```
use restaurant;

use restaurant;

-- Modifying geoplaces2 table: Replace '?' with '0' in the zip column
where needed
UPDATE geoplaces2
SET zip = REPLACE(zip, '?', '0')
WHERE zip LIKE '%?%';

-- Modifying geoplaces2 table: Altering column data types and adding
primary key
ALTER TABLE geoplaces2
MODIFY placeID INT,
MODIFY latitude REAL,
MODIFY longitude REAL,
MODIFY the_geom_meter TEXT,
MODIFY name VARCHAR(255),
MODIFY address TEXT,
MODIFY city VARCHAR(100),
MODIFY state VARCHAR(100),
MODIFY country VARCHAR(100),
MODIFY fax VARCHAR(100),
MODIFY zip INT,
MODIFY alcohol VARCHAR(50),
MODIFY smoking_area VARCHAR(50),
MODIFY dress_code VARCHAR(50),
MODIFY accessibility VARCHAR(50),
MODIFY price VARCHAR(50),
MODIFY url VARCHAR(255),
MODIFY Rambience VARCHAR(50),
MODIFY franchise VARCHAR(50),
MODIFY area VARCHAR(50),
MODIFY other_services VARCHAR(255),
ADD PRIMARY KEY (placeID);


-- Modifying userprofile table: Remove 'U' prefix from userid and alter
column data types
UPDATE userprofile
SET userid = SUBSTRING(userid, 2)
WHERE userid LIKE 'U%';

ALTER TABLE userprofile
MODIFY userID INT,
MODIFY latitude REAL,
MODIFY longitude REAL,
MODIFY smoker VARCHAR(10),
MODIFY drink_level VARCHAR(20),
MODIFY dress_preference VARCHAR(50),
MODIFY ambience VARCHAR(50),
MODIFY transport VARCHAR(50),
MODIFY marital_status VARCHAR(20),
MODIFY hijos VARCHAR(50),
```

```
MODIFY birth_year INT,
MODIFY interest VARCHAR(100),
MODIFY personality VARCHAR(100),
MODIFY religion VARCHAR(50),
MODIFY activity VARCHAR(100),
MODIFY color VARCHAR(20),
MODIFY weight INT,
MODIFY budget VARCHAR(20),
MODIFY height DECIMAL(5,2),
ADD PRIMARY KEY (userID);


-- Modifying rating_final table: Remove 'U' prefix from userid, alter
column data types, and add foreign keys
UPDATE rating_final
SET userid = SUBSTRING(userid, 2)
WHERE userid LIKE 'U%';

ALTER TABLE rating_final
MODIFY userID INT,
MODIFY placeID INT,
MODIFY rating INT CHECK (rating IN (0,1,2)),
MODIFY food_rating INT CHECK (food_rating IN (0,1,2)),
MODIFY service_rating INT CHECK (service_rating IN (0,1,2)),
ADD FOREIGN KEY (userID) REFERENCES userprofile(userID),
ADD FOREIGN KEY (placeID) REFERENCES geoplaces2(placeID);


-- Modifying userpayment table: Remove 'U' prefix from userid, alter
column data types, and add foreign key
UPDATE userpayment
SET userid = SUBSTRING(userid, 2)
WHERE userid LIKE 'U%';

ALTER TABLE userpayment
MODIFY COLUMN userID INT,
MODIFY COLUMN Upayment VARCHAR(50),
ADD FOREIGN KEY (userID) REFERENCES userprofile(userID);


-- Modifying chefmozaccepts table: Alter column data types
ALTER TABLE chefmozaccepts
MODIFY COLUMN placeID INT,
MODIFY COLUMN Rpayment VARCHAR(50);


-- Modifying chefmozcuisine table: Alter column data types
ALTER TABLE chefmozcuisine
MODIFY COLUMN placeID INT,
MODIFY COLUMN Rcuisine VARCHAR(50);


-- Modifying chefmozhours4 table: Alter column data types
ALTER TABLE chefmozhours4
```

```
MODIFY COLUMN placeID INT,
-- MODIFY COLUMN hours TIME, -- Uncomment if time data type is needed
MODIFY COLUMN days TEXT;


-- Modifying chefmozparking table: Alter column data types
ALTER TABLE chefmozparking
MODIFY COLUMN placeID INT,
MODIFY COLUMN parking_lot VARCHAR(20);


-- Modifying usercuisine table: Remove specific user data, remove 'U'
prefix from userid, and alter column data types
DELETE FROM usercuisine
WHERE userid = 'userid';

UPDATE usercuisine
SET userid = SUBSTRING(userid, 2)
WHERE userid LIKE 'U%';

ALTER TABLE usercuisine
MODIFY COLUMN userid INT,
MODIFY COLUMN Rcuisine VARCHAR(50);

#Question 1: - We need to find out the total visits to all restaurants
#under all alcohol categories available.

SELECT
     name, alcohol,
      COUNT(placeid) AS total_visits -- Selecting columns 'name',
'alcohol', and counting the occurrences of 'placeid' as 'total_visits'
FROM
   geoplaces2  -- From the 'geoplaces2' table
GROUP BY name, alcohol;  -- Grouping the results based on the 'name' and
'alcohol' columns

/*
The above query provides the total number of visits to each restaurant,
categorized by their names and the type of alcohol they serve.
*/


# Question 2: -Let's find out the average rating according to alcohol and
# price so that we can understand the rating in respective price
categories as well.

SELECT
     g.alcohol, g.price, AVG(r.rating) AS avg_rating   -- Selecting
columns: alcohol type, price, and average rating
FROM
geoplaces2 g JOIN rating_final r USING (placeid)  -- From tables:
geoplaces2 (aliased as g) and rating_final (aliased as r)
```

```
GROUP BY g.alcohol, g.price; -- Grouping the results by alcohol type and
price

/*
This query calculates the average rating for restaurants based on their
alcohol type and price category.
*/


# Question 3:  Letâ€™s write a query to quantify that what are the parking
availability as
# well in different alcohol categories along with the total number of
restaurants.

SELECT
    g.Alcohol, -- Selecting the 'Alcohol' column from the 'geoplaces2'
table
    COUNT(*) AS total_restaurants, -- Counting the number of occurrences
for each combination of 'Alcohol' and 'parking_lot'
    c.parking_lot -- Selecting the 'parking_lot' column from the
'chefmozparking' table
FROM
    geoplaces2 g -- Alias 'g' for the 'geoplaces2' table
JOIN
    chefmozparking c USING (placeid) -- Joining the 'geoplaces2' and
'chefmozparking' tables using the 'placeid' column
GROUP BY
    g.Alcohol, c.parking_lot; -- Grouping the results by 'Alcohol' and
'parking_lot'

/*
This query provides information on the number of restaurants in each
alcohol category with different parking availability.
*/


# Question 4: -Also take out the percentage of different cuisine in each
alcohol type.

-- Selecting columns for analysis
SELECT
     g.alcohol,                      -- Alcohol availability in the
restaurant
     u.rcuisine,                     -- Restaurant cuisine type
     COUNT(u.placeid) AS cuisine_count,  -- Count of restaurants for each
alcohol and cuisine combination
     (COUNT(u.placeid) * 100.0 / SUM(COUNT(u.placeid)) OVER (PARTITION BY
g.alcohol)) AS percentage  -- Calculating percentage of each cuisine type
within its alcohol category
FROM
     chefmozcuisine u                -- Using the chefmozcuisine table
with alias 'u'
JOIN
```

```
     geoplaces2 g USING (placeid)      -- Joining with geoplaces2 table on
the 'placeid' column
GROUP BY
     g.alcohol, u.rcuisine;           -- Grouping the results by alcohol
and cuisine type

/*
This query calculates the percentage of different cuisines in each alcohol
category.
*/


# Questions 5: - letâ€™s take out the average rating of each state.

SELECT
    g.state, g.alcohol,
    AVG(r.rating) AS avg_rating  -- Selecting state, alcohol, and average
rating from geoplaces2 and rating_final tables
FROM
 geoplaces2 g JOIN  rating_final r USING(placeid) -- Joining geoplaces2
and rating_final tables using the placeid column
GROUP BY g.state, g.alcohol;  -- Grouping the results by state and alcohol
to calculate the average rating for each group

/*
This query calculates the average rating for restaurants in each state,
categorized by alcohol type.
*/


# Questions 6: -' Tamaulipas' Is the lowest average rated state.
# Quantify the reason why it is the lowest rated by providing the summary
on the basis of State, alcohol, and Cuisine.

SELECT DISTINCT g.state, g.alcohol, z.rcuisine, AVG(r.rating) -- Selecting
distinct state, alcohol, cuisine, and average rating columns
FROM geoplaces2 g JOIN rating_final r USING(placeid) -- Joining the
geoplaces2 and rating_final tables using the placeid column
JOIN chefmozcuisine z USING(placeid) -- Joining the chefmozcuisine table
using the placeid column

WHERE state LIKE '%Tamaulipas%' -- Filtering results to include only
places in the state of Tamaulipas

GROUP BY g.state, g.alcohol, z.rcuisine -- Grouping the results by state,
alcohol, and cuisine to get unique combinations
ORDER BY AVG(r.rating) -- Ordering the results by the average rating in
ascending order

LIMIT 1; -- Limiting the output to only the top result

/*
The above query analyzes why 'Tamaulipas' has the lowest average rating,
considering alcohol type and cuisine.
```

```
Tamaulipas' lower average rating may be attributed to its policy of not
serving alcohol and
the relatively lower rating of its Regional cuisine (0.50),
potentially impacting its overall rating compared to other states.
*/



# Question 7:  - Find the average weight, food rating, and service rating
of the customers
# who have visited KFC and tried Mexican or Italian types of cuisine,
# and also their budget level is low. We encourage you to give it a try by
not using joins.


SELECT
AVG(p.weight) AS avg_weight,
AVG(r.food_rating) AS avg_food_rating,
AVG(r.service_rating) AS avg_service_rating  -- Selecting the average
weight of users, average food rating, and average service rating
FROM
userprofile p,
rating_final r,
usercuisine u,
geoplaces2 g   -- From the following tables: userprofile, rating_final,
usercuisine, and geoplaces2
WHERE
p.userid = r.userid           -- Matching user IDs between userprofile and
rating_final
AND p.userid = u.userid       -- Matching user IDs between userprofile and
usercuisine
AND g.placeid = r.placeid     -- Matching place IDs between geoplaces2 and
rating_final
AND g.name = 'KFC'            -- Specifying the restaurant name as 'KFC'
AND (u.Rcuisine = 'Mexican' OR u.Rcuisine = 'Italian')     -- Specifying
the cuisine types as 'Mexican' or 'Italian'
AND p.budget = 'low';         -- Specifying the budget as 'low'

/*
The above query retrieves average values for user weight, food rating, and
service rating from
the userprofile, rating_final, usercuisine, and geoplaces2 tables. It
focuses on users who have a 'low' budget,
have rated a restaurant named 'KFC,' and have a cuisine preference for
either 'Mexican' or 'Italian'.
The results provide insights into the average characteristics of users
meeting these criteria.
*/
```