UUM

Universiti Utara Malaysia

STID3014 DATABASE SYSTEM AND INFORMATION RETRIEVAL (F)

SEMESTER/SESSION A222

GROUP 4

SQL PROJECT REPORT

PREPARED FOR: DR SYAHIDA BINTI HASSAN

PREPARED BY:

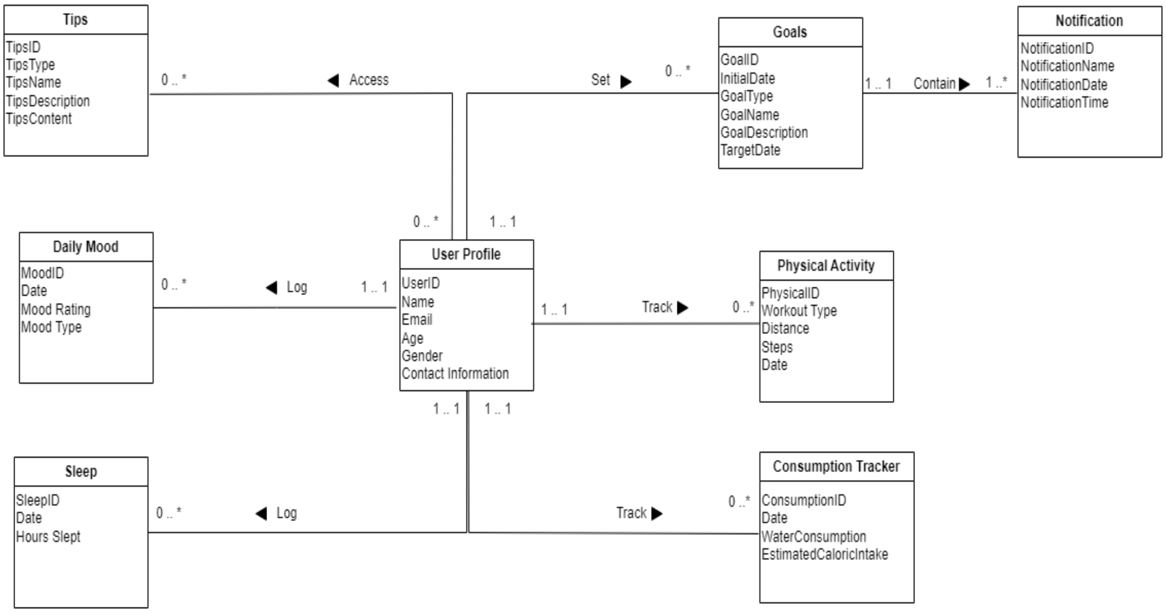
NO.	MATRIC NO.	NAME
1	278688	YAP JIA QING
2	279018	VINCENT BEH HUA EIK
3	279021	POON WAI KIT
4	279231	YAP YUN LOON

Entity Relational Diagram

Link to ERD: ERD Link

Assumptions:

- Each User can only register for one profile.
- For some entities, attributes are less than 5.



Relational Model

User_Profile (UserID, Name, Email, Age, Gender, Contact Information)

PK: UserID

Goals (GoalID, InitialDate, GoalType, GoalName, GoalDescription, TargetDate, UserID)

PK: GoalID

FK: UserID references User_Profile (UserID)

Notification (NotificationID, NotificationName, NotificationDate, NotificationTime, GoalID)

PK: NotificationID

FK: GoalID references Goals (GoalID)

Physical_Activity (PhysicalID, WorkoutType, Distance, Steps, Date, UserID)

PK: PhysicalID

FK: UserID references User_Profile (UserID)

Consumption_Tracker (ConsumptionID, Date, WaterConsumption, EstimatedCaloricIntake, UserID)

PK: ConsumptionID

FK: UserID references User_Profile (UserID)

Sleep (SleepID, Date, HoursSlept, UserID)

PK: SleepID

FK: UserID references User_Profile (UserID)

Daily_Mood (MoodID, Date, MoodRating, MoodType,UserID)

PK: MoodID

FK: UserID references User_Profile (UserID)

Tips (TipsID, TipsType, TipsName, TipsDescription, TipsContent)

PK: TipsID

User_Tips (UserID, TipsID)

PK: UserID, TipsID

FK: UserID references User_Profile (UserID)

FK: TipsID references Tips (TipsID)

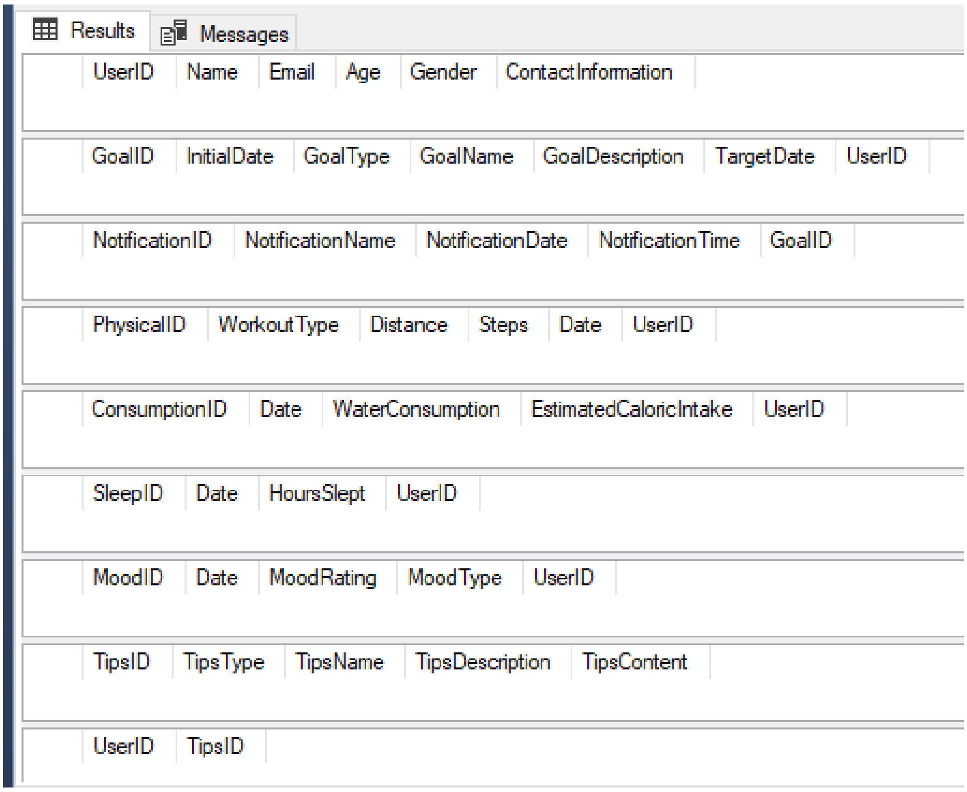
Queries

Create Tables

```
-- Create User Profile table
CREATE TABLE User_Profile (
   UserID VARCHAR(6) PRIMARY KEY,
   Name VARCHAR(255) NOT NULL,
   Email VARCHAR(255) NOT NULL,
   Age INT NOT NULL,
   Gender  CHAR NOT NULL
        CONSTRAINT User_sexCHK
        CHECK (Gender IN ('M','F')),
   ContactInformation VARCHAR(255) NOT NULL
);

-- Create Goals table
CREATE TABLE Goals (
   GoalID VARCHAR (8) PRIMARY KEY,
   InitialDate DATE NOT NULL,
   GoalType VARCHAR(50) NOT NULL,
   GoalName VARCHAR(255) NOT NULL,
   GoalDescription VARCHAR(1000) NOT NULL,
   TargetDate DATE NOT NULL,
   UserID VARCHAR(6) NOT NULL,
   FOREIGN KEY (UserID) REFERENCES User_Profile (UserID)
);
```

```
-- Create Notification table
CREATE TABLE Notification (
   NotificationID VARCHAR(8) PRIMARY KEY,
   NotificationName VARCHAR(255) NOT NULL,
   NotificationDate DATE NOT NULL,
   NotificationTime TIME NOT NULL,
   GoalID VARCHAR (8) NOT NULL,
   FOREIGN KEY (GoalID) REFERENCES Goals (GoalID)
);

-- Create Physical_Activity table
CREATE TABLE Physical_Activity (
   PhysicalID VARCHAR(8) PRIMARY KEY,
   WorkoutType CHAR NOT NULL
           CONSTRAINT User_actCHK
           CHECK (WorkoutType IN ('W','R')),
   Distance FLOAT,
   Steps INT,
   Date DATE NOT NULL,
   UserID VARCHAR(6) NOT NULL,
   FOREIGN KEY (UserID) REFERENCES User_Profile (UserID)
);

-- Create Consumption_Tracker table
CREATE TABLE Consumption_Tracker (
   ConsumptionID VARCHAR(8) PRIMARY KEY,
   Date DATE NOT NULL,
   WaterConsumption FLOAT NOT NULL,
   EstimatedCaloricIntake INT NOT NULL,
   UserID VARCHAR(6) NOT NULL,
   FOREIGN KEY (UserID) REFERENCES User_Profile (UserID)
);

-- Create Sleep table
CREATE TABLE Sleep (
   SleepID VARCHAR(8) PRIMARY KEY,
   Date DATE NOT NULL,
   HoursSlept FLOAT NOT NULL,
   UserID VARCHAR(6) NOT NULL,
   FOREIGN KEY (UserID) REFERENCES User_Profile (UserID)
);

-- Create Daily_Mood table
CREATE TABLE Daily_Mood (
   MoodID VARCHAR(8) PRIMARY KEY,
   Date DATE NOT NULL,
   MoodRating INT CHECK (MoodRating BETWEEN 1 AND 10) NOT NULL,
   MoodType CHAR NOT NULL
         CONSTRAINT User_moodCHK
         CHECK (MoodType IN ('B','G','N')),
   UserID VARCHAR(6) NOT NULL,
   FOREIGN KEY (UserID) REFERENCES User_Profile (UserID)
);

-- Create Tips table
CREATE TABLE Tips (
   TipsID VARCHAR(8) PRIMARY KEY,
   TipsType VARCHAR(50) NOT NULL,
   TipsName VARCHAR(255) NOT NULL,
   TipsDescription VARCHAR(1000) NOT NULL,
   TipsContent VARCHAR(MAX) NOT NULL
);
```

```
-- Create User_Tips table
CREATE TABLE User_Tips (
    UserID VARCHAR(6),
    TipsID VARCHAR(8),
    PRIMARY KEY (UserID, TipsID),
    FOREIGN KEY (UserID) REFERENCES User_Profile (UserID),
    FOREIGN KEY (TipsID) REFERENCES Tips (TipsID)
);
```



UserID	Name	Email	Age	Gender	ContactInfomation

GoalID	InitialDate	GoalType	GoalName	GoalDescription	TargetDate	UserID

NotificationID	NotificationName	NotificationDate	NotificationTime	GoalID

PhysicalID	WorkoutType	Distance	Steps	Date	UserID

ConsumptionID	Date	WaterConsumption	EstimatedCaloricIntake	UserID

SleepID	Date	HoursSlept	UserID

MoodID	Date	MoodRating	MoodType	UserID

TipsID	TipsType	TipsName	TipsDescription	TipsContent

UserID	TipsID

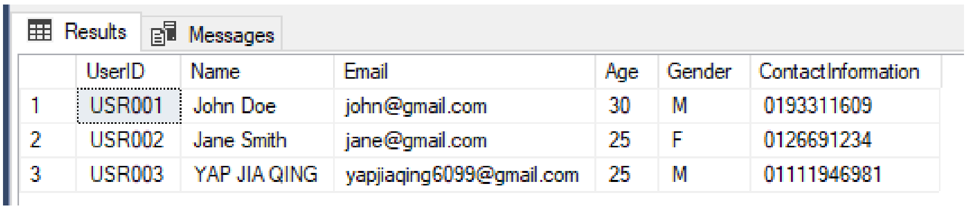
Alter Table

select * from User_Profile

ALTER TABLE User_Profile

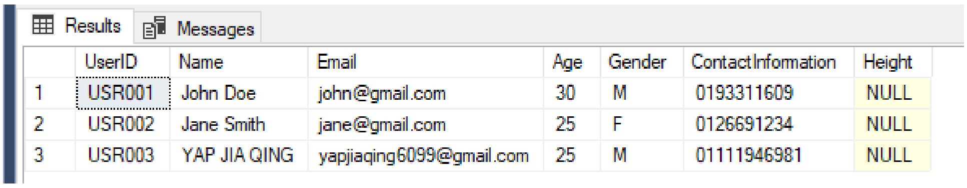
ADD Height FLOAT;

Before:



	UserID	Name	Email	Age	Gender	ContactInformation
1	USR001	John Doe	john@gmail.com	30	M	0193311609
2	USR002	Jane Smith	jane@gmail.com	25	F	0126691234
3	USR003	YAP JIA QING	yapjiaqing6099@gmail.com	25	M	01111946981

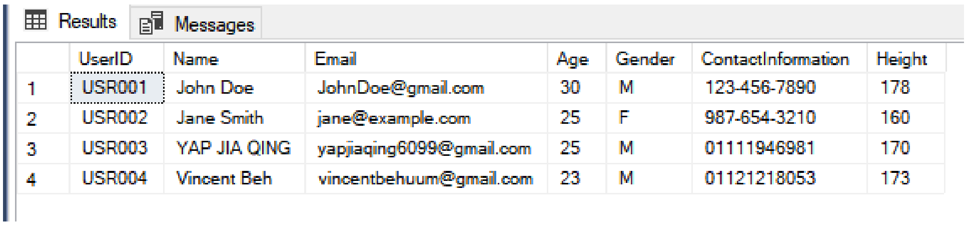
After:



	UserID	Name	Email	Age	Gender	ContactInformation	Height
1	USR001	John Doe	john@gmail.com	30	M	0193311609	NULL
2	USR002	Jane Smith	jane@gmail.com	25	F	0126691234	NULL
3	USR003	YAP JIA QING	yapjiaqing6099@gmail.com	25	M	01111946981	NULL

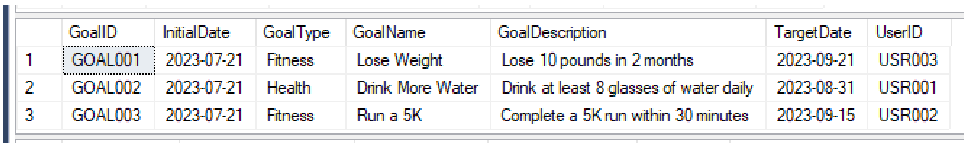
Insert Data

```
INSERT INTO User_Profile (UserID, Name, Email, Age, Gender, ContactInformation,
Height)

VALUES

 ('USR001', 'John Doe', 'john@gmail.com', 30, 'M', '0193311609',’178’),

('USR002', 'Jane Smith', 'jane@gmail.com', 25, 'F', '0126691234',’160’),

(‘USR003', 'YAP JIA QING' , 'yapjiaqing6099@gmail.com' ,
23 ,'M' ,'01111946981' , ’170’ ),

(‘USR004', 'Vincent Beh' , 'vincentbehuum@gmail.com' ,
23 ,'M' ,01121218053, ’173’ );
```



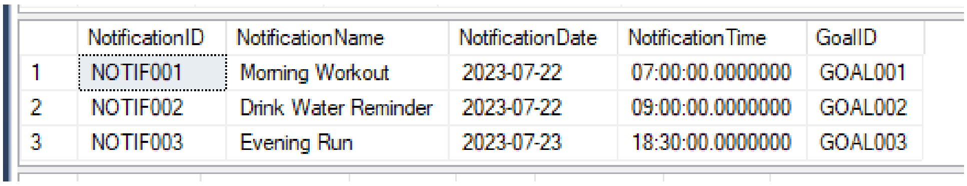
	UserID	Name	Email	Age	Gender	ContactInformation	Height
1	USR001	John Doe	JohnDoe@gmail.com	30	M	123-456-7890	178
2	USR002	Jane Smith	jane@example.com	25	F	987-654-3210	160
3	USR003	YAP JIA QING	yapjiaqing6099@gmail.com	25	M	01111946981	170
4	USR004	Vincent Beh	vincentbehuum@gmail.com	23	M	01121218053	173

```
INSERT INTO Goals (GoalID, InitialDate, GoalType, GoalName, GoalDescription,
TargetDate, UserID)

VALUES

    ('GOAL001', '2023-07-21', 'Fitness', 'Lose Weight', 'Lose 10 pounds in 2 months',
'2023-09-21', 'USR003'),

    ('GOAL002', '2023-07-21', 'Health', 'Drink More Water', 'Drink at least 8 glasses of
water daily', '2023-08-31', 'USR001'),

    ('GOAL003', '2023-07-21', 'Fitness', 'Run a 5K', 'Complete a 5K run within 30
minutes', '2023-09-15', 'USR002');
```



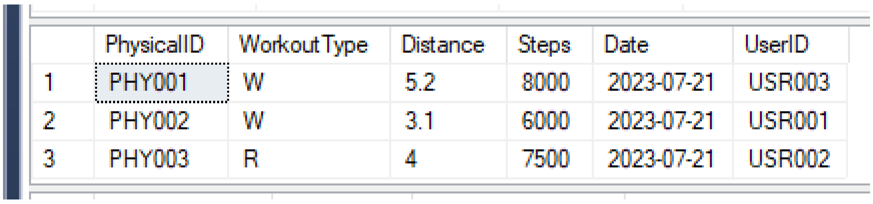
	GoalID	InitialDate	GoalType	GoalName	GoalDescription	TargetDate	UserID
1	GOAL001	2023-07-21	Fitness	Lose Weight	Lose 10 pounds in 2 months	2023-09-21	USR003
2	GOAL002	2023-07-21	Health	Drink More Water	Drink at least 8 glasses of water daily	2023-08-31	USR001
3	GOAL003	2023-07-21	Fitness	Run a 5K	Complete a 5K run within 30 minutes	2023-09-15	USR002

```
INSERT INTO Notification (NotificationID, NotificationName, NotificationDate, NotificationTime, GoalID)
VALUES
    ('NOTIF001', 'Morning Workout', '2023-07-22', '07:00:00', 'GOAL001'),
    ('NOTIF002', 'Drink Water Reminder', '2023-07-22', '09:00:00', 'GOAL002'),
    ('NOTIF003', 'Evening Run', '2023-07-23', '18:30:00', 'GOAL003');
```



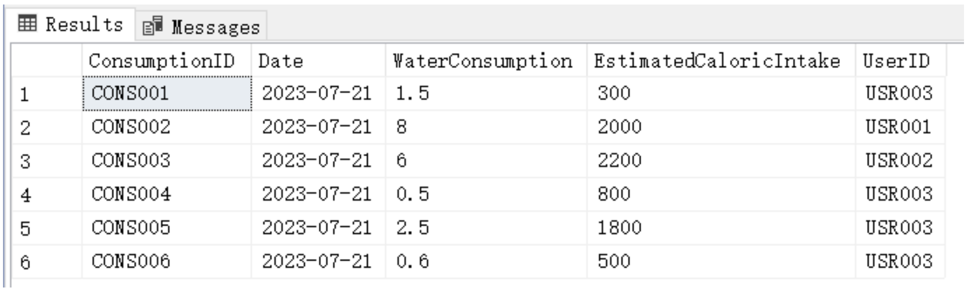
	NotificationID	NotificationName	NotificationDate	NotificationTime	GoalID
1	NOTIF001	Morning Workout	2023-07-22	07:00:00.0000000	GOAL001
2	NOTIF002	Drink Water Reminder	2023-07-22	09:00:00.0000000	GOAL002
3	NOTIF003	Evening Run	2023-07-23	18:30:00.0000000	GOAL003

```
-- Insert physical activity records for each user
INSERT INTO Physical_Activity (PhysicalID, WorkoutType, Distance, Steps, Date, UserID)
VALUES
    ('PHY001', 'W', 5.2, 8000, '2023-07-21', 'USR003'),
    ('PHY002', 'W', 3.1, 6000, '2023-07-21', 'USR001'),
    ('PHY003', 'R', 4.0, 7500, '2023-07-21', 'USR002');
```



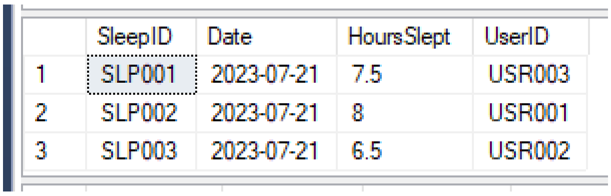
	PhysicalID	WorkoutType	Distance	Steps	Date	UserID
1	PHY001	W	5.2	8000	2023-07-21	USR003
2	PHY002	W	3.1	6000	2023-07-21	USR001
3	PHY003	R	4	7500	2023-07-21	USR002

```
-- Insert consumption tracker records for each user

INSERT INTO Consumption_Tracker (ConsumptionID, Date, WaterConsumption, EstimatedCaloricIntake, UserID)

VALUES

    ('CONS001', '2023-07-21', 2.5, 1800, 'USR003'),

    ('CONS002', '2023-07-21', 8.0, 2000, 'USR001'),

    ('CONS003', '2023-07-21', 6.0, 2200, 'USR002'),

    ('CONS004', '2023-07-21', 0.5, 800, 'USR003'),

    ('CONS005', '2023-07-21', 2.5, 1800, 'USR003'),

    ('CONS006', '2023-07-21', 0.6, 500, 'USR003');
```



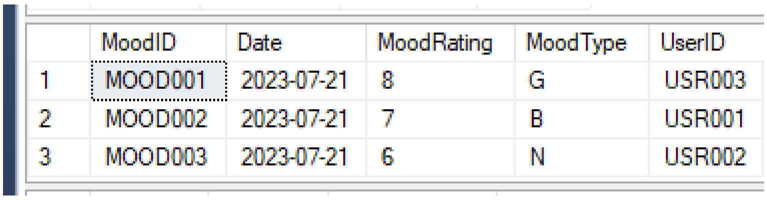
	ConsumptionID	Date	WaterConsumption	EstimatedCaloricIntake	UserID
1	CONS001	2023-07-21	1.5	300	USR003
2	CONS002	2023-07-21	8	2000	USR001
3	CONS003	2023-07-21	6	2200	USR002
4	CONS004	2023-07-21	0.5	800	USR003
5	CONS005	2023-07-21	2.5	1800	USR003
6	CONS006	2023-07-21	0.6	500	USR003

```
INSERT INTO Sleep (SleepID, Date, HoursSlept, UserID)

VALUES

    ('SLP001', '2023-07-21', 7.5, 'USR003'),

    ('SLP002', '2023-07-21', 8.0, 'USR001'),

    ('SLP003', '2023-07-21', 6.5, 'USR002');
```



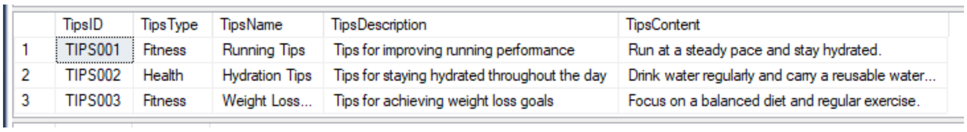
	SleepID	Date	HoursSlept	UserID
1	SLP001	2023-07-21	7.5	USR003
2	SLP002	2023-07-21	8	USR001
3	SLP003	2023-07-21	6.5	USR002

```
-- Insert daily mood records for each user
INSERT INTO Daily_Mood (MoodID, Date, MoodRating, MoodType, UserID)
VALUES
    ('MOOD001', '2023-07-21', 8, 'G', 'USR003'),
    ('MOOD002', '2023-07-21', 7, 'B', 'USR001'),
    ('MOOD003', '2023-07-21', 6, 'N', 'USR002');
```



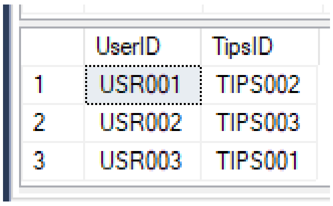
	MoodID	Date	MoodRating	MoodType	UserID
1	MOOD001	2023-07-21	8	G	USR003
2	MOOD002	2023-07-21	7	B	USR001
3	MOOD003	2023-07-21	6	N	USR002

```
-- Insert tips
INSERT INTO Tips (TipsID, TipsType, TipsName, TipsDescription, TipsContent)
VALUES
    ('TIPS001', 'Fitness', 'Running Tips', 'Tips for improving running performance', 'Run at a steady pace and stay hydrated.'),
    ('TIPS002', 'Health', 'Hydration Tips', 'Tips for staying hydrated throughout the day', 'Drink water regularly and carry a reusable water bottle.'),
    ('TIPS003', 'Fitness', 'Weight Loss Tips', 'Tips for achieving weight loss goals', 'Focus on a balanced diet and regular exercise.');
```



	TipsID	TipsType	TipsName	TipsDescription	TipsContent
1	TIPS001	Fitness	Running Tips	Tips for improving running performance	Run at a steady pace and stay hydrated.
2	TIPS002	Health	Hydration Tips	Tips for staying hydrated throughout the day	Drink water regularly and carry a reusable water...
3	TIPS003	Fitness	Weight Loss...	Tips for achieving weight loss goals	Focus on a balanced diet and regular exercise.

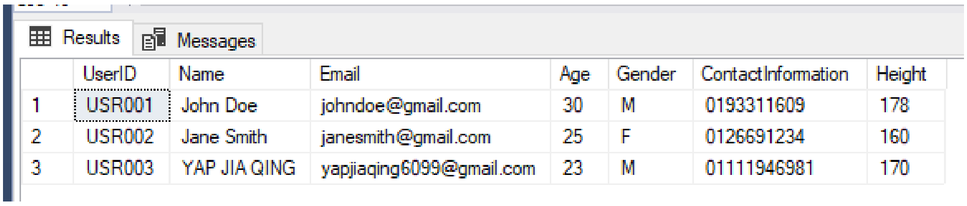
```
-- Assign tips to users
INSERT INTO User_Tips (UserID, TipsID)
VALUES
    ('USR003', 'TIPS001'),
    ('USR001', 'TIPS002'),
    ('USR002', 'TIPS003');
```



	UserID	TipsID
1	USR001	TIPS002
2	USR002	TIPS003
3	USR003	TIPS001

Update Data

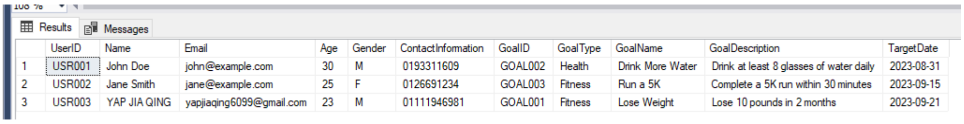
```
UPDATE User_Profile
SET Email = 'johndoe@gmail.com'
WHERE UserID = 'USR001';

UPDATE User_Profile
SET Email = 'janesmith@gmail.com'
WHERE UserID = 'USR002';
```



	UserID	Name	Email	Age	Gender	ContactInformation	Height
1	USR001	John Doe	johndoe@gmail.com	30	M	0193311609	178
2	USR002	Jane Smith	janesmith@gmail.com	25	F	0126691234	160
3	USR003	YAP JIA QING	yapjiaqing6099@gmail.com	23	M	01111946981	170

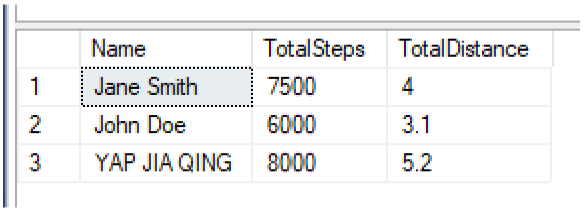
Create View

```
CREATE VIEW UserProfileWithGoals AS
SELECT u.UserID, u.Name, u.Email, u.Age, u.Gender, u.ContactInformation,
    g.GoalID, g.GoalType, g.GoalName, g.GoalDescription, g.TargetDate
FROM User_Profile u
LEFT JOIN Goals g ON u.UserID = g.UserID;
```



	UserID	Name	Email	Age	Gender	ContactInformation	GoalID	GoalType	GoalName	GoalDescription	TargetDate
1	USR001	John Doe	john@example.com	30	M	0193311609	GOAL002	Health	Drink More Water	Drink at least 8 glasses of water daily	2023-08-31
2	USR002	Jane Smith	jane@example.com	25	F	0126691234	GOAL003	Fitness	Run a 5K	Complete a 5K run within 30 minutes	2023-09-15
3	USR003	YAP JIA QING	yapjiaqing6099@gmail.com	23	M	01111946981	GOAL001	Fitness	Lose Weight	Lose 10 pounds in 2 months	2023-09-21

```
CREATE VIEW UserPhysicalActivitySummary AS
SELECT u.Name,
    SUM(Steps) AS TotalSteps,
    SUM(Distance) AS TotalDistance
FROM User_Profile u
LEFT JOIN Physical_Activity p ON u.UserID = p.UserID
GROUP BY u.Name;
```



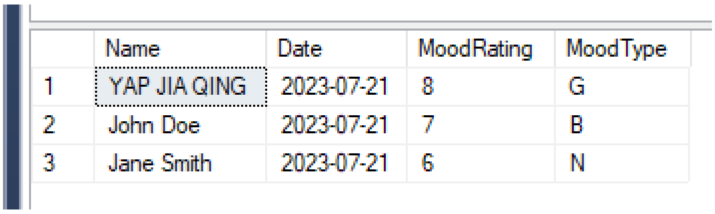
	Name	TotalSteps	TotalDistance
1	Jane Smith	7500	4
2	John Doe	6000	3.1
3	YAP JIA QING	8000	5.2

CREATE VIEW UserMoodView AS

SELECT u.Name,m.Date, m.MoodRating, m.MoodType

FROM User_Profile u

JOIN Daily_Mood m ON u.UserID = m.UserID;



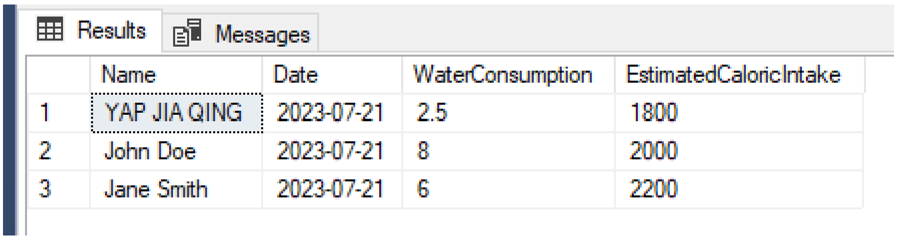
	Name	Date	MoodRating	MoodType
1	YAP JIA QING	2023-07-21	8	G
2	John Doe	2023-07-21	7	B
3	Jane Smith	2023-07-21	6	N

CREATE VIEW UserConsumptionView AS

SELECT u.Name, c.Date, c.WaterConsumption, c.EstimatedCaloricIntake

FROM User_Profile u

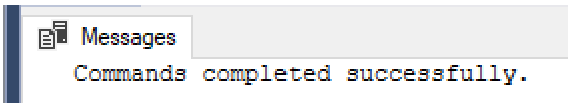
JOIN Consumption_Tracker c ON u.UserID = c.UserID;



	Name	Date	WaterConsumption	EstimatedCaloricIntake
1	YAP JIA QING	2023-07-21	2.5	1800
2	John Doe	2023-07-21	8	2000
3	Jane Smith	2023-07-21	6	2200

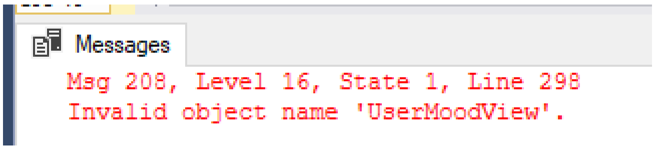
Drop View

DROP VIEW UserMoodView (Example)



```
Commands completed successfully.
```

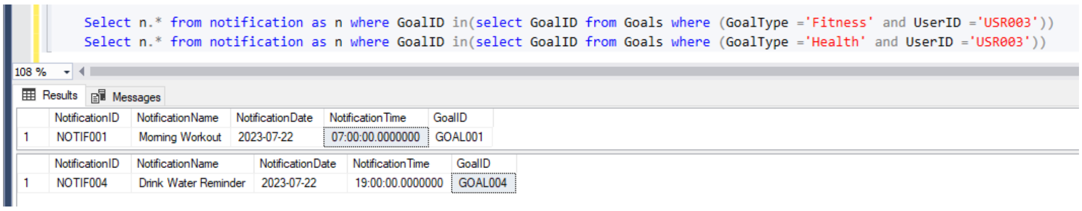
select * from UserMoodView



```
Msg 208, Level 16, State 1, Line 298
Invalid object name 'UserMoodView'.
```

Create subqueries

1. Select n.* from notification as n where GoalID in(select GoalID from Goals where (GoalType ='@GoalType' and UserID ='@UserID'))



```
Select n.* from notification as n where GoalID in(select GoalID from Goals where (GoalType ='Fitness' and UserID ='USR003'))
Select n.* from notification as n where GoalID in(select GoalID from Goals where (GoalType ='Health' and UserID ='USR003'))
```

	NotificationID	NotificationName	NotificationDate	NotificationTime	GoalID
1	NOTIF001	Morning Workout	2023-07-22	07:00:00.0000000	GOAL001

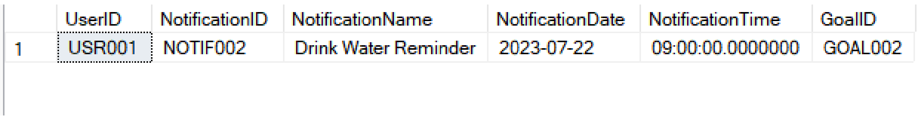
	NotificationID	NotificationName	NotificationDate	NotificationTime	GoalID
1	NOTIF004	Drink Water Reminder	2023-07-22	19:00:00.0000000	GOAL004

Justification: This query is used for the user to check the notification been created by them for notify the specific goal. As the query executed, the user can see the notification that been setting to notify specific goal to remind user to do the goal at specific time.

2. SELECT UserID, SUM(WaterConsumption) AS TotalWaterConsumed

   FROM Consumption_Tracker

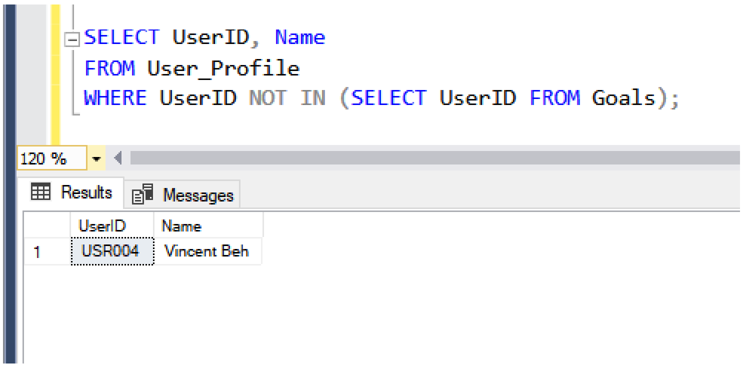
   GROUP BY UserID;



	UserID	NotificationID	NotificationName	NotificationDate	NotificationTime	GoalID
1	USR001	NOTIF002	Drink Water Reminder	2023-07-22	09:00:00.0000000	GOAL002

Justification: This subquery calculates the total water consumption for each user from the Consumption_Tracker table, providing valuable insights into individual hydration habits for health analysis and personalized recommendations.

3. SELECT UserID, Name FROM User_Profile WHERE UserID NOT IN (SELECT UserID FROM Goals);



```
SELECT UserID, Name
FROM User_Profile
WHERE UserID NOT IN (SELECT UserID FROM Goals);
```

	UserID	Name
1	USR004	Vincent Beh

Justification: This query retrieves UserID and Name of the user which does not have any set goals in the app. This information will allow the business to do targeted marketing/introduction to the Goals function of the application. Doing that will hopefully bring on more users to use said function.

4.

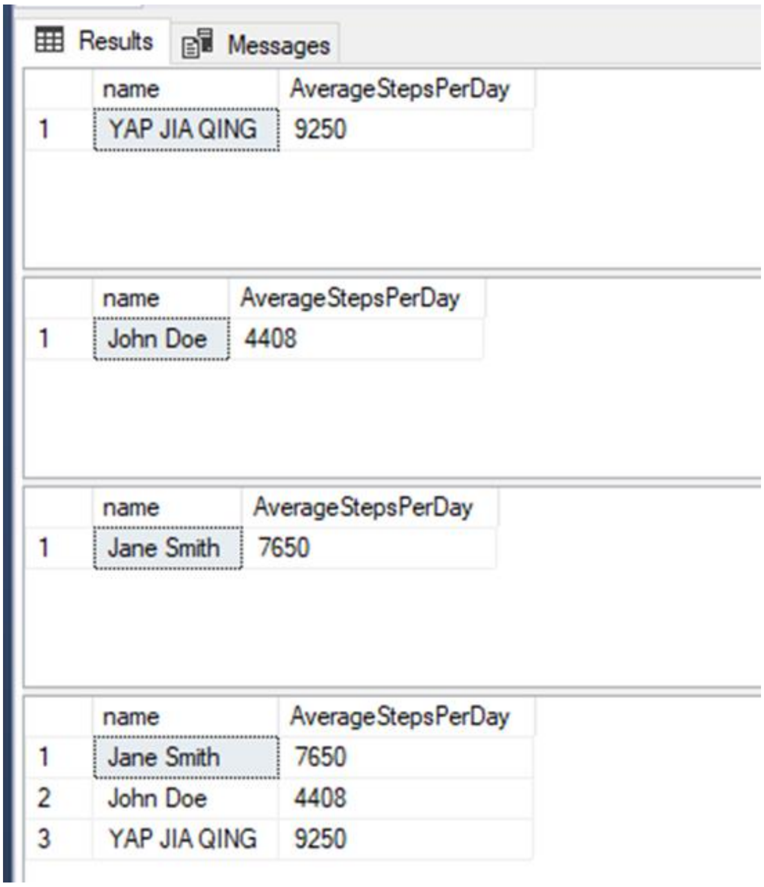
Select name ,avg(Steps) AS AverageStepsPerDay from User_Profile u, Physical_Activity p

where p.UserID IN(Select u.UserID from User_Profile where u.UserID = 'USR003')
Group by name;

Select name ,avg(Steps) AS AverageStepsPerDay from User_Profile u, Physical_Activity p
where p.UserID IN(Select u.UserID from User_Profile where u.UserID = 'USR001')
Group by name;

Select name ,avg(Steps) AS AverageStepsPerDay from User_Profile u, Physical_Activity p
where p.UserID IN(Select u.UserID from User_Profile where u.UserID = 'USR002')
Group by name;

Select name ,avg(Steps) AS AverageStepsPerDay from User_Profile u, Physical_Activity p
where p.UserID IN(Select u.UserID from User_Profile)
Group by name;



	name	AverageStepsPerDay
1	YAP JIA QING	9250

	name	AverageStepsPerDay
1	John Doe	4408

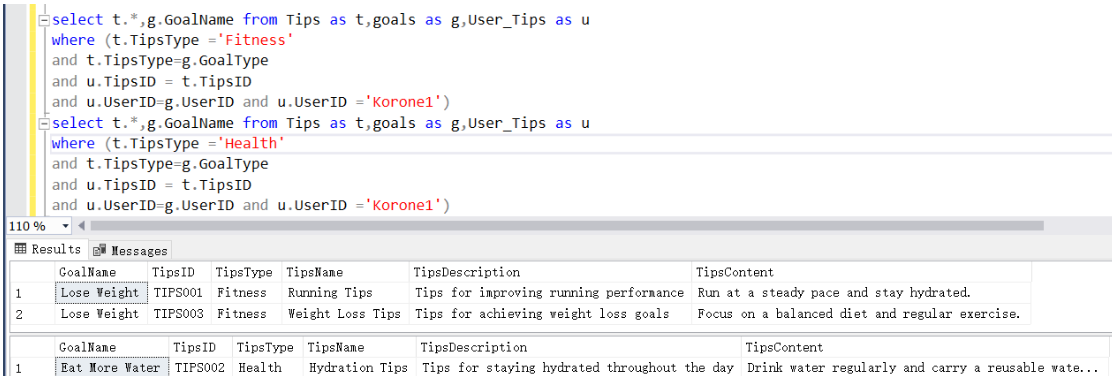
	name	AverageStepsPerDay
1	Jane Smith	7650

	name	AverageStepsPerDay
1	Jane Smith	7650
2	John Doe	4408
3	YAP JIA QING	9250

Justification: The purpose of this query is to calculate the user's average steps taken per day so the user may keep track of it to monitor their daily average steps that allows them to gauge their physical activity levels or assess progress and make necessary adjustments to their lifestyle.

Create Multi-table queries

1. Select t.*,g.GoalName from Tips as t,goals as g,User_Tips as u where (t.TipsType ='@GoalType' and t.TipsType=g.GoalType and u.TipsID = t.TipsID and u.UserID=g.UserID and u.UserID ='@UserID')



```
select t.*,g.GoalName from Tips as t,goals as g,User_Tips as u
where (t.TipsType ='Fitness'
and t.TipsType=g.GoalType
and u.TipsID = t.TipsID
and u.UserID=g.UserID and u.UserID ='Korone1')
select t.*,g.GoalName from Tips as t,goals as g,User_Tips as u
where (t.TipsType ='Health'
and t.TipsType=g.GoalType
and u.TipsID = t.TipsID
and u.UserID=g.UserID and u.UserID ='Korone1')
```

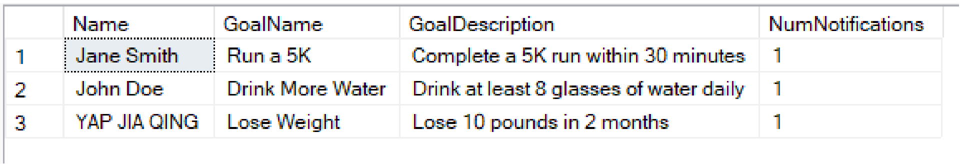
	GoalName	TipsID	TipsType	TipsName	TipsDescription	TipsContent
1	Lose Weight	TIPS001	Fitness	Running Tips	Tips for improving running performance	Run at a steady pace and stay hydrated.
2	Lose Weight	TIPS003	Fitness	Weight Loss Tips	Tips for achieving weight loss goals	Focus on a balanced diet and regular exercise.

	GoalName	TipsID	TipsType	TipsName	TipsDescription	TipsContent
1	Eat More Water	TIPS002	Health	Hydration Tips	Tips for staying hydrated throughout the day	Drink water regularly and carry a reusable wate...

Justification: This query is created in the application for the user to check and find the tips that is related to specific goal type. As the user adjust the goal type, the tips that have the same type as the goals will be display to the user to refer again.

2.

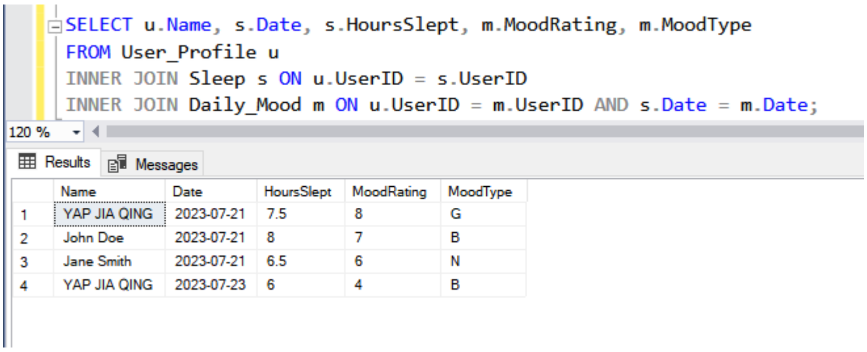
```
SELECT u.Name, g.GoalName, g.GoalDescription, COUNT(n.NotificationID) AS NumNotifications

FROM User_Profile u

LEFT JOIN Goals g ON u.UserID = g.UserID

LEFT JOIN Notification n ON g.GoalID = n.GoalID

GROUP BY u.Name, g.GoalName, g.GoalDescription;
```



	Name	GoalName	GoalDescription	NumNotifications
1	Jane Smith	Run a 5K	Complete a 5K run within 30 minutes	1
2	John Doe	Drink More Water	Drink at least 8 glasses of water daily	1
3	YAP JIA QING	Lose Weight	Lose 10 pounds in 2 months	1

Justification: The purpose of this query is to displays the 'Name' of the user, the 'GoalName' and 'GoalDescription' of their goals, along with the total number of notifications associated with each specific goal. The Count function is an aggregate function that calculates the total number of notifications associated with each specific goal for each user. The result provides a clear breakdown of each user's goals, their respective descriptions, and the number of notifications received for each goal. This information can be valuable for monitoring user engagement and progress towards their set goals.

3. SELECT u.Name, s.Date, s.HoursSlept, m.MoodRating, m.MoodType FROM User_Profile u INNER JOIN Sleep s ON u.UserID = s.UserID INNER JOIN Daily_Mood m ON u.UserID = m.UserID AND s.Date = m.Date;

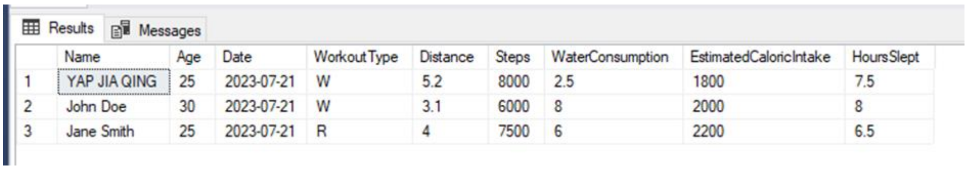


```
SELECT u.Name, s.Date, s.HoursSlept, m.MoodRating, m.MoodType
FROM User_Profile u
INNER JOIN Sleep s ON u.UserID = s.UserID
INNER JOIN Daily_Mood m ON u.UserID = m.UserID AND s.Date = m.Date;
```

	Name	Date	HoursSlept	MoodRating	MoodType
1	YAP JIA QING	2023-07-21	7.5	8	G
2	John Doe	2023-07-21	8	7	B
3	Jane Smith	2023-07-21	6.5	6	N
4	YAP JIA QING	2023-07-23	6	4	B

Justification: The purpose of this query is to obtain data that allows the health app business to analyze the correlation between users' sleep duration and their daily mood ratings. By combining information from the Sleep and Daily_Mood tables, the query ensures that the sleep and mood data are collected on the same days for each user.

4. SELECT u.Name, u.Age, p.Date, p.WorkoutType, p.Distance, p.Steps, c.WaterConsumption,c.EstimatedCaloricIntake, s.HoursSlept FROM User_Profile u, Physical_Activity p, Consumption_Tracker c, Sleep S WHERE p.Date = c.Date and c.Date = s.Date and u.UserID = c.UserID and c.UserID = p.UserID and p.userID = s.UserID

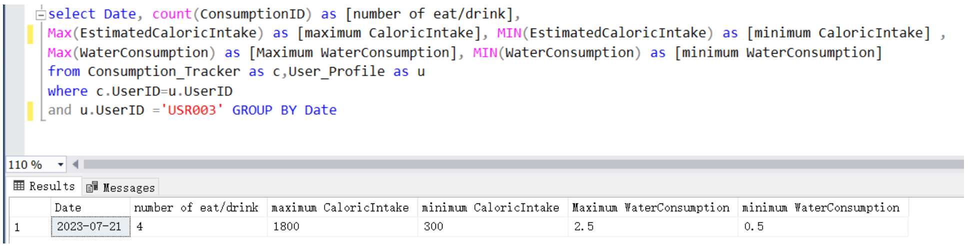


	Name	Age	Date	WorkoutType	Distance	Steps	WaterConsumption	EstimatedCaloricIntake	HoursSlept
1	YAP JIA QING	25	2023-07-21	W	5.2	8000	2.5	1800	7.5
2	John Doe	30	2023-07-21	W	3.1	6000	8	2000	8
3	Jane Smith	25	2023-07-21	R	4	7500	6	2200	6.5

Justification: The purpose of this query is to display the summary of the user’s activity report which includes the workout type, distance travelled, steps taken, water consumption, estimated caloric intake and hours slept. We can use the summarize activity report information to push out notifications to the user on which aspect to improve on whether they need more exercise, water consumption or their hour slept.

Create queries including Aggregate Function

1. select Date, count(ConsumptionID) as [number of eat/drink], Max(EstimatedCaloricIntake) as [maximum CaloricIntake], MIN(EstimatedCaloricIntake) as [minimum CaloricIntake] ,Max(WaterConsumption) as [Maximum WaterConsumption], MIN(WaterConsumption) as [minimum WaterConsumption] from Consumption_Tracker as c,User_Profile as u where c.UserID=u.UserID and u.UserID ='@UserID' GROUP BY Date

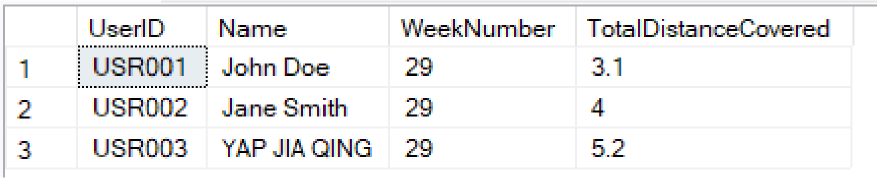


```
select Date, count(ConsumptionID) as [number of eat/drink],
Max(EstimatedCaloricIntake) as [maximum CaloricIntake], MIN(EstimatedCaloricIntake) as [minimum CaloricIntake] ,
Max(WaterConsumption) as [Maximum WaterConsumption], MIN(WaterConsumption) as [minimum WaterConsumption]
from Consumption_Tracker as c,User_Profile as u
where c.UserID=u.UserID
and u.UserID ='USR003' GROUP BY Date
```

	Date	number of eat/drink	maximum CaloricIntake	minimum CaloricIntake	Maximum WaterConsumption	minimum WaterConsumption
1	2023-07-21	4	1800	300	2.5	0.5

Justification: The COUNT() function returns the number of rows that matches a specified criterion. This query calculates the number of the specific user had do the record for drinking and eating in table Consumption_Tracker. So, the user can easily see how many times the user had eat and drank, minimum and maximum for calories intake and water consumption in a single day.

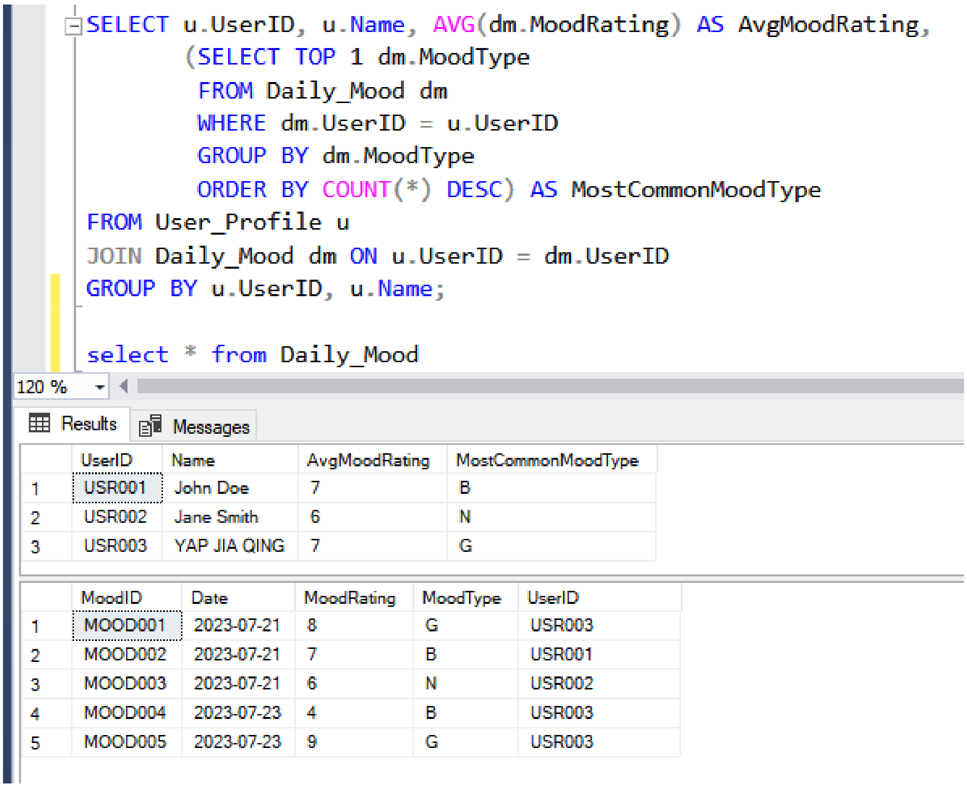
2. SELECT pa.UserID, u.Name, DATEPART(WEEK, pa.Date) AS WeekNumber, SUM(pa.Distance) AS TotalDistanceCovered FROM Physical_Activity pa JOIN User_Profile u ON pa.UserID = u.UserID GROUP BY pa.UserID, u.Name, DATEPART(WEEK, pa.Date);



	UserID	Name	WeekNumber	TotalDistanceCovered
1	USR001	John Doe	29	3.1
2	USR002	Jane Smith	29	4
3	USR003	YAP JIA QING	29	5.2

Justification: This query is to display the total distance covered by each user in the 'Physical_Activity' table by using the SUM() aggregate function. The SUM() aggregate function is applied to calculate the total distance covered by each user on a weekly basis. So, it will provide the total distance for each week to the user, which user can monitor their weekly physical activity progress.

3.
```
SELECT u.UserID, u.Name, AVG(dm.MoodRating) AS AvgMoodRating,
        (SELECT TOP 1 dm.MoodType
        FROM Daily_Mood dm
        WHERE dm.UserID = u.UserID
        GROUP BY dm.MoodType
        ORDER BY COUNT(*) DESC) AS MostCommonMoodType
FROM User_Profile u
JOIN Daily_Mood dm ON u.UserID = dm.UserID
GROUP BY u.UserID, u.Name;
```



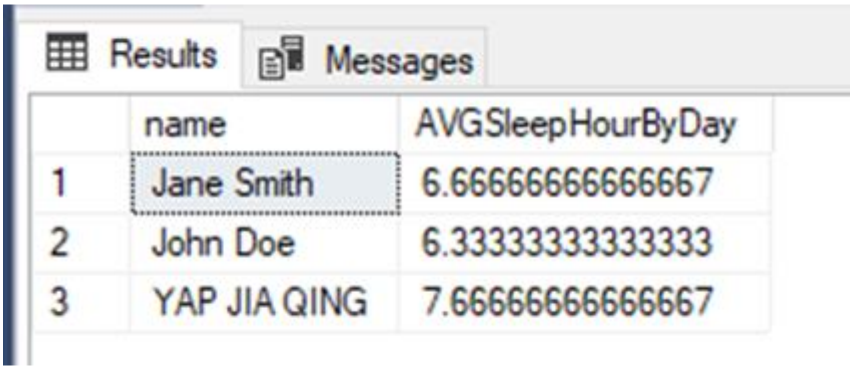
```
SELECT u.UserID, u.Name, AVG(dm.MoodRating) AS AvgMoodRating,
       (SELECT TOP 1 dm.MoodType
        FROM Daily_Mood dm
        WHERE dm.UserID = u.UserID
        GROUP BY dm.MoodType
        ORDER BY COUNT(*) DESC) AS MostCommonMoodType
FROM User_Profile u
JOIN Daily_Mood dm ON u.UserID = dm.UserID
GROUP BY u.UserID, u.Name;

select * from Daily_Mood
```

	UserID	Name	AvgMoodRating	MostCommonMoodType
1	USR001	John Doe	7	B
2	USR002	Jane Smith	6	N
3	USR003	YAP JIA QING	7	G

	MoodID	Date	MoodRating	MoodType	UserID
1	MOOD001	2023-07-21	8	G	USR003
2	MOOD002	2023-07-21	7	B	USR001
3	MOOD003	2023-07-21	6	N	USR002
4	MOOD004	2023-07-23	4	B	USR003
5	MOOD005	2023-07-23	9	G	USR003

Justification: This query is to display the average of mood rating and deduce the most common mood type for the users. The AVG() function is used to calculate the average mood ratings of users. The SELECT TOP function is then used to determine which mood type entered by the user is the most common.

4. Select name, avg(HoursSlept) AS AVGSleepHourByDay from User_Profile u, Sleep sWhere u.UserID = s.UserID Group by Name



	name	AVGSleepHourByDay
1	Jane Smith	6.66666666666667
2	John Doe	6.33333333333333
3	YAP JIA QING	7.66666666666667

Justification: The purpose of this query is to calculate the user’s accumulated daily average numbers of hours they have slept so the user may keep track of their sleeping hours and make improvement on their sleeping routine or schedule.