DATA ANALYSIS

PIZZA-SALES

By :-Kshirod Kalet

SQL QUERIES

TYPES

BASICS

INTERMEDIATE

ADVANCED

File Link : https://github.com/Ayushi0214/pizza-sales---SQL

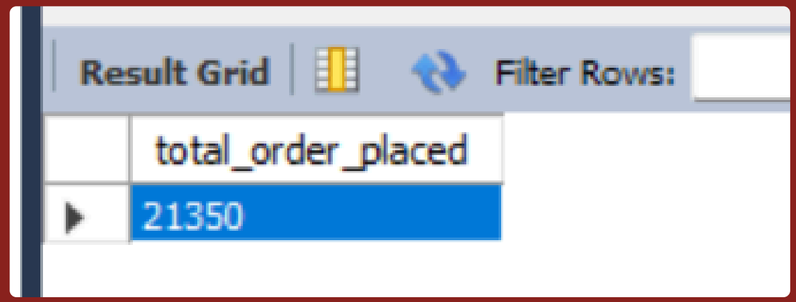
Basics

1.Retrieve the total number of orders placed

code :

```sql
-- Q1 Retrieve the total number of orders placed.
select count(order_id) as total_order_placed
from orders;
```

output :

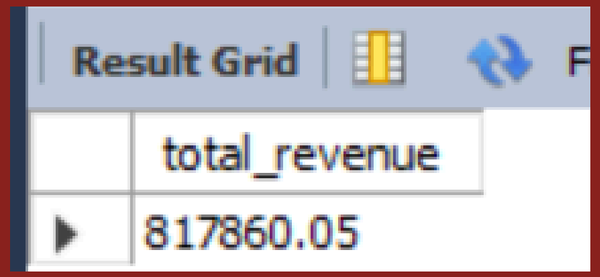


2.Calculate the total revenue generated from pizza sales

code :

```
1   -- Q2 Calculate the total revenue generated from pizza sales.
2   select  round(sum(a.price*d.quantity),2) total_revenue
3   from pizzas as a
4   join order_details d
5   on a.pizza_id=d.pizza_id
```

output :

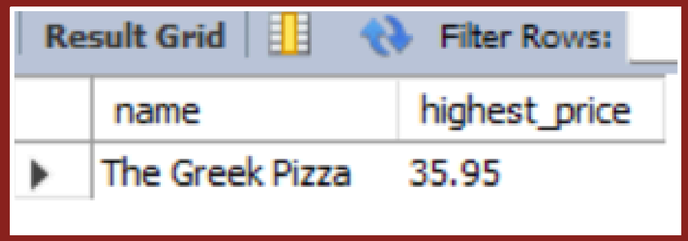


3.Identify the highest-priced pizza

code :

```sql
-- Q3 Identify the highest-priced pizza.
select c.name,a.price as highest_price
from pizzas as a
join pizza_types as c
on a.pizza_type_id=c.pizza_type_id
order by a.price desc
limit 1;
```

output :



Result Grid | Filter Rows:

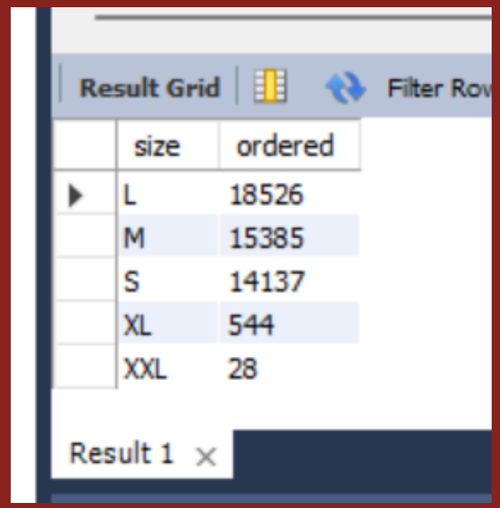
name	highest_price
The Greek Pizza	35.95

4.Identify the most common pizza size ordered.

code :

```sql
-- Q4 Identify the most common pizza size ordered.
select a.size,count(d.order_details_id)as ordered
from order_details as d
join pizzas as a
on a.pizza_id=d.pizza_id
group by a.size
order by ordered desc
```

output :



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

code :

```sql
-- Q5 List the top 5 most ordered
-- pizza types along with their quantities.
SELECT c.name,sum(d.quantity) top_ordered
FROM order_details as d
JOIN pizzas as a
ON a.pizza_id=d.pizza_id
join pizza_types as c
on c.pizza_type_id=a.pizza_type_id
group by c.name
order by sum(d.quantity) desc
limit 5
```

output :

Result Grid | Filter Rows:

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

Intermediate

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

code :

```
1   -- Q6 Join the necessary tables to find the
2   -- total quantity of each pizza category ordered.
3 • SELECT c.category,sum(d.quantity) ordered
4   FROM order_details as d
5   JOIN pizzas as a
6   ON a.pizza_id=d.pizza_id
7   join pizza_types as c
8   on c.pizza_type_id=a.pizza_type_id
9   group by c.category
```

output :

Result Grid | Filter

	category	ordered
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

2.Determine the distribution of orders by hour of the day

code :

```sql
-- Q7 Determine the distribution of orders by hour of the day.--
Select hour(b.order_time) as Hours ,count(b.order_id) as orders
from orders as b
group by hour(b.order_time)
order by hour(b.order_time) asc ;
```

output :

Result Grid

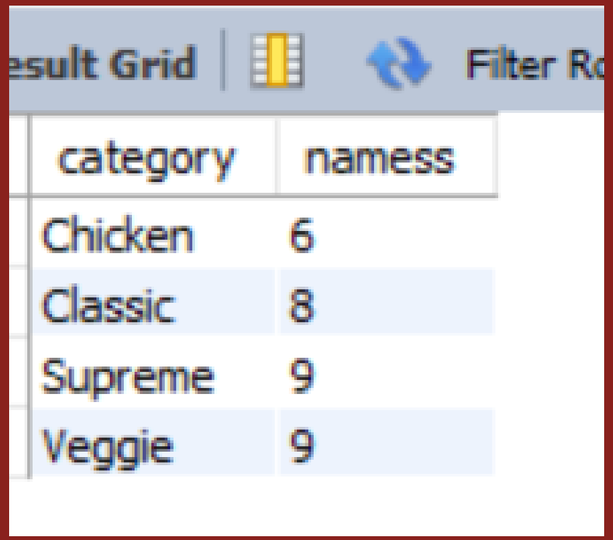
Hours	orders
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28

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

code :

```sql
-- Q 8 Join relevant tables to find the
-- category-wise distribution of pizzas.
select c.category,count(c.name) as namess
from pizza_types AS c
group by c.category ;
```

output :



category	namess
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

code :

```sql
-- Q9 Group the orders by date and
-- calculate the average number of pizzas ordered per day
SELECT
    ROUND(AVG(order_quantity.total_quantity), 2) AS average_pizzas_per_day
FROM
    (SELECT
        b.order_date, SUM(d.quantity) AS total_quantity
    FROM
        orders AS b
    JOIN order_details AS d ON b.order_id = d.order_id
    GROUP BY b.order_date) AS order_quantity;
```

output :

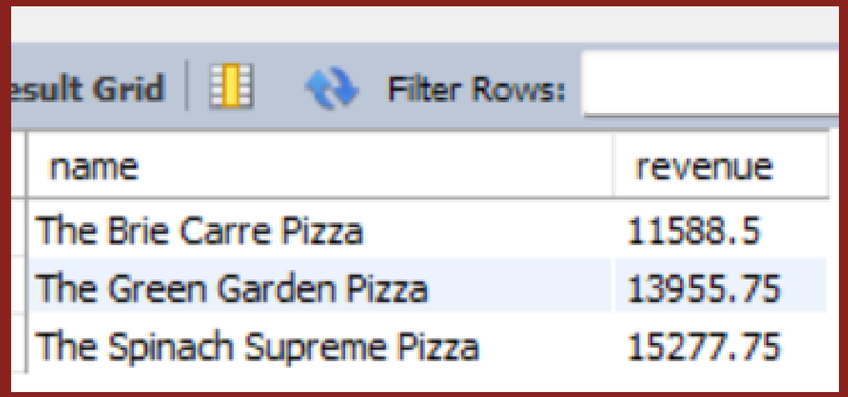
average_pizzas_per_day
138.47

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

code :

```sql
--  Q10 Determine the top 3 most ordered pizza types
-- based on revenue.
SELECT c.name,round(sum(d.quantity*a.price),2) revenue
FROM order_details as d
JOIN pizzas as a
ON a.pizza_id=d.pizza_id
join pizza_types as c
on c.pizza_type_id=a.pizza_type_id
group by c.name
order by revenue
limit 3 ;
```

output :



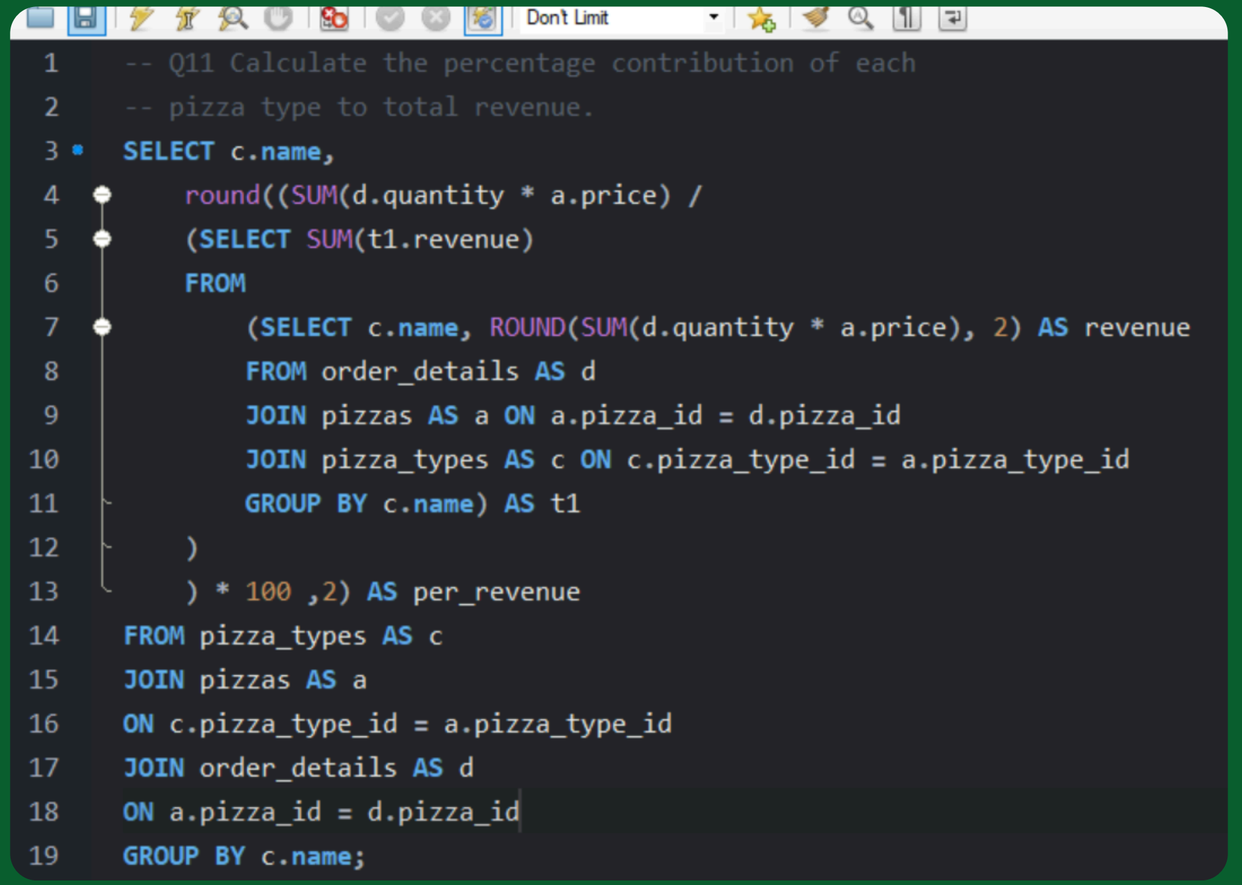
Filter Rows:

name	revenue
The Brie Carre Pizza	11588.5
The Green Garden Pizza	13955.75
The Spinach Supreme Pizza	15277.75

ADVANCED

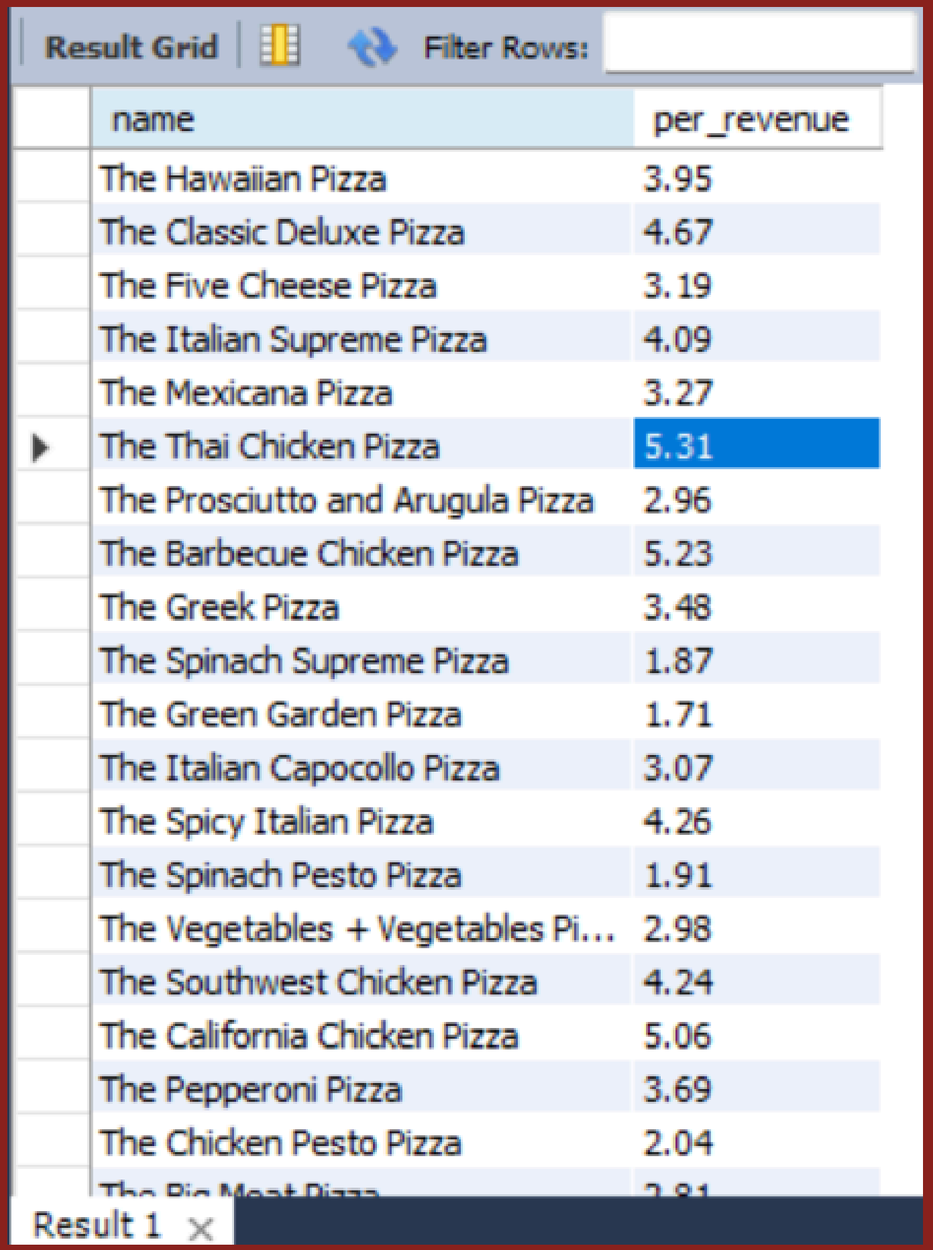
1.Calculate the percentage contribution of each pizza type to total revenue

code :



```sql
-- Q11 Calculate the percentage contribution of each
-- pizza type to total revenue.
SELECT c.name,
    round((SUM(d.quantity * a.price) /
    (SELECT SUM(t1.revenue)
    FROM
        (SELECT c.name, ROUND(SUM(d.quantity * a.price), 2) AS revenue
        FROM order_details AS d
        JOIN pizzas AS a ON a.pizza_id = d.pizza_id
        JOIN pizza_types AS c ON c.pizza_type_id = a.pizza_type_id
        GROUP BY c.name) AS t1
    )
    ) * 100 ,2) AS per_revenue
FROM pizza_types AS c
JOIN pizzas AS a
ON c.pizza_type_id = a.pizza_type_id
JOIN order_details AS d
ON a.pizza_id = d.pizza_id
GROUP BY c.name;
```

: OUTPUT



Result Grid | Filter Rows:

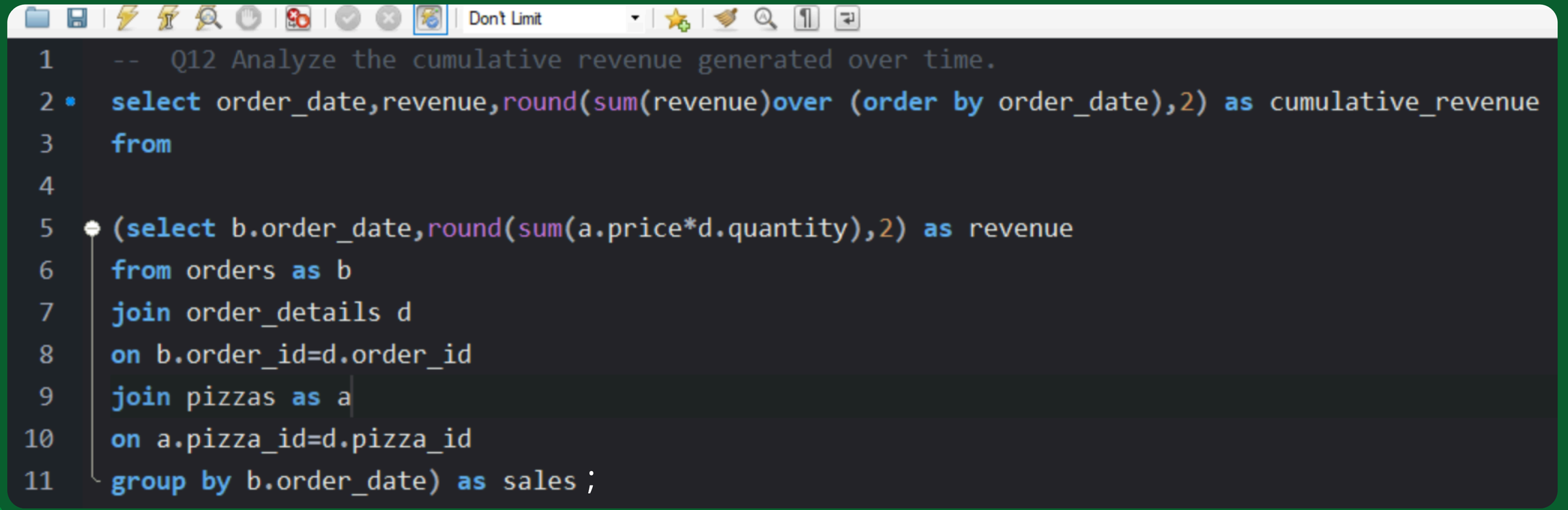
name	per_revenue
The Hawaiian Pizza	3.95
The Classic Deluxe Pizza	4.67
The Five Cheese Pizza	3.19
The Italian Supreme Pizza	4.09
The Mexicana Pizza	3.27
The Thai Chicken Pizza	5.31
The Prosciutto and Arugula Pizza	2.96
The Barbecue Chicken Pizza	5.23
The Greek Pizza	3.48
The Spinach Supreme Pizza	1.87
The Green Garden Pizza	1.71
The Italian Capocollo Pizza	3.07
The Spicy Italian Pizza	4.26
The Spinach Pesto Pizza	1.91
The Vegetables + Vegetables Pi...	2.98
The Southwest Chicken Pizza	4.24
The California Chicken Pizza	5.06
The Pepperoni Pizza	3.69
The Chicken Pesto Pizza	2.04
The Big Meat Pizza	2.81

Result 1

The Big Meat Pizza	2.81
The Soppressata Pizza	2.01
The Four Cheese Pizza	3.95
The Napolitana Pizza	2.95
The Calabrese Pizza	1.95
The Italian Vegetables Pizza	1.96
The Mediterranean Pizza	1.88
The Pepper Salami Pizza	3.12
The Spinach and Feta Pizza	2.85
The Sicilian Pizza	3.78
The Chicken Alfredo Pizza	2.07
The Pepperoni, Mushroom, and...	2.3
The Brie Carre Pizza	1.42

2.Analyze the cumulative revenue generated over time

code :



Result Grid | Filter Rows: | Export:

order_date	revenue	cumulative_revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2	16560.7
2015-01-08	2838.35	19399.05
2015-01-09	2127.35	21526.4
2015-01-10	2463.95	23990.35
2015-01-11	1872.3	25862.65
2015-01-12	1919.05	27781.7
2015-01-13	2049.6	29831.3
2015-01-14	2527.4	32358.7
2015-01-15	1984.8	34343.5
2015-01-16	2594.15	36937.65
2015-01-17	2064.1	39001.75
2015-01-18	1976.85	40978.6
2015-01-19	2387.15	43365.75
2015-01-20	2397.9	45763.65

Result 1

: OUTPUT

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

code :

```sql
-- Q 13 Determine the (top 3 most ordered ) (pizza types)
-- (based on revenue) for each (pizza category
select category,name,round(revenue,2),rn
from(

select category,name,revenue,
rank()over(partition by category order by revenue desc) as rn
from(

select c.category,c.name,sum(d.quantity*a.price) revenue
from pizza_types as c
join pizzas as a
on a.pizza_type_id=c.pizza_type_id
join order_details d
on d.pizza_id=a.pizza_id
group by c.category,c.name ) as t1    ) t2
                ;
where rn<=3;
```

Result Grid | Filter Rows: | Export: | Wrap Cel

category	name	round(revenue,2)	rn
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.7	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

: OUTPUT

Thank You