1: Retrieve the average BodyMassIndex (BMI) for users in each city with a ProPlusMembership.

```
SELECT L.City, AVG(BM.BodyMassIndex) AS AverageBMI
FROM Users U
JOIN Location L ON U.UserID = L.UserID
JOIN ProPlusMembership PP ON U.UserID = PP.UserID
JOIN BodyMeasurement BM ON U.UserID = BM.UserID
GROUP BY L.City;
```

2: Find the total number of users in each state with a BasicMembership whose renewal date is within the next 30 days.

```
SELECT L.State, COUNT(*) AS TotalUsers
FROM Users U
JOIN Location L ON U.UserID = L.UserID
JOIN BasicMembership BM ON U.UserID = BM.UserID
WHERE BM.RenewalDate BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 30 DAY)
GROUP BY L.State;
```

3: Identify the top 5 users with the highest BodyFatPercentage who have a ProMembership.

```
SELECT U.Username, BM.BodyFatPercentage
FROM Users U
JOIN ProMembership PM ON U.UserID = PM.UserID
JOIN BodyMeasurement BM ON U.UserID = BM.UserID
```

```
ORDER BY BM.BodyFatPercentage DESC
LIMIT 5;
```

4: Calculate the average number of sets and reps for weight training workouts for each ExerciseCategory.

```
SELECT W.ExerciseCategory, AVG(WT.SetNumber) AS AvgSets, AVG(WT.MinReps + WT.MaxReps) AS AvgReps
FROM Workout W
JOIN WeightTraining WT ON W.WorkoutID = WT.WorkoutID
GROUP BY W.ExerciseCategory;
```

5: Find the total number of cardio workouts completed by users in a specific city.

```
SELECT L.City, COUNT(*) AS TotalCardioWorkouts
FROM Users U
JOIN Location L ON U.UserID = L.UserID
JOIN Workout W ON U.UserID = W.UserID
WHERE W.ExerciseCategory = 'Cardio' AND L.City = 'YourCity'
GROUP BY L.City;
```

6: Retrieve the ExercisePlan with the highest average calories burned per workout.

```
SELECT W.ExercisePlan, AVG(C.CaloriesBurned) AS AvgCaloriesBurned
```

```
FROM Workout W
JOIN Cardio C ON W.WorkoutID = C.WorkoutID
GROUP BY W.ExercisePlan
ORDER BY AvgCaloriesBurned DESC
LIMIT 1;
```

7: Identify the users who have both a ProMembership and a ProPlusMembership.

```
SELECT U.Username
FROM Users U
JOIN ProMembership PM ON U.UserID = PM.UserID
JOIN ProPlusMembership PP ON U.UserID = PP.UserID;
```

8: Calculate the total fees collected from users with a BasicMembership in each state.

```
SELECT L.State, SUM(BM.Fee) AS TotalFees
FROM Users U
JOIN Location L ON U.UserID = L.UserID
JOIN BasicMembership BM ON U.UserID = BM.UserID
GROUP BY L.State;
```

9: Find the ExerciseLevel with the highest number of users who have a BasicMembership.

```
SELECT W.ExerciseLevel, COUNT(*) AS UserCount
```

```
FROM Users U
JOIN BasicMembership BM ON U.UserID = BM.UserID
JOIN Workout W ON U.UserID = W.UserID
GROUP BY W.ExerciseLevel
ORDER BY UserCount DESC
LIMIT 1;
```

10: Retrieve the top 3 cities with the highest average user age.

```
SELECT L.City, AVG(U.Age) AS AvgAge
FROM Users U
JOIN Location L ON U.UserID = L.UserID
GROUP BY L.City
ORDER BY AvgAge DESC
LIMIT 3;
```

11: Calculate the total number of users with a ProPlusMembership who have an AdvancedTechnique above a certain threshold.

```
SELECT COUNT(*) AS TotalUsers
FROM Users U
JOIN ProPlusMembership PP ON U.UserID = PP.UserID
JOIN Advanced A ON U.UserID = A.UserID
WHERE A.AdvancedTechnique > 0.8;
```

12: Identify the users with a ProMembership who have the highest total AdditionalFeatures count.

```
SELECT U.Username, PM.AdditionalFeatures
FROM Users U
JOIN ProMembership PM ON U.UserID = PM.UserID
ORDER BY PM.AdditionalFeatures DESC
LIMIT 1;
```