SQL PROJECT

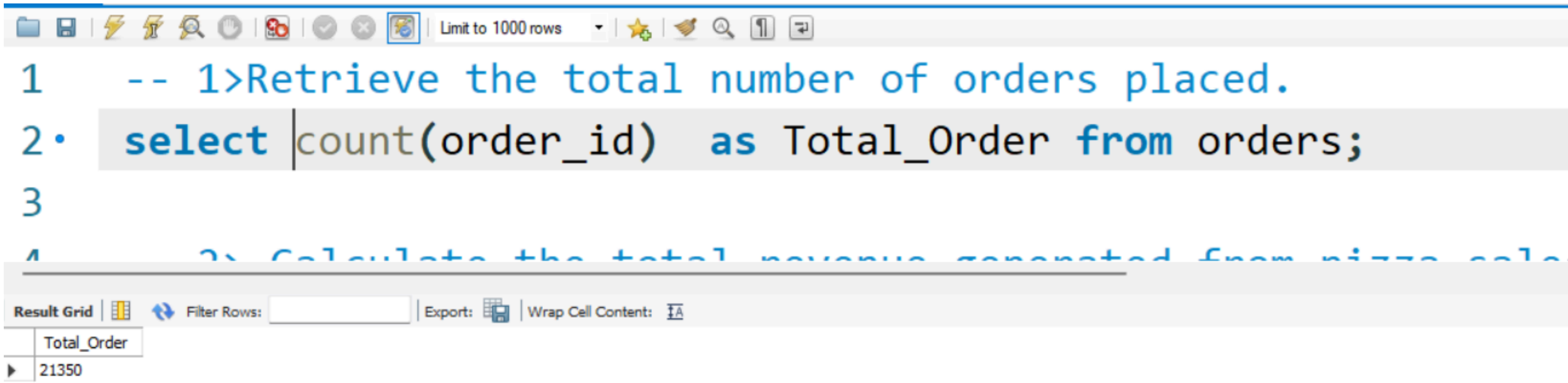
PIZZA SALES

Its about pizza sales and done queries using SQL

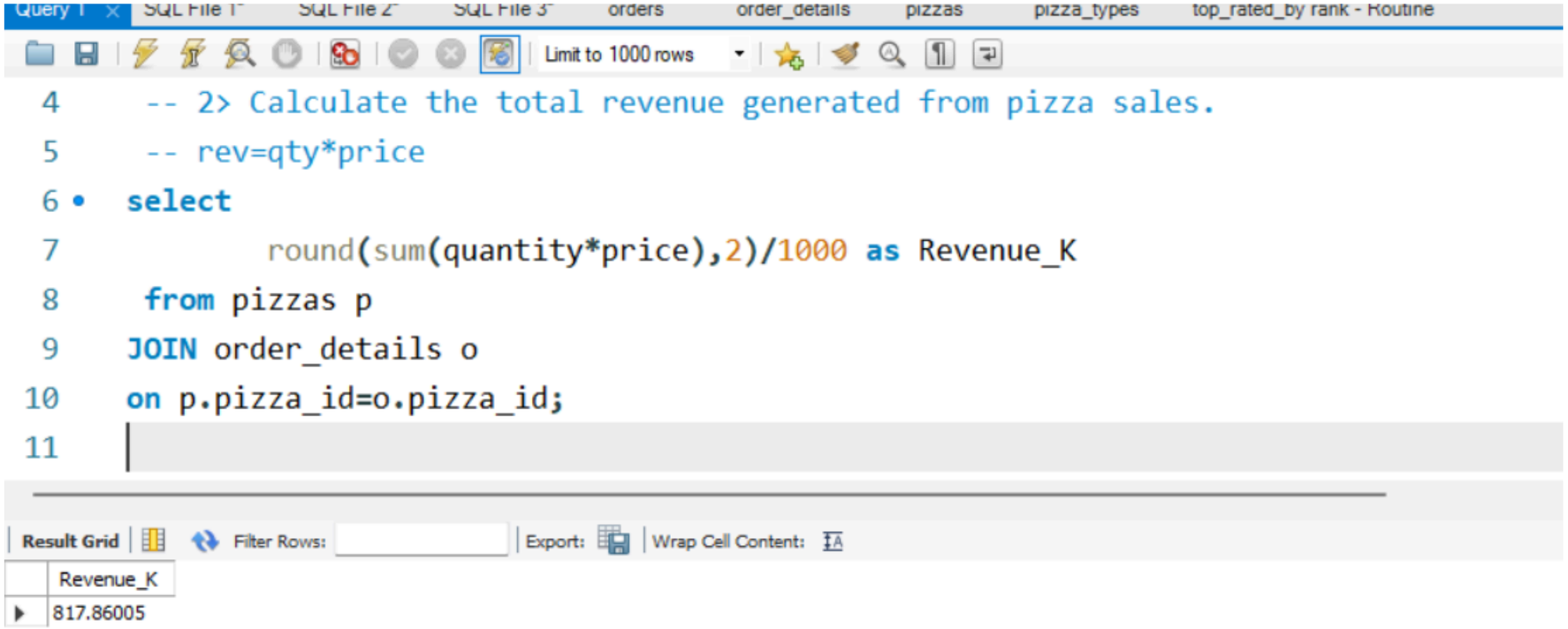
SRIHARI S

SQL PROJECT

HI
THIS IS SRIHARI , I DID PROJECT IN SQL USING DATA OF PIZZA HUT AND SOLVED ,ALL QUERIES GIVEN BY THE STACK HOLDER AND PRESENTED MY WORK IN VERY NEXT SLIDE
THANK YOU


Limit to 1000 rows
1 -- 1>Retrieve the total number of orders placed.
2 select count(order_id) as Total_Order from orders;
3
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Total_Order
21350

Problem_1


Query 1
SQL File 1*
SQL File 2*
SQL File 3*
orders
order_details
pizzas
pizza_types
top_rated_by rank - Routine
Limit to 1000 rows
-- 2> Calculate the total revenue generated from pizza sales.
-- rev=qty*price
select
round(sum(quantity*price),2)/1000 as Revenue_K
from pizzas p
JOIN order_details o
on p.pizza_id=o.pizza_id;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Revenue_K
817.86005

problem_2

```sql
-- 3>Identify the highest-priced pizza.

select p1.name,price from pizzas p join pizza_types p1
on p.pizza_type_id=p1.pizza_type_id
order by price desc
limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

problem_3

Query 1 | SQL File 1* | SQL File 2* | SQL File 3* | orders | order_details | pizzas | pizza_types | top_rated_by rank - Routine

Limit to 1000 rows

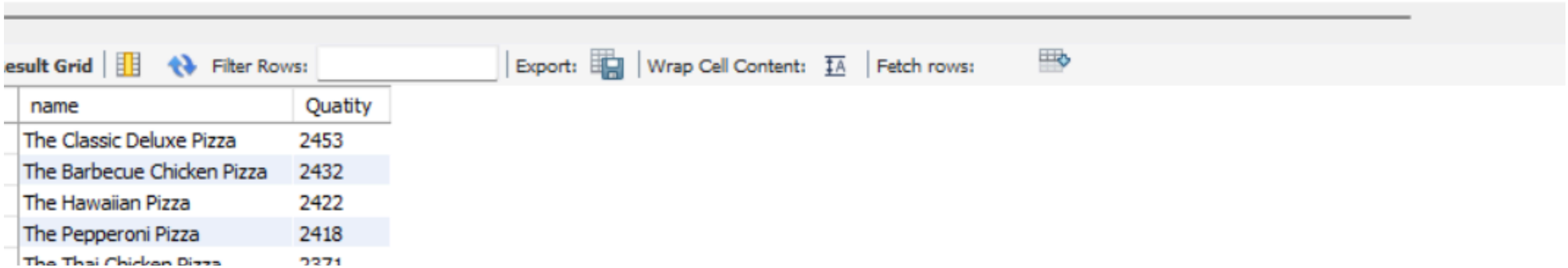
```
22  -- 4> Identify the most common pizza size ordered.
23
24  select p.size,count(o.order_details_id) as Count_Pizza from pizzas p join order_details o
25  on p.pizza_id=o.pizza_id
26  group by p.size
27  order by Count_Pizza desc;
28
29
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

problme_4

```
31    -- List the top 5 most ordered pizza types along with their quantities.
32
33 •  select p.name,sum(o.quantity) as Quatity from pizza_types p join pizzas p1
34    on p.pizza_type_id=p1.pizza_type_id
35    join order_details o
36    on o.pizza_id=p1.pizza_id
37    group by p.name
38    order by Quatity desc
39    limit 5;
```



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

problem_5

Query 1 | SQL File 1 | SQL File 2 | SQL File 3 | orders | order_details | pizzas | pizza_types | top_rated_by rank - Routine

Limit to 1000 rows

```sql
-- 1>Join the necessary tables to find the total Revenue of each pizza category ordered.

SELECT (p_h.name),
    sum(o.quantity*p.price) as Total_Revenue
    FROM pizza_hut.pizza_types p_h
    JOIN pizzas as p
    on p_h.pizza_type_id = p.pizza_type_id
    join order_details o
    on o.pizza_id=p.pizza_id
    group by p_h.name
order by Total_Revenue desc limit 3;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

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

Result 59

problem_6

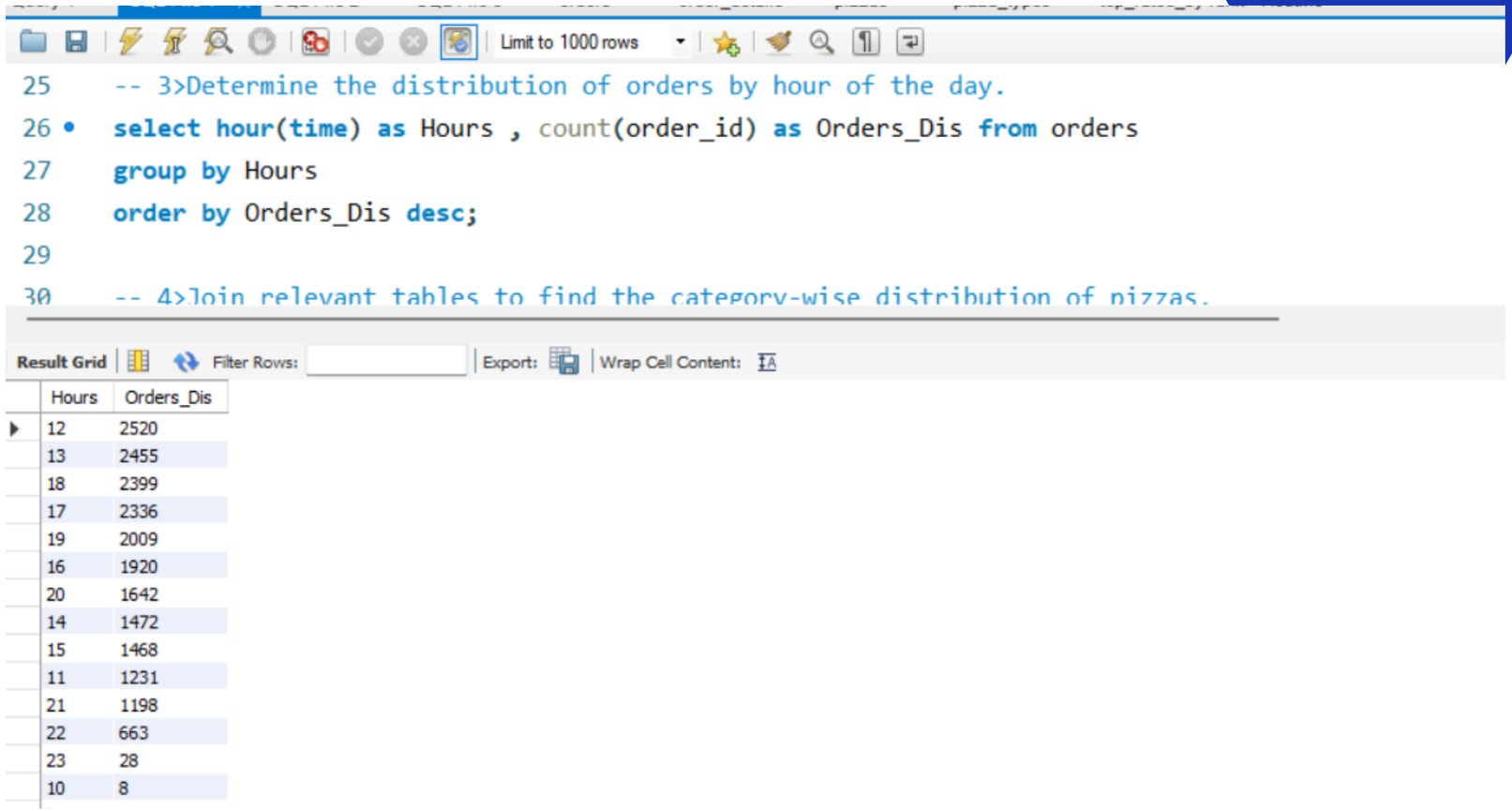
```sql
-- 2>Join the necessary tables to find the total quantity of each pizza category ordered.

SELECT (p_h.category),
     round(sum(o.quantity)/1000,2) as Total_Quauntity_K
    FROM pizza_hut.pizza_types p_h
    JOIN pizzas as p
    using(pizza_type_id)
    JOIN order_details o
    on o.pizza_id =p.pizza_id
group by p_h.category
order by Total_Quauntity_K desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	Total_Quauntity_K
Classic	14.89
Supreme	11.99
Veggie	11.65
Chicken	11.05

problme_7



```
Limit to 1000 rows

25    -- 3>Determine the distribution of orders by hour of the day.
26 •  select hour(time) as Hours , count(order_id) as Orders_Dis from orders
27    group by Hours
28    order by Orders_Dis desc;
29
30    -- 4>Join relevant tables to find the category-wise distribution of pizzas.
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Hours	Orders_Dis
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
10	8

problme_8

```
29
30    -- 4>Join relevant tables to find the category-wise distribution of pizzas.
31
32 •  select category,count(name) Name from pizza_types
33    group by category
34    order by Name desc;
35
36
37
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	Name
Supreme	9
Veggie	9
Classic	8
Chicken	6

problme_9

Query 1 | SQL File 1* | SQL File 2* | SQL File 3* | orders | order_details | pizzas | pizza_types | top_rated_by rank - Routine

Limit to 1000 rows

```
37
38    -- 5>Group the orders by date and calculate the average number of pizzas ordered per day with name.
39
40 •  select o.date,p_t.name,sum(o_d.quantity) as Quantity from order_details o_d
41    join orders o on
42    o_d.order_id=o.order_id
43    join pizzas p on p.pizza_id=o_d.pizza_id
44    join pizza_types p_t
45    on p_t.pizza_type_id=p.pizza_type_id
46    group by p_t.name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

date	name	Quantity
2015-01-01	The Hawaiian Pizza	2422
2015-01-01	The Thai Chicken Pizza	2371
2015-01-01	The Mexicana Pizza	1484
2015-01-01	The Italian Supreme Pizza	1884
2015-01-01	The Five Cheese Pizza	1409
2015-01-01	The Classic Deluxe Pizza	2453
2015-01-01	The Prosciutto and Arugula Pizza	1457
2015-01-01	The Greek Pizza	1420
2015-01-01	The Barbecue Chicken Pizza	2432

Result 63

problme_10

```sql
47
48    -- 6>Group the orders by date and calculate the average number of pizzas ordered per day.
49  • select avg(Quantity) as Average_Qty from(
50    select o.date,sum(o_.quantity) Quantity from orders o join order_details o_
51    on o.order_id=o_.order_id
52    group by o.date) as Quantity;
53
54
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Average_Qty
138.4749

problem_11

Query 1 | SQL File 1* | SQL File 2* | SQL File 3* | orders | order_details | pizzas | pizza_types | top_rated_by rank - Routine

Limit to 1000 rows

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.
select
    sub.name,
    sub.Revenue,
    round((sub.Revenue/Total_Revenue.TotalRevenue)*100,2) as Total_Revenue ,
    dense_rank() over( order by (sub.Revenue/Total_Revenue.TotalRevenue)*100  desc) as Rank_
    from(
select
    pt.name,
    round(sum(od.quantity*p.price),2) as Revenue
    from pizza_types pt
    join pizzas p
        on pt.pizza_type_id=p.pizza_type_id
    join order_details od
        on od.pizza_id=p.pizza_id
    group by pt.name)as sub
    join(
        select
        sum(od.quantity*p.price) as TotalRevenue
            from order_details od
            join pizzas p
            on od.pizza_id=p.pizza_id) as Total_Revenue
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	Revenue	Total_Revenue	Rank_
The Thai Chicken Pizza	43434.25	5.31	1
The Barbecue Chicken Pizza	42768	5.23	2
The California Chicken Pizza	41409.5	5.06	3
The Classic Deluxe Pizza	38180.5	4.67	4
The Spicy Italian Pizza	34831.25	4.26	5

Result 27

problme_12

Query 1 | SQL File 1* | SQL File 2* | SQL File 3* | orders | order_details | pizzas | pizza_types | top_rated_by rank - Routine

Limit to 1000 rows

```sql
-- Calculate the percentage contribution of each pizza type to total revenue and rank them, then filter to show top 2.
WITH ranked_pizzas AS (
    SELECT
        sub.name,sub.Revenue,
        ROUND((sub.Revenue / Total_Revenue.TotalRevenue) * 100, 2) AS Total_Revenue_Percentage,
        DENSE_RANK() OVER (ORDER BY (sub.Revenue / Total_Revenue.TotalRevenue) DESC) AS Rank_
    FROM (
        SELECT pt.name,
            ROUND(SUM(od.quantity * p.price), 2) AS Revenue
        FROM pizza_types pt
        JOIN
            pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN order_details od ON od.pizza_id = p.pizza_id
        GROUP BY pt.name
    ) AS sub
    JOIN (
        SELECT SUM(od.quantity * p.price) AS TotalRevenue
        FROM order_details od
        JOIN pizzas p ON od.pizza_id = p.pizza_id
    ) AS Total_Revenue
)
SELECT name,Revenue,Total_Revenue_Percentage,Rank_
FROM ranked_pizzas
WHERE Rank_ < 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	Revenue	Total_Revenue_Percentage	Rank_
The Thai Chicken Pizza	43434.25	5.31	1
The Barbecue Chicken Pizza	42768	5.23	2

problme_13


Thank
you