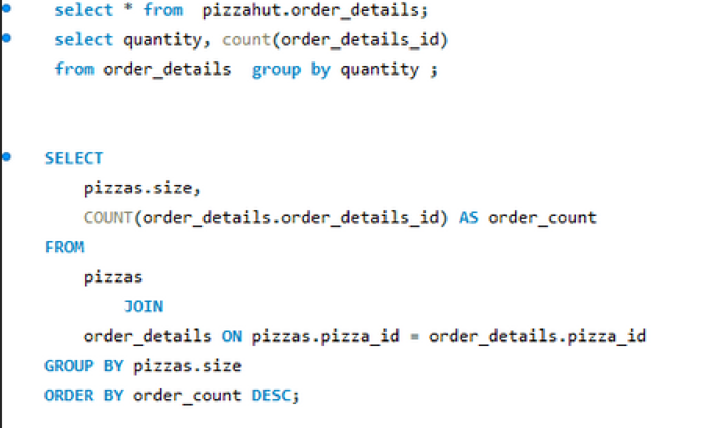
```
select max(pizzas.price) as hight_price
from  pizzas;


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;
```



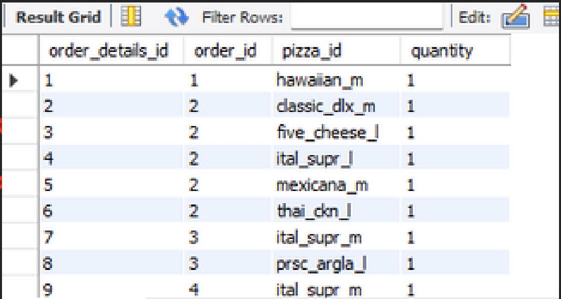
	hight_price
▶	35.95



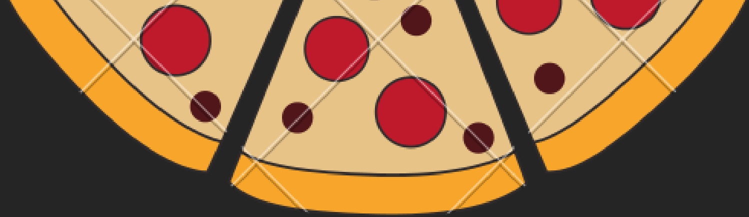
Q 3) IDENTIFY MOST COMMON PIZZA SIZE ORDERED



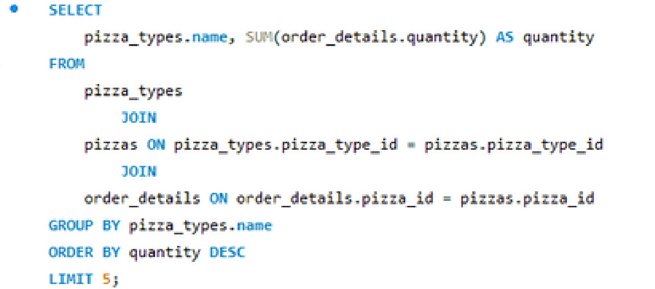
```
select * from  pizzahut.order_details;
select quantity, count(order_details_id)
from order_details  group by quantity ;


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



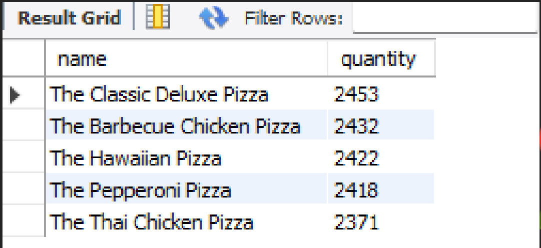
order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1
8	3	prsc_argla_l	1
9	4	ital_supr_m	1



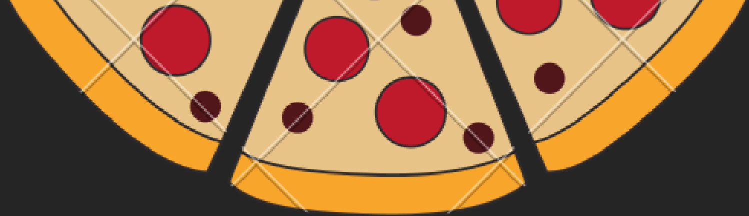
Q 4) LIST THE TOP 5 MOST ORDERED PIZZA TYPE ALONG WITH THERE QUANTITIES .



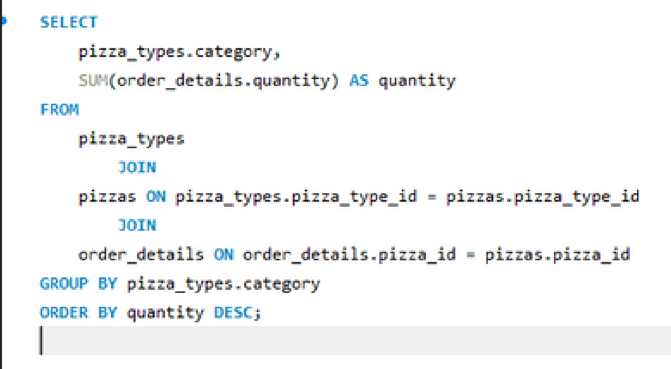
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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



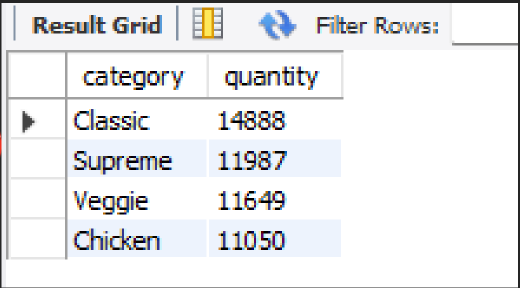
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



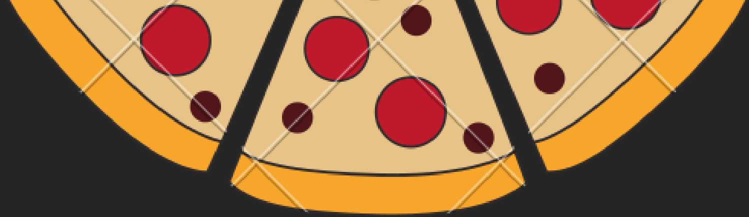
Q 5) JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED



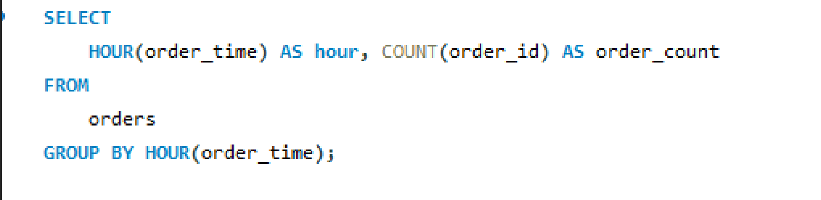
```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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



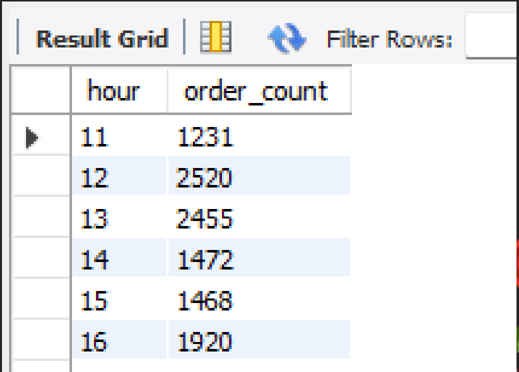
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



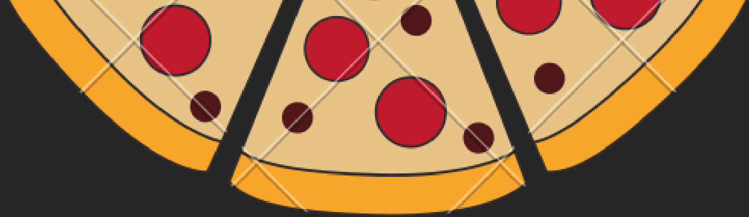
Q 6) DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY



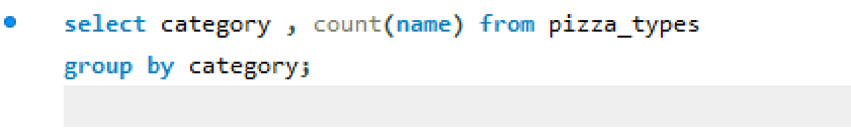
```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```



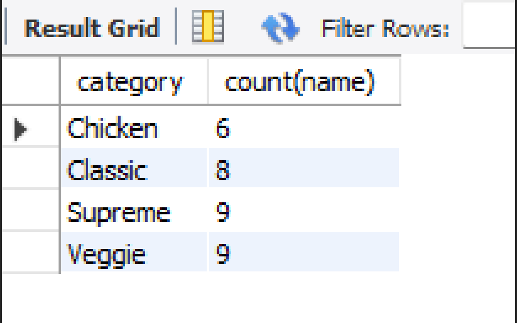
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920



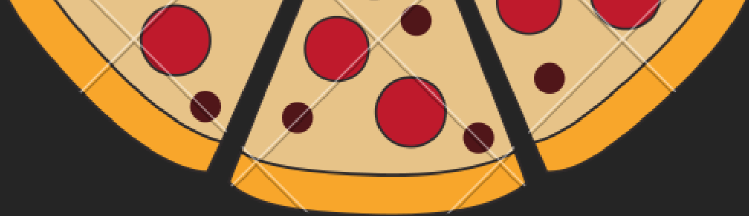
Q 7) JOIN RELEVENT TABLE TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZA



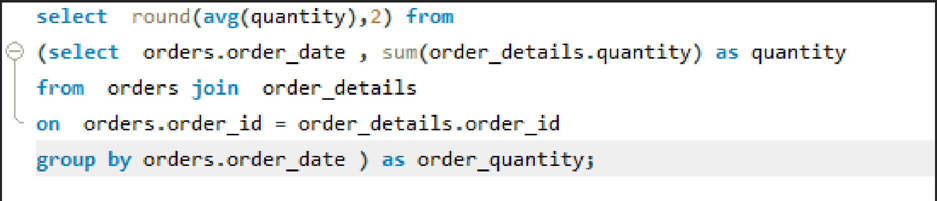
```
select category , count(name) from pizza_types
group by category;
```



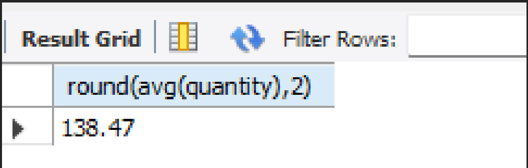
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



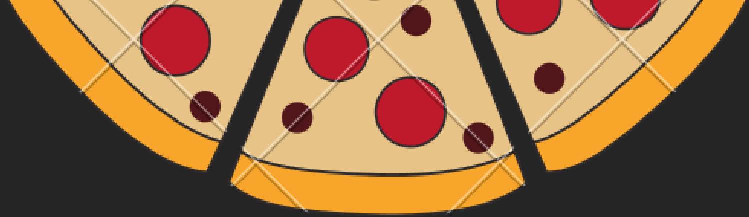
Q 8) GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY



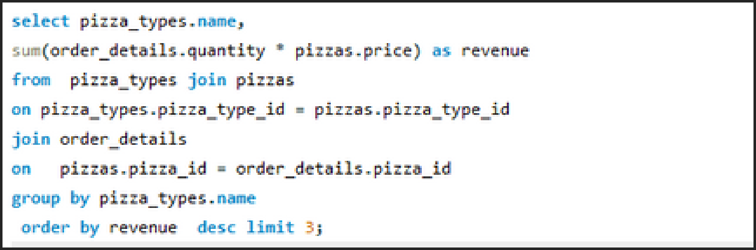
```
select  round(avg(quantity),2) from
(select  orders.order_date , sum(order_details.quantity) as quantity
from  orders join  order_details
on  orders.order_id = order_details.order_id
group by orders.order_date ) as order_quantity;
```



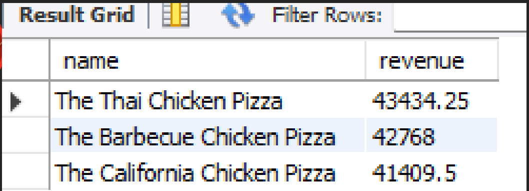
	round(avg(quantity),2)
▶	138.47



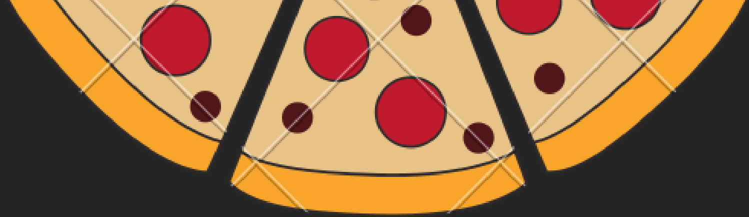
Q 9) DETERMINE THE TOP 3 MOST ORERD PIZZA TYPE BASED ON REVENUE .



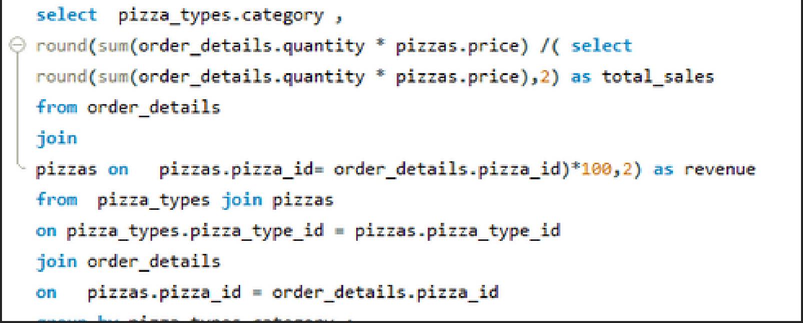
```
select pizza_types.name,
sum(order_details.quantity * pizzas.price) as revenue
from  pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on   pizzas.pizza_id = order_details.pizza_id
group by pizza_types.name
 order by revenue  desc limit 3;
```



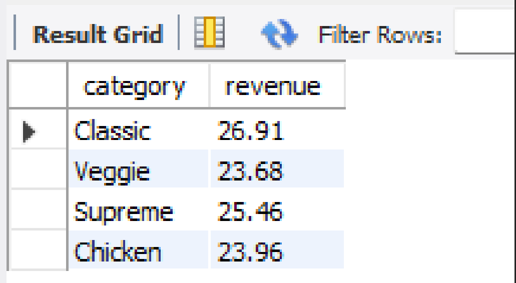
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



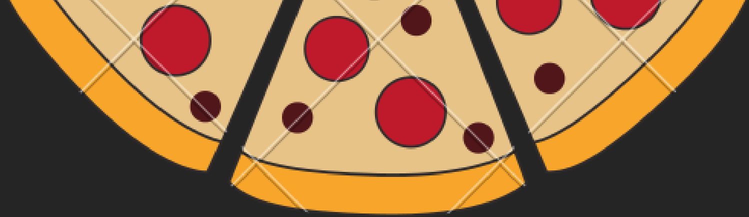
Q 10) CALCULATE TOTAL PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPES TO TOTAL REVENUE



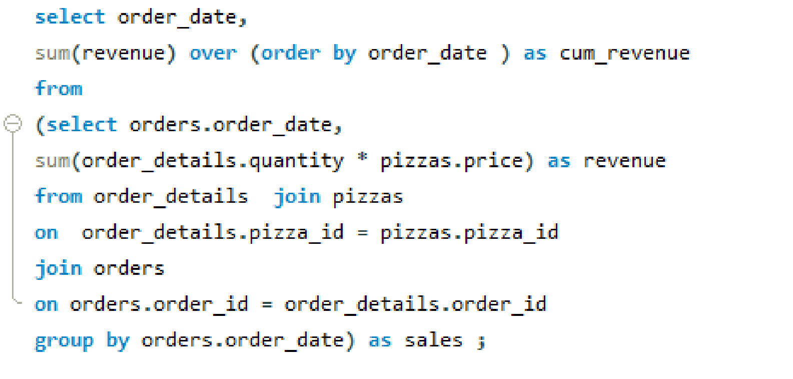
```
select  pizza_types.category ,
round(sum(order_details.quantity * pizzas.price) /( select
round(sum(order_details.quantity * pizzas.price),2) as total_sales
from order_details
join
pizzas on   pizzas.pizza_id= order_details.pizza_id)*100,2) as revenue
from  pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on   pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category ;
```



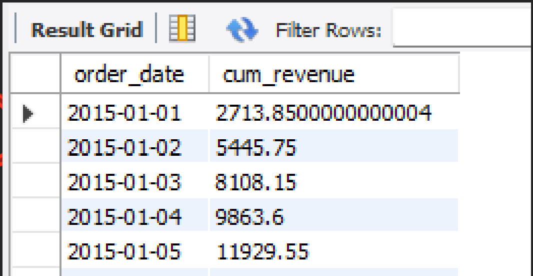
category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96



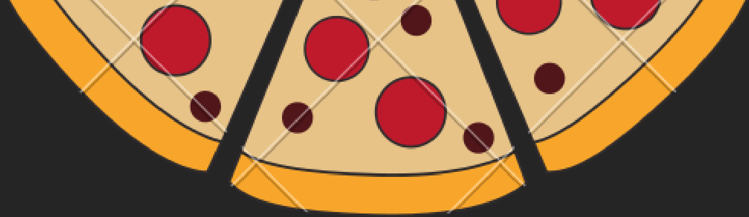
Q 11)ANALYZE THE CUMULATIVE REVENUE GENERAYED OVER TIME



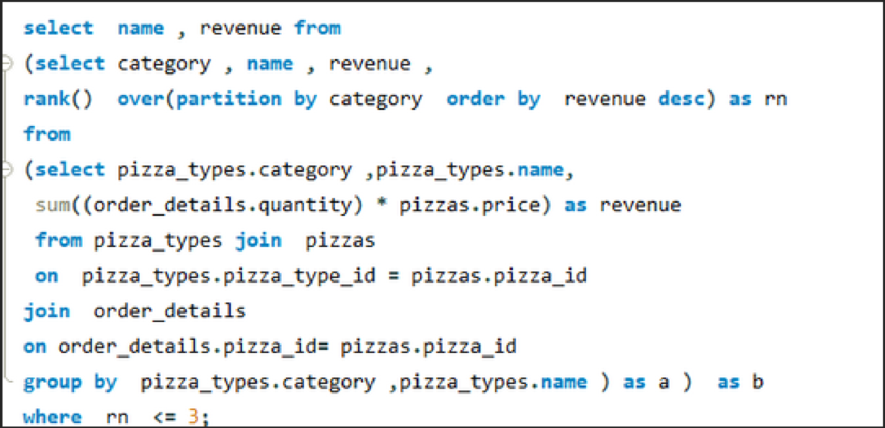
```
select order_date,
sum(revenue) over (order by order_date ) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details  join pizzas
on  order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales ;
```



```
   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
```



Q 12) DETERMINE TOP 3 MOST ORDER PIZZA TYPE BASED ON REVENUE OF EACH PIZAA CATEGORY



```
select  name , revenue from
(select category , name , revenue ,
rank()  over(partition by category  order by  revenue desc) as rn
from
(select pizza_types.category ,pizza_types.name,
 sum((order_details.quantity) * pizzas.price) as revenue
 from pizza_types join  pizzas
 on  pizza_types.pizza_type_id = pizzas.pizza_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;
```