

# 1. SQL DDL commands for Table Creation

```
CREATE TABLE Complaint (  
    Complaint_id INT NOT NULL IDENTITY(1,1),  
    Complaint_text VARCHAR(255) NOT NULL,  
    Complaint_status VARCHAR(50) NOT NULL,  
    Filed_date_time DATETIME NOT NULL,  
    User_id INT NOT NULL,  
    PRIMARY KEY(Complaint_id)  
);
```

```
Create table Complaints_On_Shop  
(Complaint_id int not null,  
Shop_id int not null,  
Primary Key (Complaint_id),  
);
```

```
Create table Complaints_On_Restaurant  
(Complaint_id int not null,  
Outlet_id int not null,  
Primary Key(Complaint_id),  
);
```

```
CREATE TABLE User_Relation(  
    User_id1 INT NOT NULL,  
    User_id2 INT NOT NULL,  
    Relationship varchar(50) NOT NULL,  
    Primary Key(User_id1, User_id2)  
);
```

```
Create table Shop_Transaction  
(User_id int not null,  
Shop_id int not null,  
Amount_spent int not null,  
Date_time_in datetime not null,  
Date_time_out datetime not null,  
Primary Key (User_id, Shop_id, Date_time_in)  
);
```

```
Create table Restaurant_Outlet  
(Outlet_id int not null,  
Restaurant_id int not null,  
Mall_id int not null,
```

Primary Key(Outlet\_id)  
);

```
CREATE TABLE User_Account (  
    User_id INTEGER NOT NULL IDENTITY(1,1),  
    Username VARCHAR(255) NOT NULL,  
    Gender VARCHAR(1) NOT NULL,  
    DOB DATE NOT NULL,  
    PRIMARY KEY(User_id)  
);
```

```
CREATE TABLE Dine (  
    User_id INTEGER NOT NULL,  
    Outlet_id INTEGER NOT NULL,  
    Amount_spent FLOAT NOT NULL,  
    Date_time_in DATETIME NOT NULL,  
    Date_time_out DATETIME NOT NULL,  
    PRIMARY KEY(User_id,Outlet_id,Date_time_in)  
);
```

```
CREATE TABLE Shop (  
    Shop_id INTEGER NOT NULL,  
    Category VARCHAR(128) NOT NULL,  
    Mall_id INTEGER NOT NULL,  
    PRIMARY KEY(Shop_id)  
);
```

```
CREATE TABLE Voucher (  
    Voucher_id INTEGER NOT NULL,  
    Details VARCHAR(255),  
    Date_issued DATE NOT NULL,  
    Date_expiring DATE NOT NULL,  
    PRIMARY KEY(Voucher_id)  
);
```

```
CREATE TABLE Purchase_Voucher (  
    Voucher_id INTEGER NOT NULL,  
    Purchase_discount FLOAT NOT NULL,  
    User_id INTEGER NOT NULL,  
    Date_time_used DATETIME,  
    PRIMARY KEY(Voucher_id)  
);
```

```
CREATE TABLE Dine_Voucher (  
    User_id INTEGER NOT NULL,  
    Outlet_id INTEGER NOT NULL,  
    Amount_spent FLOAT NOT NULL,  
    Date_time_in DATETIME NOT NULL,  
    Date_time_out DATETIME NOT NULL,  
    PRIMARY KEY(User_id,Outlet_id,Date_time_in)  
);
```

```
Voucher_id INTEGER NOT NULL,  
Cash_discount INTEGER NOT NULL,  
User_id INTEGER NOT NULL,  
Date_time_used DATETIME NOT NULL,  
PRIMARY KEY(Voucher_id)  
);
```

```
CREATE TABLE Group_Voucher(  
Voucher_id INTEGER NOT NULL,  
Group_size INTEGER NOT NULL,  
Group_discount FLOAT NOT NULL,  
Date_time_used DATETIME,  
PRIMARY KEY (Voucher_id)  
);
```

```
CREATE TABLE Package_Voucher(  
Voucher_id INTEGER NOT NULL,  
Package_discount FLOAT NOT NULL,  
PRIMARY KEY(Voucher_id)  
);
```

```
CREATE TABLE Recommendation(  
Recommendation_id INTEGER NOT NULL,  
Mall_id INTEGER NOT NULL,  
Outlet_id INTEGER NOT NULL,  
DayPackage_id INTEGER NOT NULL,  
Voucher_id INTEGER NOT NULL,  
Date_expired DATE NOT NULL,  
Date_issued DATE NOT NULL,  
PRIMARY KEY(Recommendation_id)  
);
```

```
CREATE TABLE UserAccount_Use_Recommendation(  
User_id INTEGER NOT NULL,  
Recommendation_id INTEGER NOT NULL,  
PRIMARY KEY(User_id,Recommendation_id)  
);
```

```
CREATE TABLE Mall(  
Mall_id INTEGER NOT NULL,  
Company_ID INTEGER NOT NULL,  
Address_of_mall VARCHAR(255) NOT NULL,  
NumShops INTEGER NOT NULL
```

);

```
CREATE TABLE Mall_Has_DayPackage(  
    DayPackage_id INTEGER NOT NULL,  
    Mall_id INTEGER NOT NULL,  
    PRIMARY KEY(DayPackage_id, Mall_id)  
);
```

```
CREATE TABLE Day_Package(  
    DayPackage_id INTEGER NOT NULL,  
    User_id INTEGER NOT NULL,  
    Voucher_id INTEGER NOT NULL,  
    Description VARCHAR(255) NOT NULL,  
    PRIMARY KEY(DayPackage_id)  
);
```

```
CREATE TABLE Day_Package_Has_RestaurantOutlet(  
    DayPackage_id INTEGER NOT NULL,  
    Outlet_id INTEGER NOT NULL,  
    PRIMARY KEY(DayPackage_id,Outlet_id)  
);
```

```
CREATE TABLE Mall_Mgmt_Company(  
    Company_id INTEGER NOT NULL,  
    Company_Address VARCHAR(255) NOT NULL,  
    PRIMARY KEY(Company_id)  
);
```

```
CREATE TABLE Restaurant_Chain(  
    RestaurantChain_id INTEGER NOT NULL,  
    RestaurantChain_Address VARCHAR(255) NOT NULL,  
    PRIMARY KEY(RestaurantChain_id)  
);
```

## 2.SQL statements to solve queries

- Each query should be immediately followed by the query output
- Explain how output is obtained

# 1. Most popular day packages where all participants are related to one another as either family members or members of the same club

```
SELECT
  DP.DayPackage_id,
  DP.Description,
  COUNT(*) AS ParticipantsCount
FROM
  Day_Package DP
INNER JOIN
  User_Relation UR1 ON DP.User_id = UR1.User_id1
INNER JOIN
  User_Relation UR2 ON DP.User_id = UR2.User_id2
WHERE
  (UR1.Relationship IN ('Married', 'Parent-Child', 'Siblings') OR UR2.Relationship IN ('Married',
  'Parent-Child', 'Siblings'))
  OR
  (UR1.Relationship = 'Clubmates' OR UR2.Relationship = 'Clubmates')
GROUP BY
  DP.DayPackage_id, DP.Description
ORDER BY
  ParticipantsCount DESC;
```

	DayPackage_id	Description	ParticipantsCount
1	12	City Square Mall, Funan, Liat Towers, Marina Square	4
2	4	Beauty World Centre, Beauty World Plaza, Greenri...	4
3	3	City Square Mall, Funan, Liat Towers, Marina Square	3
4	13	City Square Mall, Funan, Liat Towers, Marina Square	3
5	14	City Square Mall, Funan, Liat Towers, Marina Square	2
6	6	Fajar Shopping Centre, JCube, Lot One, Rail Mall	2
7	7	CityLink Mall, HDB Hub, Raffles City, Plaza Singap...	2
8	16	Bugis Junction, Cathay Cineleisure Orchard, Milleni...	1
9	17	Bugis Junction, Cathay Cineleisure Orchard, Milleni...	1

## Explanation:

- Three columns(DayPackage\_id, Description, Count(\*) AS ParticipantsCount) are selected from the Day\_Package table named DP
- Two inner joins are performed on two User Relations tables named UR1 and UR2, and the joins connect to DP based on the User\_id
- The WHERE clause filters the rows based on:
  1. whether the relationship in UR1 or UR2 is either 'Married', 'Siblings' or 'Parent-Child'
  2. whether the relationship in UR1 or UR2 is Clubmates

- The GROUP BY clause groups the result by DayPackage\_id and Description so that COUNT can be applied to each group
- The ORDER BY clause orders the result by ParticipantsCount in descending order so that the most popular day packages, with the highest ParticipantCount, are shown at the top

## 2. Families who frequently shopped and dined together with or without day packages

- indicate whether these families use day packages or not as part of your output
- frequently means at least 50% of the time

WITH f AS

```
( SELECT Dine.User_id, Dine.Date_time_in, Dine.Date_time_out
  FROM Dine
```

```
  UNION
```

```
  SELECT Shop_Transaction.User_id, Shop_Transaction.Date_time_in,
 Shop_Transaction.Date_time_out
  FROM Shop_Transaction
```

```
  RIGHT JOIN User_Relation ON User_id = User_Relation.User_id1 OR User_id =
 User_Relation.User_id2),
```

x AS

```
( SELECT User_id, COUNT(*) AS count1
  FROM
```

```
( SELECT User_id
```

```
  FROM f
```

```
  WHERE Date_time_in IN (
```

```
    SELECT Date_time_in
```

```
    FROM f
```

```
    GROUP BY Date_time_in
```

```
    HAVING COUNT(*) > 1)) AS sub
```

```
  GROUP BY User_id),
```

y AS

```
( SELECT User_id, COUNT(*) AS count2
```

```
  FROM
```

```
( SELECT User_id
```

```
  FROM f
```

```
  WHERE Date_time_in IN (
```

```
    SELECT Date_time_in
```

```
    FROM f
```

```
    GROUP BY Date_time_in
```

```
    HAVING COUNT(*) = 1)) AS sub
```

```
  GROUP BY User_id),
```

xy AS

```

( SELECT User_id, count1, NULL as count2
  FROM x
  UNION
  SELECT User_id, NULL as count1, count2
  FROM y)
SELECT sub.User_id, User_Account.Username,
       CASE WHEN EXISTS (SELECT *
                        FROM Day_Package
                        WHERE sub.User_id = Day_Package.User_id)
       THEN 1 ELSE 0 END AS usedPackage
FROM
( SELECT User_id, MAX(count1) AS count1, MAX(count2) AS count2
  FROM xy
  GROUP BY User_id) AS sub
LEFT JOIN User_Account ON User_Account.User_id = sub.User_id
WHERE count1 >= count2

```

	User_id	Username	usedPackage
1	5	Elijah Ang	1
2	6	Fanny Ang	1
3	7	Gilbert Ang	1
4	8	Holly Ang	1

#### Explanation:

- First create temporary table f which contains every shop and dining transaction records by users who have a relation with another user,
- Then create temporary table x which counts the amount of transactions that have overlapping date\_time\_in for each user,
- As well as temporary table y which counts the amount of transactions that does not have overlapping date\_time\_in,
- Create table xy which is a union between the two,
- Lastly select user id, username as well as a case for whether the user has a day package or not from table xy, join with day\_package, then filter out users who have more transactions alone than in a group.

### 3. Most popular recommendations from the app regarding malls

```

SELECT Recommendation.Recommendation_id, Recommendation.Mall_id,
       Recommendation.Outlet_id,
       Recommendation.DayPackage_id, Recommendation.Voucher_id,
       Recommendation.Date_expired,
       Recommendation.Date_issued

```

```

FROM Recommendation
JOIN UserAccount_Use_Recommendation ON Recommendation.Recommendation_id =
UserAccount_Use_Recommendation.Recommendation_id
JOIN Mall ON Recommendation.Mall_id = Mall.Mall_id
GROUP BY Recommendation.Recommendation_id, Recommendation.Mall_id,
Recommendation.Outlet_id,
Recommendation.DayPackage_id, Recommendation.Voucher_id,
Recommendation.Date_expired,
Recommendation.Date_issued
HAVING COUNT(DISTINCT UserAccount_Use_Recommendation.User_id) = (
    SELECT MAX(num_users)
    FROM (
        SELECT COUNT(DISTINCT User_id) AS num_users
        FROM UserAccount_Use_Recommendation
        GROUP BY Recommendation_id
    ) AS subquery
)

```

	Recommendation_id	Mall_id	Outlet_id	DayPackage_id	Voucher_id	Date_expired	Date_issued
1	1	1	1	1	1	2024-06-01	2024-04-01
2	2	3	3	1	2	2024-07-01	2024-02-01

#### Explanation:

- The select statement retrieves the desired columns from the recommendation table, which would have been joined with other tables, explained further below.
- The recommendation table is joined with UserAccount\_Use\_Recommendation based on the Recommendation\_id column, and the Mall table based on the Mall\_id column.
- The GROUP BY clause is used to allow us to filter the records by the HAVING clause.
- We filter grouped records to only include records whose count of distinct User\_id are equal to the maximum count of distinct user IDs across all the recommendation ids using the HAVING clause.
- Those grouped records mentioned above are obtained by the subquery that calculates the maximum count of distinct user IDs, grouped by Recommendation\_ids from the UserAccount\_Use\_Recommendation table.
- In the output, only the most popular recommendation is shown. In this case, there are two recommendations with the max number of users.



#### 4. Youngest shoppers who have visited a certain mall more than 5 times during December 2023 and the amount they spent in total

/\*Find 3 youngest compulsive shoppers and amount spent each\*/

```
WITH Dec_Transactions AS (  
    SELECT DISTINCT User_id, SUM(Amount_spent) AS Amount_spent  
    FROM Shop_Transaction, Shop  
    WHERE MONTH(Date_time_in)=12 AND YEAR(Date_time_in)=2023  
           AND Shop.Shop_id = Shop_Transaction.Shop_id  
    GROUP BY User_id, Mall_id  
    HAVING COUNT(User_id)>5  
)  
Compulsive_Shoppers AS (  
    SELECT TOP 3 User_Account.User_id, Username, DOB  
    FROM Dec_Transactions, User_Account  
    WHERE Dec_Transactions.User_id = User_Account.User_id  
    GROUP BY User_Account.User_id, Username, DOB  
    ORDER BY DOB DESC  
)  
SELECT Compulsive_Shoppers.User_id, Compulsive_Shoppers.Username,  
SUM(Amount_spent) AS Amount_spent  
FROM Compulsive_Shoppers, Shop_Transaction  
WHERE Compulsive_Shoppers.User_id = Shop_Transaction.User_id  
GROUP BY Compulsive_Shoppers.User_id, Compulsive_Shoppers.Username
```

	User_id	Username	Amount_spent
1	3	Cyndi Tan	414
2	11	Kelly Oubre	953
3	1	Ali Baba	762

/\*Find amount spent by the top 3 compulsive shoppers in total, in December\*/

```
WITH Dec_Transactions AS (  
    SELECT DISTINCT User_id, SUM(Amount_spent) AS Amount_spent  
    FROM Shop_Transaction, Shop  
    WHERE MONTH(Date_time_in)=12 AND YEAR(Date_time_in)=2023  
           AND Shop.Shop_id = Shop_Transaction.Shop_id  
    GROUP BY User_id, Mall_id  
    HAVING COUNT(User_id)>5  
)  
Compulsive_Shoppers AS (  
    SELECT TOP 3 User_Account.User_id, Username, DOB  
    FROM Dec_Transactions, User_Account  
    WHERE Dec_Transactions.User_id = User_Account.User_id  
    GROUP BY User_Account.User_id, Username, DOB
```

```

ORDER BY DOB DESC
), Amount_Spent_In_Dec AS (
    SELECT Compulsive_Shoppers.User_id, Compulsive_Shoppers.Username,
    SUM(Amount_spent) AS Amount_spent
    FROM Compulsive_Shoppers, Shop_Transaction
    WHERE Compulsive_Shoppers.User_id = Shop_Transaction.User_id
    GROUP BY Compulsive_Shoppers.User_id, Compulsive_Shoppers.Username
)SELECT SUM(Amount_spent) AS Total_spent
FROM Amount_Spent_In_Dec;

```

	Total_spent
1	2129

#### Explanation:

- We retrieve transactions that occurred in December 2023, using MONTH(Date\_time\_in)=12 AND YEAR(Date\_time\_in)=2023
- Then, count the number times a particular mall was visited by joining shop and mall tables with Shop.Shop\_id = Shop\_Transaction.Shop\_id, and assigning the data to users
- A user belongs to the compulsive shopper category if they visit a particular mall more than 5 times as seen by COUNT(User\_id)>5
- Then, we retrieve information about the user's age by joining the DOB column Dec\_Transactions.User\_id = User\_Account.User\_id and getting the youngest users by sorting in descending order
- The amount spent reflected in the first output is the amount spent at the respective malls in which they visited more than 5 times and does not fully account for the total amount spent as transactions in other malls have not been accounted for
- Next, to get the total amount spent by the top 3 youngest compulsive shoppers in December 2023 across all malls, we summed the amount spent for each of these users from the shop transactions table using SELECT SUM(Amount\_spent) AS Total\_spent FROM Amount\_Spent\_In\_Dec; therefore producing the second output

## 5. Users who have dined in all the restaurants in some malls and never dined in any restaurants in some other malls

-- find total number of restaurants for each mall

WITH Total\_restaurants AS

(SELECT Mall\_id, COUNT(DISTINCT Outlet\_id) AS total\_restaurants

FROM Restaurant\_Outlet

GROUP BY Mall\_id)

-- find number of unique restaurants each user has dined in for each mall

, User\_restaurants\_dined AS

(SELECT User\_id, Mall\_id, COUNT(DISTINCT Dine.Outlet\_id) AS restaurants\_dined\_in

```
FROM Dine JOIN Restaurant_Outlet ON Dine.Outlet_id = Restaurant_Outlet.Outlet_id
GROUP BY User_id, Mall_id)
```

```
-- compare counts for each user
```

```
SELECT ua.User_id, ua.Username, COUNT(DISTINCT Dine.Outlet_id) AS Outlets_dined_in
FROM User_Account AS ua
JOIN Dine ON ua.User_id = Dine.User_id
JOIN Restaurant_Outlet AS ro ON Dine.Outlet_id = ro.Outlet_id
JOIN User_restaurants_dined AS urd ON Dine.User_id = urd.User_id
JOIN Total_restaurants AS tr ON ro.Mall_id = tr.Mall_id
GROUP BY ro.Mall_id, ua.User_id, ua.Username, restaurants_dined_in, total_restaurants
HAVING restaurants_dined_in = total_restaurants -- dined in all restaurants at a mall
AND ua.User_id NOT IN (
    SELECT ua.User_id
    FROM User_Account AS ua
    JOIN Dine ON ua.User_id = Dine.User_id
    JOIN Restaurant_Outlet AS ro ON Dine.Outlet_id = ro.Outlet_id
    GROUP BY ua.User_id
    HAVING COUNT(DISTINCT ro.Mall_id) > 1 -- dined in more than 1 mall
);
```

	User_id	Username	Outlets_dined_in
1	9	Isaac Lee	3
2	10	Jack Black	3
3	11	Kelly Oubre	3

#### Explanation

- The query works by finding the total number of restaurants at each mall and the number of unique restaurants each user has dined in for each mall using the 'COUNT(DISTINCT)' aggregate function.
- The 'HAVING' clause is used to check if: the number of unique restaurants a user has dined in = the total number of restaurants at any malls.
- The subsequent 'NOT IN' clause checks for users who met the previous condition but also have dined in restaurants in other malls, and removes them from the result.

## 6. Top 3 highest earning malls and restaurants

```
SELECT TOP 3 Mall_id, SUM(Amount_spent) AS Earnings
FROM Shop, Shop_Transaction
WHERE Shop.Shop_id = Shop_Transaction.Shop_id
GROUP BY Mall_id
ORDER BY Earnings DESC;
```

	Mall_id	Earnings
1	1	2968
2	7	2489
3	3	1297

```
SELECT TOP 3 Restaurant_id, SUM(Amount_spent) AS Earnings
FROM Restaurant_Outlet, Dine
WHERE Restaurant_Outlet.Outlet_id = Dine.Outlet_id
GROUP BY Restaurant_id
ORDER BY Earnings DESC;
```

	Restau...	Earnin...
1	1	139.5
2	6	121.3
3	5	81.699999...

#### Explanation

- The query works by using the 'SUM' aggregate function to get the total amount\_spent at each shop/restaurant.
- The 'TOP 3' keyword limits the output to only 3 results. Since the results are ordered by earnings in descending order, this yields the top 3 shops/restaurants with the highest earnings.
- To get the highest earning mall, the Shop table is joined in order to retrieve the shop's Mall\_id.

## 7. Shops that received the most complaints in December 2023

```
SELECT cs.Shop_id, COUNT(*) AS Number_of_complaints
FROM Complaints_On_Shop AS cs
JOIN Complaint AS c ON cs.Complaint_id = c.Complaint_id
WHERE MONTH(c.Filed_date_time)=12 AND YEAR(c.Filed_date_time)=2023
GROUP BY cs.Shop_id
ORDER BY Number_of_complaints DESC;
```

	Shop_id	Num_complaints
1	11	4
2	1	2
3	8	1

#### Explanation

- Two columns(cs.Shop\_id, Count(\*) as Number\_of\_complaints) are selected

- The data is taken from two tables(Complaints\_On\_Shop, Complaint).
- Complaints\_On\_Shop is named 'cs' while Complaint is named 'c'
- cs is then joined with c based on Complaint\_id
- The WHERE clause then filters the rows so that only complaints in December 2023 are obtained. The month and year is obtained using Filed\_date\_time from c.
- The GROUP BY clause causes the result to be grouped based on cs.Shop\_id so that COUNT can be applied to each group to get the number of complaints per shop
- Finally, the result is sorted by the Number\_of complaints in descending order so the shop with the most complaints is at the top

### 3.Printout of all Table Records

SELECT \* FROM Complaint;

	Complaint_id	Complaint_text	Complaint_status	Filed_date_time	User_id
1	1	The service staff was racist. They called me smelly.	pending	2024-03-10 18:40:59.000	1
2	2	The food here is way too expensive! I can make b...	pending	2024-03-10 19:11:11.000	2
3	3	The prices of items are too high! I can buy all of thi...	pending	2024-03-10 20:02:02.000	3
4	4	The seats were dirty and stained. Please clean th...	pending	2024-03-10 21:01:37.000	4
5	5	The chicken they served me was raw! My tummy ...	being handled	2024-03-10 19:01:24.000	5
6	6	I think I saw a cockroach in the kitchen area.	addressed	2024-02-20 19:55:19.000	6
7	7	There are no seats in the store for me to rest my fe...	addressed	2024-01-13 11:41:49.000	7
8	8	The shop was too dirty!	addressed	2023-12-30 16:03:03.000	11
9	9	The service staff said I would fail SC2207. He hurt...	pending	2023-12-01 12:55:08.000	8
10	10	Everything is too expensive!	pending	2023-12-18 18:47:02.000	9
11	11	I think the gucci shoes sold here are fake...	being handled	2023-12-22 20:03:50.000	10
12	12	The aircon was too cold!	pending	2023-12-12 13:06:07.000	12
13	13	The aircon was not working!	pending	2023-12-12 13:06:07.000	13
14	14	They did not have stock of everything I wanted...	pending	2023-12-12 13:06:07.000	14
15	15	I think I saw some mould on one of the sofas...	pending	2023-12-12 13:06:07.000	15
16	16	A chair broke when I sat on it. It is not safe!	pending	2023-12-12 13:06:07.000	16

SELECT \* FROM Complaints\_On\_Shop;

	Complaint_id	Shop_id
1	1	1
2	3	11
3	7	5
4	8	1
5	9	6
6	10	29
7	11	1
8	12	11
9	13	11
10	14	8
11	15	11
12	16	11

SELECT \* FROM Complaints\_On\_Restaurant;

	Complaint_id	Order_id
1	2	1
2	4	1
3	5	4
4	6	10

SELECT \* FROM User\_Relation;

	User_id1	User_id2	Relationship
1	1	11	Clubmates
2	1	14	Clubmates
3	2	15	Classmates
4	2	16	Classmates
5	5	6	Marned
6	5	7	Parent-Child
7	5	8	Parent-Child
8	5	15	Classmates
9	6	7	Parent-Child
10	6	8	Parent-Child
11	6	15	Classmates
12	7	8	Clubmates
13	7	15	Clubmates
14	8	15	Clubmates
15	11	14	Clubmates
16	14	23	Clubmates
17	15	16	Classmates
18	18	19	Parent-Child
19	23	24	Clubmates
20	24	25	Colleagues

SELECT \* FROM Shop\_Transaction ;

	User_id	Shop_id	Amount_spent	Date_time_in	Date_time_out
1	1	1	85	2023-12-04 12:05:00.000	2023-12-04 12:18:00.000
2	1	2	85	2023-12-04 10:30:00.000	2023-12-04 12:01:00.000
3	1	3	102	2023-12-20 15:05:00.000	2023-12-20 16:33:00.000
4	1	4	77	2023-12-13 14:30:00.000	2023-12-13 15:53:00.000
5	1	5	32	2023-12-14 16:05:17.000	2023-12-14 16:18:25.000
6	1	10	88	2023-12-14 09:20:00.000	2023-12-14 10:58:00.000
7	1	11	15	2024-03-24 20:22:51.000	2024-03-24 20:45:36.000
8	1	13	73	2023-12-20 09:20:00.000	2023-12-20 10:38:00.000
9	1	21	64	2023-12-10 12:30:00.000	2023-12-10 14:02:00.000
10	1	24	91	2023-12-09 14:05:00.000	2023-12-09 15:43:00.000
11	1	31	50	2023-12-05 19:32:10.000	2023-12-05 20:04:55.000
12	2	1	94	2023-12-28 14:45:00.000	2023-12-28 16:24:00.000
13	2	7	80	2024-03-25 16:40:31.000	2024-03-25 16:53:20.000
14	2	16	108	2023-12-06 13:30:00.000	2023-12-06 14:57:00.000
15	2	25	103	2023-12-23 11:35:00.000	2023-12-23 13:02:00.000
16	3	2	64	2023-12-28 13:10:00.000	2023-12-28 14:36:00.000
17	3	6	98	2023-12-28 12:55:00.000	2023-12-28 14:28:00.000
18	3	21	70	2023-12-11 12:20:00.000	2023-12-11 13:49:00.000
19	3	22	10	2023-12-01 17:20:00.000	2023-12-29 18:59:00.000
20	3	23	10	2023-12-02 17:20:00.000	2023-12-29 18:59:00.000
21	3	24	8	2023-12-06 14:14:14.000	2023-12-06 14:24:34.000
22	3	25	10	2023-12-03 17:20:00.000	2023-12-29 18:59:00.000
23	3	26	10	2023-12-04 17:20:00.000	2023-12-29 18:59:00.000
24	3	27	65	2023-12-19 10:50:00.000	2023-12-19 12:16:00.000
25	3	28	69	2023-12-30 11:20:00.000	2023-12-30 12:43:00.000
26	4	5	117	2023-12-18 12:50:00.000	2023-12-18 14:13:00.000
27	4	6	67	2023-12-15 14:55:00.000	2023-12-15 16:22:00.000
28	4	15	10	2024-03-26 22:00:22.000	2024-03-26 22:06:28.000
29	4	18	12	2024-03-26 21:34:24.000	2024-03-26 21:42:55.000
30	5	5	28	2024-03-26 20:20:00.000	2024-03-26 21:02:15.000
31	5	10	10	2024-03-26 12:01:45.000	2023-03-26 12:40:31.000
32	5	15	4	2024-03-21 10:15:45.000	2023-03-21 10:18:45.000
33	5	21	10	2024-03-22 14:12:33.000	2023-12-04 14:57:29.000
34	6	10	5	2024-03-26 12:01:45.000	2023-03-26 12:40:31.000
35	6	15	4	2024-03-21 10:15:45.000	2023-12-21 10:18:45.000
36	6	21	15	2024-03-22 14:12:33.000	2023-12-04 14:57:29.000
37	7	10	8	2024-03-26 12:01:45.000	2023-03-26 12:40:31.000
38	7	15	4	2024-03-21 10:15:45.000	2023-12-21 10:18:45.000

39	7	21	5	2024-03-22 14:12:33.000	2023-12-04 14:57:29.000
40	8	10	12	2024-03-26 12:01:45.000	2023-03-26 12:40:31.000
41	8	15	4	2024-03-21 10:15:45.000	2023-12-21 10:18:45.000
42	8	21	8	2024-03-22 14:12:33.000	2023-12-04 14:57:29.000
43	9	2	22	2024-03-24 20:02:44.000	2024-03-24 20:30:30.000
44	9	10	63	2023-12-28 13:30:00.000	2023-12-28 14:51:00.000
45	9	27	92	2023-12-27 13:20:00.000	2023-12-27 14:52:00.000
46	10	2	84	2023-12-24 15:30:00.000	2023-12-24 16:56:00.000
47	10	4	55	2023-12-11 16:10:00.000	2023-12-11 17:49:00.000
48	10	12	78	2023-12-14 13:20:00.000	2023-12-14 14:41:00.000
49	10	27	9	2024-03-24 12:48:25.000	2024-03-24 12:57:46.000
50	11	3	69	2023-12-26 15:50:00.000	2023-12-26 17:17:00.000
51	11	5	69	2023-12-22 12:10:00.000	2023-12-22 13:36:00.000
52	11	6	99	2023-12-28 11:10:00.000	2023-12-28 12:39:00.000
53	11	10	102	2023-12-26 11:40:00.000	2023-12-26 13:08:00.000
54	11	15	78	2023-12-27 14:20:00.000	2023-12-27 15:45:00.000
55	11	23	112	2023-12-19 15:20:00.000	2023-12-19 16:58:00.000
56	11	24	70	2023-12-01 12:06:21.000	2023-12-01 12:50:31.000
57	11	25	112	2023-12-11 15:20:00.000	2023-12-19 16:58:00.000
58	11	26	103	2023-12-25 10:35:00.000	2023-12-25 12:16:00.000
59	11	27	40	2023-12-18 15:16:17.000	2023-12-18 16:17:18.000
60	11	28	99	2023-12-14 16:10:00.000	2023-12-14 17:28:00.000
61	12	7	56	2024-03-25 15:58:30.000	2024-03-25 16:15:16.000
62	12	10	97	2023-12-21 11:10:00.000	2023-12-21 12:36:00.000
63	12	25	79	2023-12-29 17:20:00.000	2023-12-29 18:59:00.000
64	12	27	55	2023-12-06 10:22:00.000	2023-12-06 11:35:00.000
65	13	2	81	2023-12-17 15:40:00.000	2023-12-17 17:05:00.000
66	13	7	67	2023-12-10 15:40:00.000	2023-12-10 17:09:00.000
67	13	12	114	2023-12-06 15:10:00.000	2023-12-06 16:35:00.000
68	13	21	75	2023-12-07 09:30:00.000	2023-12-07 11:05:00.000
69	13	22	96	2024-03-24 17:15:18.000	2024-03-24 17:40:34.000
70	13	24	110	2023-12-04 09:30:00.000	2023-12-04 10:56:00.000
71	13	26	103	2023-12-25 10:35:00.000	2023-12-25 12:16:00.000
72	14	9	98	2023-12-18 12:35:00.000	2023-12-18 14:05:00.000
73	14	14	27	2024-03-25 16:45:53.000	2024-03-25 16:49:40.000
74	14	22	125	2023-12-24 11:15:00.000	2023-12-24 12:44:00.000
75	14	26	88	2023-12-15 14:35:00.000	2023-12-15 15:57:00.000
76	14	27	67	2023-12-02 11:45:00.000	2023-12-02 13:24:00.000



77	15	7	32	2024-03-25 15:58:32.000	2024-03-25 16:15:10.000
78	15	10	88	2023-12-14 09:20:00.000	2023-12-14 10:58:00.000
79	15	11	62	2023-12-03 10:15:00.000	2023-12-03 11:41:00.000
80	15	18	118	2023-12-30 12:45:00.000	2023-12-30 14:10:00.000
81	16	2	115	2023-12-20 10:15:00.000	2023-12-20 11:39:00.000
82	16	4	101	2023-12-19 14:25:00.000	2023-12-19 15:56:00.000
83	16	10	10	2024-03-27 18:46:23.000	2024-03-27 18:56:29.000
84	16	14	68	2023-12-29 09:40:00.000	2023-12-29 11:09:00.000
85	16	18	105	2023-12-18 14:15:00.000	2023-12-18 15:52:00.000
86	16	19	72	2023-12-21 13:20:00.000	2023-12-21 14:44:00.000
87	16	24	94	2023-12-19 09:30:00.000	2023-12-19 10:52:00.000
88	16	26	105	2023-12-16 11:25:00.000	2023-12-16 12:53:00.000
89	17	5	120	2023-12-21 11:20:00.000	2023-12-21 12:59:00.000
90	17	22	72	2023-12-22 15:20:00.000	2023-12-22 16:45:00.000
91	17	23	72	2023-12-06 16:30:00.000	2023-12-06 18:09:00.000
92	17	25	44	2024-03-25 20:37:15.000	2024-03-25 20:52:48.000
93	18	2	80	2023-12-16 11:50:00.000	2023-12-16 13:14:00.000
94	18	3	105	2023-12-08 15:15:00.000	2023-12-08 16:44:00.000
95	18	5	96	2024-03-26 19:22:06.000	2024-03-26 20:03:37.000
96	18	6	110	2023-12-25 13:30:00.000	2023-12-25 14:59:00.000
97	18	12	93	2023-12-22 13:50:00.000	2023-12-22 15:26:00.000
98	18	19	89	2023-12-16 12:40:00.000	2023-12-16 14:01:00.000
99	18	21	112	2023-12-14 12:55:00.000	2023-12-14 14:24:00.000
100	18	27	62	2023-12-15 15:25:00.000	2023-12-15 16:56:00.000
101	18	30	62	2024-03-26 21:09:26.000	2024-03-26 21:33:07.000
102	19	5	75	2023-12-01 09:30:00.000	2023-12-01 11:07:00.000
103	19	6	120	2023-12-27 12:30:00.000	2023-12-27 14:02:00.000
104	19	13	107	2023-12-11 09:30:00.000	2023-12-11 11:04:00.000
105	19	18	20	2024-03-27 19:05:34.000	2024-03-27 19:25:28.000
106	19	23	7	2024-03-25 18:55:27.000	2024-03-25 19:02:18.000
107	20	1	94	2023-12-28 14:45:00.000	2023-12-28 16:24:00.000
108	20	2	66	2023-12-13 12:45:00.000	2023-12-13 14:25:00.000
109	20	3	108	2023-12-18 10:40:00.000	2023-12-18 12:03:00.000
110	20	4	66	2023-12-13 12:45:00.000	2023-12-13 14:25:00.000
111	20	5	78	2023-12-18 11:10:00.000	2023-12-18 12:29:00.000
112	20	11	4	2024-03-24 10:33:16.000	2024-03-24 10:55:27.000
113	20	12	109	2023-12-27 09:55:00.000	2023-12-27 11:23:00.000
114	20	17	71	2023-12-01 14:55:00.000	2023-12-01 16:21:00.000

115	20	19	67	2023-12-09 12:25:00.000	2023-12-09 13:53:00.000
116	20	27	67	2023-12-02 11:45:00.000	2023-12-02 13:24:00.000
117	20	28	118	2023-12-27 12:20:00.000	2023-12-27 13:59:00.000
118	20	31	100	2023-12-03 15:16:40.000	2023-12-03 16:17:18.000
119	21	1	91	2023-12-09 14:05:00.000	2023-12-09 15:43:00.000
120	21	2	79	2023-12-29 17:20:00.000	2023-12-29 18:59:00.000
121	21	3	65	2023-12-16 15:45:00.000	2023-12-16 17:19:00.000
122	21	4	115	2023-12-25 10:40:00.000	2023-12-25 12:11:00.000
123	21	5	121	2023-12-19 13:50:00.000	2023-12-19 15:14:00.000
124	21	8	11	2024-03-24 19:44:40.000	2024-03-24 20:01:00.000
125	21	11	116	2023-12-08 10:45:00.000	2023-12-08 12:12:00.000
126	21	23	66	2023-12-09 10:30:00.000	2023-12-09 11:59:00.000
127	21	25	114	2023-12-02 10:40:00.000	2023-12-02 12:06:00.000
128	21	28	73	2023-12-11 10:25:00.000	2023-12-11 11:50:00.000
129	21	31	100	2023-12-03 15:26:17.000	2023-12-03 16:35:33.000
130	22	14	74	2024-03-26 18:28:01.000	2024-03-26 19:40:12.000
131	22	15	66	2023-12-12 09:35:00.000	2023-12-12 11:03:00.000
132	22	19	90	2023-12-10 14:05:00.000	2023-12-10 15:43:00.000
133	22	22	84	2023-12-26 14:50:00.000	2023-12-26 16:16:00.000
134	22	28	78	2023-12-26 09:30:00.000	2023-12-26 11:01:00.000
135	23	6	72	2023-12-03 14:20:00.000	2023-12-03 15:47:00.000
136	23	16	108	2024-03-24 13:08:50.000	2024-03-24 13:11:29.000
137	23	18	102	2023-12-13 13:15:00.000	2023-12-13 14:47:00.000
138	24	10	12	2024-03-26 19:17:32.000	2024-03-26 19:27:22.000
139	24	11	10	2024-03-26 19:34:03.000	2024-03-26 19:38:41.000
140	24	19	67	2023-12-09 13:45:00.000	2023-12-09 15:14:00.000
141	24	21	124	2023-12-29 12:15:00.000	2023-12-29 13:43:00.000
142	24	25	68	2023-12-25 15:20:00.000	2023-12-25 16:48:00.000
143	24	27	16	2024-03-24 14:04:13.000	2024-03-24 14:29:02.000
144	24	30	8	2024-03-24 18:08:35.000	2024-03-24 18:33:06.000
145	25	5	79	2023-12-17 14:30:00.000	2023-12-17 15:57:00.000
146	25	6	8	2024-03-27 22:24:38.000	2024-03-27 22:28:53.000
147	25	7	76	2023-12-07 15:00:00.000	2023-12-07 16:28:00.000
148	25	10	103	2023-12-14 09:45:00.000	2023-12-14 11:11:00.000
149	26	14	56	2024-03-25 19:18:49.000	2024-03-25 19:27:04.000
150	26	30	80	2024-03-25 20:09:56.000	2024-03-25 20:14:30.000

SELECT \* FROM Restaurant\_Outlet ;

	Outlet_id	Restaurant_id	Mall_id
1	1	1	1
2	2	3	1
3	3	5	1
4	4	1	2
5	5	6	2
6	6	3	2
7	7	4	3
8	8	2	3
9	9	5	3
10	10	1	4
11	11	2	4
12	12	5	4
13	13	4	5
14	14	3	5
15	15	6	5
16	16	1	6
17	17	2	6
18	18	6	6
19	19	6	7
20	20	3	7
21	21	4	7
22	22	5	8
23	23	2	8
24	24	4	8

SELECT \* FROM User\_Account;

	User_id	Username	Gender	DOB
1	1	Ali Baba	M	2000-01-01
2	2	Betty Crocker	F	2003-10-02
3	3	Cyndi Tan	F	2002-06-04
4	4	Damien Lee	M	2001-10-01
5	5	Elijah Ang	M	1969-04-16
6	6	Fanny Ang	F	1967-07-12
7	7	Gilbert Ang	M	2004-10-11
8	8	Holly Ang	F	2006-12-29
9	9	Isaac Lee	M	1998-02-16
10	10	Jack Black	M	2015-05-19
11	11	Kelly Oubre	F	2000-08-26
12	12	Lewis Kang	M	1984-08-21
13	13	Milly Brown	F	1977-09-01
14	14	Nicholas C...	M	2001-09-11
15	15	Oliver Twist	M	2003-01-28
16	16	Pete David...	M	2002-10-30
17	17	Quentin Lee	M	2010-08-13
18	18	Romeo Tan	M	2001-11-05
19	19	Susan Tan	F	1963-06-16
20	20	Timothy Tan	M	1961-03-03
21	21	Ulysses Ang	M	1999-05-18
22	22	Victoria Maria	F	1988-09-30
23	23	Wendy Tan	F	2010-08-13
24	24	Xenia Ang	F	1969-10-01
25	25	Yasmine Tan	M	1969-03-07
26	26	Zachary Liew	M	1984-04-16

SELECT \* FROM Dine ;

	User_id	Outlet_id	Amount_spent	Date_time_in	Date_time_out
1	1	1	14.1	2024-06-01 18:06:02.000	2024-06-01 19:09:02.000
2	1	3	9.3	2024-11-27 18:32:02.000	2024-11-27 19:35:02.000
3	2	2	6.1	2024-10-02 17:17:19.000	2024-10-02 18:10:55.000
4	2	4	10.7	2024-12-28 18:33:02.000	2024-12-28 19:36:02.000
5	3	3	6.5	2024-10-03 18:08:20.000	2024-10-03 19:11:15.000
6	3	5	12.2	2024-01-29 17:34:16.000	2024-01-29 18:37:01.000
7	4	4	13.6	2024-01-04 18:09:02.000	2024-01-04 19:12:02.000
8	4	6	7.7	2024-02-28 17:35:16.000	2024-02-28 18:38:01.000
9	5	1	12.2	2024-10-30 18:02:02.000	2024-10-30 19:01:02.000
10	5	5	12.2	2024-02-05 17:10:16.000	2024-02-05 18:13:01.000
11	5	5	13.2	2024-11-12 17:02:16.000	2024-11-12 18:05:01.000
12	5	7	8.3	2024-03-01 17:36:16.000	2024-03-01 18:39:01.000
13	6	1	13.1	2024-10-30 18:02:02.000	2024-10-30 19:02:02.000
14	6	5	10.1	2024-11-12 17:03:16.000	2024-11-12 18:06:01.000
15	6	6	9.1	2024-03-06 17:11:16.000	2024-03-06 18:14:01.000
16	6	8	7.3	2024-04-02 17:37:16.000	2024-04-02 18:40:01.000
17	7	1	9.2	2024-10-30 18:02:02.000	2024-10-30 19:03:02.000
18	7	5	8.6	2024-11-12 17:04:16.000	2024-11-12 18:07:01.000
19	7	7	8.2	2024-04-07 17:12:16.000	2024-04-07 18:15:01.000
20	7	9	12.8	2024-05-30 18:38:02.000	2024-05-30 19:13:02.000
21	8	1	10.6	2024-10-30 18:02:02.000	2024-10-30 19:04:02.000
22	8	5	7.3	2024-11-12 17:05:16.000	2024-11-12 18:08:01.000
23	8	8	7.1	2024-05-08 17:13:16.000	2024-05-08 18:16:01.000
24	8	10	13	2024-06-03 18:39:02.000	2024-06-03 19:41:02.000
25	9	1	7.6	2024-07-04 18:44:02.000	2024-07-04 19:44:02.000
26	9	2	7.6	2024-08-04 18:40:02.000	2024-08-04 19:42:02.000
27	9	3	7.6	2024-09-04 18:40:02.000	2024-09-04 19:42:02.000
28	10	4	8.8	2024-07-06 18:41:02.000	2024-07-06 19:43:02.000
29	10	5	8.8	2024-08-07 18:41:02.000	2024-08-07 19:43:02.000
30	10	6	8.8	2024-09-08 18:41:02.000	2024-09-08 19:43:02.000
31	11	7	9.9	2024-09-06 17:42:16.000	2024-09-06 18:44:01.000
32	11	8	9.9	2024-09-07 17:42:16.000	2024-09-07 18:44:01.000
33	11	9	9.9	2024-09-08 17:42:16.000	2024-09-08 18:44:01.000
34	12	1	10.1	2024-10-07 17:43:16.000	2024-10-07 18:45:01.000
35	12	2	10.1	2024-10-08 17:43:16.000	2024-10-08 18:45:01.000
36	12	3	10.1	2024-10-09 17:43:16.000	2024-10-09 18:45:01.000
37	12	4	10.1	2024-10-10 17:43:16.000	2024-10-10 18:45:01.000
38	13	1	6.6	2024-11-07 17:44:16.000	2024-11-07 18:46:01.000
39	13	2	6.6	2024-11-08 17:44:16.000	2024-11-08 18:46:01.000
40	13	3	6.6	2024-11-09 17:44:16.000	2024-11-09 18:46:01.000
41	13	7	6.6	2024-11-10 17:44:16.000	2024-11-10 18:46:01.000

42	15	15	6.6	2024-11-15 17:20:16.000	2024-11-15 18:23:01.000
43	15	17	12.5	2024-01-10 18:46:02.000	2024-01-10 19:48:02.000
44	16	16	9.3	2024-12-16 17:21:16.000	2024-12-16 18:24:01.000
45	16	18	13.5	2024-02-11 18:47:02.000	2024-02-11 19:49:02.000
46	17	17	10.2	2024-01-17 18:22:02.000	2024-01-17 19:25:02.000
47	17	19	9.5	2024-03-12 18:48:02.000	2024-03-12 19:50:02.000
48	18	18	12.1	2024-02-18 18:23:02.000	2024-02-18 19:26:02.000
49	18	20	10.5	2024-04-13 18:49:02.000	2024-04-13 19:51:02.000
50	19	19	9.4	2024-03-19 18:24:02.000	2024-03-19 19:27:02.000
51	19	21	12.6	2024-05-14 17:50:16.000	2024-05-14 18:52:01.000
52	20	20	10.1	2024-04-20 18:25:02.000	2024-04-20 19:28:02.000
53	20	22	10.6	2024-06-15 17:51:16.000	2024-06-15 18:53:01.000
54	21	21	13.2	2024-05-21 17:26:16.000	2024-05-21 18:29:01.000
55	21	23	7.6	2024-07-16 17:52:16.000	2024-07-16 18:54:01.000
56	22	22	10.1	2024-06-22 17:27:16.000	2024-06-22 18:30:01.000
57	22	24	7.3	2024-08-17 17:53:16.000	2024-08-17 18:10:01.000
58	23	23	8	2024-07-23 17:28:16.000	2024-07-23 18:31:01.000
59	24	24	7.4	2024-08-24 17:29:16.000	2024-08-24 18:32:01.000
60	25	1	12.4	2024-09-25 18:30:02.000	2024-09-25 19:33:02.000
61	26	2	13.3	2024-10-26 18:31:02.000	2024-10-26 19:34:02.000

SELECT \* FROM Shop ;

	Shop_id	Category	Mall_id
1	1	fashion	1
2	2	furniture and decor	1
3	3	pet care	1
4	4	health and welln...	1
5	5	fashion	1
6	6	electronics	2
7	7	medical	2
8	8	books	2
9	9	fashion	3
10	10	education	3
11	11	furniture and decor	3
12	12	fashion	3
13	13	fashion	4
14	14	electronics	4
15	15	toys and hobbies	4
16	16	groceries	5
17	17	electronics	5
18	18	fashion	5
19	19	books	6
20	20	fashion	6
21	21	pet care	6
22	22	health and welln...	6
23	23	toys and hobbies	7
24	24	fashion	7
25	25	education	7
26	26	electronics	7
27	27	fashion	7
28	28	furniture and decor	7
29	29	fashion	8
30	30	fashion	8
31	31	fashion	1

SELECT \* FROM Voucher ;

	Voucher_id	Details	Date_issued	Date_expiring
1	1	Food Voucher	2024-04-01	2024-06-01
2	2	Grocery Voucher	2024-11-01	2024-12-01
3	3	Food Voucher	2024-03-01	2024-05-01
4	4	Food Voucher	2024-01-01	2024-03-01
5	5	Food Voucher	2024-01-01	2024-04-01
6	6	Grocery Voucher	2024-05-01	2024-07-01
7	7	Grocery Voucher	2024-06-01	2024-09-01
8	8	Shop Voucher	2024-08-01	2024-09-01
9	9	Food Voucher	2024-03-01	2024-04-01
10	10	Shop Voucher	2024-04-01	2024-05-01
11	11	Food Voucher	2024-01-01	2024-04-01
12	12	Book Voucher	2024-04-01	2024-05-01
13	13	Book Voucher	2024-02-01	2024-05-01
14	14	Food Voucher	2024-04-01	2024-06-01
15	15	Book Voucher	2024-05-01	2024-08-01
16	16	Shop Voucher	2024-07-01	2024-11-01
17	17	Food Voucher	2024-03-01	2024-05-01
18	18	Food Voucher	2024-03-01	2024-05-01
19	19	Food Voucher	2024-03-01	2024-05-01
20	20	Food Voucher	2024-03-01	2024-05-01
21	21	Food Voucher	2024-03-01	2024-04-01
22	22	Shop Voucher	2024-04-01	2024-05-01
23	23	Shop Voucher	2024-04-01	2024-05-01

SELECT \* FROM Purchase\_Voucher ;

	Voucher_id	Purchase_discount	User_id	Date_time_used
1	1	0.2	1	2024-10-30 20:02:02.000
2	5	0.4	5	2024-03-30 18:03:20.000
3	9	0.1	9	2024-03-29 15:04:30.000
4	13	0.5	13	2024-04-30 17:03:06.000
5	21	0.1	9	2024-03-29 15:04:30.000

SELECT \* FROM Dine\_Voucher ;

	Voucher_id	Cash_discount	User_id	Date_time_used
1	2	0	2	2024-11-30 14:06:10.000
2	6	0	6	2024-06-30 13:05:32.000
3	10	0	10	2024-04-29 08:04:14.000
4	14	0	14	2024-05-28 12:30:30.000
5	22	0	10	2024-04-29 08:04:14.000
6	23	0	10	2024-04-29 08:04:14.000

SELECT \* FROM Group\_Voucher;



	Voucher_id	Group_size	Group_discount	Date_time_used
1	3	4	0.2	2024-04-30 14:06:10.000
2	7	6	0.2	2024-08-28 13:07:11.000
3	11	10	0.5	2024-03-29 17:30:12.000
4	15	3	0.2	2024-07-30 11:09:12.000
5	17	4	0.2	2024-04-30 14:06:10.000
6	18	4	0.2	2024-04-30 14:06:10.000
7	19	4	0.2	2024-04-30 14:06:10.000
8	20	4	0.2	2024-04-30 14:06:10.000

SELECT \* FROM Package\_Voucher;

	Voucher_id	Package_discount
1	4	0.3
2	8	0.2
3	12	0.4
4	16	0.1

SELECT \* FROM Recommendation;

	Recommendation_id	Mall_id	Outlet_id	DayPackage_id	Voucher_id	Date_expired	Date_issued
1	1	1	1	1	1	2024-06-01	2024-04-01
2	2	3	3	1	2	2024-07-01	2024-02-01
3	3	5	1	2	3	2024-12-01	2024-03-01
4	4	6	4	2	4	2024-11-01	2024-04-01
5	5	1	5	4	5	2024-10-01	2024-01-01

SELECT \* FROM UserAccount\_Use\_Recommendation;

	User_id	Recommendation_id
1	1	1
2	2	1
3	5	2
4	6	2
5	13	3
6	18	4
7	20	5

SELECT \* FROM Mall;

	Mall_id	Company_ID	Address_of_mall	NumShops
1	1	1	1 HarbourFront Walk, Singapore 098585	102
2	2	1	2 Orchard Tum, Singapore 238801	70
3	3	1	180 Kitchener Rd, Singapore 208539	65
4	4	2	290 Orchard Rd, Singapore 238859	47
5	5	2	2 Jurong East Street 21, Singapore 60...	36
6	6	3	1 Stadium Pl, Singapore 397628	57
7	7	4	1 Jurong West Central 2, Singapore 64...	64
8	8	4	6 Raffles Blvd, Singapore 039594	46

SELECT \* FROM Mall\_Has\_DayPackage;

	DayPackage_id	Mall_id
1	1	1
2	1	2
3	1	3
4	1	4
5	2	5
6	2	6
7	2	7
8	2	8
9	3	9
10	3	10
11	3	11
12	3	12
13	4	13
14	4	14
15	4	15
16	4	16
17	5	17
18	5	18
19	5	19
20	5	20
21	6	21
22	6	22
23	6	23
24	6	24
25	7	25
26	7	26
27	7	27
28	7	28
29	8	29
30	8	30
31	8	31
32	8	32
33	9	33
34	9	34
35	9	35
36	9	36
37	10	37

38	10	38
39	10	39
40	10	40
41	11	9
42	11	10
43	11	11
44	11	12
45	12	9
46	12	10
47	12	11
48	12	12
49	13	9
50	13	10
51	13	11
52	13	12
53	14	9
54	14	10
55	14	11
56	14	12
57	15	33
58	15	34
59	15	35
60	15	36
61	16	37
62	16	38
63	16	39
64	16	40
65	17	37
66	17	38
67	17	39
68	17	40

SELECT \* FROM Day\_Package;

	DayPackage_id	User_id	Voucher_id	Description
1	1	1	1	Bugis+, ION Orchard, JEM, West Mall
2	2	3	2	Buangkok Square, Junction 8, Lucky Plaza, Marina...
3	3	6	3	City Square Mall, Funan, Liat Towers, Marina Square
4	4	7	4	Beauty World Centre, Beauty World Plaza, Greenrid...
5	5	10	5	Bukit Panjang Plaza, Bukit Timah Plaza, Hillion Mall,...
6	6	14	6	Fajar Shopping Centre, JCube, Lot One, Rail Mall
7	7	15	7	CityLink Mall, HDB Hub, Raffles City, Plaza Singapura
8	8	17	8	Mustafa Shopping Centre, Sim Lim Square, Square ...
9	9	19	9	Compass One, NEX, The Seletar Mall, Waterway P...
10	10	20	10	Bugis Junction, Cathay Cineleisure Orchard, Millenia...
11	11	5	17	City Square Mall, Funan, Liat Towers, Marina Square
12	12	7	18	City Square Mall, Funan, Liat Towers, Marina Square
13	13	8	19	City Square Mall, Funan, Liat Towers, Marina Square
14	14	15	20	City Square Mall, Funan, Liat Towers, Marina Square
15	15	18	21	Compass One, NEX, The Seletar Mall, Waterway P...
16	16	23	22	Bugis Junction, Cathay Cineleisure Orchard, Millenia...
17	17	24	23	Bugis Junction, Cathay Cineleisure Orchard, Millenia...

SELECT \* FROM Day\_Package\_Has\_RestaurantOutlet;

	DayPackage_id	Outlet_id
1	1	1
2	1	2
3	1	3
4	2	1
5	2	4
6	3	1
7	3	3
8	4	5
9	5	4
10	6	1
11	7	5
12	8	1
13	9	3
14	10	2
15	11	1
16	11	3
17	12	1
18	12	3
19	13	1
20	13	3
21	14	1
22	14	3
23	15	3
24	16	2
25	17	2

SELECT \* FROM Mall\_Mgmt\_Company;

	Company_id	Company_Address
1	1	298 Tiong Bahru Rd, #08-01 Central Plaza, Singapo...
2	2	168 Robinson Rd, #30-01 Capital Tower, Singapore...
3	3	480 Lor 6 Toa Payoh, Singapore 310480
4	4	9 Penang Road, Singapore 238459

SELECT \* FROM Restaurant\_Chain;

	RestaurantChain_id	RestaurantChain_Address
1	1	123 Sesame Street
2	2	512 Comet Drive
3	3	728 Edgeroad Plains
4	4	522 Beverly Road
5	5	222 Piles Avenue
6	6	922 Coral Street

## 4. Description of any additional efforts made(optional)

### Auto incrementing primary keys

Usage of keyword "IDENTITY(1,1)" to auto-increment primary keys. The parameters indicate the starting index and the number to increment by. This was used for some tables like User\_Account and Complaint