Restaurant Data Analysis

Project by Nandini Agarwal

SQL Case study

CONTENT

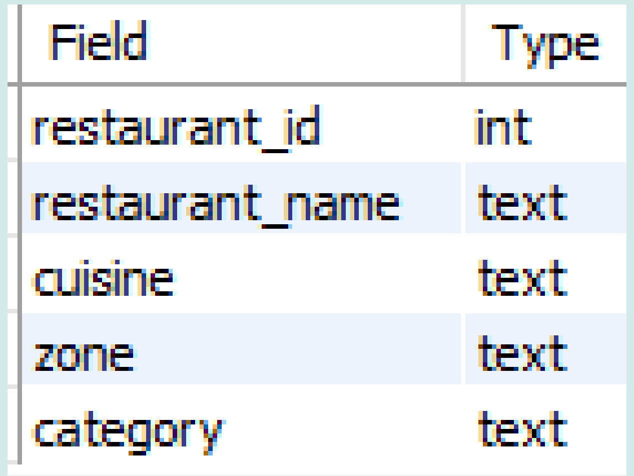
- DATASET
- OBJECTIVES
- INSIGHTS
- DATA ANALYSIS USING SQL

DATASET

The dataset contains two Tables.

The first table namely **restaurant** consists of restaurant details along with the zone and category they fall into (5 columns).

This Table consists of **20 different restaurants** which can be identified using the restaurant_id and **8 different cuisine.**

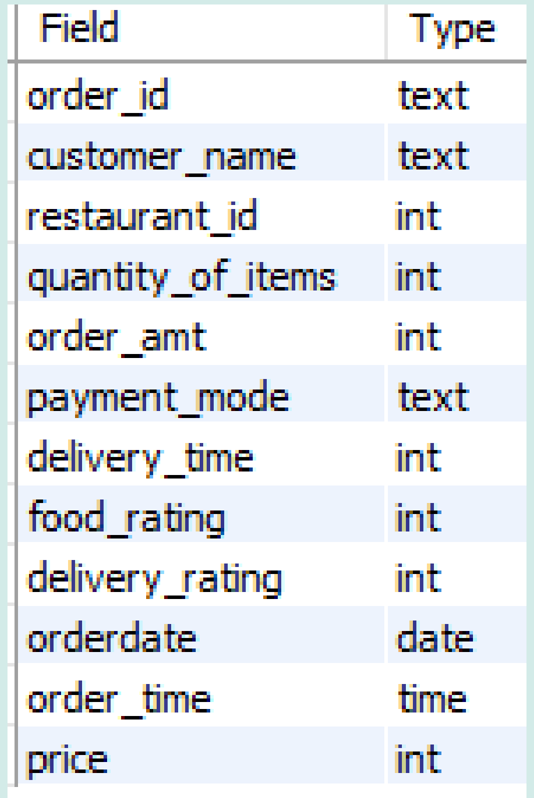


Field	Type
restaurant_id	int
restaurant_name	text
cuisine	text
zone	text
category	text

The second table namely **order_details** consists of all order details along with the delivery timings and customer

name (10 columns).

This Table consists of **500 distinct orders** which can be identified using the order_id.

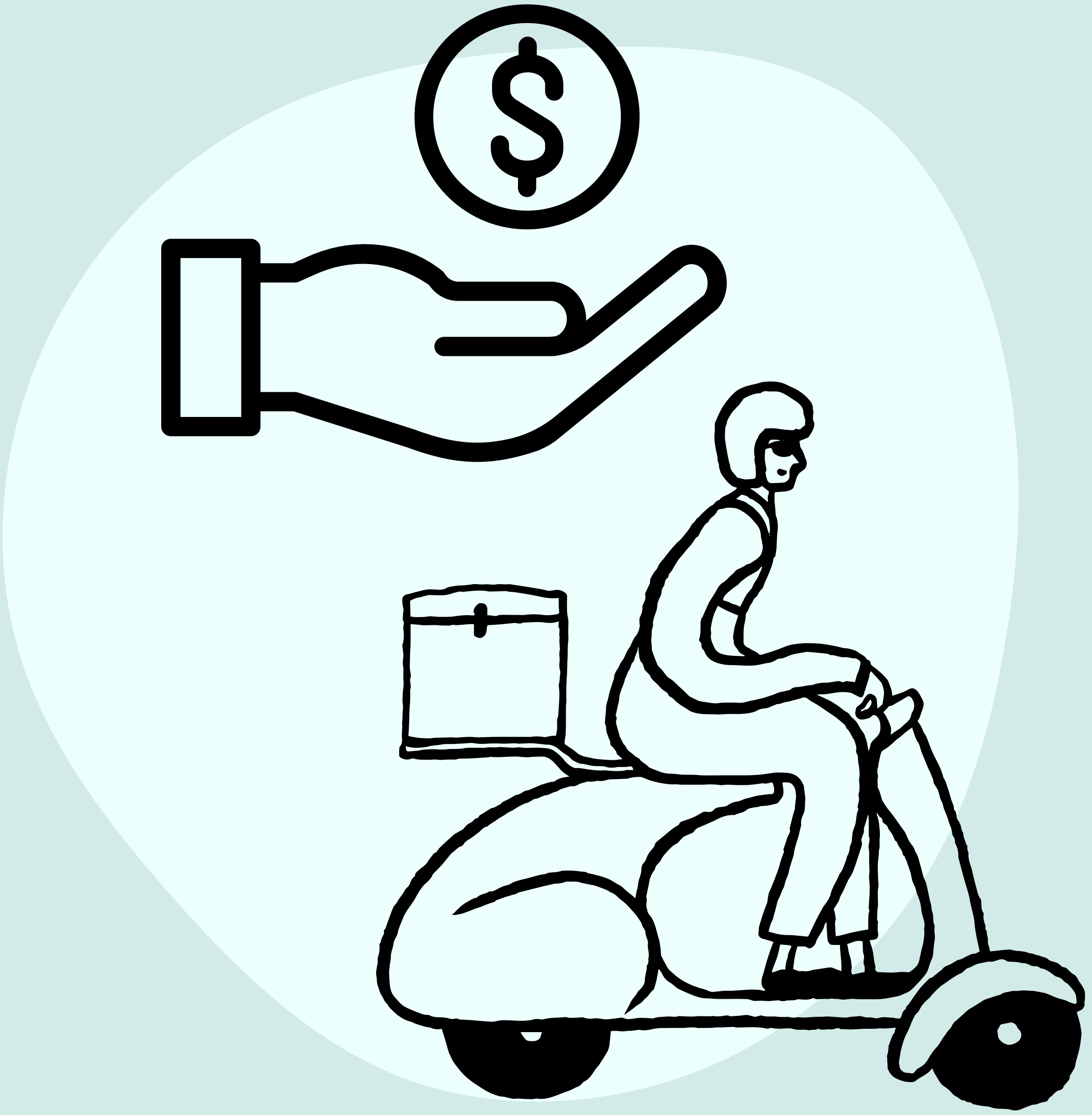


Field	Type
order_id	text
customer_name	text
restaurant_id	int
quantity_of_items	int
order_amt	int
payment_mode	text
delivery_time	int
food_rating	int
delivery_rating	int
orderdate	date
order_time	time
price	int

This is a Restaurant orders and sales case study. Here, we need to analyze the data and provide crucial insights.

Here are the topics analyzed in the case study:

- Category-wise average sales, orders, and rating.
- Sales, orders, and restaurants in each zone.
- Customer who ordered the most.
- Most ordered cuisine in each zone.
- Top 5 restaurants based on food and delivery rating.
- Bottom 5 restaurants based on food delivery rating.
- Number of orders received by zones during rush hours by each zone and category.



OBJECTIVES

01.

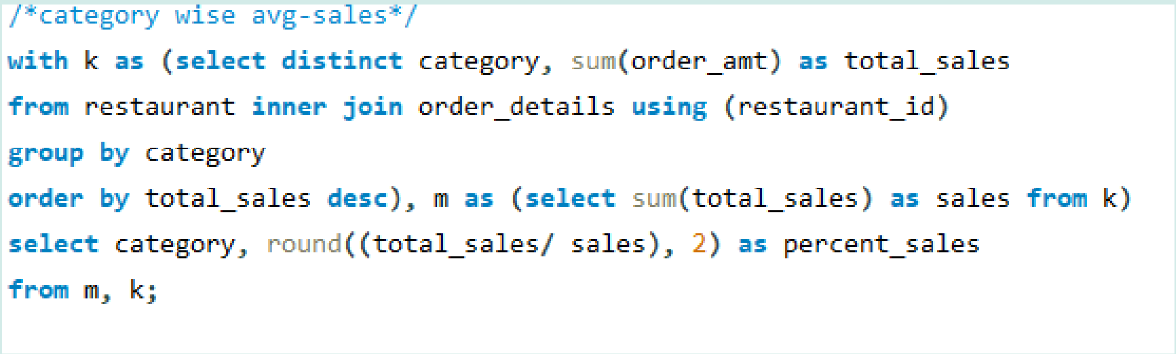
Analyze the orders and sales on the basis of zone, category and restaurant

02.

Publish the findings based on the analysis.

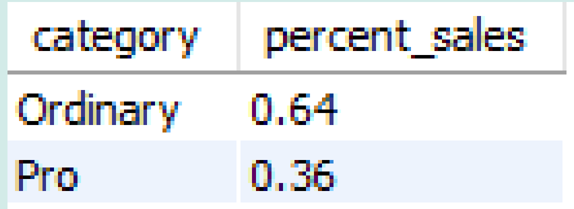
CATEGORY WISE AVERAGE SALES, RESTAURANTS, ORDERS AND RATING

SQL Query

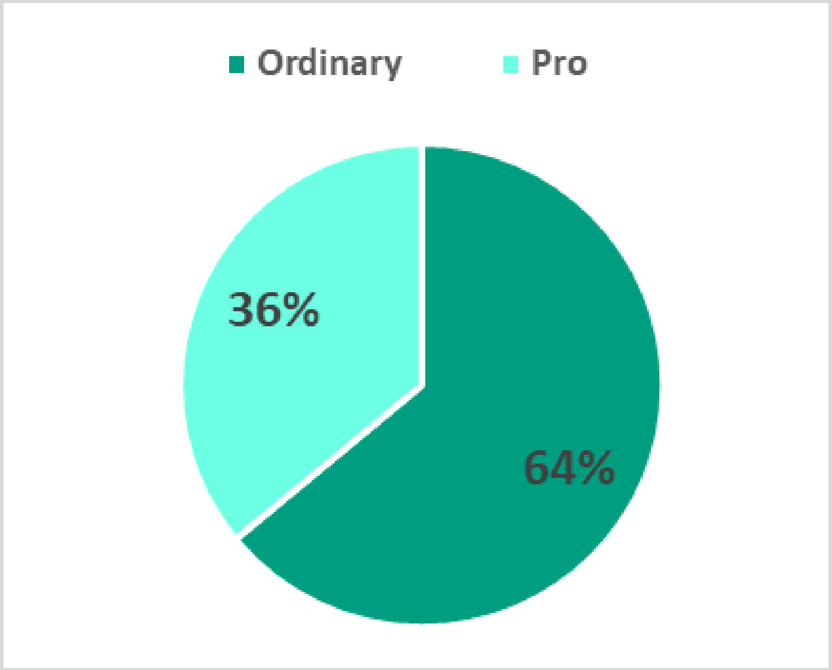


```
/*category wise avg-sales*/
with k as (select distinct category, sum(order_amt) as total_sales
from restaurant inner join order_details using (restaurant_id)
group by category
order by total_sales desc), m as (select sum(total_sales) as sales from k)
select category, round((total_sales/ sales), 2) as percent_sales
from m, k;
```

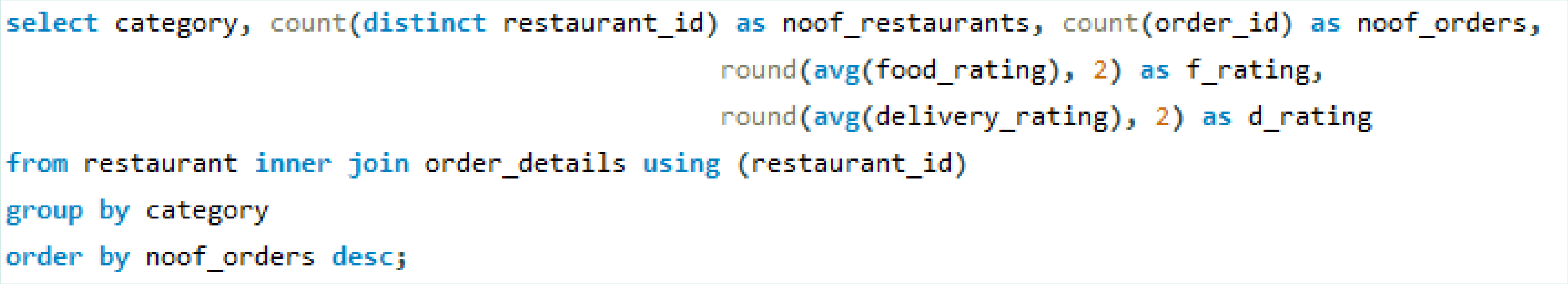
Output



category	percent_sales
Ordinary	0.64
Pro	0.36

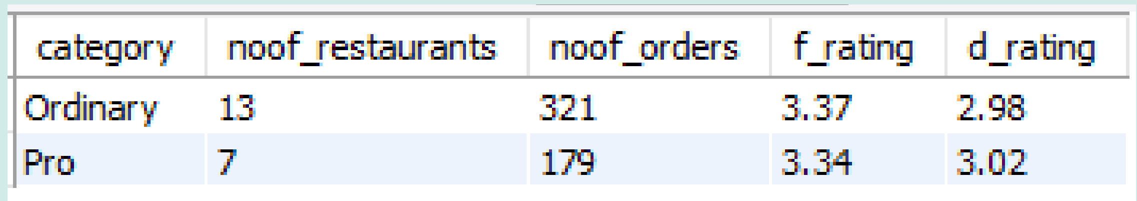


SQL Query



```
select category, count(distinct restaurant_id) as noof_restaurants, count(order_id) as noof_orders,
                                               round(avg(food_rating), 2) as f_rating,
                                               round(avg(delivery_rating), 2) as d_rating
from restaurant inner join order_details using (restaurant_id)
group by category
order by noof_orders desc;
```

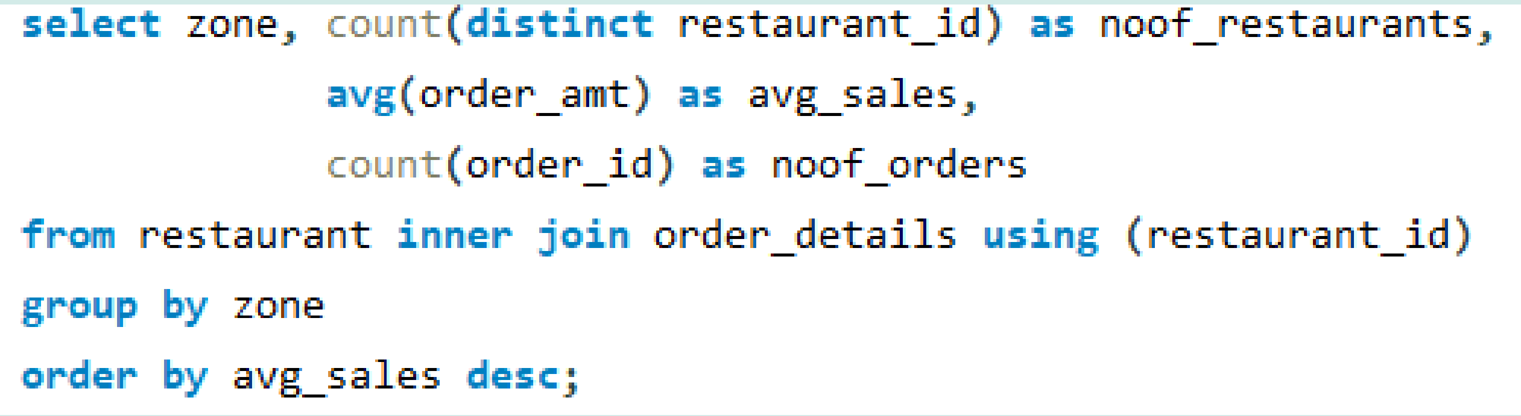
Output



category	noof_restaurants	noof_orders	f_rating	d_rating
Ordinary	13	321	3.37	2.98
Pro	7	179	3.34	3.02

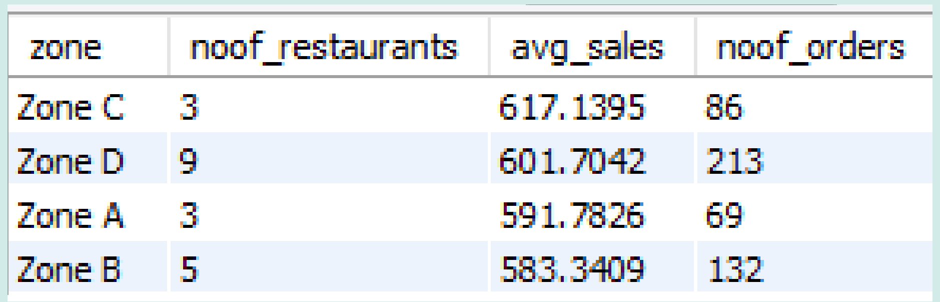
RESTAURANTS, SALES AND ORDERS IN EACH ZONE

SQL Query

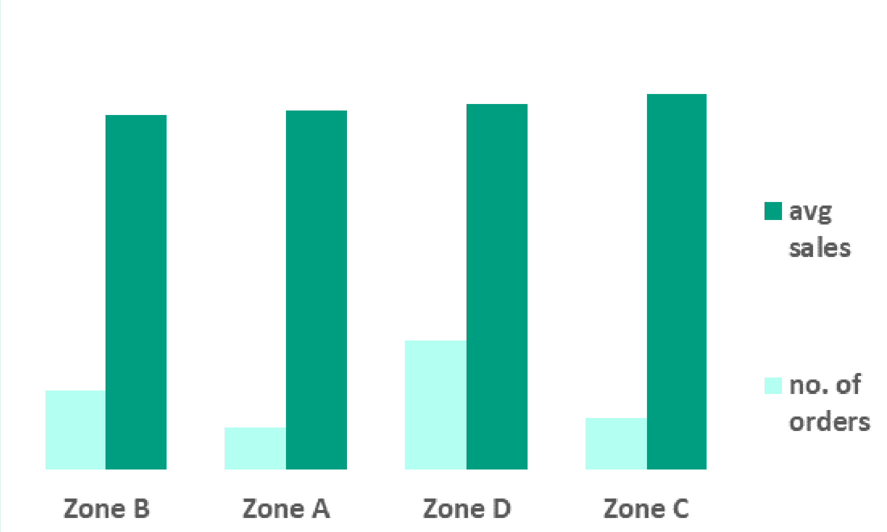


```sql
select zone, count(distinct restaurant_id) as noof_restaurants,
             avg(order_amt) as avg_sales,
             count(order_id) as noof_orders
from restaurant inner join order_details using (restaurant_id)
group by zone
order by avg_sales desc;
```

Output

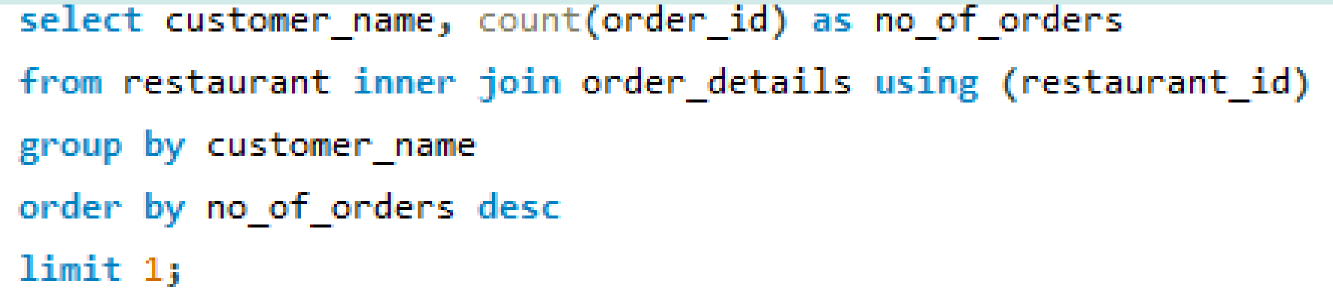


zone	noof_restaurants	avg_sales	noof_orders
Zone C	3	617.1395	86
Zone D	9	601.7042	213
Zone A	3	591.7826	69
Zone B	5	583.3409	132

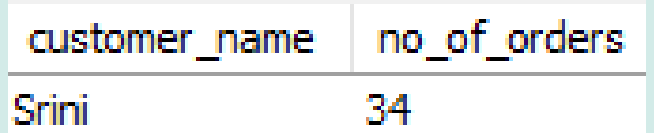


Orders and Sales in each zone.

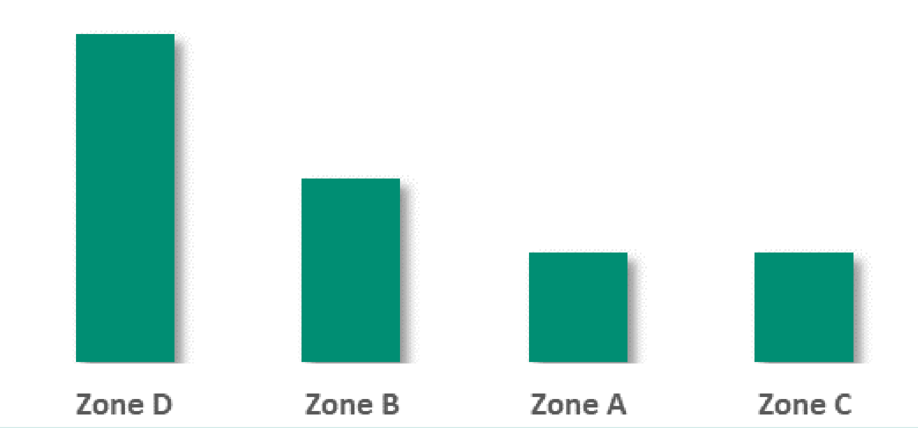
CUSTOMER WHO ORDERED THE MOST



```sql
select customer_name, count(order_id) as no_of_orders
from restaurant inner join order_details using (restaurant_id)
group by customer_name
order by no_of_orders desc
limit 1;
```



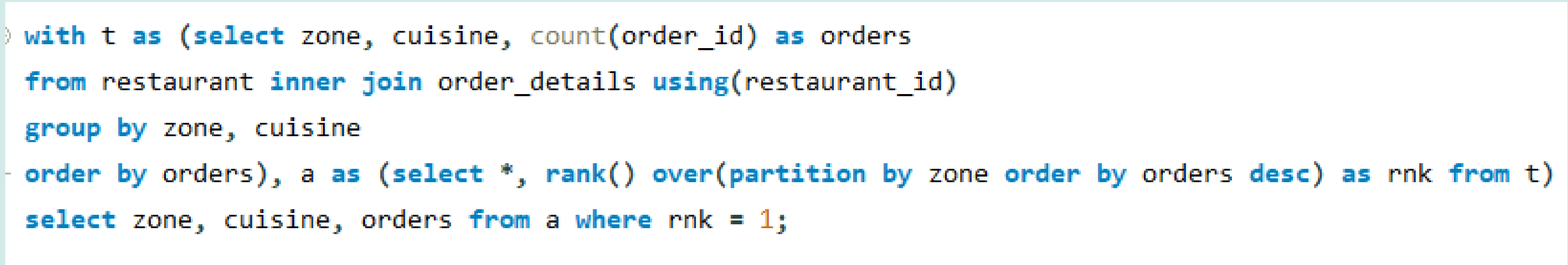
customer_name	no_of_orders
Srini	34



Number of Restaurants in each zone

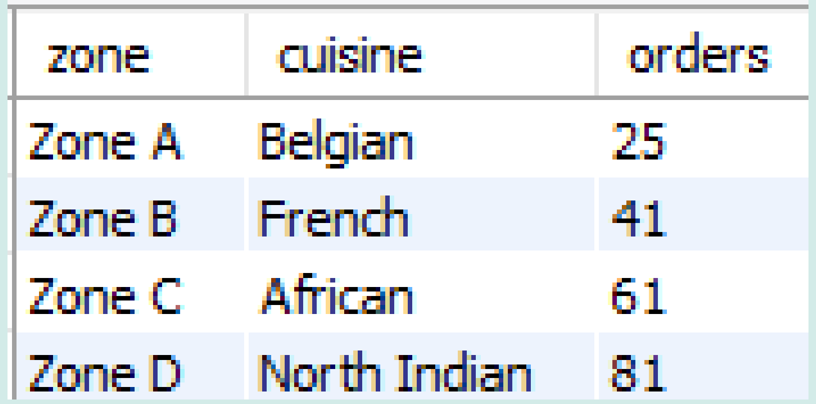
MOST ORDERED CUISINE IN EACH ZONE

SQL Query



```
with t as (select zone, cuisine, count(order_id) as orders
from restaurant inner join order_details using(restaurant_id)
group by zone, cuisine
order by orders), a as (select *, rank() over(partition by zone order by orders desc) as rnk from t)
select zone, cuisine, orders from a where rnk = 1;
```

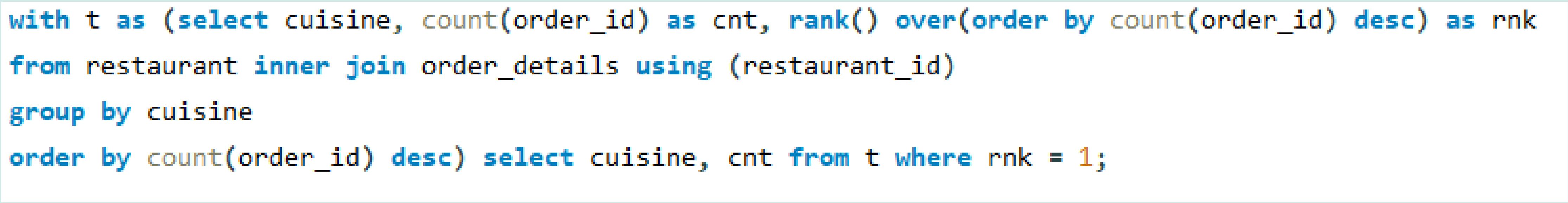
Output



zone	cuisine	orders
Zone A	Belgian	25
Zone B	French	41
Zone C	African	61
Zone D	North Indian	81

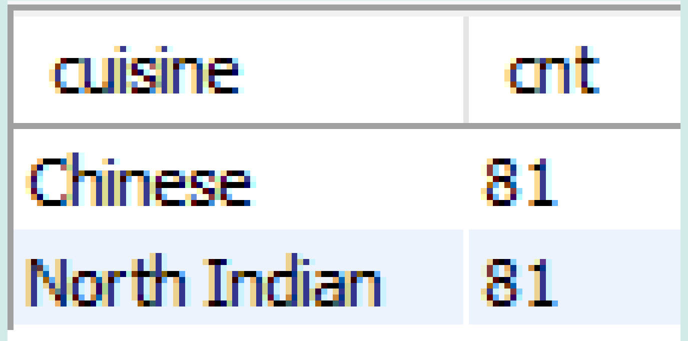
OVERALL MOST ORDERED CUISINE

SQL Query



```
with t as (select cuisine, count(order_id) as cnt, rank() over(order by count(order_id) desc) as rnk
from restaurant inner join order_details using (restaurant_id)
group by cuisine
order by count(order_id) desc) select cuisine, cnt from t where rnk = 1;
```

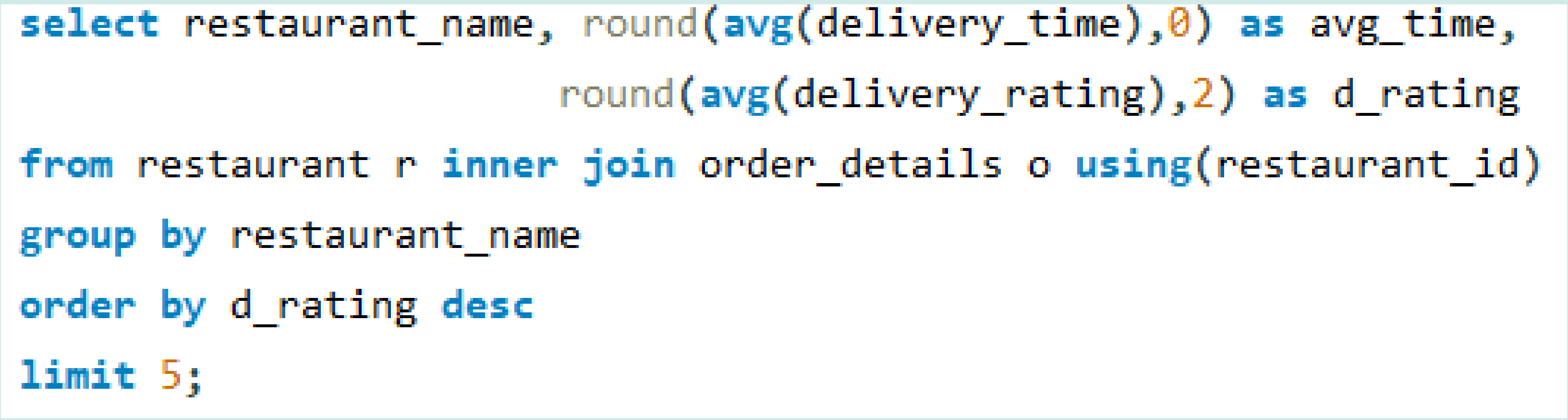
Output



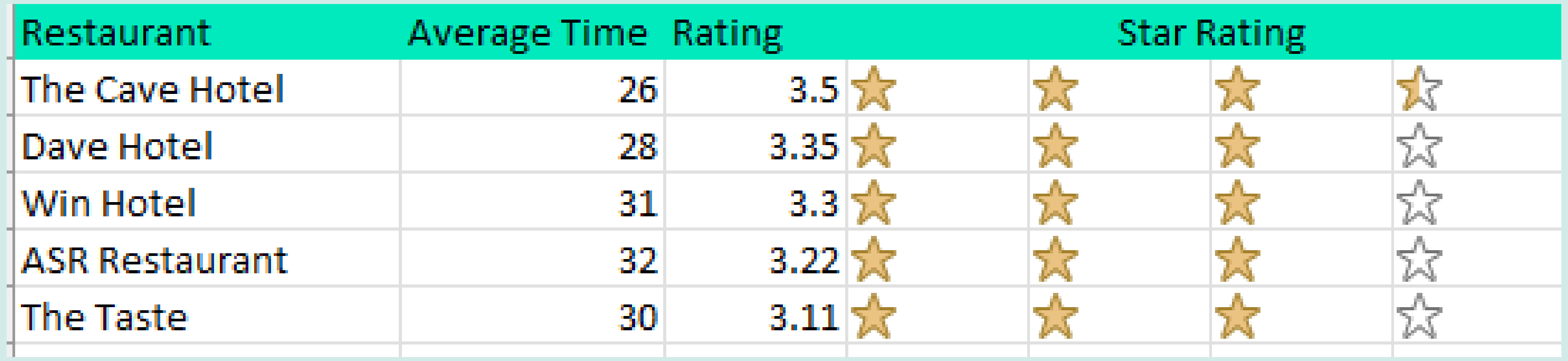
cuisine	cnt
Chinese	81
North Indian	81

TOP-5 RESTAURANTS ON THE BASIS OF DELIVERY-RATING

SQL Query



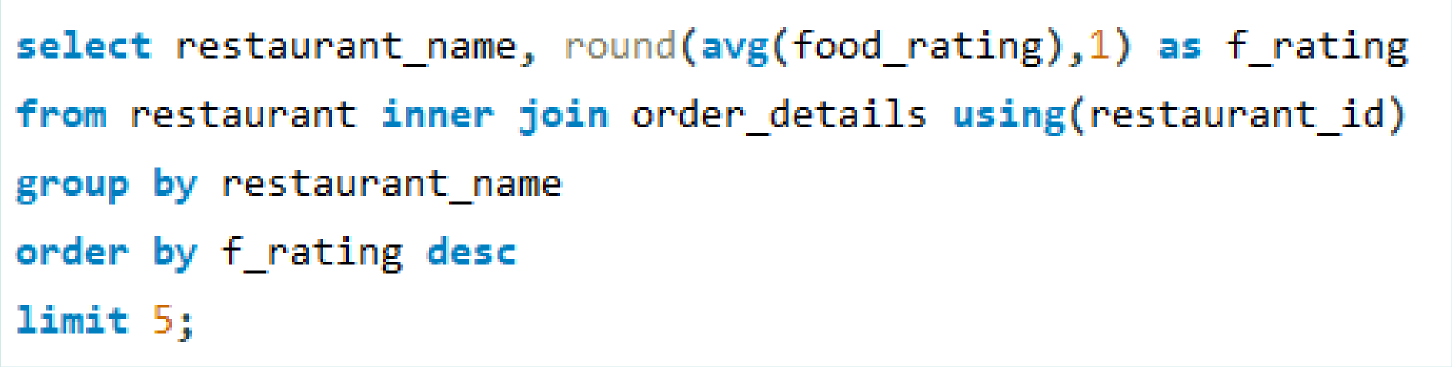
```
select restaurant_name, round(avg(delivery_time),0) as avg_time,
                        round(avg(delivery_rating),2) as d_rating
from restaurant r inner join order_details o using(restaurant_id)
group by restaurant_name
order by d_rating desc
limit 5;
```



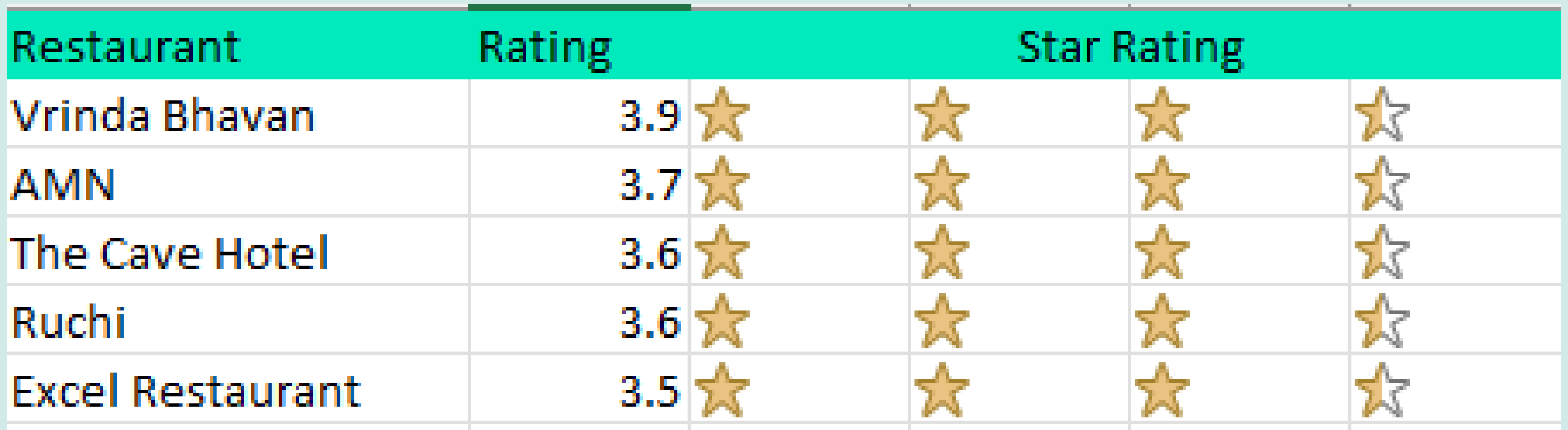
Restaurant	Average Time	Rating	Star Rating			
The Cave Hotel	26	3.5	★	★	★	⯪
Dave Hotel	28	3.35	★	★	★	☆
Win Hotel	31	3.3	★	★	★	☆
ASR Restaurant	32	3.22	★	★	★	☆
The Taste	30	3.11	★	★	★	☆

TOP-5 RESTAURANTS ON THE BASIS OF FOOD-RATING

SQL Query



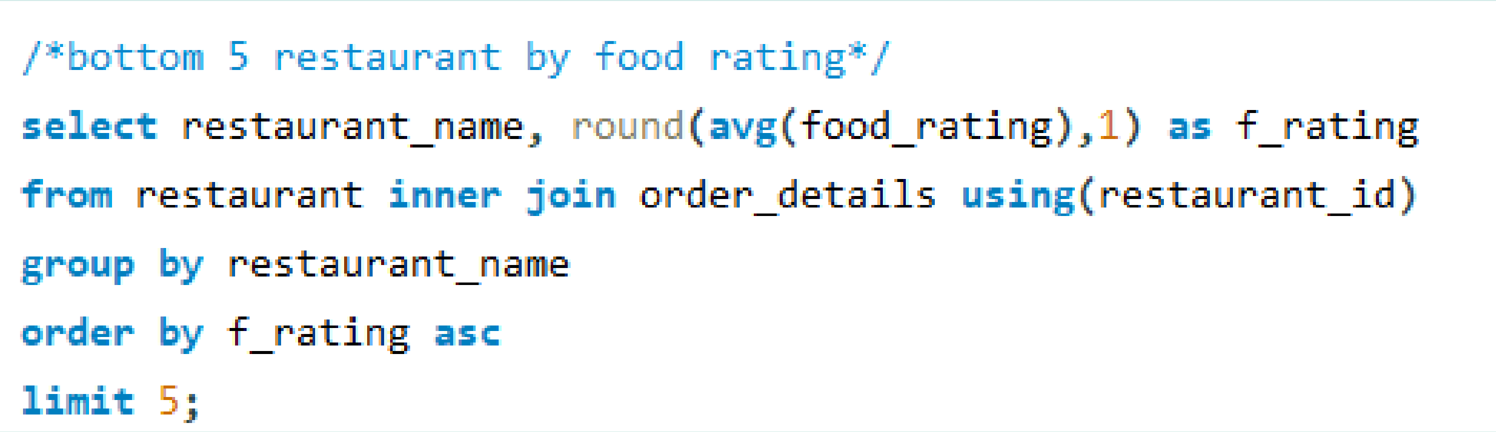
```
select restaurant_name, round(avg(food_rating),1) as f_rating
from restaurant inner join order_details using(restaurant_id)
group by restaurant_name
order by f_rating desc
limit 5;
```



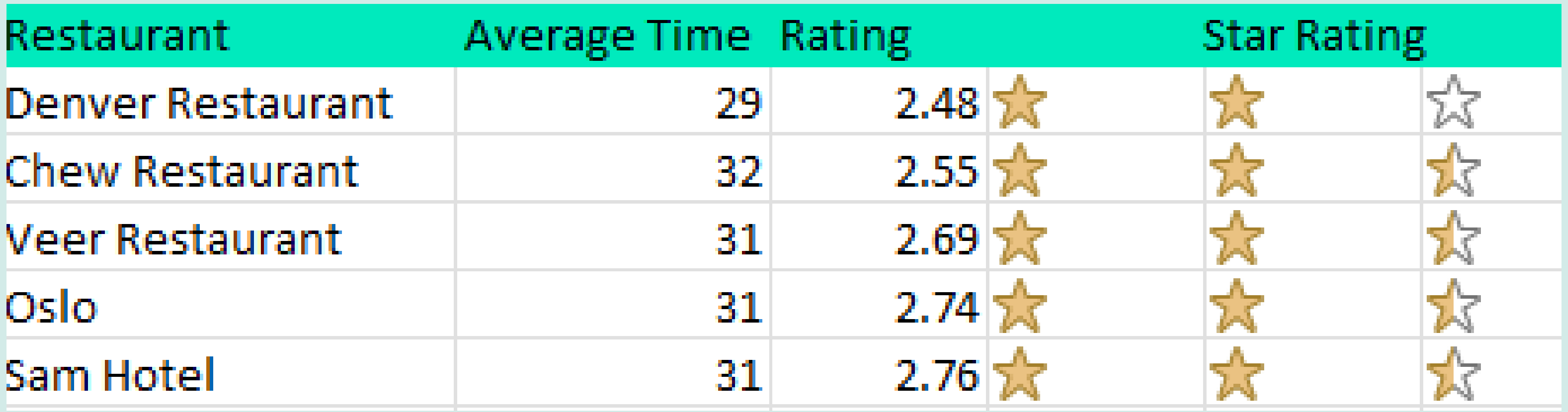
Restaurant	Rating	Star Rating			
Vrinda Bhavan	3.9	★	★	★	⯪
AMN	3.7	★	★	★	⯪
The Cave Hotel	3.6	★	★	★	⯪
Ruchi	3.6	★	★	★	⯪
Excel Restaurant	3.5	★	★	★	⯪

BOTTOM-5 RESTAURANTS ON THE BASIS OF DELIVERY-RATING

SQL Query



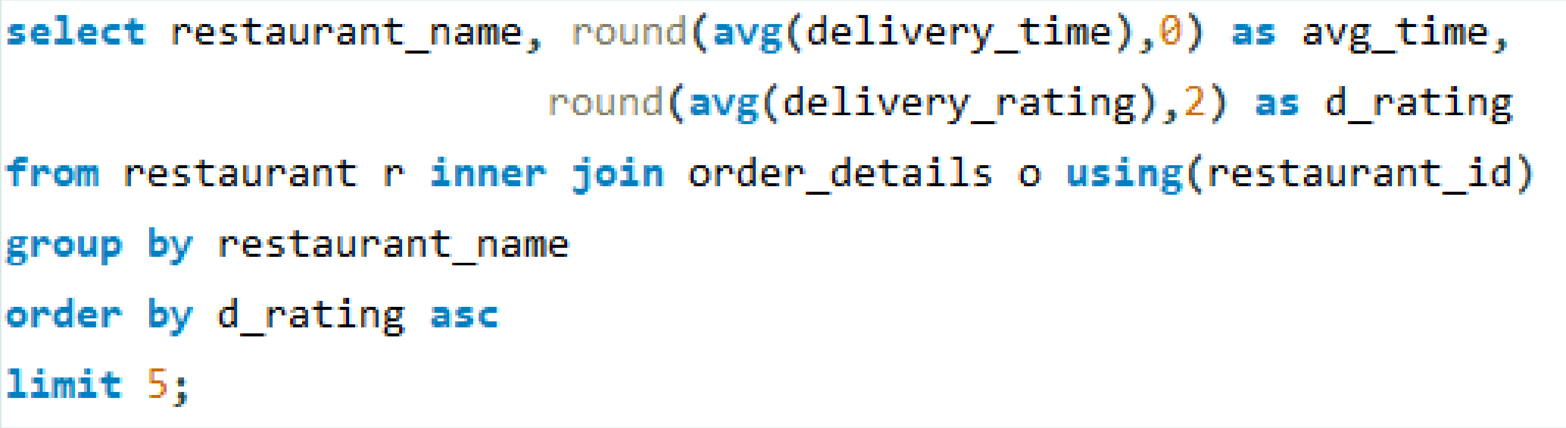
```
/*bottom 5 restaurant by food rating*/
select restaurant_name, round(avg(food_rating),1) as f_rating
from restaurant inner join order_details using(restaurant_id)
group by restaurant_name
order by f_rating asc
limit 5;
```



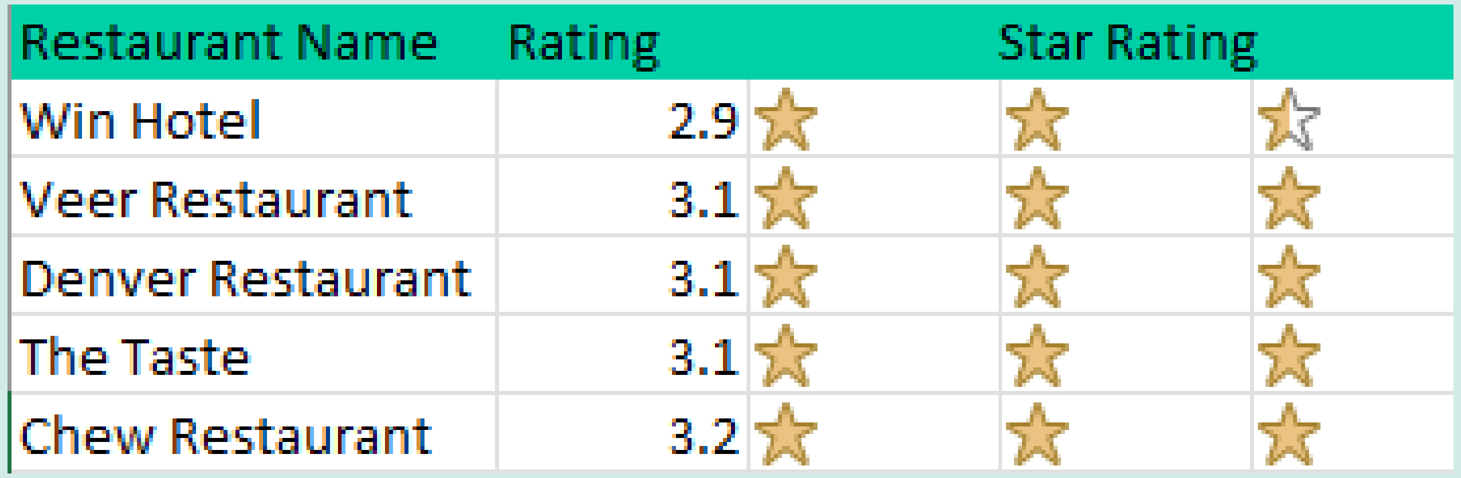
Restaurant	Average Time	Rating		Star Rating	
Denver Restaurant	29	2.48	★	★	☆
Chew Restaurant	32	2.55	★	★	⯨
Veer Restaurant	31	2.69	★	★	⯨
Oslo	31	2.74	★	★	⯨
Sam Hotel	31	2.76	★	★	⯨

BOTTOM-5 RESTAURANTS ON THE BASIS OF FOOD-RATING

SQL Query



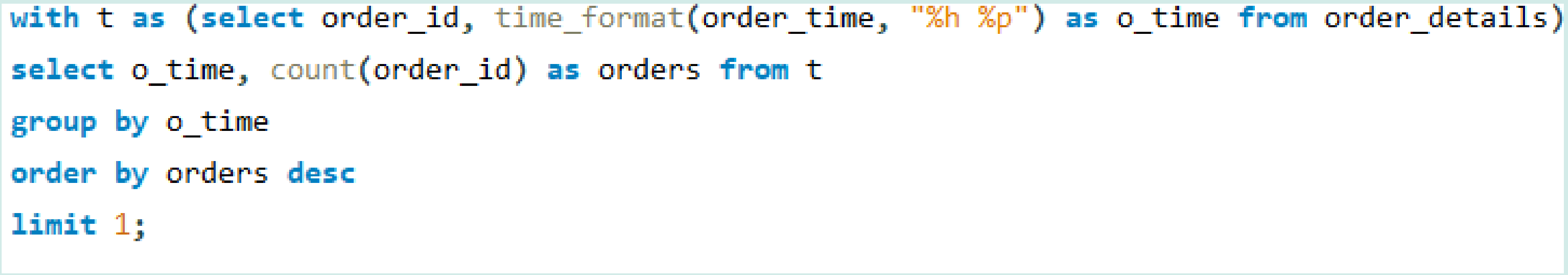
```
select restaurant_name, round(avg(delivery_time),0) as avg_time,
                        round(avg(delivery_rating),2) as d_rating
from restaurant r inner join order_details o using(restaurant_id)
group by restaurant_name
order by d_rating asc
limit 5;
```



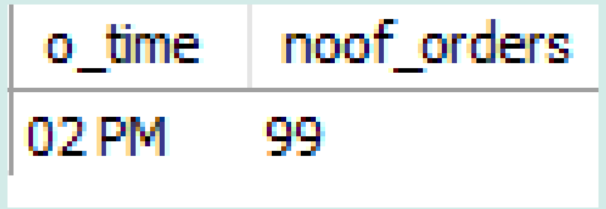
Restaurant Name	Rating		Star Rating	
Win Hotel	2.9	★	★	⯨
Veer Restaurant	3.1	★	★	★
Denver Restaurant	3.1	★	★	★
The Taste	3.1	★	★	★
Chew Restaurant	3.2	★	★	★

Rush Hour

SQL Query

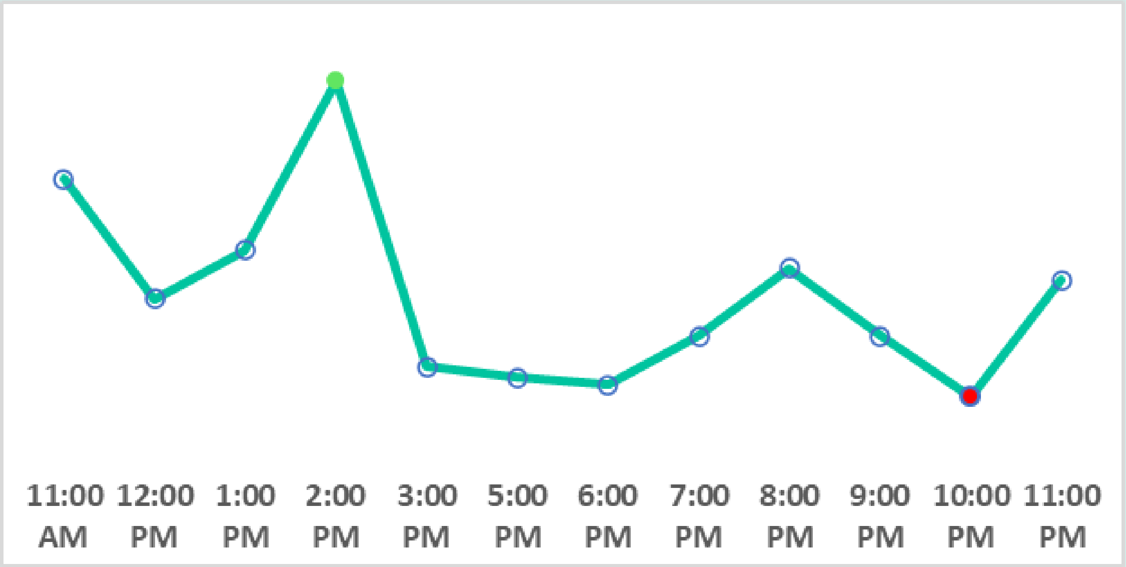


```
with t as (select order_id, time_format(order_time, "%h %p") as o_time from order_details)
select o_time, count(order_id) as orders from t
group by o_time
order by orders desc
limit 1;
```



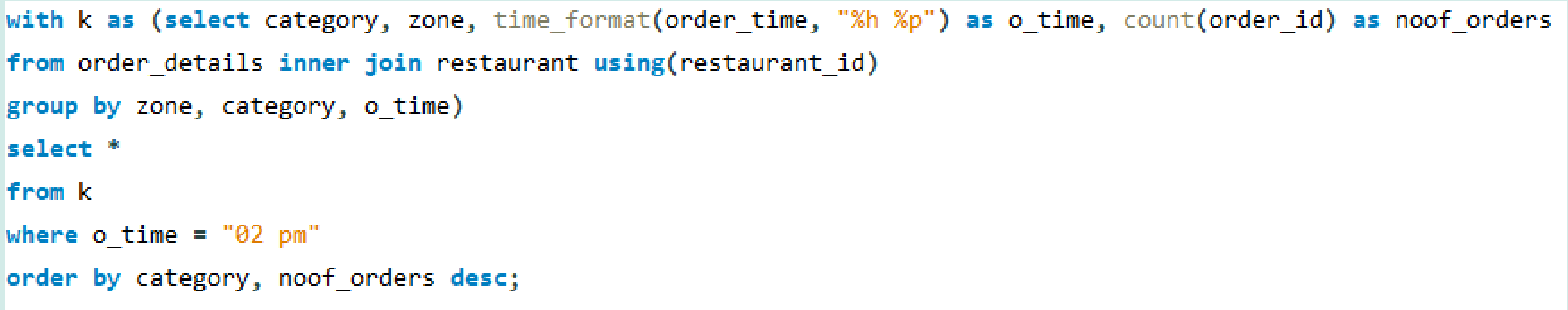
o_time	noof_orders
02 PM	99

Output

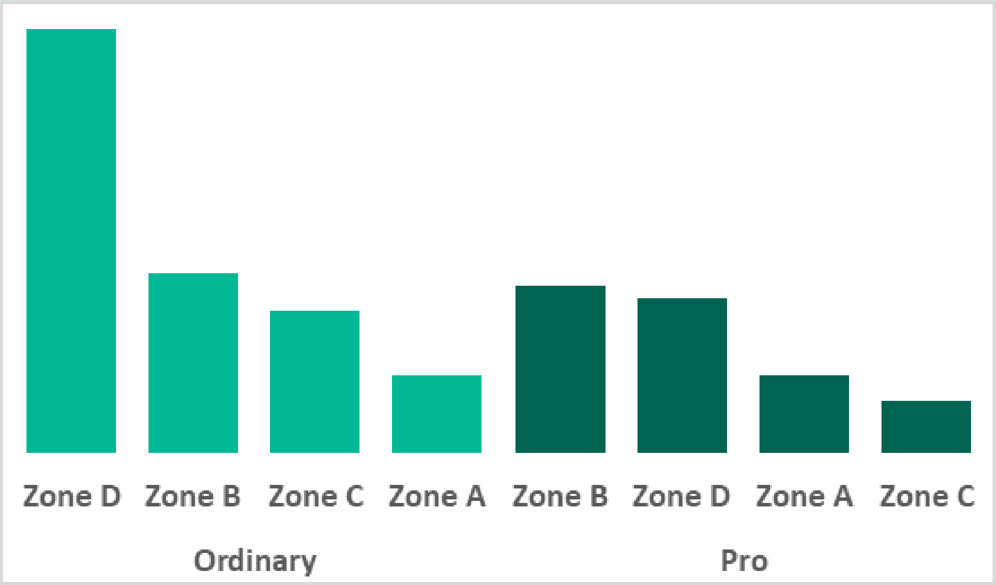


Orders received in each zone during the rush hour.

SQL Query



```
with k as (select category, zone, time_format(order_time, "%h %p") as o_time, count(order_id) as noof_orders
from order_details inner join restaurant using(restaurant_id)
group by zone, category, o_time)
select *
from k
where o_time = "02 pm"
order by category, noof_orders desc;
```



INSIGHTS

- **Zone C** has **maximum sales** even though the number of orders and restaurants is very less.
- **Zone D** is performing well in terms of both sales and orders and even has the **most number of restaurants**. The lower sales might be due to price differences. It can be concluded that the price in Zone D restaurants is more reasonable.
- **Srini** ordered the most.
- **Ordinary** category has superb performance.
- Overall, the most ordered cuisine is **"Chinese"** and **"North Indian"**. Cuisines famous in each zone are **"North Indian", "Belgian", "French" and "African".**
- **"The Cave Hotel"** has the best reputation.
- **"Denver Restaurant", "Chew Restaurant", and "Veer Restaurant"** needs to improve both food and delivery service.
- Rush hour is **2 pm**. Zone D in ordinary and Zone B in Pro receives the **Max orders**.

THANK YOU