EZ Training

Team 18
Adithya Bhat
Chuyun Deng
Gerson Aaron Morales Deras
Emily Qiu
Saivenkata Nagavyjayanthi Polapragada
Shiyunyang Zhao

A Quick Recap

EZ Training combines immersive virtual reality (VR) experiences with personalized workout and nutrition plans to create a holistic fitness environment.

Berkeley

Populating Data

Data: Populated using **INSERT** statements in SQL with synthetic values created to reflect diverse scenarios.

Tables:

- A total of **22 tables** were populated with data to ensure comprehensive testing and meaningful query results.
- Employee, Employee_Role, Exercise, Group_Trainer, Group_Trainer_Conducts_Workout_Session, Individual, Medical_Condition, Member, Member_Consults_Nutritionist, Member_Participates_Workout_Session, Member_VR_Preferences, Nutrition_Plan, Nutritionist, Payment, Role, Shift, VR_Equipment, Workout_Log, Workout_Session, Workout_Session_Exercise, Workout_Session_Uses_Equipment, Member_Measurememnt

Berkeley

Interesting Query 1 - SQL

Identifying top nutritionists based on metrics like client health improvement and engagement.

Sub Query1:

This query joins the **Member_Measurements** table to itself to compare the earliest and latest recorded measurements for each member and calculates changes in BMI, body fat percentage, muscle mass, and visceral fat level as the difference between the latest and earliest values for each member.

```sql
-- Interesting Query 1: Top N Nutritionists
-- 1. View: Member_Changes
CREATE OR REPLACE VIEW Member_Changes AS
SELECT
    mm_latest.Member_ID,
    (mm_latest.BMI - mm_first.BMI) AS BMI_Change,
    (mm_latest.Body_Fat_Percentage - mm_first.Body_Fat_Percentage) AS Body_Fat_Change,
    (mm_latest.Muscle_Mass - mm_first.Muscle_Mass) AS Muscle_Mass_Change,
    (mm_latest.Visceral_Fat_Level - mm_first.Visceral_Fat_Level) AS Visceral_Fat_Change
FROM
    ieor215_project.MEMBER_MEASUREMENTS mm_latest
JOIN
    (
        SELECT
            mm1.Member_ID,
            mm1.BMI,
            mm1.Body_Fat_Percentage,
            mm1.Muscle_Mass,
            mm1.Visceral_Fat_Level
        FROM
            ieor215_project.MEMBER_MEASUREMENTS mm1
        WHERE
            mm1.Record_Date = (
                SELECT MIN(mm2.Record_Date)
                FROM ieor215_project.MEMBER_MEASUREMENTS mm2
                WHERE mm2.Member_ID = mm1.Member_ID
            )
    ) mm_first
    ON mm_latest.Member_ID = mm_first.Member_ID
WHERE
    mm_latest.Record_Date = (
        SELECT MAX(mm3.Record_Date)
        FROM ieor215_project.MEMBER_MEASUREMENTS mm3
        WHERE mm3.Member_ID = mm_latest.Member_ID
    );
```

Berkeley

Interesting Query 1 - SQL

Sub Query2: This query builds on the **Member_Changes** view by retrieving each member's changes in BMI, body fat percentage, muscle mass, and visceral fat levels and computes the **Health_Improvement_Score** using a weighted formula.

```sql
-- 2. View: Client_Health_Scores
CREATE OR REPLACE VIEW Client_Health_Scores AS
SELECT
    Member_ID,
    ((-1 * BMI_Change)
     + (-1 * Body_Fat_Change)
     + (1 * Muscle_Mass_Change)
     + (-1 * Visceral_Fat_Change)) AS Health_Improvement_Score
FROM Member_Changes;
```

Berkeley

Interesting Query 1 - SQL

Sub Query3: This query extracts data by joining the **Client_Health_Scores**, **Member_Consults_Nutritionist**, and **Member** tables to associate health improvement scores with nutritionists. It then aggregates metrics such as active client count, total client count, and total health improvement scores for each nutritionist, grouped by their unique ID.

```sql
-- 3. View: Nutritionist_Performance
CREATE OR REPLACE VIEW Nutritionist_Performance AS
SELECT
    mcn.Employee_ID AS Nutritionist_ID,
    COUNT(DISTINCT CASE WHEN me.Membership_status = 'Active' THEN chs.Member_ID END) AS Active_Client_Count,
    COUNT(DISTINCT chs.Member_ID) AS Total_Client_Count,
    SUM(chs.Health_Improvement_Score) AS Total_Health_Improvement
FROM Client_Health_Scores chs
JOIN ieor215_project.Member_Consults_Nutritionist mcn
    ON chs.Member_ID = mcn.Member_ID
JOIN ieor215_project.Member me
    ON chs.Member_ID = me.Member_ID
GROUP BY mcn.Employee_ID;
```

Berkeley

Interesting Query 1 - SQL

Sub Query4: This query extracts data by joining the **Member_Measurements** and **Member** tables to retrieve the average of their BMI based on their Record_Date and their Membership_Status (considered only for the Active members). This allows us to track the change in the BMI of all the active members over a period of time.

```sql
-- 4. View: Avg_BMI_Trend
CREATE OR REPLACE VIEW Avg_BMI_Trend AS
SELECT
    DATE(mm.Record_Date) AS Measurement_Date,
    AVG(mm.BMI) AS Avg_BMI
FROM ieor215_project.MEMBER_MEASUREMENTS mm
JOIN ieor215_project.Member me
    ON mm.Member_ID = me.Member_ID
WHERE me.Membership_status = 'Active'
GROUP BY DATE(mm.Record_Date)
ORDER BY DATE(mm.Record_Date);
```

Berkeley

Interesting Query 1 - SQL Output

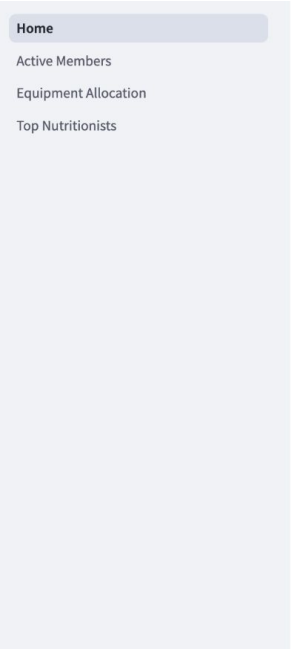
Example rows:

Nutritionist_...	Active_Client_Cou...	Total_Client_Cou...	Total_Health_Improvem...
8	3	5	5.24
9	3	5	5.52
11	3	5	-3.06
12	4	5	3.45
14	3	5	3.05
15	2	5	0.47
18	2	4	3.35
19	2	4	4.81
20	2	4	12.50

Berkeley

Interesting Query 1 - Analysis

Interactive Dashboard: https://ez-training.streamlit.app/



EZ Training Dashboard

Overview

Welcome to the **EZ Training Dashboard**! This platform provides a comprehensive interface for analyzing and optimizing nutritionist and member performance metrics, empowering you to make data-driven decisions with ease.

What You'll Find Here

- **Top Nutritionists Dashboard**: Discover the top-performing nutritionists based on metrics like client health improvement and engagement.
- **Active Members Analysis**: Dive into detailed analytics on active members, including BMI trends, workout frequency, and overall progress.

Features

- **Interactive Visualizations**: Easily filter and visualize data to uncover actionable insights.
- **Data Exploration Tools**: Gain deep insights into member health trends and nutritionist performance.
- **Customizable Filters**: Tailor your analysis with date ranges, performance metrics, and more.

Setup Details

- **Database**: The data is hosted on Aiven, ensuring reliable and scalable database management.
- **GitHub**: The codebase is hosted on GitHub for easy collaboration and version control.
- **Streamlit App Deployment**: The dashboard is deployed on Streamlit's free tier, allowing for seamless access and interaction. Secret keys and other sensitive data are stored in Streamlit's secure environment.

Berkeley

Interesting Query 1 - Analysis

Filtered Top N Nutritionists Table: This table highlights the top 5 nutritionists, ranked by their total health improvement scores, showcasing those who have made the most significant impact on their active clients' well-being.

🏆 Top Nutritionists Dashboard

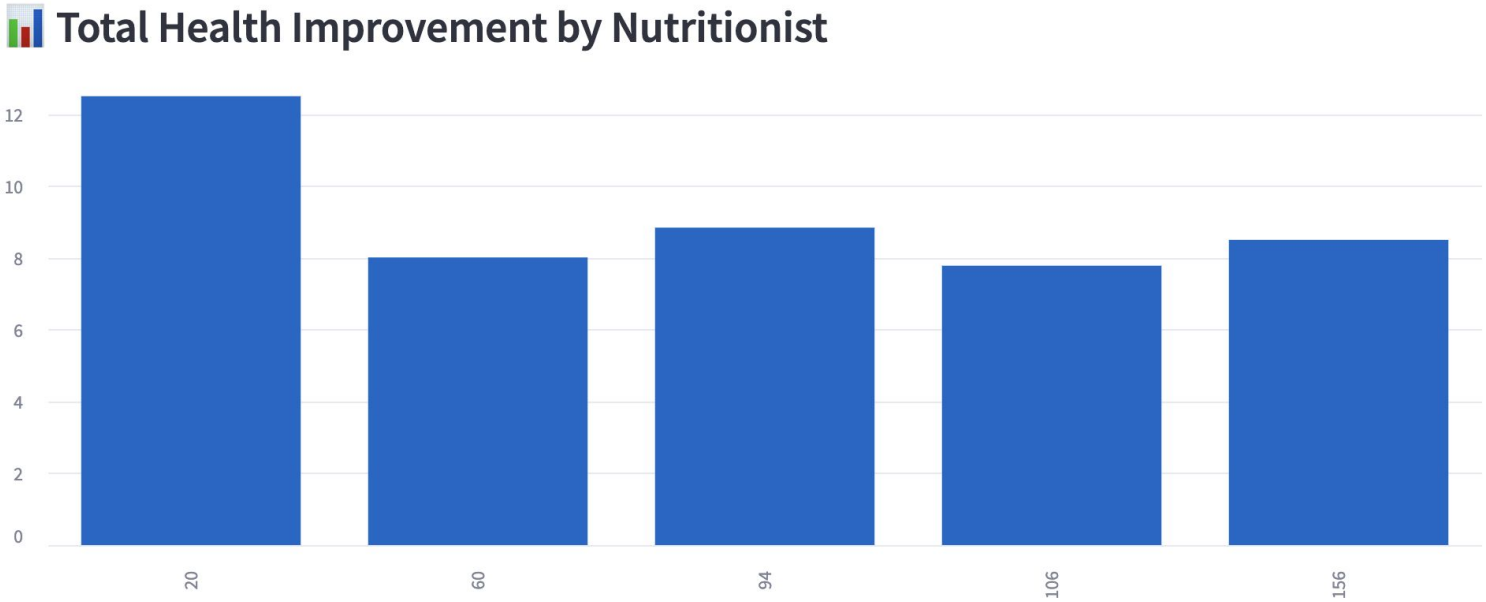
📋 Top Nutritionists Table

	Nutritionist_ID	Pay_rate	Active_Client_Count	Total_Client_Count	Total_Health_Improvement
0	20	$29.60	2	4	12.50
1	94	$39.33	3	4	8.84
2	156	$27.49	3	3	8.50
3	60	$22.40	2	4	8.01
4	106	$46.34	0	4	7.78

Berkeley

Interesting Query 1 - Analysis

Visualizations: The bar graph illustrates the disparity in total health improvement scores, with the top nutritionists demonstrating substantially higher cumulative impacts compared to others.



Berkeley

Interesting Query 1 - Analysis

Visualizations: The steady line in the average BMI trend indicates consistent health maintenance among active clients under the guidance of the top nutritionists.

Berkeley

Interesting Query 1 - Analysis

How This Helps the Client:

- Recognizes and rewards top-performing nutritionists.
- Enhances service quality by identifying effective practices.
- Provides insights to develop training programs for other staff.

Berkeley

Interesting Query 2 - SQL

Identifying the most frequently used VR equipment across all workout sessions.

SQL Query: This query joins **Workout_Session**, **Workout_Session_Uses_Equipment**, and **VR_Equipment** tables to calculate the usage count for each available VR equipment grouped by program type. The **WHERE** clause ensures only available equipment is included, and the results are ordered by program type and equipment name

```sql
SELECT
    ws.Program_type,
    vr.Name AS Equipment_Name,
    COUNT(wsu.Equipment_ID) AS Usage_Count
FROM
    Workout_Session ws
JOIN
    Workout_Session_Uses_Equipment wsu ON ws.Workout_ID = wsu.Session_ID
JOIN
    VR_Equipment vr ON wsu.Equipment_ID = vr.Equipment_ID
WHERE
    vr.Status = 'Available'  -- Only include equipment that is available
GROUP BY
    ws.Program_type, vr.Name
ORDER BY
    ws.Program_type, Equipment_Name;
```

Berkeley

Interesting Query 2 - SQL Output

Example rows:

Program_type	Equipment_Name	Usage_Count
Cardio	Beige Yoga VR	1
Cardio	DarkCyan Boxing VR	2
Cardio	DarkSeaGreen Yoga VR	3
Cardio	DarkViolet Boxing VR	5
Cardio	Gainsboro Treadmill VR	1
Cardio	GoldenRod Cycling VR	4
Cardio	LawnGreen Cycling VR	4
Cardio	LightGreen Cycling VR	7
Cardio	RoyalBlue Treadmill VR	2

Berkeley

Interesting Query 2 - ILP Formulation

Optimization Formulation

Objective:

$$\text{Maximize} \sum_{i \in \text{program\_types}} \sum_{j \in \text{equipment\_names}} x_{ij} \cdot \text{Usage\_Count}_{ij}$$

Where:

- x_{ij} is a binary decision variable representing whether equipment j is assigned to program type i (1 if assigned, 0 if not).
- Usage_Count_{ij} is the usage count for equipment j in program type i, taken from the dataset.

Berkeley

Interesting Query 2 - ILP Formulation

Optimization Formulation

Constraints:

Equipment Availability Constraint: This ensures that no equipment is allocated more thar once to any program type. Each equipment can only be assigned to one program type.

$$\sum_{i \in \text{program\_types}} x_{ij} \leq 1, \quad \forall j \in \text{equipment\_names}$$

Program Type Capacity Constraint: This ensures that each program type does not exceed its capacity (in terms of the number of equipment available). Here, we assume that the number of equipment assigned to any program type is limited by the number of equipment available.

$$\sum_{j \in \text{equipment\_names}} x_{ij} \leq \text{len}(\text{equipment\_names}), \quad \forall i \in \text{program\_types}$$

Berkeley

Interesting Query 2 - ILP Output

Gurobi Results: The optimization results evenly allocate VR equipment across program types, ensuring that each piece of equipment is utilized in its respective program type with an allocation of 1.0, achieving balanced usage and no underutilization.

```
Optimal Equipment Allocation:
Program Type: Cardio, Equipment: DarkViolet Boxing VR, Allocation: 1.0
Program Type: Cardio, Equipment: LawnGreen Cycling VR, Allocation: 1.0
Program Type: Cardio, Equipment: LightGreen Cycling VR, Allocation: 1.0
Program Type: Cardio, Equipment: RoyalBlue Treadmill VR, Allocation: 1.0
Program Type: Cardio, Equipment: Yellow Cycling VR, Allocation: 1.0
Program Type: HIIT, Equipment: DarkSeaGreen Yoga VR, Allocation: 1.0
Program Type: HIIT, Equipment: Gainsboro Treadmill VR, Allocation: 1.0
Program Type: HIIT, Equipment: GoldenRod Cycling VR, Allocation: 1.0
Program Type: HIIT, Equipment: SandyBrown Yoga VR, Allocation: 1.0
Program Type: HIIT, Equipment: Wheat Treadmill VR, Allocation: 1.0
Program Type: Strength Training, Equipment: Beige Yoga VR, Allocation: 1.0
Program Type: Strength Training, Equipment: DarkCyan Boxing VR, Allocation: 1.0
Program Type: Strength Training, Equipment: SeaGreen Yoga VR, Allocation: 1.0
Program Type: Strength Training, Equipment: Turquoise Yoga VR, Allocation: 1.0
```

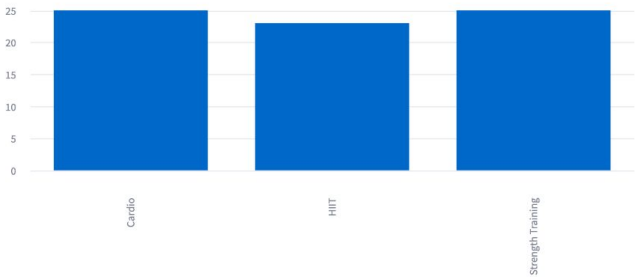
Berkeley

Interesting Query 2 - Optimal Equipment Allocation

Optimal Equipment Allocation

	Program Type	Equipment	Usage Count
0	Cardio	DarkViolet Boxing VR	5
1	Cardio	LawnGreen Cycling VR	4
2	Cardio	LightGreen Cycling VR	7
3	Cardio	RoyalBlue Treadmill VR	2
4	Cardio	Yellow Cycling VR	7
5	HIIT	DarkSeaGreen Yoga VR	5
6	HIIT	Gainsboro Treadmill VR	3
7	HIIT	GoldenRod Cycling VR	4
8	HIIT	SandyBrown Yoga VR	4
9	HIIT	Wheat Treadmill VR	7
10	Strength Training	Beige Yoga VR	5
11	Strength Training	DarkCyan Boxing VR	6
12	Strength Training	SeaGreen Yoga VR	7
13	Strength Training	Turquoise Yoga VR	7

Allocation Summary



Berkeley

Interesting Query 2 - ILP

How This Helps the Client:

- Ensures equitable distribution of VR equipment across sessions.
- Prevents underutilization and improves overall user experience.
- Maximizes the value of VR technology investments.
- Increases customer satisfaction and operational efficiency.

Berkeley

Interesting Query 3 - SQL

Retrieving the data on active members' BMI changes and their workout frequency.

Sub Query1: This query joins **Member_Measurements** with **Member** to filter out inactive members, groups the measurements by date, and calculates the average BMI for all active members on each day.

```sql
CREATE OR REPLACE VIEW Avg_BMI_Trend AS
SELECT
    DATE(mm.Record_Date) AS Measurement_Date,
    AVG(mm.BMI) AS Avg_BMI
FROM ieor215_project.MEMBER_MEASUREMENTS mm
JOIN ieor215_project.Member me
    ON mm.Member_ID = me.Member_ID
WHERE me.Membership_status = 'Active'
GROUP BY DATE(mm.Record_Date)
ORDER BY DATE(mm.Record_Date);
```

Berkeley

Interesting Query 3 - SQL

Sub Query2: This query joins the **Member**, **Member_Measurements**, and **Member_Participates_Workout_Session** tables to link each active member's BMI changes with their workout participation and calculates metrics like average BMI, BMI change, and BMI change per session for active members.

```sql
CREATE OR REPLACE VIEW Active_Member_BMI_Workout_View AS
SELECT
    M.Member_ID,
    AVG(MM.BMI) AS Average_BMI,
    (MAX(MM.BMI) - MIN(MM.BMI)) AS BMI_Change,
    COUNT(DISTINCT MPWS.Workout_ID) AS Workout_Session_Count,
    CASE
        WHEN COUNT(DISTINCT MPWS.Workout_ID) = 0 THEN 0
        ELSE (MAX(MM.BMI) - MIN(MM.BMI)) / COUNT(DISTINCT MPWS.Workout_ID)
    END AS BMI_Change_Per_Session
FROM Member M
JOIN MEMBER_MEASUREMENTS MM ON M.Member_ID = MM.Member_ID
LEFT JOIN Member_Participates_Workout_Session MPWS ON M.Member_ID = MPWS.Member_ID
LEFT JOIN Workout_Session WS ON MPWS.Workout_ID = WS.Workout_ID
WHERE M.Membership_status = 'Active'
GROUP BY M.Member_ID;
```

Berkeley

Interesting Query 3 - SQL Output

Example rows:

Member_ID	Average_BMI	BMI_Change	Workout_Session_Co...	BMI_Change_Per_Session
1	25.451667	0.10	1	0.100000
3	27.913333	0.43	2	0.215000
4	23.024667	1.45	3	0.483333
7	27.802941	2.40	4	0.600000
9	22.550000	0.21	1	0.210000
10	33.778571	1.07	2	0.535000
14	15.900000	0.12	4	0.030000
15	21.235263	1.11	0	0.000000
16	26.932105	0.45	3	0.150000

Berkeley

Interesting Query 3 - Analysis

Interactive Dashboard: https://ez-training.streamlit.app/

Explanation: This enhanced dashboard uses the **Active_Member_BMI_Workout_View** to focus on:

Average_BMI: Gives an idea of the member's BMI trend.

BMI_Change: How much their BMI has changed over recorded measurements.

Workout_Session_Count: How many sessions they've attended.

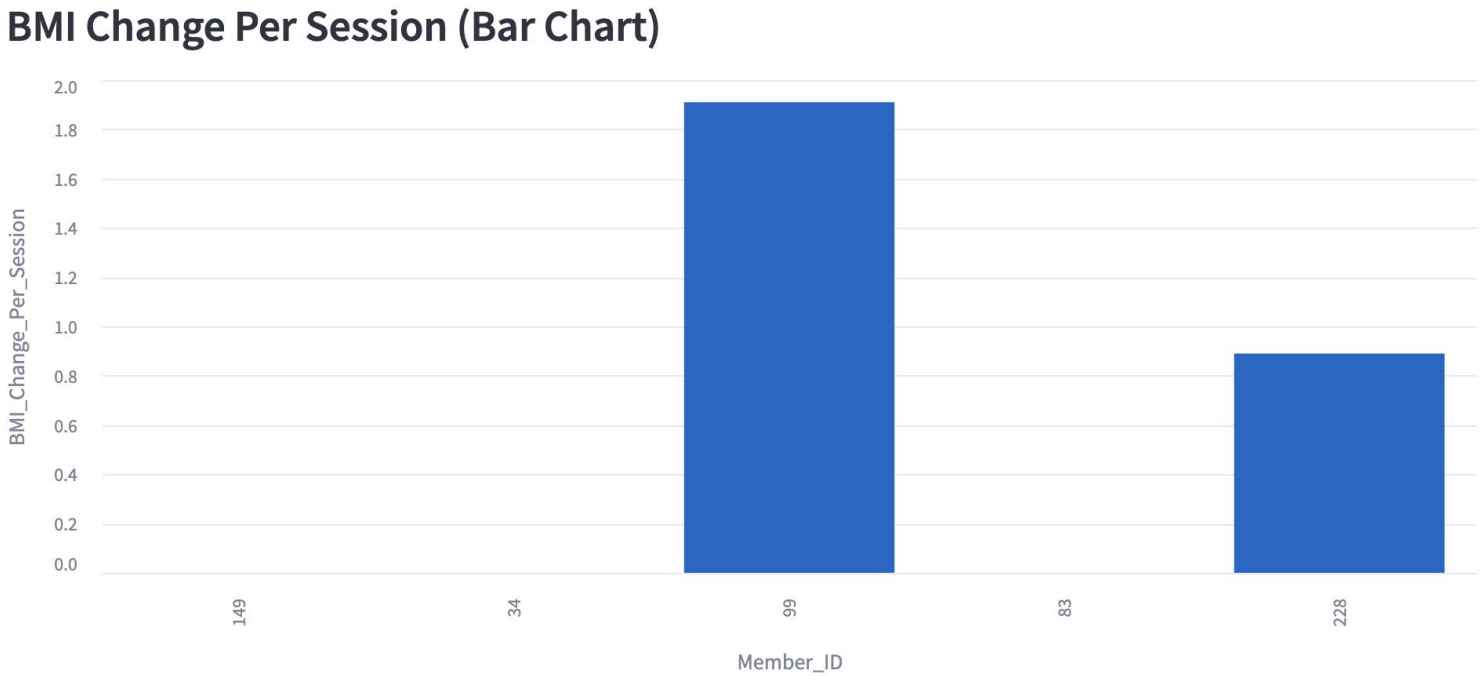
BMI_Change_Per_Session: Efficiency metric indicating how much BMI changes per workout session.

The filters allow for a detailed investigation of specific subgroups of members, and the visualizations offer additional perspectives on the data.

Berkeley

Interesting Query 3 - Analysis

Visualizations: The bar chart helps identify members with the greatest BMI change per session.



Berkeley

Interesting Query 3 - Analysis

Filtered Top N Active Members' BMI Change and Workout Frequency Table

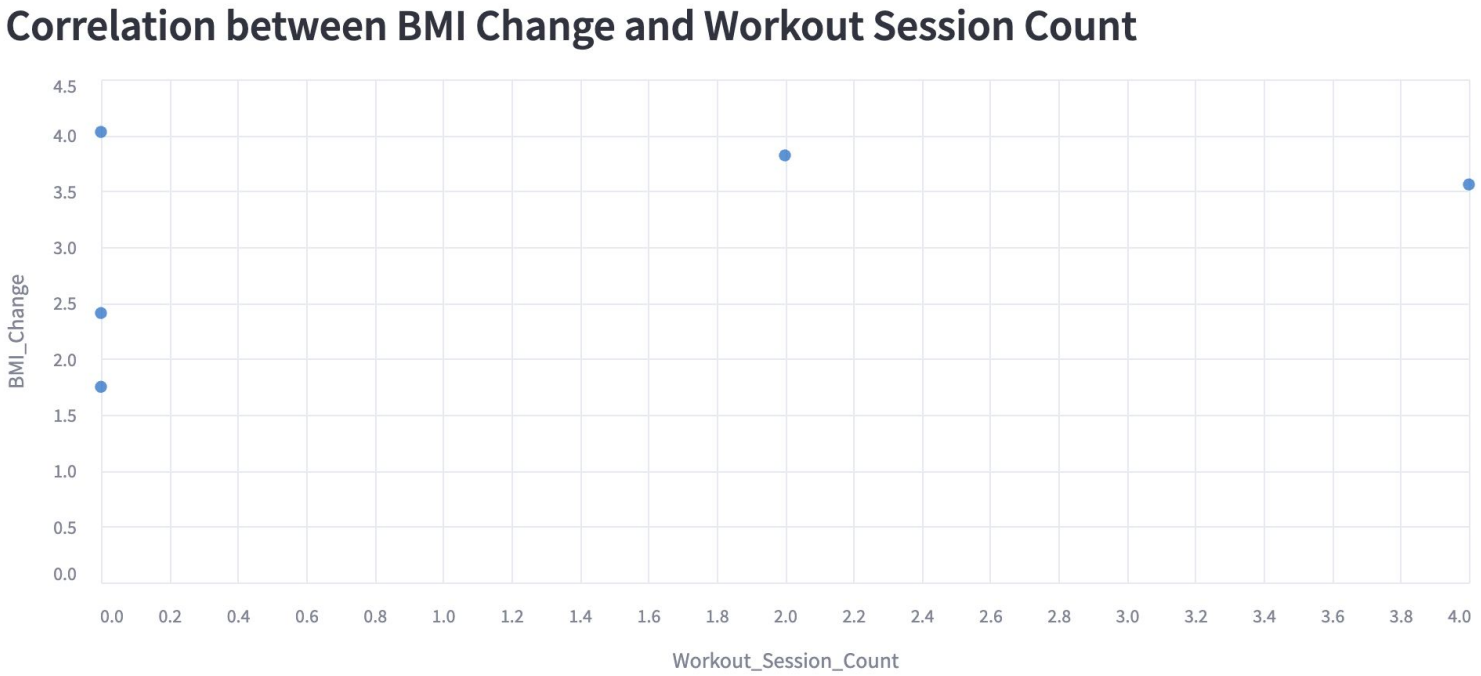
Filtered Results

	Member_ID	Average_BMI	BMI_Change	Workout_Session_Count	BMI_Change_Per_Session
39	99	51.5475	3.82	2	1.91
87	202	40.7075	2.28	2	1.14
88	205	36.5407	2.91	2	1.455
111	247	35.1673	2.66	2	1.33
137	296	29.395	2.8	2	1.4
51	125	29.385	2.52	2	1.26
22	47	28.5242	2.42	2	1.21

Berkeley

Interesting Query 3 - Analysis

Visualizations: The scatter plot shows if there's a correlation between attending more workout sessions and achieving higher BMI changes.



Berkeley

Interesting Query 3 - Analysis

How This Helps the Client:

- Enables the design of more effective and personalized workout programs.
- Identifies trends in member activity and health improvement.
- Helps improve member outcomes and satisfaction.
- Increases client retention through targeted interventions.

Berkeley

Questions?

Berkeley

Thank you

Berkeley