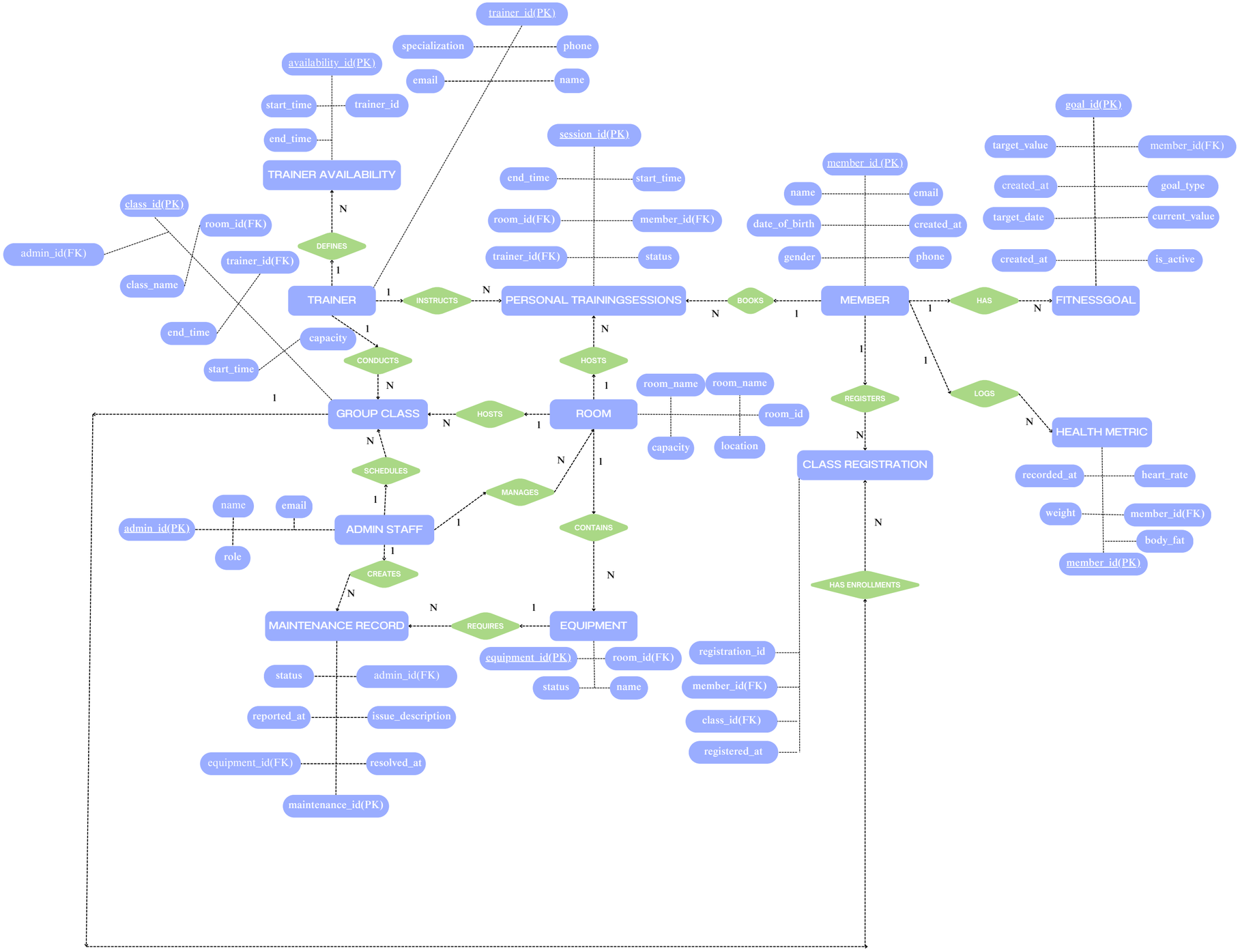
trainer_id(PK)
specialization
phone
availability_id(PK)
email
name
start_time
trainer_id
end_time
TRAINER AVAILABILITY
session_id(PK)
end_time
start_time
room_id(FK)
member_id(FK)
trainer_id(FK)
status
member_id (PK)
name
email
date_of_birth
created_at
gender
phone
goal_id(PK)
target_value
member_id(FK)
created_at
goal_type
target_date
current_value
created_at
is_active
class_id(PK)
room_id(FK)
admin_id(FK)
class_name
trainer_id(FK)
end_time
capacity
start_time
DEFINES
TRAINER
INSTRUCTS
PERSONAL TRAININGSESSIONS
BOOKS
MEMBER
HAS
FITNESSGOAL
CONDUCTS
HOSTS
REGISTERS
LOGS
GROUP CLASS
HOSTS
ROOM
room_name
room_name
room_id
capacity
location
HEALTH METRIC
SCHEDULES
MANAGES
CONTAINS
CLASS REGISTRATION
name
email
admin_id(PK)
role
ADMIN STAFF
recorded_at
heart_rate
weight
member_id(FK)
body_fat
member_id(PK)
CREATES
HAS ENROLLMENTS
MAINTENANCE RECORD
REQUIRES
EQUIPMENT
registration_id
equipment_id(PK)
room_id(FK)
member_id(FK)
status
admin_id(FK)
status
name
class_id(FK)
reported_at
issue_description
registered_at
equipment_id(FK)
resolved_at
maintenance_id(PK)

NORMALIZATION

1. MEMBER table
   Attributes: member_id (PK), name, email, phone, gender, date_of_birth, created_at
   The MEMBER table is in 1NF because all of the attributes have atomic values—a single name, email address, etc. It's also in 2NF because it has one primary key column (member_id), meaning partial dependencies are impossible. It's in 3NF because all non-key attributes are only dependent on the member_id and none of them determine any of the others. In conclusion, this table is fully normalized to 3NF.

2. TRAINER table
   Attributes: trainer_id (PK), name, email, phone, specialization
   This TRAINER table is in 1NF because all fields hold single, indivisible values. It is in 2NF because it uses a single-column primary key. It is in 3NF because name, email, phone, and specialization depend only on the trainer_id and do not determine one another. This table is fully normalized.

3. TRAINER AVAILABILITY table
   Attributes: availability_id (PK), trainer_id (FK), start_time, end_time
   The TRAINER AVAILABILITY table is in 1NF because each attribute value holds only one value—one trainer, one start time, one end time per row. It is in 2NF because it has a single-attribute primary key, availability_id. It's in 3NF because start_time and end_time depend only on the availability_id, but not any other non-key attribute. There are no transitive dependencies, so the table is fully normalized.

4. GROUP CLASS table
   Attributes: class_id (PK), class_name, trainer_id (FK), room_id (FK), start_time, end_time, capacity
   The GROUP CLASS table follows 1NF because all fields are atomic. It is also in 2NF because this table has only one simple primary key. It is in 3NF because class_name, start_time, end_time, and capacity are all fully dependent on class_id and are independent of each other. There are no transitive dependencies, so the table is fully in 3NF.

5. CLASS REGISTRATION table
   Attributes: registration_id (PK), member_id (FK), class_id (FK), registered_at
   The CLASS REGISTRATION table is in 1NF because all values are atomic. It's in 2NF because it has a single-column primary key. It is in 3NF because registered_at depends only on registration_id and not on member_id or class_id. There are no non-key-to-non-key dependencies, so the table is fully normalized.

6. PERSONAL TRAINING SESSION table
   Attributes: session_id (PK), member_id (FK), trainer_id (FK), room_id (FK), start_time,

end_time, status
This table is in 1NF since all attributes are atomic. It is in 2NF as session_id is a simple primary key. It is in 3NF as the start_time, end_time, and status depend only on session_id and are independent of each other. There are no transitive dependencies, so the table is fully normalized.

7. ROOM table
   Attributes: room_id (PK), room_name, capacity, location
   The ROOM table is in 1NF since it holds atomic values in each column—one name, one capacity, one location per row. It is in 2NF since there is a single-column primary key. It meets 3NF since room_name, capacity, and location depend only on the room_id and don't determine one another. That means it confirms the ROOM table as fully normalized.

8. EQUIPMENT table
   Attributes: equipment_id (PK), room_id (FK), name, status
   The EQUIPMENT table satisfies 1NF with atomic fields. It fulfills the 2nd normal form because the primary key is single-column. It follows 3NF too: name and status are directly dependent on equipment_id and not on any other non-key attribute. There are no transitive dependencies introduced; hence the table is fully normalized.

9. ADMIN STAFF table
   Attributes: admin_id (PK), name, email, role
   The ADMIN STAFF table is in 1NF because it has only one value in each field. It is in 2NF because it is using a single-column primary key. It's in 3NF because name, email, and role depend only on admin_id but do not determine each other. Thus, the table is also fully normalized.

10. MAINTENANCE RECORD table
    Attributes: maintenance_id (PK), equipment_id (FK), admin_id (FK), issue_description, status, reported_at, resolved_at
    This table is already in 1NF, as its attributes are atomic. It is automatically in 2NF since it has a single simple primary key. It follows 3NF because issue_description, status, reported_at, and resolved_at depend on maintenance_id alone. No non-key attributes determine others, confirming full 3NF compliance for this table.

11. HEALTH METRIC table
    Attributes: metric_id (PK), member_id (FK), recorded_at, heart_rate, weight, body_fat
    The HEALTH METRIC table is in 1NF because all measurements are atomic. It is in 2NF because the primary key is a single attribute. It is in 3NF because recorded_at, heart_rate, weight, and body_fat each depend only on metric_id, without any transitive dependencies. The table is fully normalized.

12. FITNESS GOAL table

Attributes: goal_id (PK), member_id (FK), goal_type, target_value, current_value, created_at, is_active

The FITNESS GOAL table is in 1NF because the attribute values are atomic. It is automatically in 2NF because the primary key is not composite. It is in 3NF because goal_type, target_value, current_value, created_at, and is_active depend directly on goal_id and are independent of one another. That means this table is fully normalized.

MAPPING

1. TRAINER

Relational Table Columns:

- trainer_id (PK)
- name
- email
- phone
- specialization

Relationships:

- One trainer has many trainer availability records
- One trainer conducts many group classes
- One trainer instructs many personal training sessions

2. TRAINER AVAILABILITY

Relational Table Columns:

- availability_id (PK)
- trainer_id (FK → Trainer)
- start_time
- end_time

Relationship:

- Many availability entries belong to one trainer

3. MEMBER

Relational Table Columns:

- member_id (PK)
- name
- email
- phone
- gender
- date_of_birth
- created_at

Relationships:

- One member registers for many group classes
- One member logs many health metrics
- One member maintains many fitness goals
- One member can book many personal training sessions

4. PERSONAL TRAINING SESSION

Relational Table Columns:

- session_id (PK)
- member_id (FK → Member)
- trainer_id (FK → Trainer)
- room_id (FK → Room)
- start_time
- end_time
- status

Relationships:

- Many sessions belong to a single trainer
- Many sessions belong to a single member
- Many sessions take place in a single room

5. GROUP CLASS

Relational Table Columns:

- class_id (PK)
- trainer_id (FK → Trainer)
- room_id (FK → Room)
- class_name
- start_time
- end_time
- capacity

Relationships:

- One trainer conducts many group classes
- One room hosts many group classes
- A group class can have many enrolled members (via Class Registration)

6. CLASS REGISTRATION

Relational Table Columns:

- registration_id (PK)
- member_id (FK → Member)
- class_id (FK → Group Class)

- registered_at

Relationship:

- Many registrations belong to a member
- Many registrations belong to a group class

This acts as the bridging entity between Member and GroupClass.

7. ROOM

Relational Table Columns:

- room_id (PK)
- room_name
- capacity
- location

Relationships:

- One room hosts many group classes
- One room hosts many personal training sessions
- One room contains many equipment items

8. EQUIPMENT

Relational Table Columns:

- equipment_id (PK)
- room_id (FK → Room)
- name
- status

Relationships:

- One room contains many equipment units
- One equipment item can have many maintenance records

9. MAINTENANCE RECORD

Relational Table Columns:

- maintenance_id (PK)
- equipment_id (FK → Equipment)
- admin_id (FK → Admin Staff)
- reported_at
- resolved_at
- issue_description
- status

Relationship:

- Many maintenance records belong to one equipment item

10. ADMIN STAFF

Relational Table Columns:

- admin_id (PK)
- name
- email
- role

Relationship:

- One admin staff member can create many maintenance records

11. HEALTH METRIC

Relational Table Columns:

- metric_id (PK)
- member_id (FK → Member)
- recorded_at
- heart_rate
- weight
- body_fat

Relationship:

- Many health metric entries belong to one member

12. FITNESS GOAL

Relational Table Columns:

- goal_id (PK)
- member_id (FK → Member)
- goal_type
- target_value
- current_value
- created_at
- is_active

Relationship:

- Many fitness goals belong to one member