EZ Training

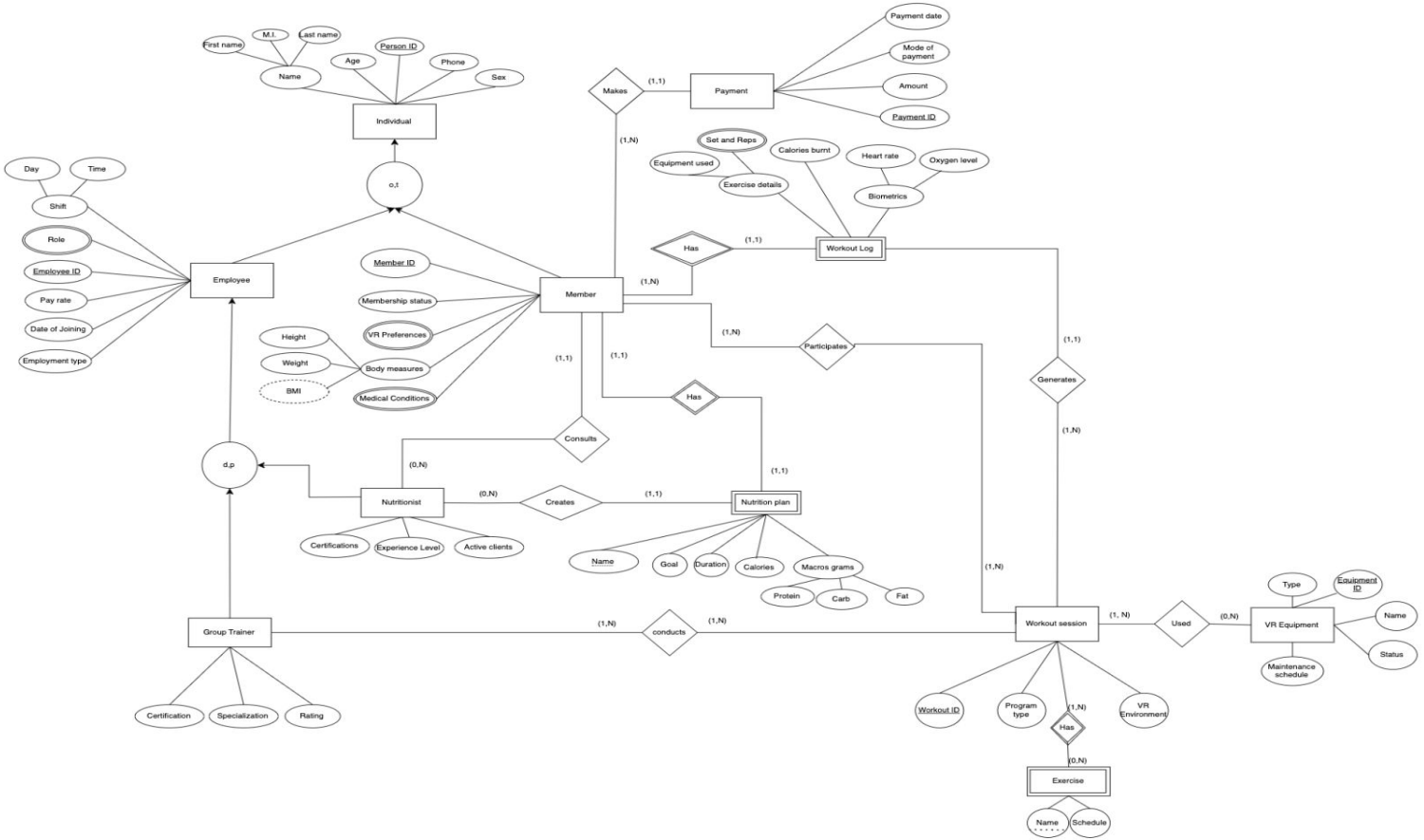
Team 18
Adithya Bhat
Chuyun Deng
Gerson Aaron Morales Deras
Emily Qiu
Saivenkata Nagavyjayanthi Polapragada
Shiyunyang Zhao

A Quick Recap

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

Berkeley

EER Diagram



Berkeley

Normalization

1NF:

Multivalued Attributes - Created separate tables:

- Role
- Medical Conditions
- Sets and Reps

Composite attributes - Decomposed:

- Name → First name, MI, Last name
- Shift → Day, Time (Created separate table to avoid storing multiple shift details in a single attribute)
- Body Measures → Height, Weight, BMI
- Macro grams → Proteins, Carbs, Fat
- Exercise details → Equipment, Sets, Reps
- Biometrics → Oxygen level, Heart rate

Berkeley

Normalization

2NF:

- Checked for partial dependencies in relations with multiple primary keys (composite primary keys).
- In relations like **Nutrition_Plan** and **Exercise**, we ensured that each non-key attribute depends on the entire composite primary key, not just part of it.

3NF:

- Checked for transitive dependencies and ensured that every non-key attribute depends only on the primary key.
- Separated **Role** into its own table to eliminate transitive dependencies and used **Role_ID** as a foreign key in the Employee table.
- Introduced **Workout_Log_Exercise** table to break transitive dependencies in **Workout_Log**.

BCNF:

- Checked each table to ensure there are no dependencies based on non-superkey determinants.
- Ensured that every determinant in each relation is a superkey, satisfying the requirements of BCNF.

Berkeley

Denormalization

- Merged the **Workout_Log** and **Workout_Log_Exercise** tables into a single table.

Reasoning:

- This denormalization reduces the need for multiple joins, streamlining data retrieval for frequently accessed combined data.
- This is especially beneficial for performance as it contains related data and can be queried together often.

Benefits:

- Faster query execution for frequently accessed data, as there is no need to join two separate tables each time a query is run.
- Improved efficiency for reports or queries that retrieve workout and exercise data together, especially in use cases like generating workout logs or member progress tracking.

Berkeley

Derived Attributes

- **BMI** (Body Mass Index) is stored as a derived attribute.

Reasoning:

- BMI is calculated from Height and Weight **(BMI = Weight (kg) / Height (m)^2)**.
- Instead of recalculating it every time it’s needed, we store the value to improve query performance.

Benefits:

- Storing BMI directly saves processing time during queries, which is especially useful when dealing with large datasets or frequent access to BMI data.

Berkeley

Relational Schema

- **Individual** (Person_ID, First_name, MI, Last_name, Age, Phone, Sex)
- **Member** (Member_ID, Person_ID(FK), Membership_status, Height, Weight, BMI)
- **Member_VR_Preferences** (Member_ID, Person_ID(FK), VR_Preferences)
- **Medical_Condition** (Condition_ID, Member_ID(FK), Person_ID(FK), Condition_name)
- **Employee(**Employee_ID,Person_ID(FK),Role_ID(FK),Shift_ID(FK),Pay_rate,Date_of_joining, Employment_type)
- **Role** (Role_ID, Role_name)
- **Employee_Role** (Employee_ID(FK), Role_ID(FK))
- **Shift** (Shift_ID,Employee_ID(FK), Day, Time)
- **Group_Trainer** (Employee_ID(FK), Certification, Specialization, Rating)

Berkeley

Relational Schema

- **Nutritionist** (Employee_ID(FK), Certifications, Experience_level, Active_clients)
- **Workout_Log** (Log_ID,Member_ID(FK),Workout_Session_ID(FK),Calories_burnt,Heart_rate,Oxygen_level, Exercise_ID(FK), Equipment_ID(FK), Sets, Reps, Weight)
- **Workout_Log_Sets&Reps**(Log_ID, Member_ID(FK), Workout_Session_ID(FK), Sets, Reps)
- **Workout_Session** (Workout_ID, Member_ID(FK), Workout_Log_ID(FK), Program_type, VR_Environment)
- **Exercise**(Exercise_ID, Workout_ID(FK), Name, Schedule)
- **Workout_Session_Exercise** (Session_ID(FK), Exercise_ID(FK))
- **Workout_Session_Uses_Equipment** (Session_ID(FK), Equipment_ID(FK))
- **Nutrition_Plan** (Name, Member_ID(FK), Goal, Duration, Calories, Protein, Carbs, Fat, Employee_ID(FK))

Berkeley

Relational Schema

- **Payment** (Payment_ID, Member_ID(FK), Payment_date, Mode_of_payment, Amount)
- **VR_Equipment** (Equipment_ID, Type, Name, Status, Maintenance_schedule)
- **Member_Consults_Nutritionist** (Member_ID(FK), Employee_ID(FK))
- **Member_Participates_Workout_Session**(MemberID(FK), Workout_ID(FK))
- **Group_Trainer_Conducts_Workout_Session**(Employee_ID(FK), Workout_ID (FK))

Berkeley

Interesting Queries

Query1: Identifying Top Nutritionists Based on BMI Improvement

Objective:

This query helps identify top nutritionists based on their effectiveness in improving client health metrics (BMI).

Benefits to the Client:

- **Promotions and Pay Adjustments:** The performance data can be used by management to recognize high-performing nutritionists and consider them for promotions or pay increases.
- **Improvement in Client Outcomes:** By identifying nutritionists who make the most significant impact, the company can replicate successful strategies across other employees.
- **Business Decisions:** This query can also help in staffing decisions, where the company might allocate more clients to top nutritionists or provide them with additional training opportunities to further improve the results of others.
- **Retention and Motivation:** Recognizing top performers helps improve morale and retention, motivating nutritionists to work even harder to achieve better client outcomes.

Berkeley

Interesting Queries

Query1: Identifying Top Nutritionists Based on BMI Improvement

Explanation: This subquery calculates the change in BMI for each member (who is active) after consulting with a nutritionist. The change is calculated as the difference between the highest and lowest recorded BMI for each member.

```
WITH Client_BMI_Change AS (
    SELECT
        M.Member_ID,
        MCN.Employee_ID AS Nutritionist_ID,
        (MAX(M.BMI) - MIN(M.BMI)) AS BMI_Change
    FROM
        Member M
    JOIN
        Member_Consults_Nutritionist MCN
    ON
        M.Member_ID = MCN.Member_ID
    WHERE
        M.Membership_status = 'Active'
    GROUP BY
        M.Member_ID, MCN.Employee_ID
),
```

Berkeley

Interesting Queries

Query1: Identifying Top Nutritionists Based on BMI Improvement

Explanation: This subquery aggregates the BMI change data by nutritionist. It sums the total BMI change for each nutritionist across all of their active clients and counts the number of active clients they have worked with.

```
Nutritionist_Performance AS (
    SELECT
        Nutritionist_ID,
        SUM(BMI_Change) AS Total_BMI_Change,
        COUNT(Member_ID) AS Active_Client_Count
    FROM
        Client_BMI_Change
    GROUP BY
        Nutritionist_ID
),
```

Berkeley

Interesting Queries

Query1: Identifying Top Nutritionists Based on BMI Improvement

Explanation: This subquery ranks nutritionists based on their total impact on BMI changes and sorts them in descending order. It returns the top 5 nutritionists with the most significant total BMI change. This is useful for recognizing the top performers.

```
Top_Nutritionists AS (
    SELECT
        NP.Nutritionist_ID,
        E.Pay_rate,
        NP.Total_BMI_Change,
        NP.Active_Client_Count
    FROM
        Nutritionist_Performance NP
    JOIN
        Employee E
    ON
        NP.Nutritionist_ID = E.Employee_ID
    ORDER BY
        NP.Total_BMI_Change DESC
    LIMIT 5
)
```

Berkeley

Interesting Queries

Query1: Identifying Top Nutritionists Based on BMI Improvement

Explanation: The final output selects the top nutritionists, displaying their ID, pay rate, total BMI change, and active client count.

```
SELECT
    TN.Nutritionist_ID,
    TN.Pay_rate,
    TN.Total_BMI_Change,
    TN.Active_Client_Count
FROM
    Top_Nutritionists TN;
```

Berkeley

Interesting Queries

Query2: Most Frequently Used VR Equipment in Workout Sessions

Objective:

This query is used to identify the most frequently used VR equipment across all workout sessions.

Benefits to the Client:

- **Inventory Management:** The results of this query help management identify which VR equipment is used most frequently in workout sessions. This information can help decide which equipment needs more frequent maintenance or restocking, ensuring that the most popular equipment is always available.
- **Operational Efficiency:** By understanding the frequency of use, the management can plan maintenance schedules to minimize downtime and improve member experience.
- **Data-Driven Decision Making:** With this data, the business can make better decisions about purchasing additional equipment or replacing outdated or underused equipment.

Berkeley

Interesting Queries

Query2: Most Frequently Used VR Equipment in Workout Sessions

Explanation: This SQL query retrieves the type and name of VR equipment used during workout sessions and counts how many times each piece of equipment has been used. The result is ordered by usage count in descending order to highlight the most frequently used equipment.

```
SELECT
    VE.Type AS Equipment_Type,
    VE.Name AS Equipment_Name,
    COUNT(WSE.Equipment_ID) AS Usage_Count
FROM
    Workout_Session_Uses_Equipment WSE
JOIN
    VR_Equipment VE
ON
    WSE.Equipment_ID = VE.Equipment_ID
GROUP BY
    VE.Type, VE.Name
ORDER BY
    Usage_Count DESC;
```

Berkeley

Interesting Queries

Query3: Active Members' BMI Change and Workout Frequency Analysis

Objective:

This query is used to retrieve the data on active members' BMI changes and their workout frequency.

Benefits to the Client:

- **Data for Analysis:** This query provides data that can be used to analyze how the number of workout sessions influences BMI changes for members. It helps in understanding the impact of workout routines on fitness progress.
- **Identify Areas for Improvement:** Based on the data, fitness trainers can identify members who are not seeing expected BMI changes and consider adjusting their workout plans.
- **Personalized Recommendations:** With this data, personalized advice can be provided to members on how to optimize their workout routines to achieve better BMI outcomes.

Berkeley

Interesting Queries

Query3: Active Members' BMI Change and Workout Frequency Analysis

Explanation: This query retrieves the average BMI for each active member and counts how many workout sessions they've attended. It also calculates the ratio of BMI change to workout sessions, showing how much change in BMI a member has experienced per session. The results are sorted by the highest average BMI, helping identify which members have experienced the greatest BMI changes in relation to their workout frequency.

```
SELECT
    M.Member_ID,
    AVG(M.BMI) AS Average_BMI,
    COUNT(WS.Workout_ID) AS Workout_Session_Count,
    AVG(M.BMI) / NULLIF(COUNT(WS.Workout_ID), 0) AS BMI_Per_Workout_Session
FROM
    Member M
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
ORDER BY
    Average_BMI DESC;
```

Berkeley

Questions?

Berkeley

Thank you

Berkeley