SQL

PIZZA SALES

PROJECT

Overview of the Project

In this project four tablets/datasets of the pizza sales of a company are used and by analysing these datasets we are trying to find the useful insights from it with the help of SQL (Structured Query Language).As i used the csv data for this project, so firstly i need to upload all the four files. But before uploading the files to my SQL we need to create a new database. For this project I created a database named pizza_sales . Now it's time to import the datasets. But Some files, having less data and are small in size , are imported easily in a few minutes . While others , which contained a bit of large data, needed another method to import.

For importing such a file, first we need to prepare a raw structure of the table according to the existing file and then , import the file in that defined structure easily. It can be good for some other reasons as we need not worry about data types as we already defined them all carefully.

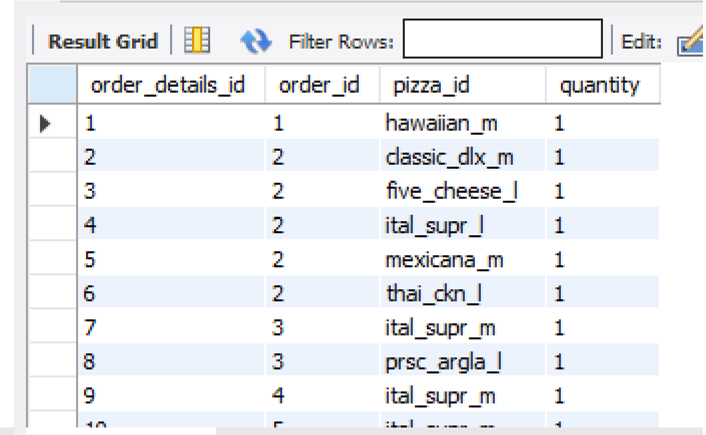
I will classify all the analysis objectives in the form of quotations , so it will become easier and clearer

I will provide the datasets through the github link .

Github Link

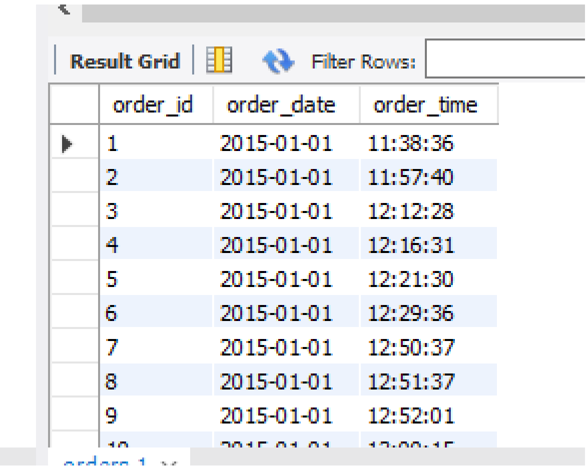
As there are four tables in this project as the dataset for the analysis. Here we introduce all the four tables one by one. So we can get familiar with the data and easily got the the hint, what it want to say;

Table 01 - ‘Order_details’ has four columns order_details_id , order_id, pizza_id , quantity.



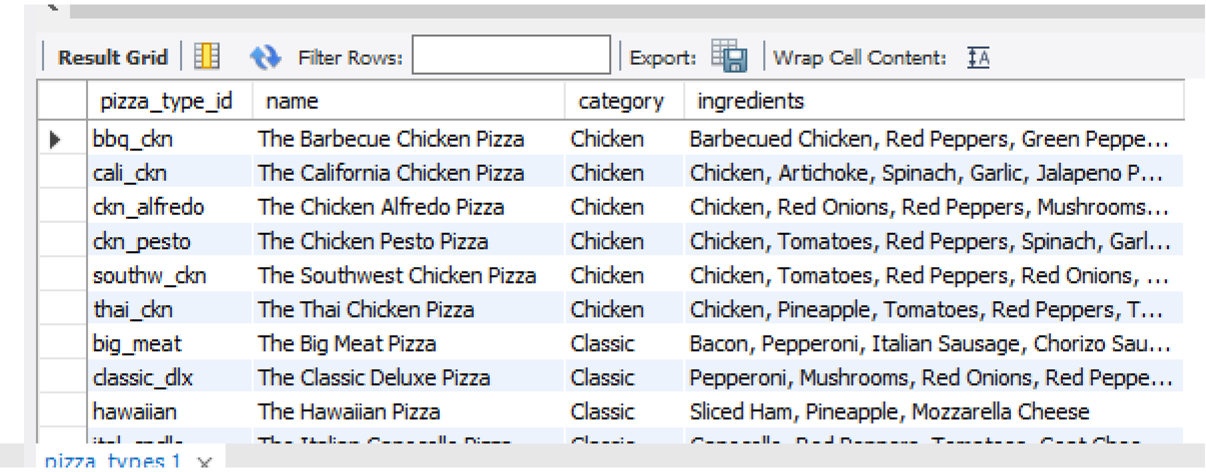
order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1
8	3	prsc_argla_l	1
9	4	ital_supr_m	1
10	5	ital_supr_m	1

Table 02 - ‘orders’ has three columns order_id , order_date , order_time .



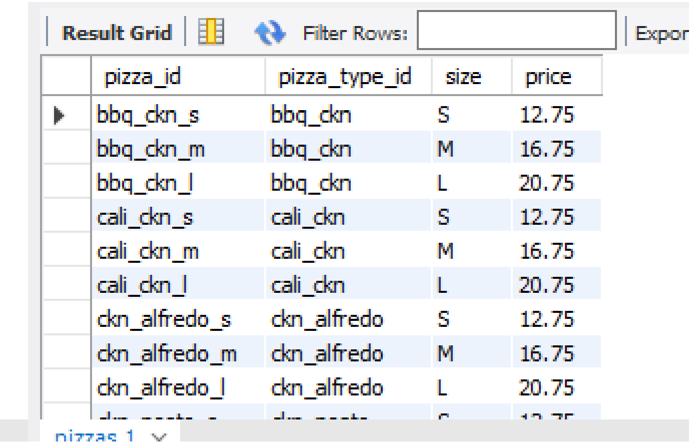
order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30
6	2015-01-01	12:29:36
7	2015-01-01	12:50:37
8	2015-01-01	12:51:37
9	2015-01-01	12:52:01
10	2015-01-01	13:00:15

Table 03 - ‘pizza_types’ has four column pizza_type_id , name , category , ingredients



pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...
big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sau...
classic_dlx	The Classic Deluxe Pizza	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppe...
hawaiian	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
ital_cpcllo	The Italian Capocollo Pizza	Classic	Capocollo, Red Peppers, Tomatoes, Goat Chee...

Table 04 - ‘pizzas’ has four columns pizza_id, pizza_type_id, size, price



pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75
ckn_alfredo_s	ckn_alfredo	S	12.75
ckn_alfredo_m	ckn_alfredo	M	16.75
ckn_alfredo_l	ckn_alfredo	L	20.75
ckn_pesto_s	ckn_pesto	S	12.75

NOTE;

*(useful while deciding the Primary key and Foreign Keys.)

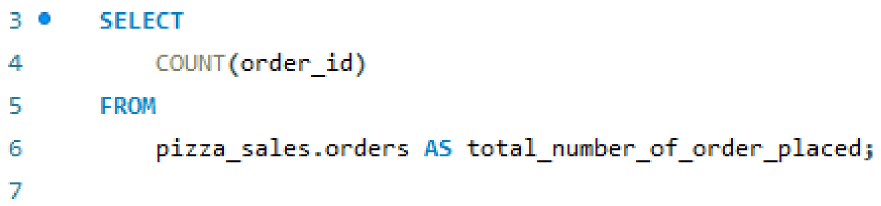
Pizza_type_id is common in pizzas and pizza_types

Pizza_id is common in pizzas and order_details

⇩

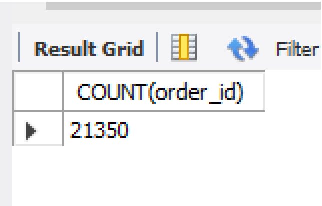
Question 1 ;

Retrieve the total number of orders placed.



```
SELECT
    COUNT(order_id)
FROM
    pizza_sales.orders AS total_number_of_order_placed;
```

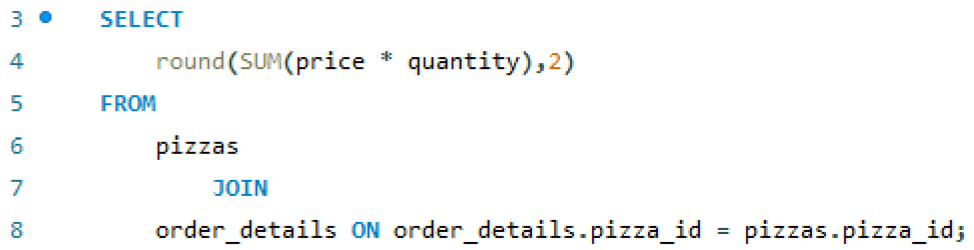
Output



COUNT(order_id)
21350

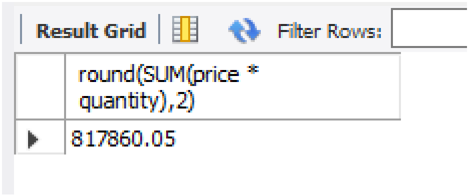
Question 2

Calculate the total revenue generated from pizza sales.



```sql
SELECT
    round(SUM(price * quantity),2)
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id;
```

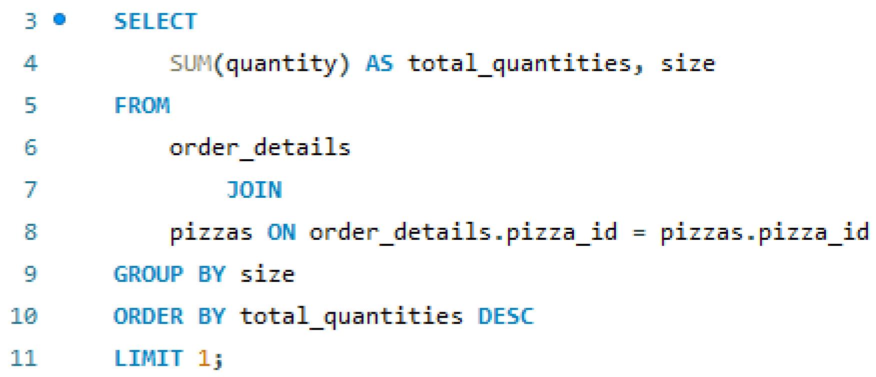
Output



round(SUM(price * quantity),2)
817860.05

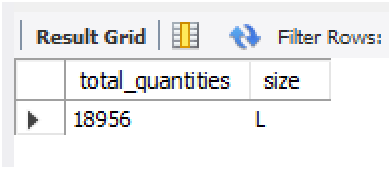
Question 3;

Identify the most common pizza size ordered.



```
SELECT
    SUM(quantity) AS total_quantities, size
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY size
ORDER BY total_quantities DESC
LIMIT 1;
```

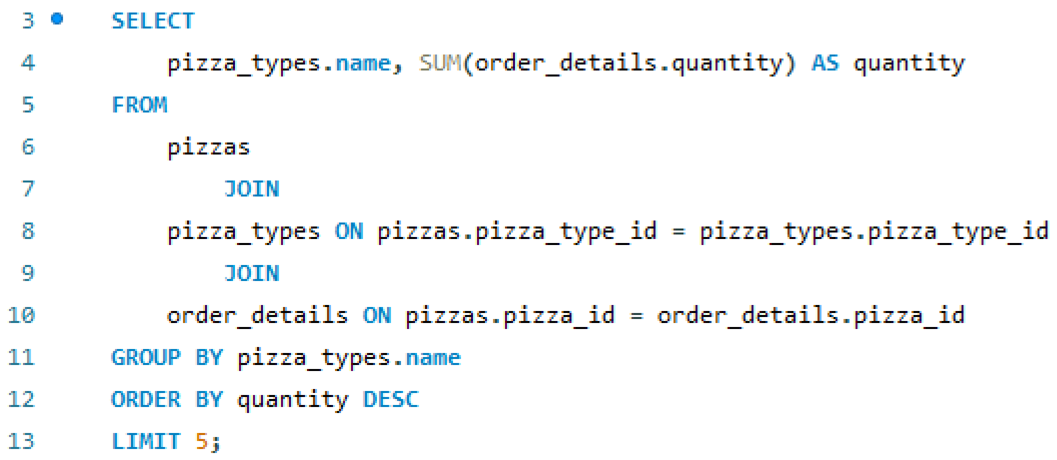
Output



total_quantities	size
18956	L

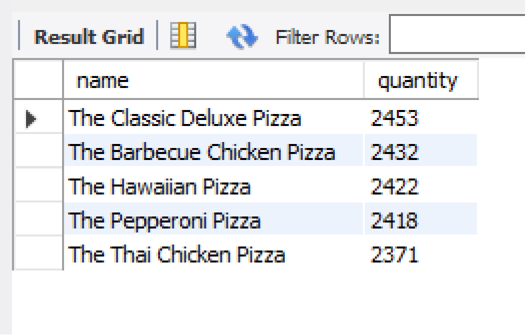
Question 04 ;

List the top 5 most ordered pizza types along with their quantities.



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

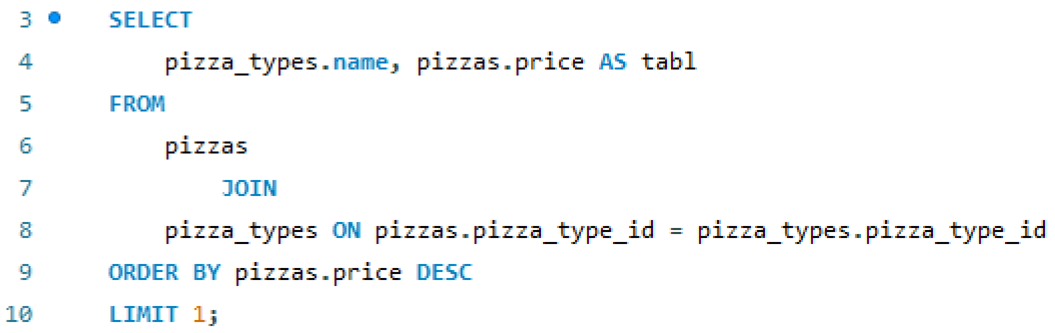
Output



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

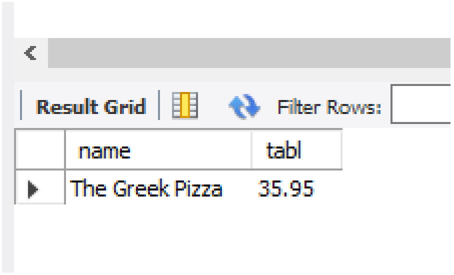
Question 05 ;

Identify the highest-priced pizza.



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

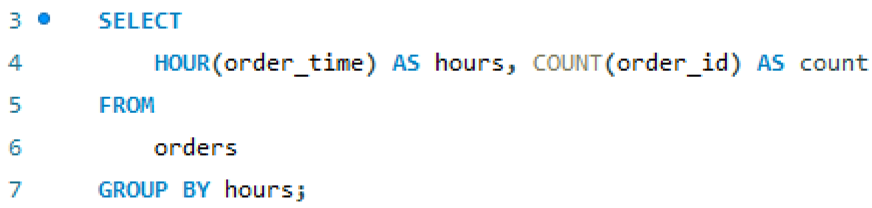
Output



name	tabl
The Greek Pizza	35.95

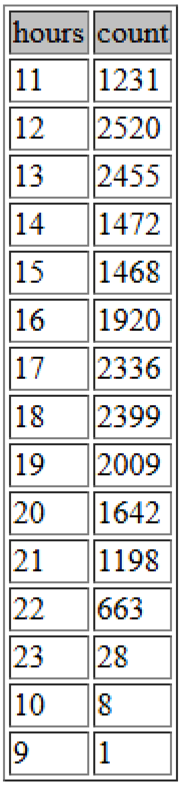
Question 06 ;

Determine the distribution of orders by hour of the day.



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

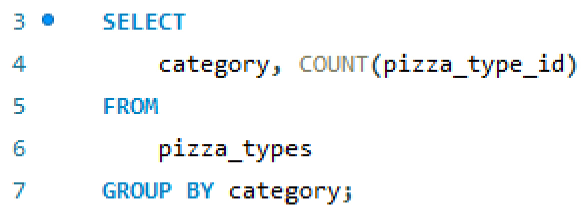
Output



hours	count
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

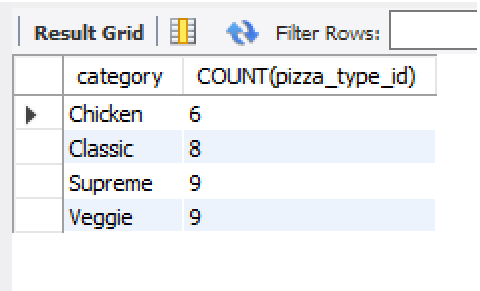
Question 07 ;

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



```sql
SELECT
    category, COUNT(pizza_type_id)
FROM
    pizza_types
GROUP BY category;
```

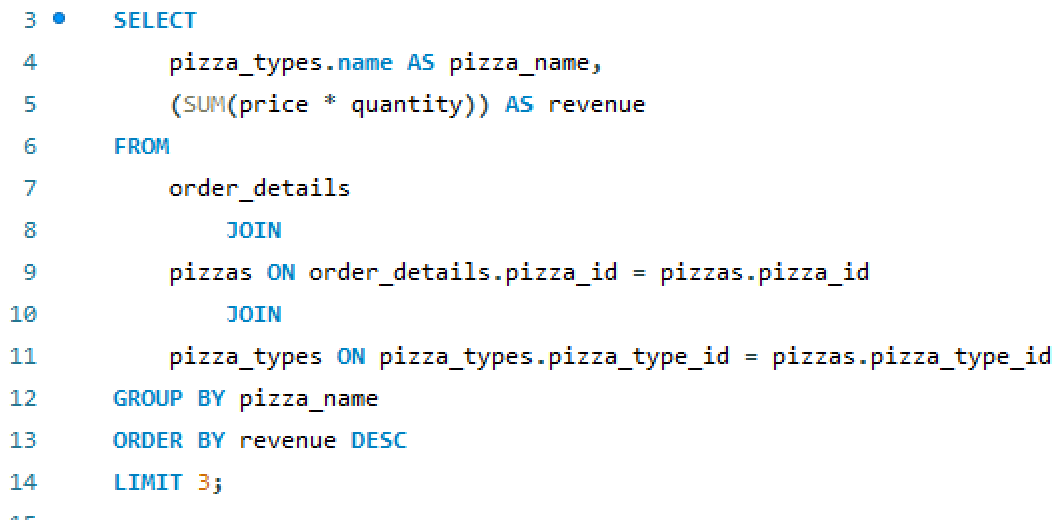
Output



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

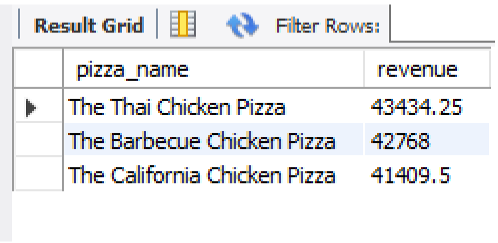
Question 8 ;

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



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

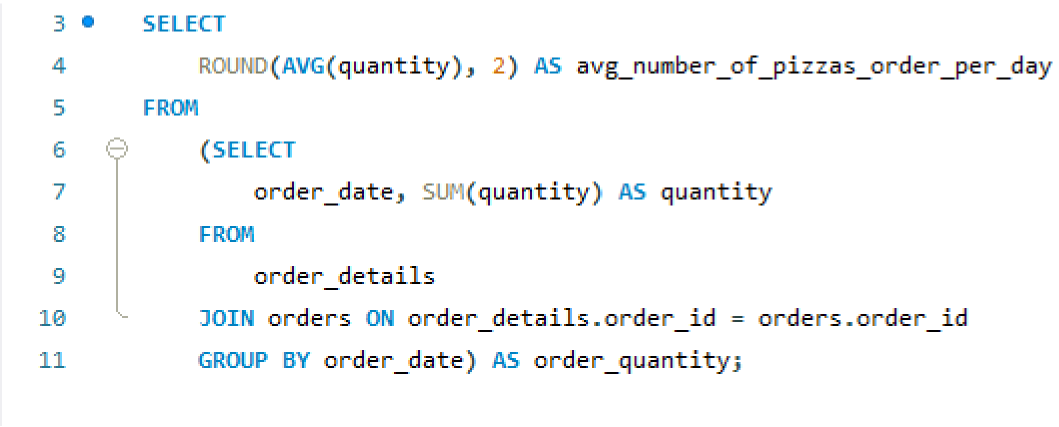
Output



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

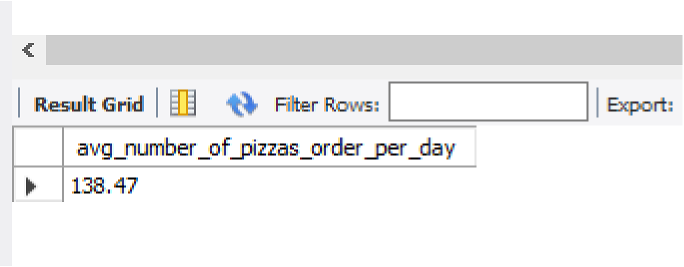
Question 09 ;

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



```
SELECT
    ROUND(AVG(quantity), 2) AS avg_number_of_pizzas_order_per_day
FROM
    (SELECT
        order_date, SUM(quantity) AS quantity
    FROM
        order_details
    JOIN orders ON order_details.order_id = orders.order_id
    GROUP BY order_date) AS order_quantity;
```

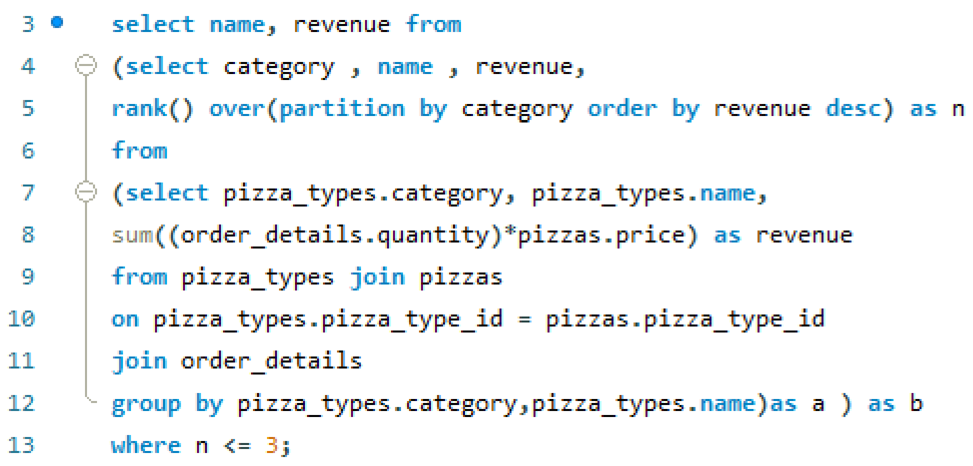
Output



avg_number_of_pizzas_order_per_day
138.47

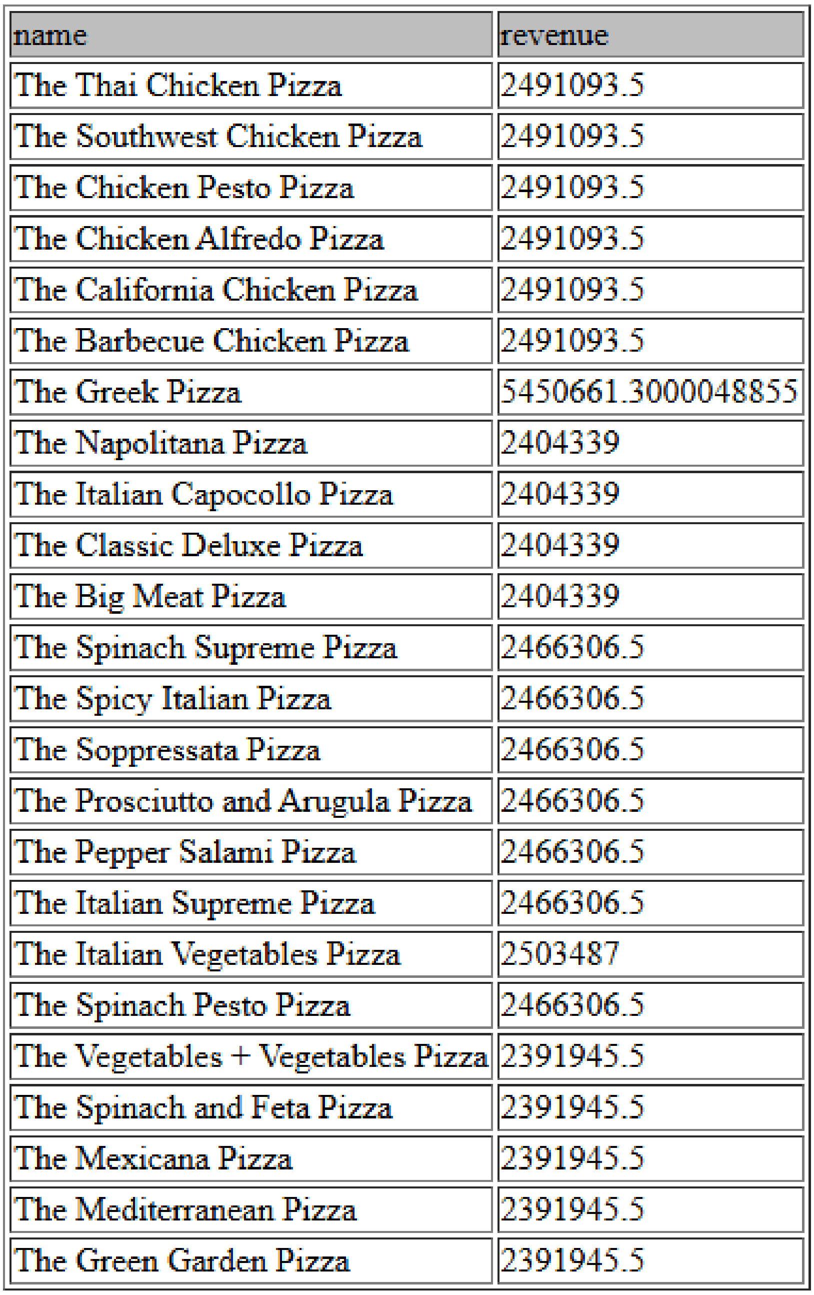
Question 10;

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



```sql
select name, revenue from
(select category , name , revenue,
rank() over(partition by category order by revenue desc) as n
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_type_id
join order_details
group by pizza_types.category,pizza_types.name)as a ) as b
where n <= 3;
```

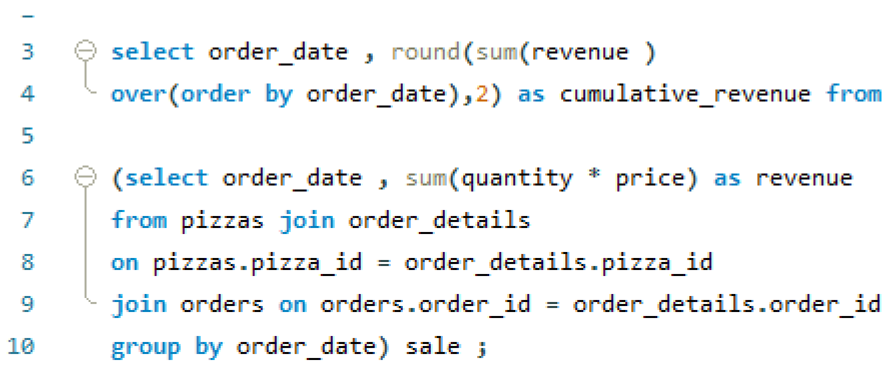
Output



name	revenue
The Thai Chicken Pizza	2491093.5
The Southwest Chicken Pizza	2491093.5
The Chicken Pesto Pizza	2491093.5
The Chicken Alfredo Pizza	2491093.5
The California Chicken Pizza	2491093.5
The Barbecue Chicken Pizza	2491093.5
The Greek Pizza	5450661.3000048855
The Napolitana Pizza	2404339
The Italian Capocollo Pizza	2404339
The Classic Deluxe Pizza	2404339
The Big Meat Pizza	2404339
The Spinach Supreme Pizza	2466306.5
The Spicy Italian Pizza	2466306.5
The Soppressata Pizza	2466306.5
The Prosciutto and Arugula Pizza	2466306.5
The Pepper Salami Pizza	2466306.5
The Italian Supreme Pizza	2466306.5
The Italian Vegetables Pizza	2503487
The Spinach Pesto Pizza	2466306.5
The Vegetables + Vegetables Pizza	2391945.5
The Spinach and Feta Pizza	2391945.5
The Mexicana Pizza	2391945.5
The Mediterranean Pizza	2391945.5
The Green Garden Pizza	2391945.5

Question 11 ;

Analyse the cumulative revenue generated over time.



```
select order_date , round(sum(revenue )
over(order by order_date),2) as cumulative_revenue from

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

Output

order_date	cumulative_revenue
2015-01-01	2713.85
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
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.2
2015-01-22	50300.9
2015-01-23	52724.6
2015-01-24	55013.85
2015-01-25	56631.4
2015-01-26	58515.8
2015-01-27	61043.85
2015-01-28	63059.85
2015-01-29	65105.15
2015-01-30	67375.45
2015-01-31	69793.3
2015-02-01	72982.5
2015-02-02	75311.1
2015-02-03	77925.9
2015-02-04	80159.8
2015-02-05	82375.6
2015-02-06	84885.55
2015-02-07	87123.2
2015-02-08	89158.2

2015-02-09	91353.55
2015-02-10	93410.05
2015-02-11	95870.05
2015-02-12	98028.85
2015-02-13	100783.35
2015-02-14	103102.5
2015-02-15	105243.75
2015-02-16	107212.55
2015-02-17	109334.45
2015-02-18	111977.3
2015-02-19	114007.55
2015-02-20	116898.7
2015-02-21	119009.7
2015-02-22	120589.65
2015-02-23	122758.2
2015-02-24	124952.75
2015-02-25	127294.05
2015-02-26	129555.35
2015-02-27	132413.3
2015-02-28	134952.9
2015-03-01	136551.45
2015-03-02	138930.5
2015-03-03	141218.4
2015-03-04	143662.7
2015-03-05	146013.35
2015-03-06	148527.3
2015-03-07	150927.75
2015-03-08	153115.9
2015-03-09	155450.45
2015-03-10	157839.15
2015-03-11	160046.85
2015-03-12	162041.75
2015-03-13	164828.4
2015-03-14	166867.85
2015-03-15	168936.45
2015-03-16	171231.5
2015-03-17	174196.8
2015-03-18	176272.2
2015-03-19	178660.8
2015-03-20	181122.05

2015-03-21	183389.45
2015-03-22	184648.7
2015-03-23	186881.25
2015-03-24	189043.55
2015-03-25	190971.3
2015-03-26	193186.8
2015-03-27	195931.6
2015-03-28	198183.7
2015-03-29	200337.95
2015-03-30	202593.4
2015-03-31	205350
2015-04-01	207526.85
2015-04-02	210074
2015-04-03	212612.2
2015-04-04	215379.75
2015-04-05	217289.6
2015-04-06	219911.95
2015-04-07	222146.2
2015-04-08	224440.15
2015-04-09	226487.45
2015-04-10	228912.4
2015-04-11	231456.15
2015-04-12	233450.45
2015-04-13	235946.65
2015-04-14	238452.35
2015-04-15	241031.2
2015-04-16	243049.15
2015-04-17	245724.8
2015-04-18	248011
2015-04-19	249538.95
2015-04-20	251998.4
2015-04-21	254211.55
2015-04-22	256405
2015-04-23	258831.15
2015-04-24	261810.35
2015-04-25	263899.55
2015-04-26	265666.95
2015-04-27	267847.75
2015-04-28	269590.55
2015-04-29	271419.3

2015-04-30	274086.8
2015-05-01	276658.75
2015-05-02	279058.95
2015-05-03	280891.2
2015-05-04	283180.1
2015-05-05	284893.7
2015-05-06	287203.5
2015-05-07	289432.35
2015-05-08	292484.65
2015-05-09	294853.05
2015-05-10	297141.4
2015-05-11	299529.45
2015-05-12	301829.15
2015-05-13	304090.95
2015-05-14	306785.45
2015-05-15	310171.6
2015-05-16	312452.7
2015-05-17	314281.1
2015-05-18	316490.75
2015-05-19	318477.75
2015-05-20	320850.75
2015-05-21	322913.3
2015-05-22	325548.4
2015-05-23	327992.55
2015-05-24	330189.5
2015-05-25	332293.9
2015-05-26	334170.35
2015-05-27	336267.35
2015-05-28	338283.75
2015-05-29	341284.95
2015-05-30	343771.9
2015-05-31	345489.55
2015-06-01	348557.3
2015-06-02	351007.25
2015-06-03	352914.3
2015-06-04	355197.9
2015-06-05	357898.05
2015-06-06	360179
2015-06-07	362139.75
2015-06-08	364404.7

2015-06-09	366847.25
2015-06-10	368866.65
2015-06-11	371517.15
2015-06-12	373655.75
2015-06-13	376164.65
2015-06-14	378023.65
2015-06-15	380619.25
2015-06-16	382517.55
2015-06-17	384654.65
2015-06-18	386639.15
2015-06-19	389432.6
2015-06-20	391493.2
2015-06-21	393418.4
2015-06-22	395737.7
2015-06-23	397780.45
2015-06-24	400107.95
2015-06-25	402507.1
2015-06-26	405252.6
2015-06-27	408065.5
2015-06-28	409635.2
2015-06-29	411508.8
2015-06-30	413719.75
2015-07-01	415951.25
2015-07-02	418246.05
2015-07-03	421689.05
2015-07-04	425553.25
2015-07-05	427144.7
2015-07-06	429261.6
2015-07-07	431636
2015-07-08	434032.05
2015-07-09	436329.8
2015-07-10	438762.2
2015-07-11	440847.75
2015-07-12	443033.4
2015-07-13	445092.9
2015-07-14	447049.4
2015-07-15	449551.2
2015-07-16	452015.1
2015-07-17	455146.75
2015-07-18	457268.95

2015-07-19	459291.65
2015-07-20	461792.65
2015-07-21	463823.5
2015-07-22	466115.6
2015-07-23	468330.1
2015-07-24	471534.5
2015-07-25	473771.75
2015-07-26	475643.3
2015-07-27	477815.3
2015-07-28	479909.85
2015-07-29	481833.1
2015-07-30	484182.25
2015-07-31	486277.65
2015-08-01	488718.2
2015-08-02	490628.35
2015-08-03	492610.6
2015-08-04	494700.75
2015-08-05	496795.6
2015-08-06	498894.85
2015-08-07	501521.25
2015-08-08	504237.65
2015-08-09	506240.3
2015-08-10	508379.75
2015-08-11	510669.75
2015-08-12	513035.5
2015-08-13	515109.65
2015-08-14	518126.25
2015-08-15	520378.6
2015-08-16	522517.9
2015-08-17	525143.9
2015-08-18	527245.1
2015-08-19	529578.05
2015-08-20	531485.75
2015-08-21	534087.15
2015-08-22	536493.15
2015-08-23	538194.75
2015-08-24	539891.8
2015-08-25	541850.7
2015-08-26	544180.1
2015-08-27	546297.75

2015-08-28	548944.95
2015-08-29	550979.95
2015-08-30	552474.55
2015-08-31	554555.9
2015-09-01	556908.75
2015-09-02	558774.3
2015-09-03	561026.9
2015-09-04	563987.85
2015-09-05	566525.65
2015-09-06	568017.3
2015-09-07	570300.65
2015-09-08	572550.15
2015-09-09	575130.25
2015-09-10	577546.1
2015-09-11	580308
2015-09-12	582896.15
2015-09-13	584734.3
2015-09-14	586899.55
2015-09-15	589449.75
2015-09-16	591635
2015-09-17	593877.05
2015-09-18	596598.6
2015-09-19	598885.15
2015-09-20	600714.15
2015-09-21	602845.6
2015-09-22	605016
2015-09-23	607179
2015-09-26	609425.85
2015-09-27	611740.55
2015-09-28	613775.85
2015-09-29	616537.9
2015-09-30	618735.95
2015-10-01	621938.1
2015-10-02	624012.95
2015-10-03	626413.9
2015-10-04	628556.1
2015-10-06	630772.05
2015-10-07	632864.4
2015-10-08	634840.25
2015-10-09	637352.85

2015-10-10	639663.05
2015-10-11	641579.3
2015-10-13	643905.25
2015-10-14	646051.6
2015-10-15	650371.8
2015-10-16	652926.9
2015-10-17	655266.7
2015-10-18	657062
2015-10-20	659499.15
2015-10-21	661959.65
2015-10-22	664360.55
2015-10-23	666971.2
2015-10-24	669650.7
2015-10-25	671487.75
2015-10-27	673476.4
2015-10-28	675112.35
2015-10-29	677282.1
2015-10-30	680018.7
2015-10-31	682763.55
2015-11-01	684750.2
2015-11-02	687049.3
2015-11-03	688877.85
2015-11-04	690843.95
2015-11-05	693024.3
2015-11-06	696181.8
2015-11-07	698762.75
2015-11-08	700872.9
2015-11-09	703356.65
2015-11-10	705399.25
2015-11-11	707345.45
2015-11-12	709910.25
2015-11-13	712174.8
2015-11-14	714499.55
2015-11-15	716321.2
2015-11-16	718577.85
2015-11-17	720534.4
2015-11-18	722646.1
2015-11-19	725341
2015-11-20	727729.1
2015-11-21	729813.05

2015-11-22	731181.75
2015-11-23	733646.9
2015-11-24	735876.95
2015-11-25	738240.2
2015-11-26	742646.15
2015-11-27	747068.6
2015-11-28	749036.65
2015-11-29	750935.65
2015-11-30	753158.9
2015-12-01	755235.6
2015-12-02	757449.7
2015-12-03	759692.9
2015-12-04	762571.25
2015-12-05	765199.2
2015-12-06	767549.45
2015-12-07	769964.25
2015-12-08	771820.5
2015-12-09	774392.05
2015-12-10	776377.65
2015-12-11	779011.65
2015-12-12	780971.8
2015-12-13	783216.95
2015-12-14	785389.55
2015-12-15	787777
2015-12-16	790011.8
2015-12-17	791892.55
2015-12-18	794778.85
2015-12-19	797083.05
2015-12-20	799187.95
2015-12-21	801288.65
2015-12-22	803171.6
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05