BUSA8090 - Pulse Fitness - Database Design and Development, Business Analytics Report

Name: Do Minh Huu Nguyen

Student Number: 47772360

Table of Contents

A. Database Design and Development

1. Entity-Relationship Diagram (ERD)

1.1 Identify Entities

The primary entities are:

- Member
- Membership_Plan
- Trainer
- Fitness_class
- Class_booking (associative)
- Training_Session
- Specialisation
- Payment

1.2 Identify Business Rules

- A Member “has” a Membership_plan
- A Member “make” a Payment
- A Trainer “hosts” a Fitness_class
- A Member “books” a Fitness_class → associative entity = Class_booking
- A Trainer “conducts” a Training_session
- A Member “attends” a Training_session
- A Trainer “has” a Specialisation
- A Fitness_Class “belongs to” a Specialisation
- A Training_Session “belongs to” a Specialisation

1.3 Define Relationships and Represent Cardinality

Relationship	**Type**	**Cardinality**
Member - Membership_plan	Many-to-One	Many members can have one membership plan
Trainer - Fitness_class	One-to-Many	One trainer can host many Fitness-class
Member-Fitness_class (via Class_booking)	Many-to-Many	One member can book many class One class can be booked by many members
Trainer-Training_session	One-to-Many	One trainer can conduct many training sessions
Member-Training_session	One-to-Many	One member can attend many training sessions.
Trainer-Specialisation	Many-to-One	Many trainers can specialize in one specialty
Fitness_Class-Specialisation	Many-to-One	One specialty can be associated with many fitness classes.
Training_Session-Specialisation	Many-to-One	One specialisation can be applied to many training sessions
Member-Payment	One-to-Many	One member can make many payments

1.4 Identify Attributes

Entity	Primary Key	Attributes
MEMBER	Mem_ID	First_Name, Last_Name, DOB, Gender, Email, Phone, Registered_Date, Cancellation_Date, Plan_ID (FK)
MEMBERSHIP_PLAN	Plan_ID	Type_Name, Price, Class_Limit, OffPeak_Access, Free_Session_per_month
TRAINER	Trainer_ID	First_Name, Last_Name, Email, Phone, Spec_ID(FK)
FITNESS_CLASS	Class_ID	Class_Name, Day_of_Week, Start_Time, End_Time, Duration, Capacity, Trainer_ID (FK), Spec_ID (FK)
CLASS_BOOKING	Booking_ID	Mem_ID (FK), Class_ID (FK), Booking_Date, Attendance_Status, Rating
TRAINING_SESSION	Session_ID	Session_Date, Session_Dur, Spec_ID (FK), Mem_ID (FK), Trainer_ID (FK), Rating
SPECIALISATION	Spec_ID	Specialisation
PAYMENT	Payment_ID	Mem_ID, Payment_Date, Payment_Type, Amount, Payment_Method

1.5 Entity-Relationship Diagram (ERD)

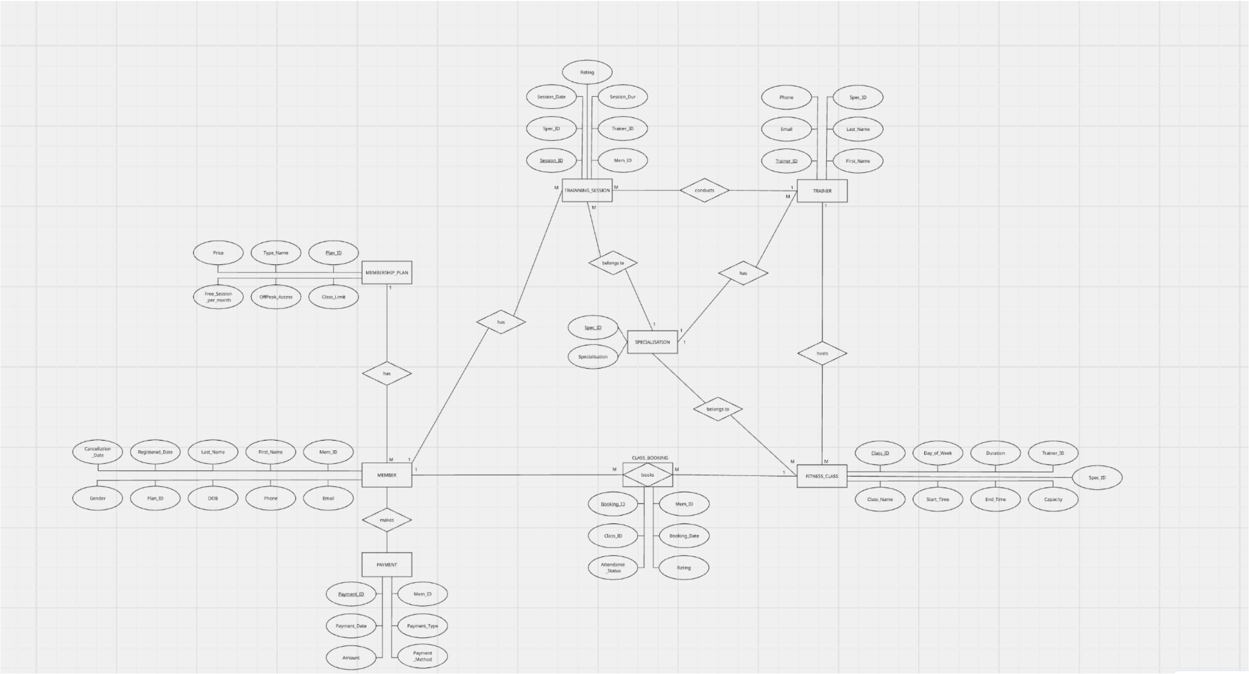
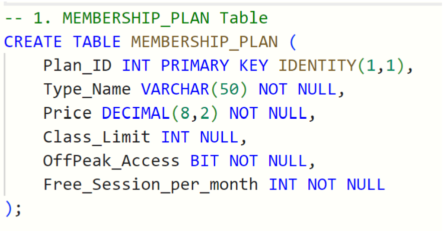


Figure A1.1 *ERD for Pulse Fitness*

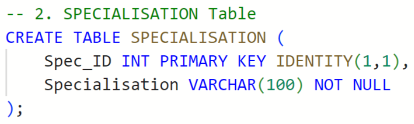
2. Database Design and Implementation for Pulse Fitness

2.1 SQL Queries for database implementation



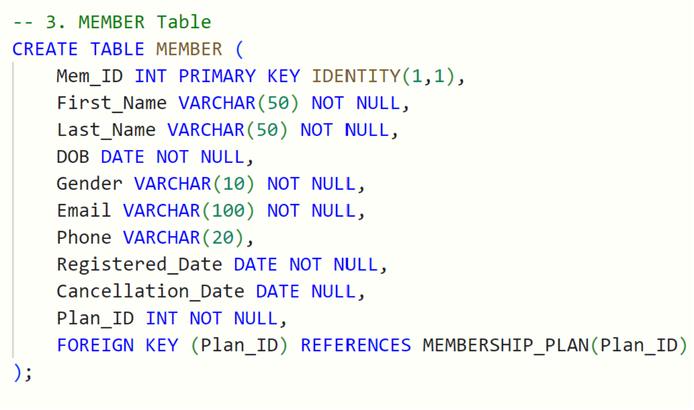
```sql
-- 1. MEMBERSHIP_PLAN Table
CREATE TABLE MEMBERSHIP_PLAN (
    Plan_ID INT PRIMARY KEY IDENTITY(1,1),
    Type_Name VARCHAR(50) NOT NULL,
    Price DECIMAL(8,2) NOT NULL,
    Class_Limit INT NULL,
    OffPeak_Access BIT NOT NULL,
    Free_Session_per_month INT NOT NULL
);
```

Figure A2.1 *Membership_Plan - Create Table Query*



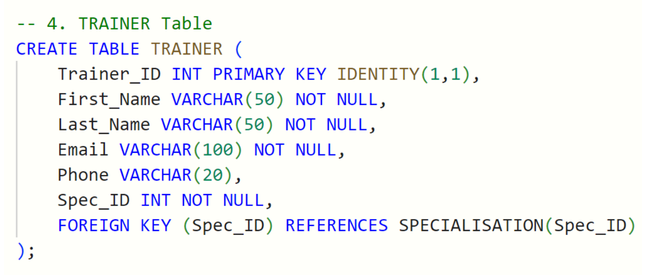
```sql
-- 2. SPECIALISATION Table
CREATE TABLE SPECIALISATION (
    Spec_ID INT PRIMARY KEY IDENTITY(1,1),
    Specialisation VARCHAR(100) NOT NULL
);
```

Figure A2.2 *Specialisation - Create Table Query*



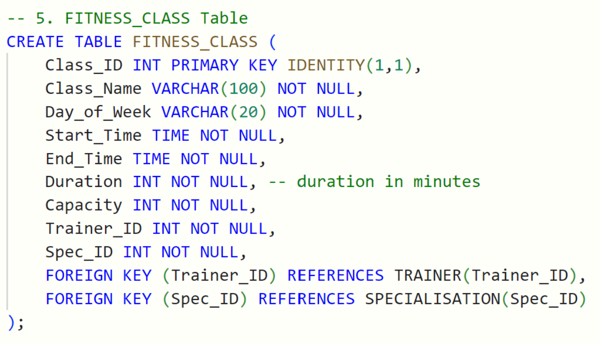
```sql
-- 3. MEMBER Table
CREATE TABLE MEMBER (
    Mem_ID INT PRIMARY KEY IDENTITY(1,1),
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    DOB DATE NOT NULL,
    Gender VARCHAR(10) NOT NULL,
    Email VARCHAR(100) NOT NULL,
    Phone VARCHAR(20),
    Registered_Date DATE NOT NULL,
    Cancellation_Date DATE NULL,
    Plan_ID INT NOT NULL,
    FOREIGN KEY (Plan_ID) REFERENCES MEMBERSHIP_PLAN(Plan_ID)
);
```

Figure A2.3 *Member - Create Table Query*



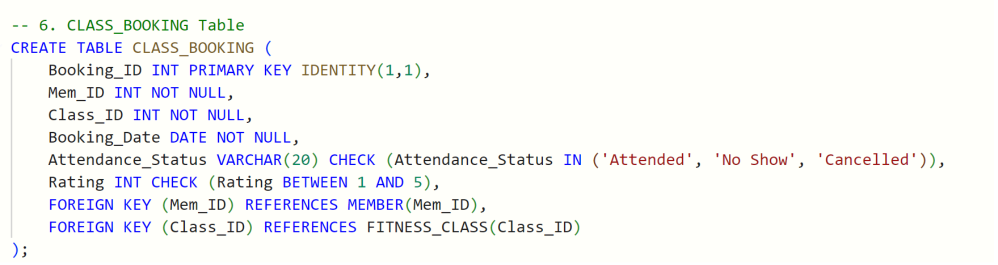
```sql
-- 4. TRAINER Table
CREATE TABLE TRAINER (
    Trainer_ID INT PRIMARY KEY IDENTITY(1,1),
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL,
    Phone VARCHAR(20),
    Spec_ID INT NOT NULL,
    FOREIGN KEY (Spec_ID) REFERENCES SPECIALISATION(Spec_ID)
);
```

Figure A2.4 *Trainer- Create Table Query*



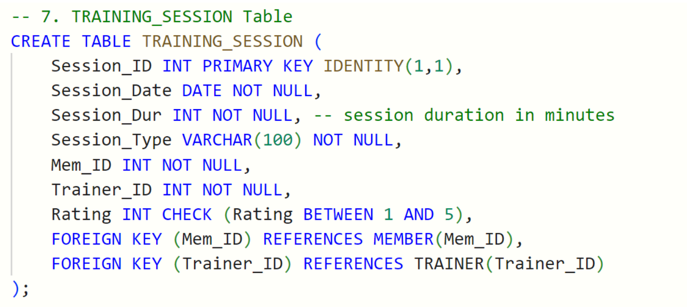
```
-- 5. FITNESS_CLASS Table
CREATE TABLE FITNESS_CLASS (
    Class_ID INT PRIMARY KEY IDENTITY(1,1),
    Class_Name VARCHAR(100) NOT NULL,
    Day_of_Week VARCHAR(20) NOT NULL,
    Start_Time TIME NOT NULL,
    End_Time TIME NOT NULL,
    Duration INT NOT NULL, -- duration in minutes
    Capacity INT NOT NULL,
    Trainer_ID INT NOT NULL,
    Spec_ID INT NOT NULL,
    FOREIGN KEY (Trainer_ID) REFERENCES TRAINER(Trainer_ID),
    FOREIGN KEY (Spec_ID) REFERENCES SPECIALISATION(Spec_ID)
);
```

Figure A2.5 *Fitness_Class- Create Table Query*



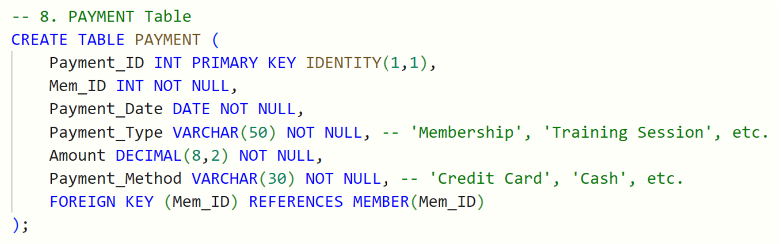
```
-- 6. CLASS_BOOKING Table
CREATE TABLE CLASS_BOOKING (
    Booking_ID INT PRIMARY KEY IDENTITY(1,1),
    Mem_ID INT NOT NULL,
    Class_ID INT NOT NULL,
    Booking_Date DATE NOT NULL,
    Attendance_Status VARCHAR(20) CHECK (Attendance_Status IN ('Attended', 'No Show', 'Cancelled')),
    Rating INT CHECK (Rating BETWEEN 1 AND 5),
    FOREIGN KEY (Mem_ID) REFERENCES MEMBER(Mem_ID),
    FOREIGN KEY (Class_ID) REFERENCES FITNESS_CLASS(Class_ID)
);
```

Figure A2.6 *Class_Booking - Create Table Query*



```
-- 7. TRAINING_SESSION Table
CREATE TABLE TRAINING_SESSION (
    Session_ID INT PRIMARY KEY IDENTITY(1,1),
    Session_Date DATE NOT NULL,
    Session_Dur INT NOT NULL, -- session duration in minutes
    Session_Type VARCHAR(100) NOT NULL,
    Mem_ID INT NOT NULL,
    Trainer_ID INT NOT NULL,
    Rating INT CHECK (Rating BETWEEN 1 AND 5),
    FOREIGN KEY (Mem_ID) REFERENCES MEMBER(Mem_ID),
    FOREIGN KEY (Trainer_ID) REFERENCES TRAINER(Trainer_ID)
);
```

Figure A2.7 *Training_Session - Create Table Query*

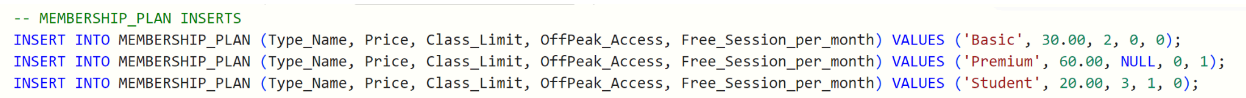


```
-- 8. PAYMENT Table
CREATE TABLE PAYMENT (
    Payment_ID INT PRIMARY KEY IDENTITY(1,1),
    Mem_ID INT NOT NULL,
    Payment_Date DATE NOT NULL,
    Payment_Type VARCHAR(50) NOT NULL, -- 'Membership', 'Training Session', etc.
    Amount DECIMAL(8,2) NOT NULL,
    Payment_Method VARCHAR(30) NOT NULL, -- 'Credit Card', 'Cash', etc.
    FOREIGN KEY (Mem_ID) REFERENCES MEMBER(Mem_ID)
);
```

Figure A2.8 *Payment - Create Table Query*

2.2 Dummy Data Integration

Membership_Plan

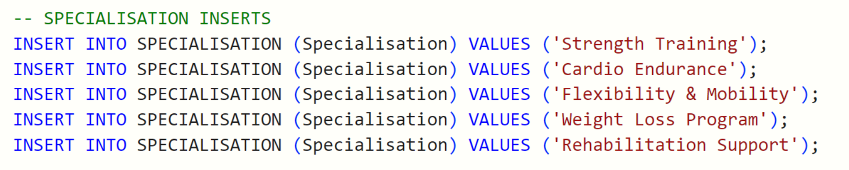


```sql
-- MEMBERSHIP_PLAN INSERTS
INSERT INTO MEMBERSHIP_PLAN (Type_Name, Price, Class_Limit, OffPeak_Access, Free_Session_per_month) VALUES ('Basic', 30.00, 2, 0, 0);
INSERT INTO MEMBERSHIP_PLAN (Type_Name, Price, Class_Limit, OffPeak_Access, Free_Session_per_month) VALUES ('Premium', 60.00, NULL, 0, 1);
INSERT INTO MEMBERSHIP_PLAN (Type_Name, Price, Class_Limit, OffPeak_Access, Free_Session_per_month) VALUES ('Student', 20.00, 3, 1, 0);
```

Figure A2.9 *Membership_Plan - Insert Query*

This table contains predefined membership plans such as Basic, Premium, and Student. Each plan specifies the price, class limit, off-peak access rights, and number of free sessions per month. Regarding the Class_Limit, null means that the Premium plan has unlimited access.

Specialisation

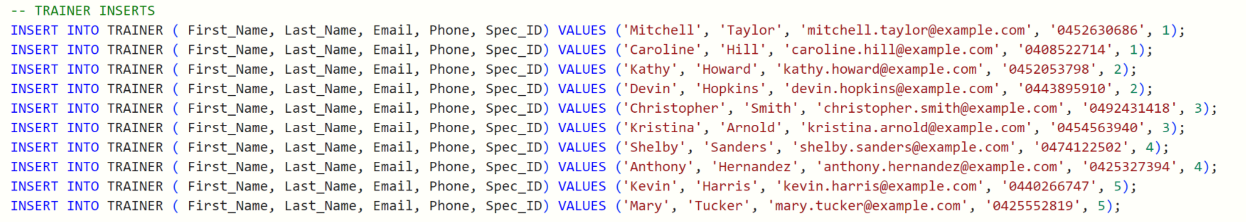


```sql
-- SPECIALISATION INSERTS
INSERT INTO SPECIALISATION (Specialisation) VALUES ('Strength Training');
INSERT INTO SPECIALISATION (Specialisation) VALUES ('Cardio Endurance');
INSERT INTO SPECIALISATION (Specialisation) VALUES ('Flexibility & Mobility');
INSERT INTO SPECIALISATION (Specialisation) VALUES ('Weight Loss Program');
INSERT INTO SPECIALISATION (Specialisation) VALUES ('Rehabilitation Support');
```

Figure A2.10 *Specialisation - Insert Query*

This table stores various fitness training specialisations like Strength Training and Cardio Endurance, which are linked to trainers through foreign keys (Spec_ID).

Trainer

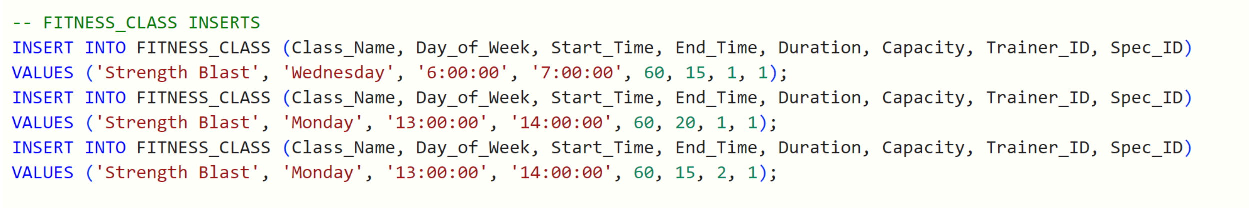


```sql
-- TRAINER INSERTS
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Mitchell', 'Taylor', 'mitchell.taylor@example.com', '0452630686', 1);
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Caroline', 'Hill', 'caroline.hill@example.com', '0408522714', 1);
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Kathy', 'Howard', 'kathy.howard@example.com', '0452053798', 2);
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Devin', 'Hopkins', 'devin.hopkins@example.com', '0443895910', 2);
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Christopher', 'Smith', 'christopher.smith@example.com', '0492431418', 3);
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Kristina', 'Arnold', 'kristina.arnold@example.com', '0454563940', 3);
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Shelby', 'Sanders', 'shelby.sanders@example.com', '0474122502', 4);
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Anthony', 'Hernandez', 'anthony.hernandez@example.com', '0425327394', 4);
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Kevin', 'Harris', 'kevin.harris@example.com', '0440266747', 5);
INSERT INTO TRAINER ( First_Name, Last_Name, Email, Phone, Spec_ID) VALUES ('Mary', 'Tucker', 'mary.tucker@example.com', '0425552819', 5);
```

Figure A2.11 *Trainer - Insert Query*

Trainer records include names, contact details, and associated specialisation IDs. The specialisation of each trainer is represented by the foreign key (Spec_ID).

Fitness_Class

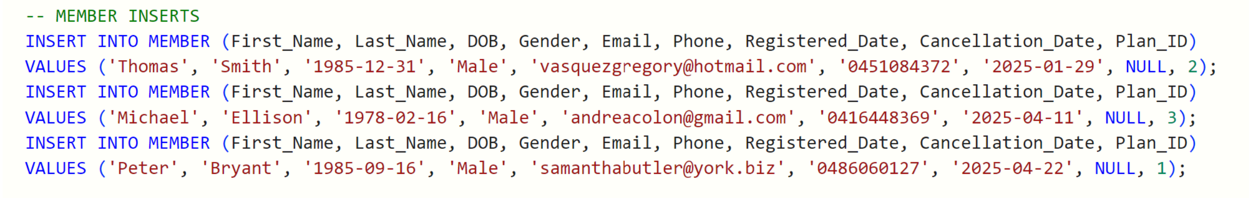


```sql
-- FITNESS_CLASS INSERTS
INSERT INTO FITNESS_CLASS (Class_Name, Day_of_Week, Start_Time, End_Time, Duration, Capacity, Trainer_ID, Spec_ID)
VALUES ('Strength Blast', 'Wednesday', '6:00:00', '7:00:00', 60, 15, 1, 1);
INSERT INTO FITNESS_CLASS (Class_Name, Day_of_Week, Start_Time, End_Time, Duration, Capacity, Trainer_ID, Spec_ID)
VALUES ('Strength Blast', 'Monday', '13:00:00', '14:00:00', 60, 20, 1, 1);
INSERT INTO FITNESS_CLASS (Class_Name, Day_of_Week, Start_Time, End_Time, Duration, Capacity, Trainer_ID, Spec_ID)
VALUES ('Strength Blast', 'Monday', '13:00:00', '14:00:00', 60, 15, 2, 1);
```

Figure A2.12 *Fitness_Class - Insert Query*

This table lists all fitness classes offered, including class name, schedule, duration, trainer, and specialisation, with constraints to ensure valid trainer and specialisation references.

Member

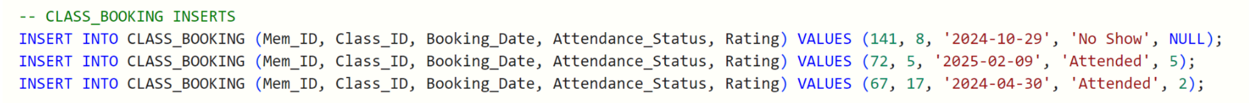


```sql
-- MEMBER INSERTS
INSERT INTO MEMBER (First_Name, Last_Name, DOB, Gender, Email, Phone, Registered_Date, Cancellation_Date, Plan_ID)
VALUES ('Thomas', 'Smith', '1985-12-31', 'Male', 'vasquezgregory@hotmail.com', '0451084372', '2025-01-29', NULL, 2);
INSERT INTO MEMBER (First_Name, Last_Name, DOB, Gender, Email, Phone, Registered_Date, Cancellation_Date, Plan_ID)
VALUES ('Michael', 'Ellison', '1978-02-16', 'Male', 'andreacolon@gmail.com', '0416448369', '2025-04-11', NULL, 3);
INSERT INTO MEMBER (First_Name, Last_Name, DOB, Gender, Email, Phone, Registered_Date, Cancellation_Date, Plan_ID)
VALUES ('Peter', 'Bryant', '1985-09-16', 'Male', 'samanthabutler@york.biz', '0486060127', '2025-04-22', NULL, 1);
```

Figure A2.13 *Member - Insert Query*

There are 150 members in the database with each member represented with personal details, contact information, registration and cancellation dates, and their chosen membership plan via a foreign key (Plan_ID).

Class_Booking

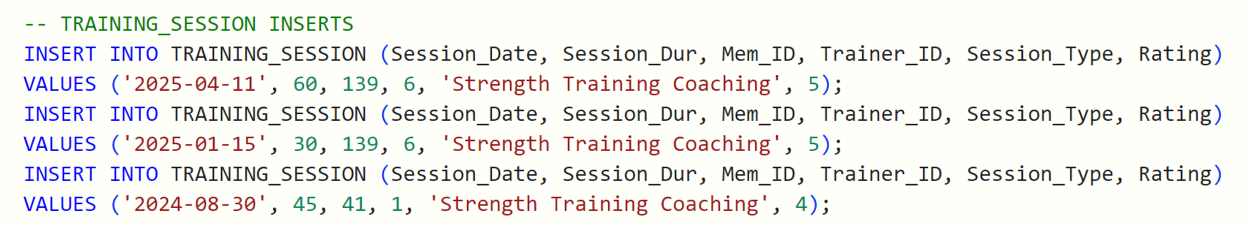


```sql
-- CLASS_BOOKING INSERTS
INSERT INTO CLASS_BOOKING (Mem_ID, Class_ID, Booking_Date, Attendance_Status, Rating) VALUES (141, 8, '2024-10-29', 'No Show', NULL);
INSERT INTO CLASS_BOOKING (Mem_ID, Class_ID, Booking_Date, Attendance_Status, Rating) VALUES (72, 5, '2025-02-09', 'Attended', 5);
INSERT INTO CLASS_BOOKING (Mem_ID, Class_ID, Booking_Date, Attendance_Status, Rating) VALUES (67, 17, '2024-04-30', 'Attended', 2);
```

Figure A2.14 *Class_Booking - Insert Query*

Records in this table represent member bookings for classes, tracking booking dates, attendance status, and optional ratings. It links to both the MEMBER and FITNESS_CLASS tables through Mem_ID and Class_ID.

Training_Session

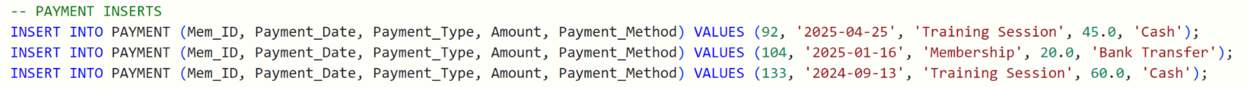


```sql
-- TRAINING_SESSION INSERTS
INSERT INTO TRAINING_SESSION (Session_Date, Session_Dur, Mem_ID, Trainer_ID, Session_Type, Rating)
VALUES ('2025-04-11', 60, 139, 6, 'Strength Training Coaching', 5);
INSERT INTO TRAINING_SESSION (Session_Date, Session_Dur, Mem_ID, Trainer_ID, Session_Type, Rating)
VALUES ('2025-01-15', 30, 139, 6, 'Strength Training Coaching', 5);
INSERT INTO TRAINING_SESSION (Session_Date, Session_Dur, Mem_ID, Trainer_ID, Session_Type, Rating)
VALUES ('2024-08-30', 45, 41, 1, 'Strength Training Coaching', 4);
```

Figure A2.15 *Training_Session - Insert Query*

This table logs one-on-one sessions between members and trainers, recording session date, duration, type, and satisfaction rating.

Payment



```sql
-- PAYMENT INSERTS
INSERT INTO PAYMENT (Mem_ID, Payment_Date, Payment_Type, Amount, Payment_Method) VALUES (92, '2025-04-25', 'Training Session', 45.0, 'Cash');
INSERT INTO PAYMENT (Mem_ID, Payment_Date, Payment_Type, Amount, Payment_Method) VALUES (104, '2025-01-16', 'Membership', 20.0, 'Bank Transfer');
INSERT INTO PAYMENT (Mem_ID, Payment_Date, Payment_Type, Amount, Payment_Method) VALUES (133, '2024-09-13', 'Training Session', 60.0, 'Cash');
```

Figure A2.16 *Payment - Insert Query*

Each payment record stores the transaction made by a member, specifying the payment type (e.g., membership or session), amount, date, and method used.

B. Business Concerns Analysis and Recommendations

The following section outlines nine key organizational challenges currently facing Pulse Fitness. Each challenge is explored through targeted SQL queries, developed to uncover meaningful insights that can guide informed and strategic decision-making by the management team.

1. Operational Efficiency

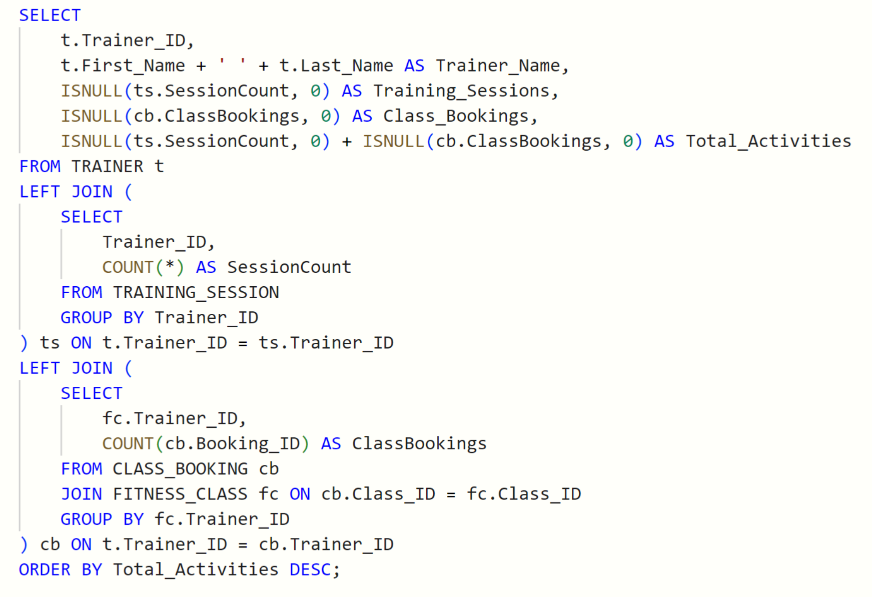
1.1 Trainer Workload Distribution

Business Concern

Trainers are one of the most important assets in Pulse Assets, in delivery performance to the customers. Pulse Fitness needs to ensure that its trainers are being allocated effectively across both personal training sessions and fitness classes. If some trainers are continuously underutilised while the others are suffering high workload, this could lead to inefficiencies in performance. A complete view of trainer workload helps ensure that staff time and customer demand are better aligned, eventually enhancing both operational efficiency and service quality.

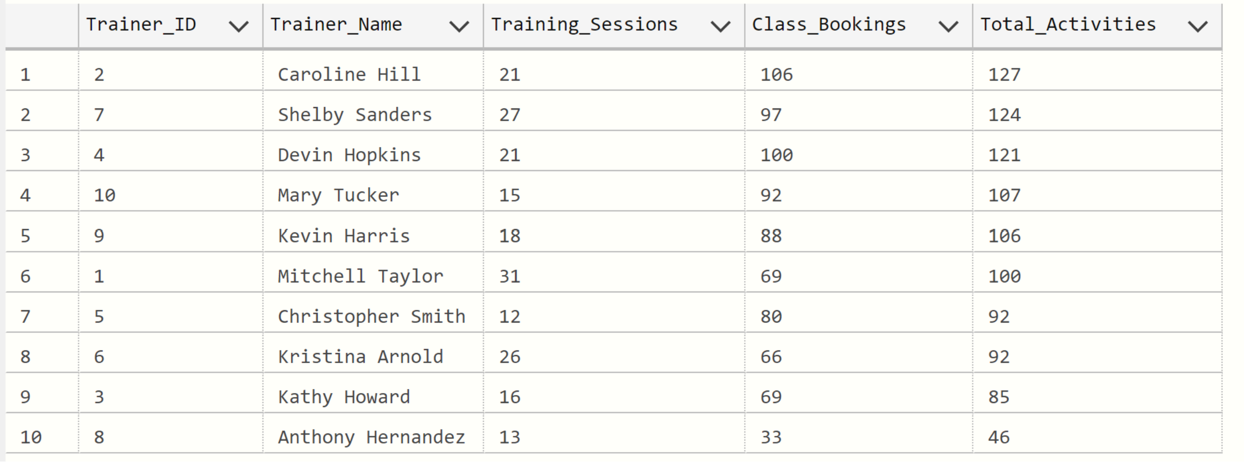
SQL Query

This query aggregates the total number of personal training sessions and group class booking handled by each trainer. Then it uses *left join* operations to ensure all trainers are included, even if they have zero activities (as we use *isnull* to replace null with 0), and calculates the total activities.



```
SELECT
    t.Trainer_ID,
    t.First_Name + ' ' + t.Last_Name AS Trainer_Name,
    ISNULL(ts.SessionCount, 0) AS Training_Sessions,
    ISNULL(cb.ClassBookings, 0) AS Class_Bookings,
    ISNULL(ts.SessionCount, 0) + ISNULL(cb.ClassBookings, 0) AS Total_Activities
FROM TRAINER t
LEFT JOIN (
    SELECT
        Trainer_ID,
        COUNT(*) AS SessionCount
    FROM TRAINING_SESSION
    GROUP BY Trainer_ID
) ts ON t.Trainer_ID = ts.Trainer_ID
LEFT JOIN (
    SELECT
        fc.Trainer_ID,
        COUNT(cb.Booking_ID) AS ClassBookings
    FROM CLASS_BOOKING cb
    JOIN FITNESS_CLASS fc ON cb.Class_ID = fc.Class_ID
    GROUP BY fc.Trainer_ID
) cb ON t.Trainer_ID = cb.Trainer_ID
ORDER BY Total_Activities DESC;
```

Output



	Trainer_ID	Trainer_Name	Training_Sessions	Class_Bookings	Total_Activities
1	2	Caroline Hill	21	106	127
2	7	Shelby Sanders	27	97	124
3	4	Devin Hopkins	21	100	121
4	10	Mary Tucker	15	92	107
5	9	Kevin Harris	18	88	106
6	1	Mitchell Taylor	31	69	100
7	5	Christopher Smith	12	80	92
8	6	Kristina Arnold	26	66	92
9	3	Kathy Howard	16	69	85
10	8	Anthony Hernandez	13	33	46

Interpretation

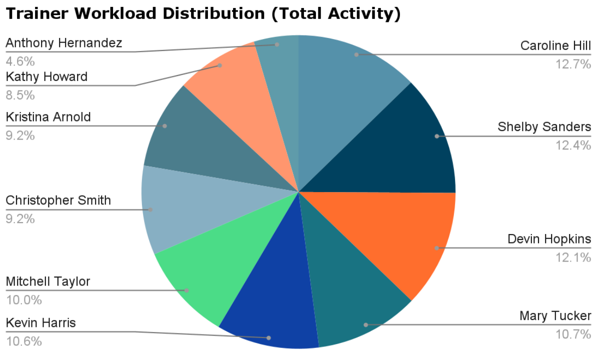


Figure B1.1 *Trainer Workload Distribution*

The output illustrates how many activities each trainer has handled, including both training sessions and class bookings. For instance, Caroline Hill recorded the highest overall activity (127), followed closely by Shelby Sanders (124) and Devin Hopkins (121). On the other hand, Anthony Hernandex had only 46 total activities, which was significantly lower than the rest of the trainers. This indicates that not all trainers are being equally utilised, which raises the concerns about the fairness in scheduling or differences in trainer popularity, specialisations or shift availability.

Implications

Understanding the trainer workload tendency allows the management team to take proactive steps in making staffing decisions. Active trainers like Caroline and Shelmy may require additional support or lighter future rosters to ensure consistent performance. Meanwhile, trainers with fewer total activities like Anthony, may require reassessment in terms of their class offerings or time slots. By addressing these imbalances, Pulse Fitness can ensure all trainers are being utilised effectively, eventually improving the quality of services, operational efficiency and customer satisfaction.

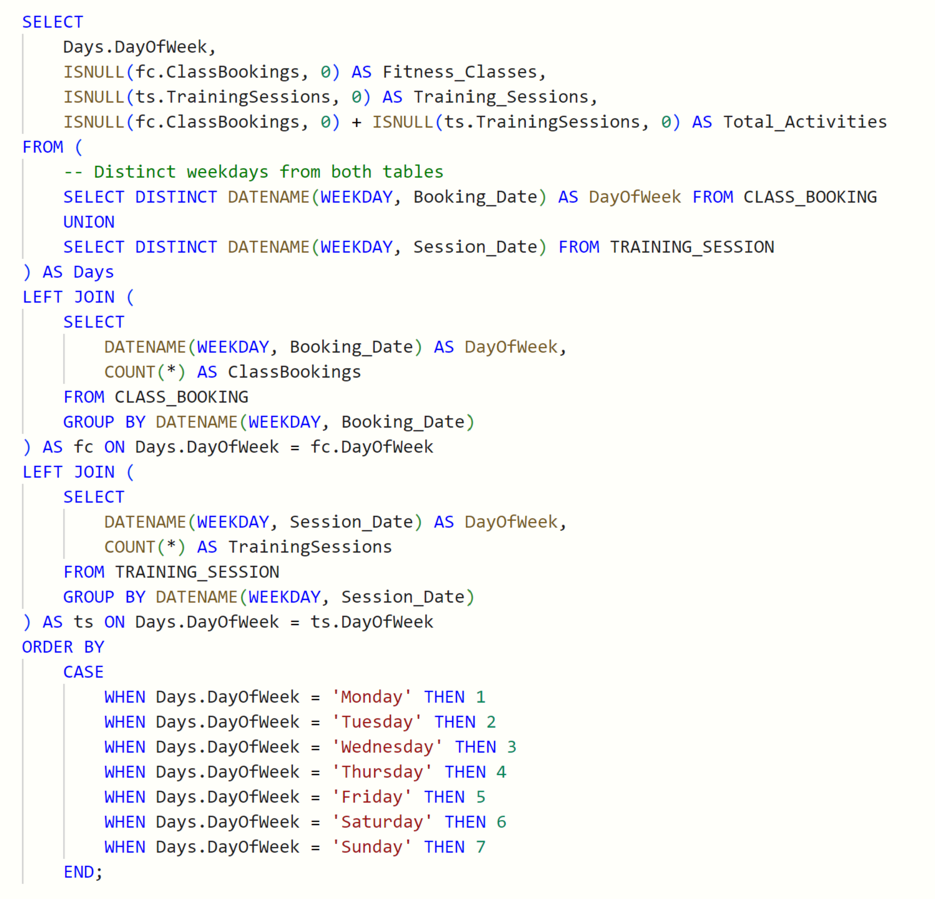
1.2 Daily Activity Load by Day of Week

Business Concern

Pulse Fitness needs to assess how member engagement through both fitness classes and personal training sessions is distributed across the week. This information is critical for optimising resource utilisation, including trainers, time slots and space. By understanding the overperformed and underperformed days of week, management can tailor the operations to meet member demand more effectively and enhance the operations regarding time and facilities.

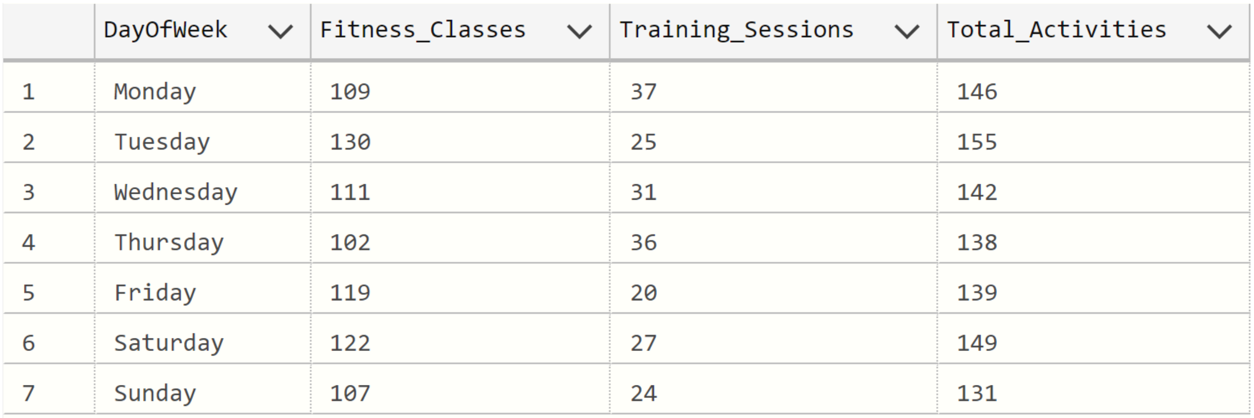
SQL Query

The SQL Query below calculates the number of class bookings and training sessions that occur on each day of week through *group by* and *datename* functions. Regarding the fitness class, this will be based on the booking_date from CLASS_BOOKING table. Meanwhile, for the training session, the day of week will be calculated from the Session_Date in TRAINING_SESSION table. By combining results from both activities and ordering the days correctly (from Monday to Sunday - encoding them with number), it provides a clear picture of daily workload patterns.



```
SELECT
    Days.DayOfWeek,
    ISNULL(fc.ClassBookings, 0) AS Fitness_Classes,
    ISNULL(ts.TrainingSessions, 0) AS Training_Sessions,
    ISNULL(fc.ClassBookings, 0) + ISNULL(ts.TrainingSessions, 0) AS Total_Activities
FROM (
    -- Distinct weekdays from both tables
    SELECT DISTINCT DATENAME(WEEKDAY, Booking_Date) AS DayOfWeek FROM CLASS_BOOKING
    UNION
    SELECT DISTINCT DATENAME(WEEKDAY, Session_Date) FROM TRAINING_SESSION
) AS Days
LEFT JOIN (
    SELECT
        DATENAME(WEEKDAY, Booking_Date) AS DayOfWeek,
        COUNT(*) AS ClassBookings
    FROM CLASS_BOOKING
    GROUP BY DATENAME(WEEKDAY, Booking_Date)
) AS fc ON Days.DayOfWeek = fc.DayOfWeek
LEFT JOIN (
    SELECT
        DATENAME(WEEKDAY, Session_Date) AS DayOfWeek,
        COUNT(*) AS TrainingSessions
    FROM TRAINING_SESSION
    GROUP BY DATENAME(WEEKDAY, Session_Date)
) AS ts ON Days.DayOfWeek = ts.DayOfWeek
ORDER BY
    CASE
        WHEN Days.DayOfWeek = 'Monday' THEN 1
        WHEN Days.DayOfWeek = 'Tuesday' THEN 2
        WHEN Days.DayOfWeek = 'Wednesday' THEN 3
        WHEN Days.DayOfWeek = 'Thursday' THEN 4
        WHEN Days.DayOfWeek = 'Friday' THEN 5
        WHEN Days.DayOfWeek = 'Saturday' THEN 6
        WHEN Days.DayOfWeek = 'Sunday' THEN 7
    END;
```

Output



	DayOfWeek	Fitness_Classes	Training_Sessions	Total_Activities
1	Monday	109	37	146
2	Tuesday	130	25	155
3	Wednesday	111	31	142
4	Thursday	102	36	138
5	Friday	119	20	139
6	Saturday	122	27	149
7	Sunday	107	24	131

Interpretation

Based on the output, Tuesday is the busiest day overall, with 155 combined activities, including 130 fitness class bookings (ranked first) and 25 training sessions. Saturday followed closely with 149 total activities. Surprisingly, while Sunday had the lowest activity (131), the difference across the week was relatively small, indicating consistent engagement throughout the week. Training Sessions were notably most popular on Monday and Thursday, while class bookings were spread more evenly across the week.

Implication

With this insight, Pulse Fitness can make decisions regarding resource planning and class scheduling. On high volume days like Tuesday and Saturday, management can consider increasing the staff availability or expanding class offerings to meet the high demand and avoid overcrowding. In addition, Pulse Fitness can allocate some high-demand classes or run special promotional campaigns to attract more members to lighter days such as Thursday or Sunday. Moreover, knowing which days training sessions are more popular can also support the scheduling of trainers more efficiently. These suggested adjustments help ensure the resources are being used effectively and members are getting the support they need.

2. Revenue Generation

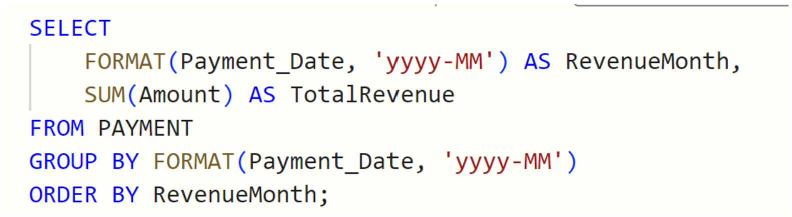
2.1 Revenue Over Time

Business Concern

Monitoring month revenue trends is essential for Pulse Fitness to optimise cash flows and plan for seasonal changes. With the following query, the management team can identify patterns related to promotions, membership patterns and member behavior throughout the year, therefore helping them to develop marketing campaigns, pricing adjustments and staffing strategies.

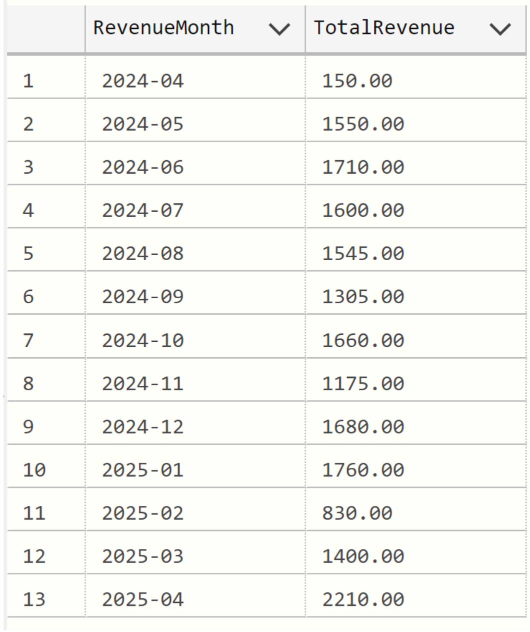
SQL Query

The query below calculates total monthly revenue by formatting the Payment_Date as year_month format, then aggregating the total amount of revenue by Payment_Date.



```
SELECT
    FORMAT(Payment_Date, 'yyyy-MM') AS RevenueMonth,
    SUM(Amount) AS TotalRevenue
FROM PAYMENT
GROUP BY FORMAT(Payment_Date, 'yyyy-MM')
ORDER BY RevenueMonth;
```

Output



	RevenueMonth	TotalRevenue
1	2024-04	150.00
2	2024-05	1550.00
3	2024-06	1710.00
4	2024-07	1600.00
5	2024-08	1545.00
6	2024-09	1305.00
7	2024-10	1660.00
8	2024-11	1175.00
9	2024-12	1680.00
10	2025-01	1760.00
11	2025-02	830.00
12	2025-03	1400.00
13	2025-04	2210.00

Interpretation

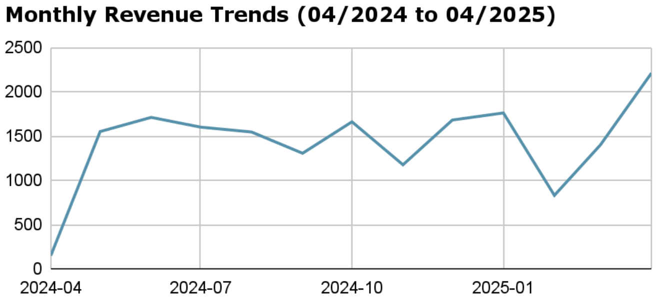


Figure B2.1 *Member Distribution by Membership Plan*

The revenue data indicates fluctuations across the year, from 04/2024 to 04/2025, influenced by seasonal customer behavior. From 05/2024 to 08/2024, the revenue levels remained steady during this mid-year period. However, at the end of the year, ranging from 10/2024 to 01/2025, although it experienced a decline in 11/2024, the revenue stayed high and reached a New Year spike, indicating the New Year sales boost which could be attributed to the New Year campaigns or signup/renewal period. Following this, the revenue just sharply declined to $830 after a high revenue month. At the end of 04/2025, this is the highest monthly revenue among the other months in thes dataset, suggesting peak renewal period (starting of second quarter) or new promotions.

Implications

Based on the observed trends, Pulse Fitness can strategically align its operational and marketing decisions around different revenue periods:

- **During the low-revenue months,** Pulse Fitness can launch promotional campaigns such as discounted plans, referral policy, etc. so as to smooth out the revenue compared to other periods of time.
- **During the high-revenue months,** as the customers tend to spend more during these months due to the renewal programs, promotional programs, etc., Pulse Fitness can implement upselling strategies (extra training session bundles or add-ons) because the members are already engaged and spending. Furthermore, our centre can ensure more trainer availability and staff support during peak sign-up periods. Before these peaks, we can launch campaigns in order to build momentum for revenue generation.
- **During the after-peak and stable months,** we can introduce the retention campaigns/policies in order to maintain consistent service quality and staff availability.

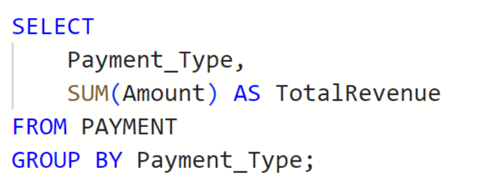
2.2 Revenue and Revenue per Transaction by Payment Type

Business Concerns

To effectively manage and grow its revenue streams, Pulse Fitness needs to understand not only the breakdown of revenue, whether from memberships or pay-per-use services like training sessions, but also the value generated per offerings. This analysis provides key insights of performance of different revenue sources for effective pricing strategy, bundling and service positioning to enhance future profitability.

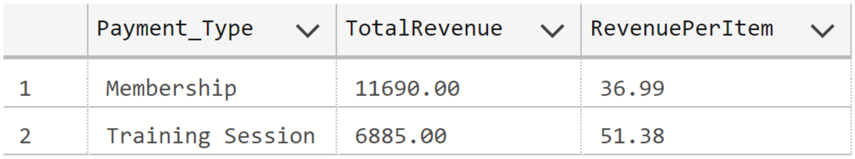
SQL Query

The query below calculates the total revenue generated by each payment_type through group by function.



```
SELECT
    Payment_Type,
    SUM(Amount) AS TotalRevenue
FROM PAYMENT
GROUP BY Payment_Type;
```

Output



	Payment_Type	TotalRevenue	RevenuePerItem
1	Membership	11690.00	36.99
2	Training Session	6885.00	51.38

Interpretation

The output reveals Membership plans generated $11690.00 (around 63% of overall revenue) an, while Training Sessions contributed $6885.00 or 37% Despite memberships being the prominent source of revenue, the average revenue per transaction is significantly higher for training sessions compared to memberships. While membership plans remained as a stable source of income, training sessions also show a significant additional income source.

Implications

This analysis suggests that while Memberships offer predictable recurring income and form the financial foundation of Pulse Fitness, training sessions are a high-margin service with high value per transaction. Hence, with the purpose of maximising revenue, management should consider following strategies:

- Upselling the training sessions to the basic or student members by promoting their high value and potential for personalisation
- Bundled services to raise overall transaction value, in order to utilise the demand for membership plans along with the high-margin products for added revenue.
- Positioning the value propositions of personal training as premium and personalised experiences.
- Adjust the pricing tiers or enhance the value of existing plans through added plans, in order to encourage members to register Premium Plan.

3. Membership Pricing Strategies

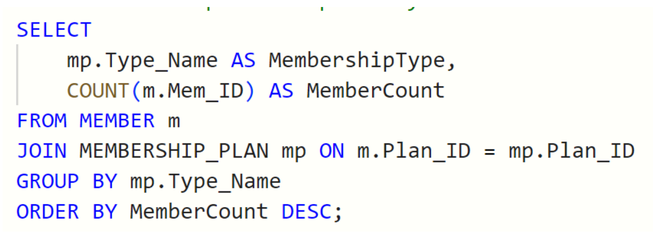
3.1 Membership Plan Popularity

Business Concern

The membership plan is one of the main products of Pulse Fitness. Therefore, understanding how members are distributed across different membership plans including Basic, Premium and Student plan, is essential for shaping effective pricing models, identifying upselling opportunities and tailoring marketing strategies.

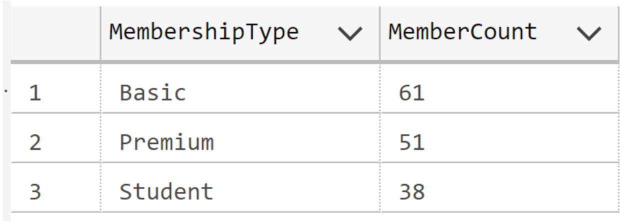
SQL Query

The following query retrieves the number of members enrolled in each membership plan, providing a breakdown by plan type by categorising by the type_name (represented membership plan) and calculating the total members for each plan through count function of Mem_ID.



```sql
SELECT
    mp.Type_Name AS MembershipType,
    COUNT(m.Mem_ID) AS MemberCount
FROM MEMBER m
JOIN MEMBERSHIP_PLAN mp ON m.Plan_ID = mp.Plan_ID
GROUP BY mp.Type_Name
ORDER BY MemberCount DESC;
```

Output



	MembershipType	MemberCount
1	Basic	61
2	Premium	51
3	Student	38

Interpretation

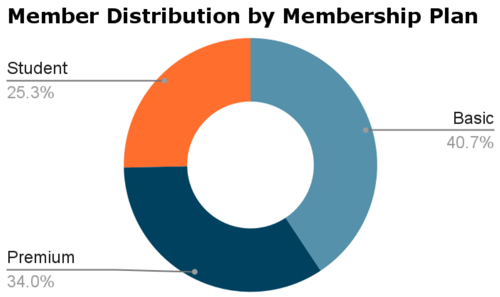


Figure B3.1 *Member Distribution by Membership Plan*

Among 150 members, there are 40.7% of members subscribing to the Basic plan, and 34% of them subscribing to the Premium plan. The rest of them (25.3%) are on the Student plan. While the Basic plan has the highest uptake, the overall distribution is relatively balanced , suggesting although many members gravitate towards affordable plans, a significant portion still value the extended benefits of Premium membership.

Implications

The Basic plan is the most subscribed, suggesting that this is the most affordable and safest option for new members. The Premium plan uptake is also strong, nearly meeting the proportion of basic plan members, indicating that the tendency of members are willing to pay for higher value such as more free sessions or unlimited fitness classes despite being cost-sensitive. This indicates effective and useful services delivery from Pulse Fitness. The Student plan is the least popular, which could reflect ineffective student targeting strategy. Based on the current trend, we can review the customer preferences, especially for the members with Basic plan, in order to develop an upselling strategy towards Premium. Furthermore, considering the weak student plan proportions, we can consider the price adjustment for this kind of plan along with modified benefits (based on surveys for students) and targeted marketing strategy towards schools or universities with time-limited student promotions.

4. Customer Retention

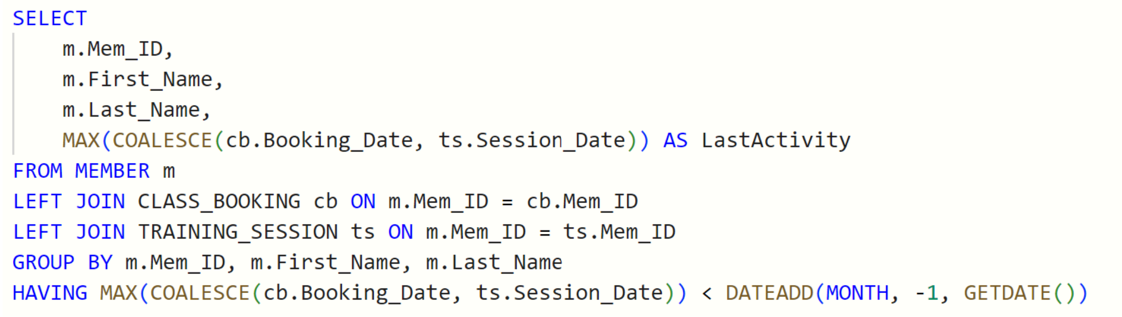
4.1 Member Inactivity Detection

Business Concern

To maintain strong customer retention, Pulse Fitness must proactively monitor member activity to prevent them being inactive (without attending classes or training sessions for extended periods as they have a tendency of cancelling their membership plan. By addressing these risks in a timely manner, the business can initiate timely intervention to mitigate potential churn.

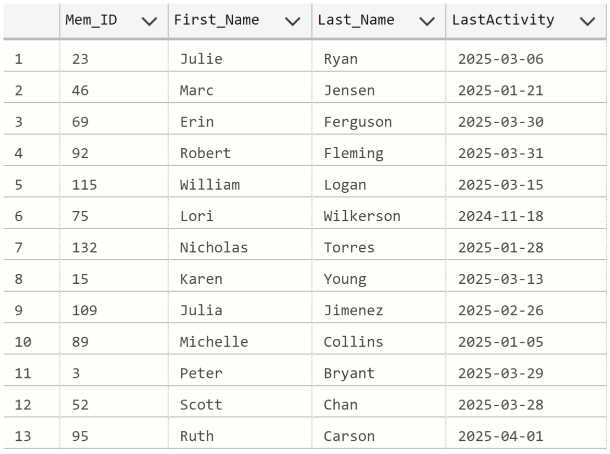
SQL Query

This query identifies members whose most recent activity (either class or training session) occurred more than a month ago. This can be achieved by using max(coalesce()) to identify the recent date of activity, then filtering with max(coalesce()) less than a month from the current date (retrieved from getdate()).



```sql
SELECT
    m.Mem_ID,
    m.First_Name,
    m.Last_Name,
    MAX(COALESCE(cb.Booking_Date, ts.Session_Date)) AS LastActivity
FROM MEMBER m
LEFT JOIN CLASS_BOOKING cb ON m.Mem_ID = cb.Mem_ID
LEFT JOIN TRAINING_SESSION ts ON m.Mem_ID = ts.Mem_ID
GROUP BY m.Mem_ID, m.First_Name, m.Last_Name
HAVING MAX(COALESCE(cb.Booking_Date, ts.Session_Date)) < DATEADD(MONTH, -1, GETDATE())
```

Output



	Mem_ID	First_Name	Last_Name	LastActivity
1	23	Julie	Ryan	2025-03-06
2	46	Marc	Jensen	2025-01-21
3	69	Erin	Ferguson	2025-03-30
4	92	Robert	Fleming	2025-03-31
5	115	William	Logan	2025-03-15
6	75	Lori	Wilkerson	2024-11-18
7	132	Nicholas	Torres	2025-01-28
8	15	Karen	Young	2025-03-13
9	109	Julia	Jimenez	2025-02-26
10	89	Michelle	Collins	2025-01-05
11	3	Peter	Bryant	2025-03-29
12	52	Scott	Chan	2025-03-28
13	95	Ruth	Carson	2025-04-01

Interpretation

The query identifies members whose last activity, either a fitness class or training session, occurred a month ago. The results include members such as Julie Ryan, Marc Jensen, etc., whose last activity was around January, March of 2025. This list provides a clear view of potential churn candidates, enabling Pulse Fitness to track customer health and behavioral drop-offs.

Implication

The presence of inactive members represents a risk to both revenue stability and customer retention. Management should treat these members as a potential segment for targeted re-engagement strategy. By executing retention strategies such as sending reminder emails, exclusive reactivation offers or personal check-in messages from trainers, Pulse Fitness could understand the rationale why members are inactive. As a result, this can increase the likelihood of reactivation. Furthermore, by examining the membership plan or which classes they attended during their last activity period, management will be able to identify problems in service delivery or satisfaction issues.

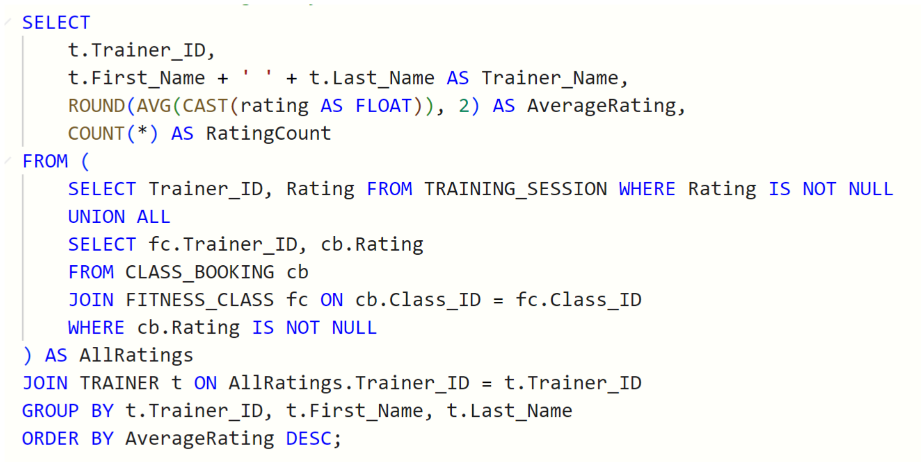
4.2 Trainer Rating Analysis

Business Concern

While it is important to monitor the number of sessions and classes each trainer conducts for operational planning, it is essential to understand their performance using their ratings from members. A high workload does not always mean high satisfaction. Hence, Pulse Fitness needs to assess the quality of their deliveries to see if the workload impacts member satisfaction or not. Therefore, the findings will guide decisions regarding staffing, coaching and recognition.

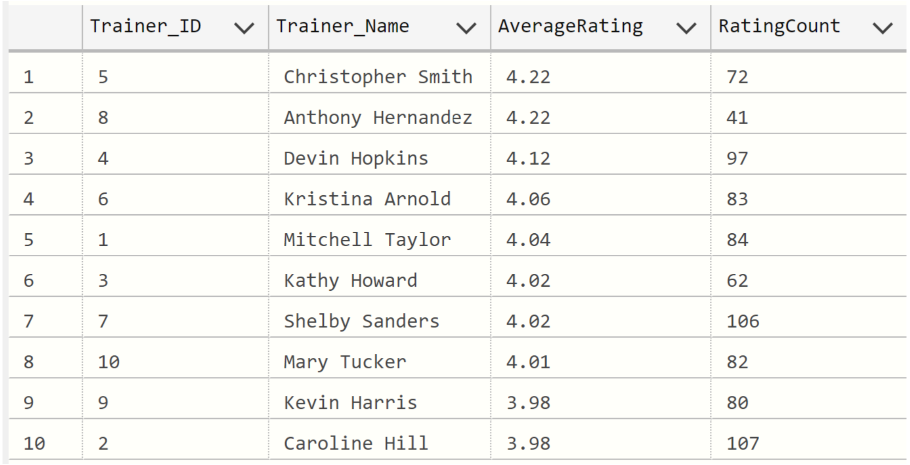
SQL Query

This query calculates the average rating and total number of ratings for each trainer by combining feedback from both training sessions and fitness classes. It merges the two sources using a UNION ALL, then groups the results by trainer to assess overall performance. By ordering trainers by their average rating, the query highlights top performers based on member satisfaction.



```sql
SELECT
    t.Trainer_ID,
    t.First_Name + ' ' + t.Last_Name AS Trainer_Name,
    ROUND(AVG(CAST(rating AS FLOAT)), 2) AS AverageRating,
    COUNT(*) AS RatingCount
FROM (
    SELECT Trainer_ID, Rating FROM TRAINING_SESSION WHERE Rating IS NOT NULL
    UNION ALL
    SELECT fc.Trainer_ID, cb.Rating
    FROM CLASS_BOOKING cb
    JOIN FITNESS_CLASS fc ON cb.Class_ID = fc.Class_ID
    WHERE cb.Rating IS NOT NULL
) AS AllRatings
JOIN TRAINER t ON AllRatings.Trainer_ID = t.Trainer_ID
GROUP BY t.Trainer_ID, t.First_Name, t.Last_Name
ORDER BY AverageRating DESC;
```

Output



	Trainer_ID	Trainer_Name	AverageRating	RatingCount
1	5	Christopher Smith	4.22	72
2	8	Anthony Hernandez	4.22	41
3	4	Devin Hopkins	4.12	97
4	6	Kristina Arnold	4.06	83
5	1	Mitchell Taylor	4.04	84
6	3	Kathy Howard	4.02	62
7	7	Shelby Sanders	4.02	106
8	10	Mary Tucker	4.01	82
9	9	Kevin Harris	3.98	80
10	2	Caroline Hill	3.98	107

Interpretation

The data shows that Christopher Smith and Anthony Hernandex share the highest average ratings (4.22) but their total workloads are relatively modest compared to others. In contrast, for the active trainers like Caroline, Shelby and Devin all have high workload but average ratings just more time and energy to focus on client satisfaction, leading to better feedback. However, rating volume is also meaningful. Caroline may have the lowest rating score but the highest number ratings, suggesting her rating coming from broad feedback. However, Anthony has only 41 suggesting that his high average rating score may be less stable.

Implication

This analysis allows Pulse Fitness to evaluate trainer performance besides their workload analysis. For trainers with high workloads but underperforming, it may be the time to reduce the workloads for them, offer assisting efforts or implement burnout prevention strategies. On the other hand, highly rated but underused trainers could be featured more prominently in marketing or be assigned to personal sessions with high-value members/premium sessions.

In addition, by leveraging rating insights, Pulse Fitness can assign high-performing trainers to members at high-risk churn identified by the aforementioned query, in order to reinforce member satisfaction and loyalty. Furthermore, mentoring programs can be established where top-rated trainers mentor underperformed trainers, helping uplift the overall quality of services and supporting long-term retention

5. Resource Utilisation

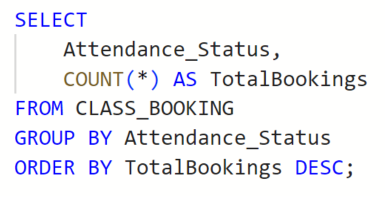
5.1 Attendance Behavior Analysis

Business Concern

Understanding the attendance behavior is important for managing class utilisation, optimising staff preparation and reducing last-minute cancellations or no-shows tendency. High rates of absenteeism can lead to underutilised time slots, wasted resources and reduced class availability for committed members. With this analysis as foundation, the management team can identify key pain points in the booking and engagement process, aiming for better member service delivery.

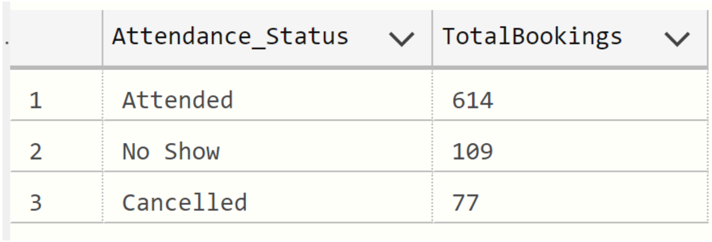
SQL Query

The query retrieves a summary of attendance behavior by counting the total number of class bookings grouped by their Attendance_Status. It categorises the results into three possible outcomes including **Attended, No Show, and Cancelled,** in order to to help quantify how many members actually attended versus missed or cancelled their booked classes.



```sql
SELECT
    Attendance_Status,
    COUNT(*) AS TotalBookings
FROM CLASS_BOOKING
GROUP BY Attendance_Status
ORDER BY TotalBookings DESC;
```

Output



	Attendance_Status	TotalBookings
1	Attended	614
2	No Show	109
3	Cancelled	77

Interpretation

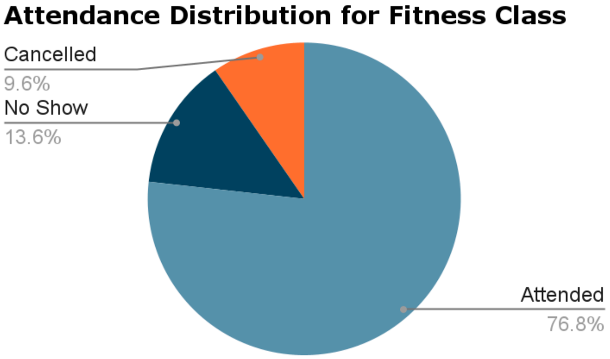


Figure A3.1 *Member Distribution by Membership Plan*

The query examines the attendance behavior across all fitness class booking based on their attendance_status. The output reveals how many members attended, missed or cancelled their classes. Although 614 of the bookings (around 76.8%) resulted in actual attendance, there were 23.2% of them unable to attend. This non-attendance rate is still significant enough to affect class utilisation, trainer preparation, revenue and overall scheduling.

Implications

This insight helps management identify areas of improvement in attendance reliability and class engagement as high rates of no-shows can lead to wasted resources and preparations, especially for high-demand classes or peak-hour classes. In order to reduce the rate, Pulse Fitness can address this by implementing reminders to reduce forgetfulness, reallocation of schedule instead of cancelling, or introducing programs for consistent commitment. Furthermore, reviewing structural issues related to class scheduling, formats or capacity regarding these high-rates of cancellations/no-show can reveal insights for improvement. In addition, management can introduce an early reminder message for the registered member so that the others, who were not able to book the class, would have the opportunity to attend. This helps the centre in class utilisation, avoiding wasting resources while maximising revenue.

5.2 Specialisation Demand

Business Concern

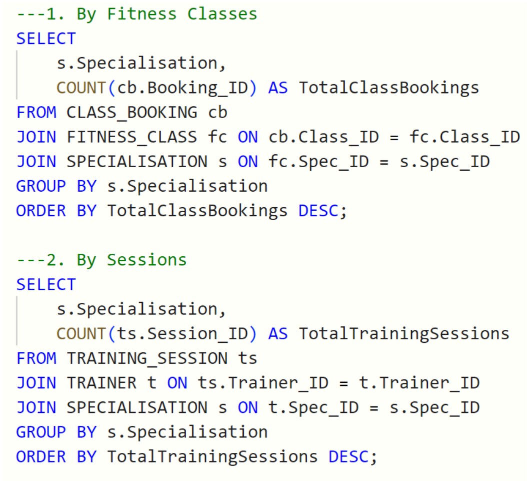
Pulse Fitness must ensure that its service offerings align with the customer demand across different areas of fitness concerns (such as Strength Training, Cardio Endurance, Rehabilitation Support, etc.), in both Training Sessions and Fitness Classes. By analysing which specialisations are most frequently booked allows management to allocate the trainer time effectively, optimise scheduling and make decisions regarding hiring and program development.

SQL Query

These two queries analyse the number of bookings for each specialisation:

- The first query counts how many times each specialisation was booked through group fitness classes.
- The second query counts the number of bookings for each specialisation through personal training sessions.

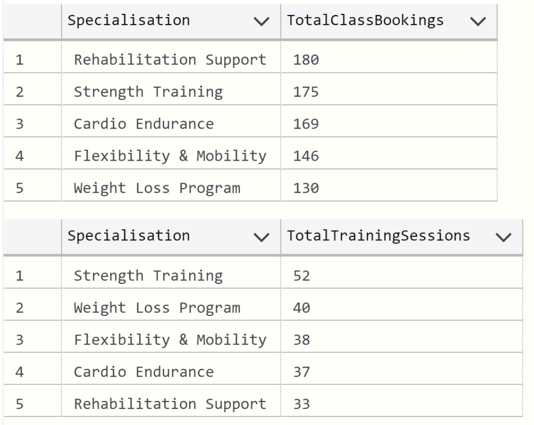
The goal is to compare demand patterns across both service formats and identify where member preferences lie.



```sql
---1. By Fitness Classes
SELECT
    s.Specialisation,
    COUNT(cb.Booking_ID) AS TotalClassBookings
FROM CLASS_BOOKING cb
JOIN FITNESS_CLASS fc ON cb.Class_ID = fc.Class_ID
JOIN SPECIALISATION s ON fc.Spec_ID = s.Spec_ID
GROUP BY s.Specialisation
ORDER BY TotalClassBookings DESC;

---2. By Sessions
SELECT
    s.Specialisation,
    COUNT(ts.Session_ID) AS TotalTrainingSessions
FROM TRAINING_SESSION ts
JOIN TRAINER t ON ts.Trainer_ID = t.Trainer_ID
JOIN SPECIALISATION s ON t.Spec_ID = s.Spec_ID
GROUP BY s.Specialisation
ORDER BY TotalTrainingSessions DESC;
```

Output



	Specialisation	TotalClassBookings
1	Rehabilitation Support	180
2	Strength Training	175
3	Cardio Endurance	169
4	Flexibility & Mobility	146
5	Weight Loss Program	130

	Specialisation	TotalTrainingSessions
1	Strength Training	52
2	Weight Loss Program	40
3	Flexibility & Mobility	38
4	Cardio Endurance	37
5	Rehabilitation Support	33

Interpretation

These 2 queries separately assess demand for each specialisation in group fitness class and in training sessions. As evidenced by the output, “Rehabilitation Support” has the highest number of class bookings (180) but a low number of personal training sessions (33), suggesting that people gravitate towards this kind of training in groups rather than individuals. On the other hand, Strength Training with 175 class bookings, which is ranked second, has the highest total sessions. Furthermore, while the weight loss program ranked second among the other specialisation in terms of personal sessions (40), it has the least class bookings (130). To conclude, some specialisations such as Rehabilitation Support, are more attractive in group format, while Strength Training and Weight Loss Program may be more appealing to those seeking for personalised coaching.

Implication

These queries provide management with targeted direction for service planning and marketing strategies. Specialisations like Strength Training, which show high demand across both formats, may need investment in time slots, resources and bundled programs of fitness classes and personal sessions. In contrast, for specialisation that underperforms in group format or personal format only, can be considered to be promoted with other formats, in which it underperforms for revenue maximisation. Last but not least, these insights help inform trainer hiring priorities, ensuring the recruitment aligns with service demand.