**SQL Assignment 2**

**Dataset:** Jomato

**About the dataset:**

You work for a data analytics company, and your client is a food delivery platform similar to Jomato. They have provided you with a dataset containing information about various restaurants in a city. Your task is to analyze this dataset using SQL queries to extract valuable insights and generate reports for your client.

/\*Tasks to be performed:\*/

--1. Create a user-defined functions to stuff the Chicken into ‘Quick Bites’.

Eg: ‘Quick Chicken Bites’.

select \* from Jomato

/\*This creates a new user-defined function called udf\_StuffChicken

that accepts a varchar input parameter called @RestaurantType and returns a varchar output. \*/

CREATE FUNCTION udf\_StuffChicken(@RestaurantType varchar(50))

RETURNS varchar (100) AS

BEGIN

/\*This declares a variable to hold the output string. It checks if the input @RestaurantType is 'Quick Bites', and if so, sets @Stuffed to the modified string 'Quick Chicken Bites'.Otherwise, it just sets @Stuffed to the original input \*/

DECLARE @Stuffed varchar(100)

IF @RestaurantType = 'Quick Bites'

SET @Stuffed = 'Quick Chicken Bites'

ELSE

SET @Stuffed = @RestaurantType

/\*Returns the @Stuffed variable as the output of the function.\*/

RETURN @Stuffed

END

GO

-- call the function

SELECT dbo.udf\_StuffChicken(RestaurantType) AS RestaurantType from Jomato

SELECT OrderId,RestaurantName,dbo.udf\_StuffChicken(RestaurantType) AS RestaurantType,Rating,No\_of\_Rating,

AverageCost, OnlineOrder, TableBooking, CuisinesType, Area, LocalAddress, Delivery\_time FROM Jomato

/\*2. Use the function to display the restaurant name and cuisine type which has the maximum number of rating.\*/

select \* from Jomato

select orderid,RestaurantName,CuisinesType,Rating,No\_of\_Rating

from Jomato

where rating = (select Max(rating)Rating from Jomato)

/\*use of subquery to ckeck out all the rows of the table\*/

order by No\_of\_Rating desc

-- 5. Display the restaurant type and total average cost using rollup.

-- The COALESCE function replaces any NULL values in the RestaurantType column with 'Total' for the grand total row

SELECT COALESCE(RestaurantType, 'Total')RestaurantType,

AVG(AverageCost) AvgCost

FROM jomato

GROUP BY RestaurantType WITH ROLLUP

order by AvgCost desc

select \* from Jomato

SELECT RestaurantType, AVG(AverageCost) AvgCost FROM jomato

GROUP BY RestaurantType WITH ROLLUP

order by AvgCost desc

/\*3. Create a Rating Status column to display the rating as ‘Excellent’ if it has more the 4 start rating,

‘Good’ if it has above 3.5 and below 4 star rating, ‘Average’ if it is above 3

and below 3.5 and ‘Bad’ if it is below 3 star rating \*/

SELECT RestaurantName,RestaurantType,Rating,Area,

CASE

WHEN Rating > 4 THEN 'Excellent'

WHEN Rating > 3.5 AND Rating <= 4 THEN 'Good'

WHEN Rating > 3 AND Rating <= 3.5 THEN 'Average'

ELSE 'Bad'

END

Rating\_Status FROM jomato

select \* from Jomato

/\* 4. Find the Ceil, floor and absolute values of the rating column and display the current date and separately display the year, month\_name and day.\*/

SELECT OrderId,RestaurantName,Rating,No\_of\_Rating,Area,

CASE

WHEN Rating > 4 THEN 'Excellent'

WHEN Rating > 3.5 AND Rating <= 4 THEN 'Good'

WHEN Rating > 3 AND Rating <= 3.5 THEN 'Average'

ELSE 'Bad'

END AS RatingStatus,

CEILING(Rating) AS CeilRating,

FLOOR(Rating) AS FloorRating,

ABS(Rating) AS AbsRating,

GETDATE() AS CurrentDate,

YEAR(GETDATE()) AS CurrentYear,

DATENAME(month, GETDATE()) AS CurrentMonth,

DAY(GETDATE()) AS CurrentDay

FROM jomato

--1. Create a stored procedure to display the restaurant name, type and cuisine where the table booking is not zero.

select \* from Jomato

CREATE PROCEDURE get\_restaurants\_with\_table\_bookings

AS

BEGIN

SELECT RestaurantName, RestaurantType, CuisinesType

FROM jomato

WHERE TableBooking ='yes'

END

GO

EXEC get\_restaurants\_with\_table\_bookings;

drop procedure get\_restaurants\_with\_table\_bookings

--2. Create a transaction and update the cuisine type ‘Cafe’ to ‘Cafeteria’. Check the result and rollback it.

select \* from Jomato

BEGIN TRANSACTION;

-- Update the cuisine type 'Cafe' to 'Cafeteria'

UPDATE Jomato

SET CuisinesType = REPLACE(CuisinesType, 'Cafe', 'Cafeteria')

WHERE CuisinesType LIKE '%Cafe%';

-- Check the result

SELECT CuisinesType, COUNT(\*) AS Count

FROM jomato

GROUP BY CuisinesType

ORDER BY CuisinesType desc;

-- Rollback the transaction

ROLLBACK TRANSACTION;

-- Check the result after rollback

SELECT CuisinesType, COUNT(\*) AS Count

FROM jomato

GROUP BY CuisinesType

ORDER BY CuisinesType;

--3. Generate a row number column and find the top 5 areas with the highest rating of restaurants.

--use of subquery to mark row number wrt rating

SELECT Area, Rating, ROW\_NUMBER() OVER (PARTITION BY Area ORDER BY Rating DESC) AS RowNum FROM jomato

SELECT top 5 Area, AVG(Rating) AS AvgRating

FROM ( SELECT Area, Rating, ROW\_NUMBER() OVER (PARTITION BY Area ORDER BY Rating DESC) AS RowNum

FROM jomato ) y

WHERE RowNum <= 5

GROUP BY Area

ORDER BY AvgRating DESC

--4. Use the while loop to display the 1 to 50.

DECLARE @Counter INT = 1;

WHILE @Counter <= 50

BEGIN

PRINT @Counter;

SET @Counter = @Counter + 1;

END;

--creating a function

CREATE FUNCTION DisplayNumbers()

RETURNS @Numbers TABLE (Number INT)

AS

BEGIN

DECLARE @Counter INT = 1;

WHILE @Counter <= 50

BEGIN

INSERT INTO @Numbers (Number)

VALUES (@Counter);

SET @Counter = @Counter + 1;

END

RETURN;

END

GO

-- Call the function

SELECT Number

FROM DisplayNumbers()

ORDER BY Number;

--5. Write a query to Create a Top rating view to store the generated top 5 highest rating of restaurants.

select \* from Jomato

CREATE VIEW TopRatedRestaurants

AS

SELECT top 5 OrderId, RestaurantName, RestaurantType, Rating,No\_of\_Rating, AverageCost,OnlineOrder, TableBooking ,CuisinesType, Area, LocalAddress, Delivery\_time FROM Jomato

ORDER BY Rating DESC

SELECT \* FROM TopRatedRestaurants

--6. CREATE TRIGGER new\_record\_inserted

CREATE TRIGGER tr\_Insert\_Message

ON jomato

AFTER INSERT

AS

BEGIN

DECLARE @Message VARCHAR(255)

SET@Message ='A new record has been inserted into the jomato table,Thank you .'

PRINT @Message

END

GO

--drop trigger tr\_InsertMessage

--select \* from Jomato

--check the trigger by inserting a new record

INSERT INTO jomato (RestaurantName, RestaurantType, Rating, Area)

VALUES ('New Restaurant', 'Italian', 4.5, 'Downtown');