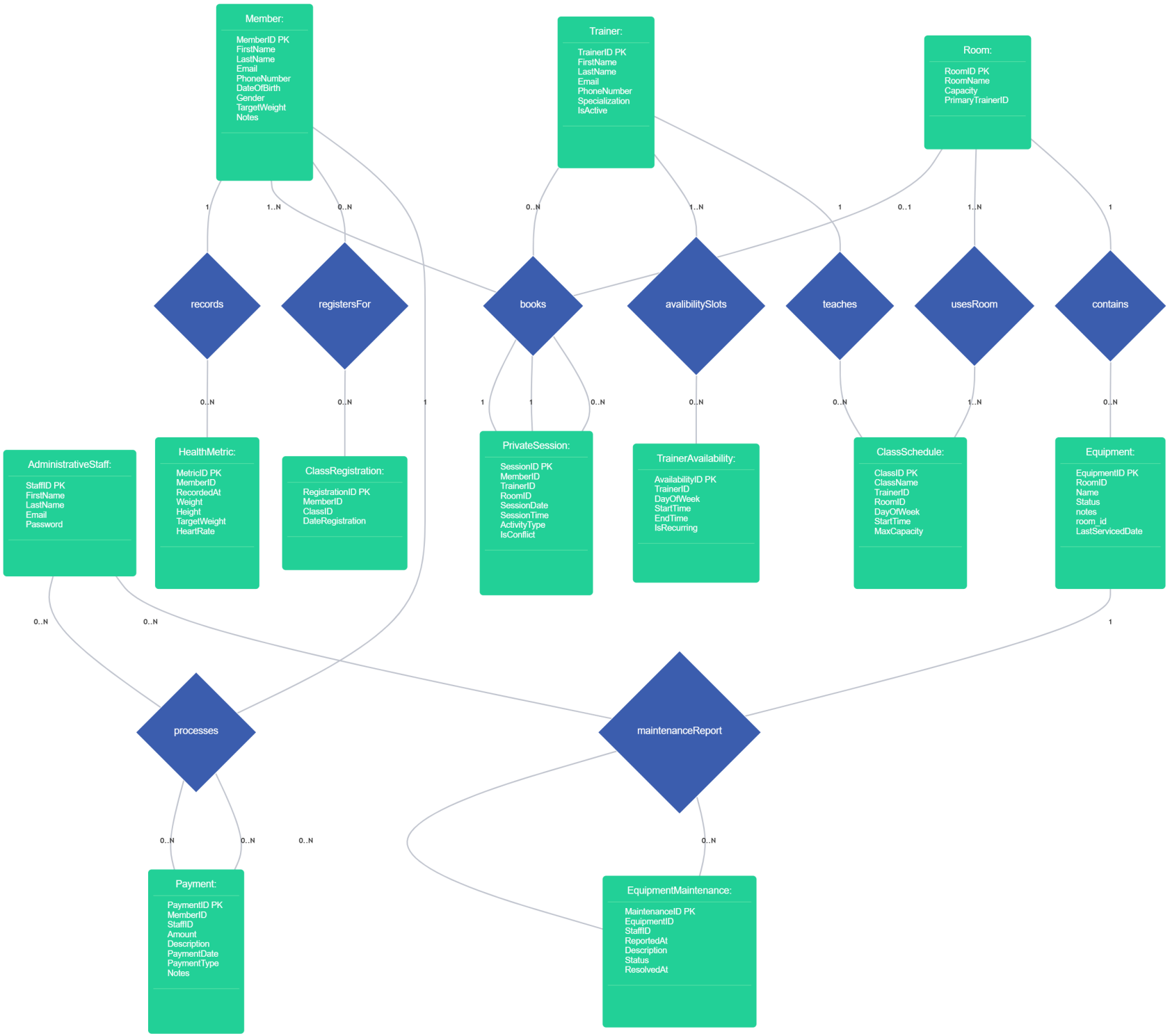
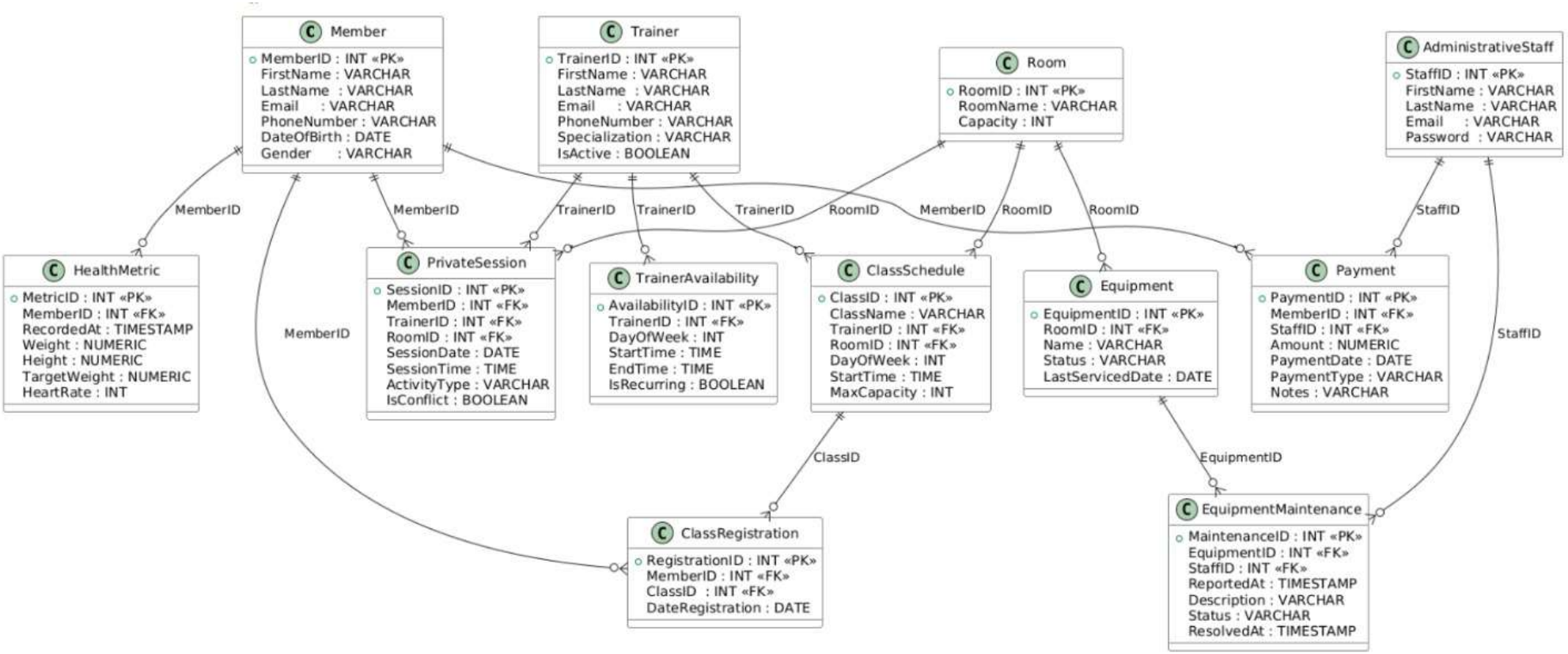

Member:
MemberID PK
FirstName
LastName
Email
PhoneNumber
DateOfBirth
Gender
TargetWeight
Notes
Trainer:
TrainerID PK
FirstName
LastName
Email
PhoneNumber
Specialization
IsActive
Room:
RoomID PK
RoomName
Capacity
PrimaryTrainerID
records
registersFor
books
availibilitySlots
teaches
usesRoom
contains
HealthMetric:
MetricID PK
MemberID
RecordedAt
Weight
Height
TargetWeight
HeartRate
ClassRegistration:
RegistrationID PK
MemberID
ClassID
DateRegistration
PrivateSession:
SessionID PK
MemberID
TrainerID
RoomID
SessionDate
SessionTime
ActivityType
IsConflict
TrainerAvailability:
AvailabilityID PK
TrainerID
DayOfWeek
StartTime
EndTime
IsRecurring
ClassSchedule:
ClassID PK
ClassName
TrainerID
RoomID
DayOfWeek
StartTime
MaxCapacity
Equipment:
EquipmentID PK
RoomID
Name
Status
notes
room_id
LastServicedDate
AdministrativeStaff:
StaffID PK
FirstName
LastName
Email
Password
processes
maintenanceReport
Payment:
PaymentID PK
MemberID
StaffID
Amount
Description
PaymentDate
PaymentType
Notes
EquipmentMaintenance:
MaintenanceID PK
EquipmentID
StaffID
ReportedAt
Description
Status
ResolvedAt


Member
MemberID : INT «PK»
FirstName : VARCHAR
LastName : VARCHAR
Email : VARCHAR
PhoneNumber : VARCHAR
DateOfBirth : DATE
Gender : VARCHAR
Trainer
TrainerID : INT «PK»
FirstName : VARCHAR
LastName : VARCHAR
Email : VARCHAR
PhoneNumber : VARCHAR
Specialization : VARCHAR
IsActive : BOOLEAN
Room
RoomID : INT «PK»
RoomName : VARCHAR
Capacity : INT
AdministrativeStaff
StaffID : INT «PK»
FirstName : VARCHAR
LastName : VARCHAR
Email : VARCHAR
Password : VARCHAR
HealthMetric
MetricID : INT «PK»
MemberID : INT «FK»
RecordedAt : TIMESTAMP
Weight : NUMERIC
Height : NUMERIC
TargetWeight : NUMERIC
HeartRate : INT
PrivateSession
SessionID : INT «PK»
MemberID : INT «FK»
TrainerID : INT «FK»
RoomID : INT «FK»
SessionDate : DATE
SessionTime : TIME
ActivityType : VARCHAR
IsConflict : BOOLEAN
TrainerAvailability
AvailabilityID : INT «PK»
TrainerID : INT «FK»
DayOfWeek : INT
StartTime : TIME
EndTime : TIME
IsRecurring : BOOLEAN
ClassSchedule
ClassID : INT «PK»
ClassName : VARCHAR
TrainerID : INT «FK»
RoomID : INT «FK»
DayOfWeek : INT
StartTime : TIME
MaxCapacity : INT
Equipment
EquipmentID : INT «PK»
RoomID : INT «FK»
Name : VARCHAR
Status : VARCHAR
LastServicedDate : DATE
Payment
PaymentID : INT «PK»
MemberID : INT «FK»
StaffID : INT «FK»
Amount : NUMERIC
PaymentDate : DATE
PaymentType : VARCHAR
Notes : VARCHAR
ClassRegistration
RegistrationID : INT «PK»
MemberID : INT «FK»
ClassID : INT «FK»
DateRegistration : DATE
EquipmentMaintenance
MaintenanceID : INT «PK»
EquipmentID : INT «FK»
StaffID : INT «FK»
ReportedAt : TIMESTAMP
Description : VARCHAR
Status : VARCHAR
ResolvedAt : TIMESTAMP
MemberID
MemberID
MemberID
TrainerID
TrainerID
TrainerID
RoomID
MemberID
RoomID
RoomID
StaffID
StaffID
ClassID
EquipmentID

COMP3005A: ER Model, Relational Mapping, and Normalization

Sanvi Kaushik (101303354)
Pardis Ehsani (101300400)
Seham Khalifa (101295726)

December 1, 2025

1 Entity–Relationship (ER) Model

Figure 1 shows the final ER model for the Health & Fitness Club Management System. The diagram (generated in Gleek.io) summarizes all entities, attributes, and relationships required by the project, including cardinalities, participation, and associative entities. The section does not repeat the full explanation of each component; instead, it provides a visual summary of the conceptual design that drives the relational schema.

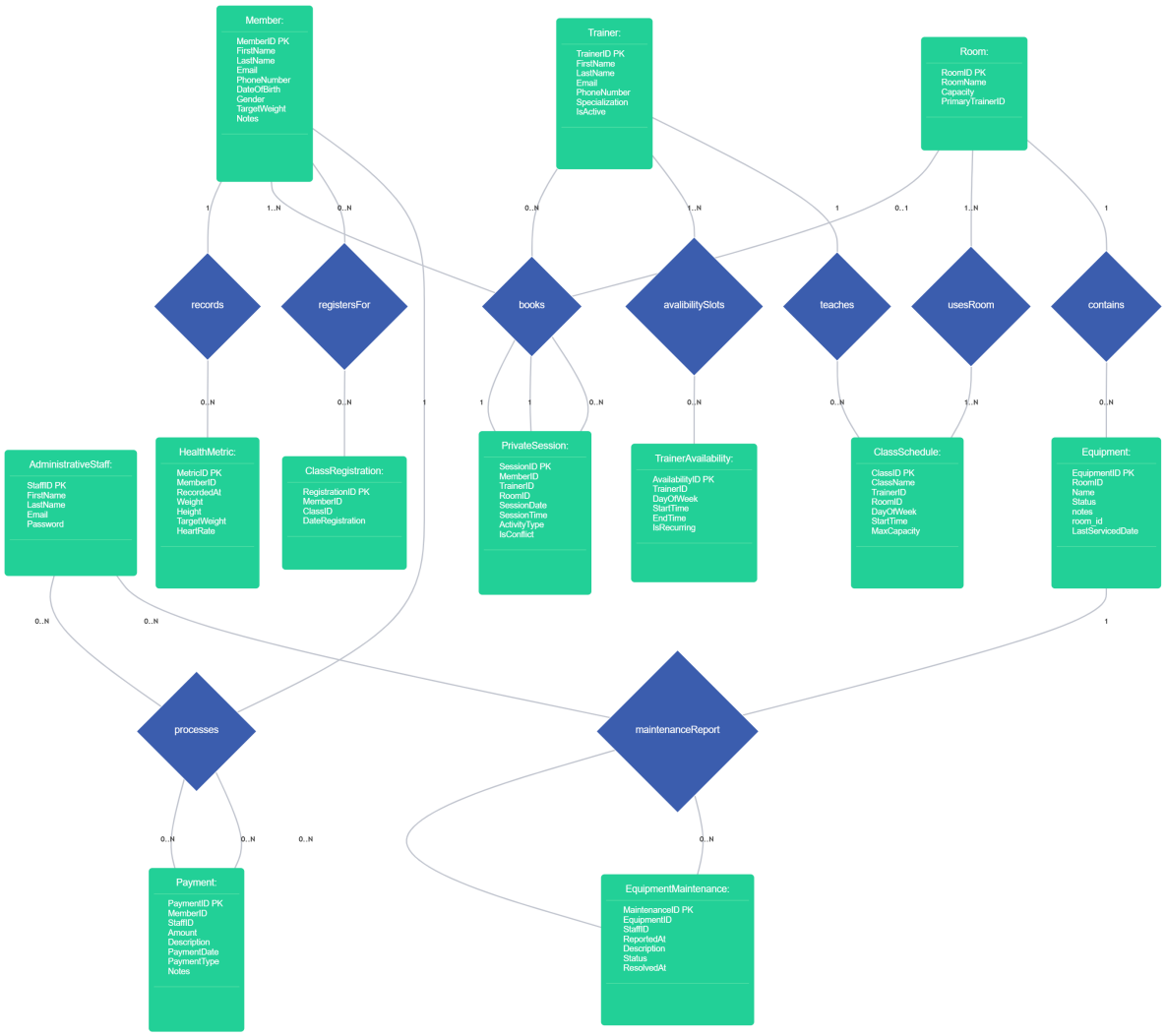


Figure 1: Final ER Diagram for the Health & Fitness Club Management System.

1.1 Model Overview

The ER model contains:

- Regular entities such as Member, Trainer, Room, Equipment, Payment, and their primary attributes.
- History tables (HealthMetric, EquipmentMaintenance) capturing multi-valued temporal information.
- Associative entities (ClassRegistration, PrivateSession) for M:N and composite interactions.
- Relationship diamonds representing booking, registering, processing, teaching, availability, and maintenance relationships.

This model serves as the conceptual foundation for the relational schema shown next.

2 ER-to-Relational Mapping

Figure 2 shows the relational schema derived from the ER diagram. Every entity becomes a table with a surrogate primary key, and all relationships are implemented via foreign keys. Associative entities remain as

independent tables because they have their own attributes (e.g., dates, status fields).

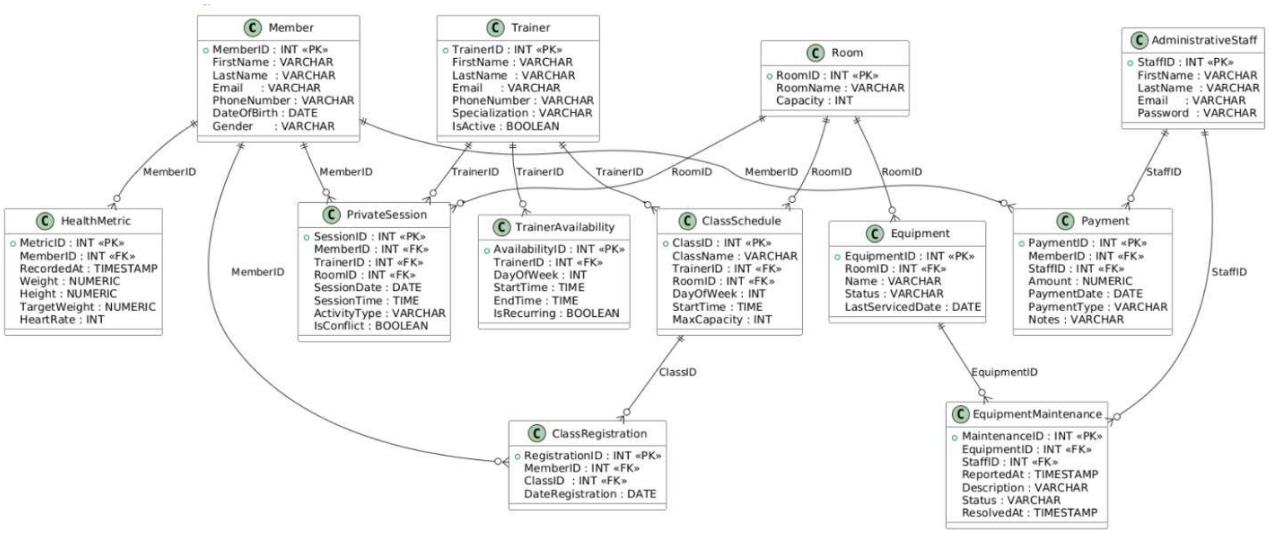


Figure 2: Relational mapping generated from the ER model.

2.1 Mapping Highlights

- 1:N relationships map to foreign keys on the N-side (e.g., `HealthMetric.MemberID`, `ClassSchedule.TrainerID`).
- The single M:N relationship (Member–ClassSchedule) maps to `ClassRegistration`.
- PrivateSession is modeled as an associative entity with additional attributes (date, time, activity, status).
- All tables use atomic, single-valued attributes to support 1NF.

This schema reflects the exact structure implemented in PostgreSQL through SQLAlchemy ORM in the final project.

3 Normalization

This section demonstrates that all relations in the schema are in **Third Normal Form (3NF)**. Proofs are shown for representative tables (covering both simple and associative cases); the remainder follow the same logic.

3.1 Normalization Framework

A relation is in:

- **1NF**: if all attributes are atomic.
- **2NF**: if it is in 1NF and no non-key attribute depends on a proper subset of a composite key.
- **3NF**: if for every FD $X \to A$, either X is a superkey, or A is part of a key (no transitive dependencies).

All tables use single-attribute surrogate primary keys; therefore:

- No composite keys $\Rightarrow$ no partial dependencies (2NF satisfied).
- 3NF reduces to checking that no non-key attribute determines another.

3.2 Sample Normalization Proofs

3.2.1 Member(MemberID, FirstName, LastName, Email, PhoneNumber, DateOfBirth, Gender, TargetWeight, Notes)

FDs:

$$MemberID \to allotherattributes$$

$$Email \to MemberID, FirstName, ...$$

Keys: {MemberID} (primary), {Email} (unique)
3NF Proof:

- Left side of each FD is a key or candidate key.
- No non-key attribute determines another non-key attribute.

Thus, Member is in **3NF**.

3.2.2 ClassSchedule(ClassID, ClassName, TrainerID, RoomID, DayOfWeek, StartTime, MaxCapacity)

FD:

$$ClassID \to allnon - keyattributes$$

There is no FD such as RoomID $\to$ MaxCapacity or TrainerID $\to$ DayOfWeek.
Thus, ClassSchedule is in **3NF**.

3.2.3 ClassRegistration(RegistrationID, MemberID, ClassID, DateRegistration)

FDs:

$$RegistrationID \to MemberID, ClassID, DateRegistration$$

$$(MemberID, ClassID) \to DateRegistration$$

Keys: {RegistrationID}, {MemberID, ClassID}
3NF Proof:

- Both LHS of the above FDs are keys or candidate keys.
- No non-key attribute determines another non-key attribute.

Therefore, ClassRegistration is in **3NF**.

3.2.4 PrivateSession(SessionID, MemberID, TrainerID, RoomID, SessionDate, SessionTime, ActivityType, IsConflict)

FD:

$$SessionID \to allattributes$$

No non-key attribute determines another (e.g., ActivityType does not determine RoomID).
Thus, PrivateSession is in **3NF**.

3.3 Conclusion

- All relations use atomic fields (1NF).
- No relations contain composite keys (2NF).
- All non-trivial FDs originate from keys; no non-key determines another (3NF).

Therefore, the full schema is normalized to **Third Normal Form (3NF)**.