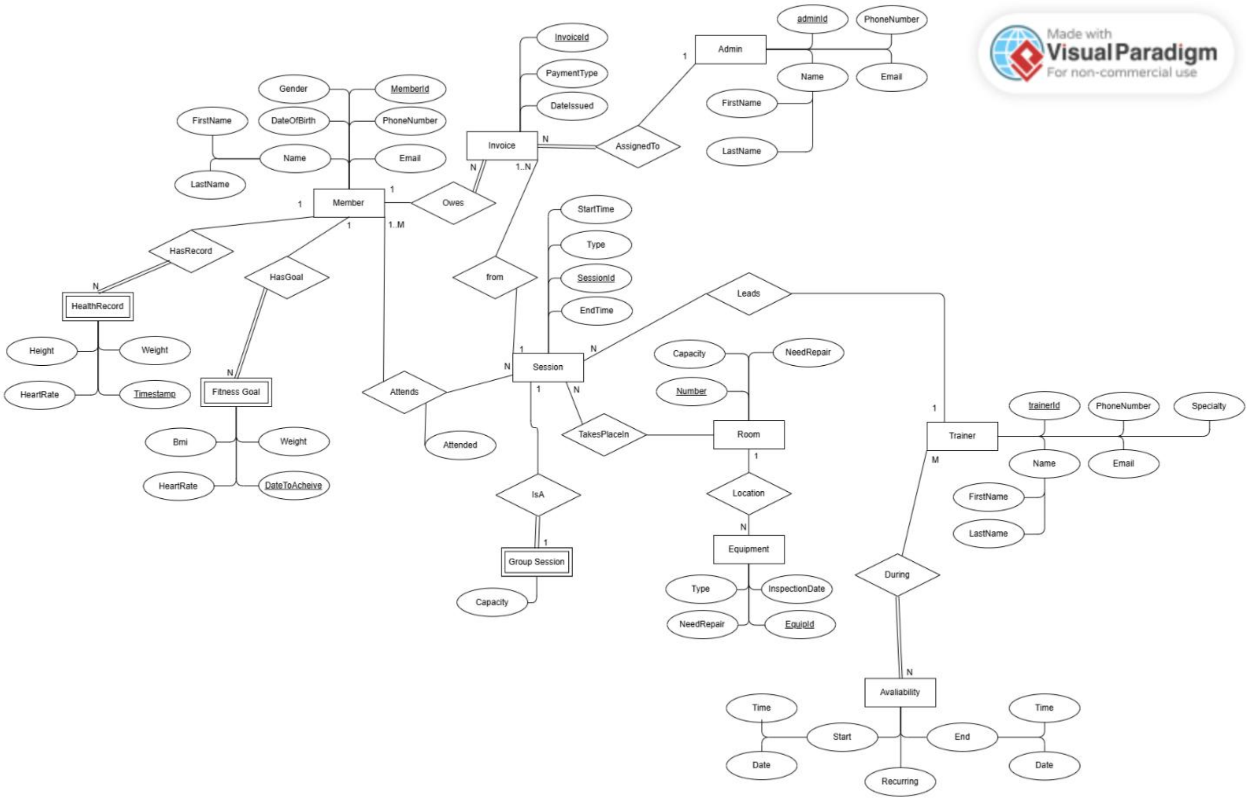

Figure 1: Entity Relationship Diagram

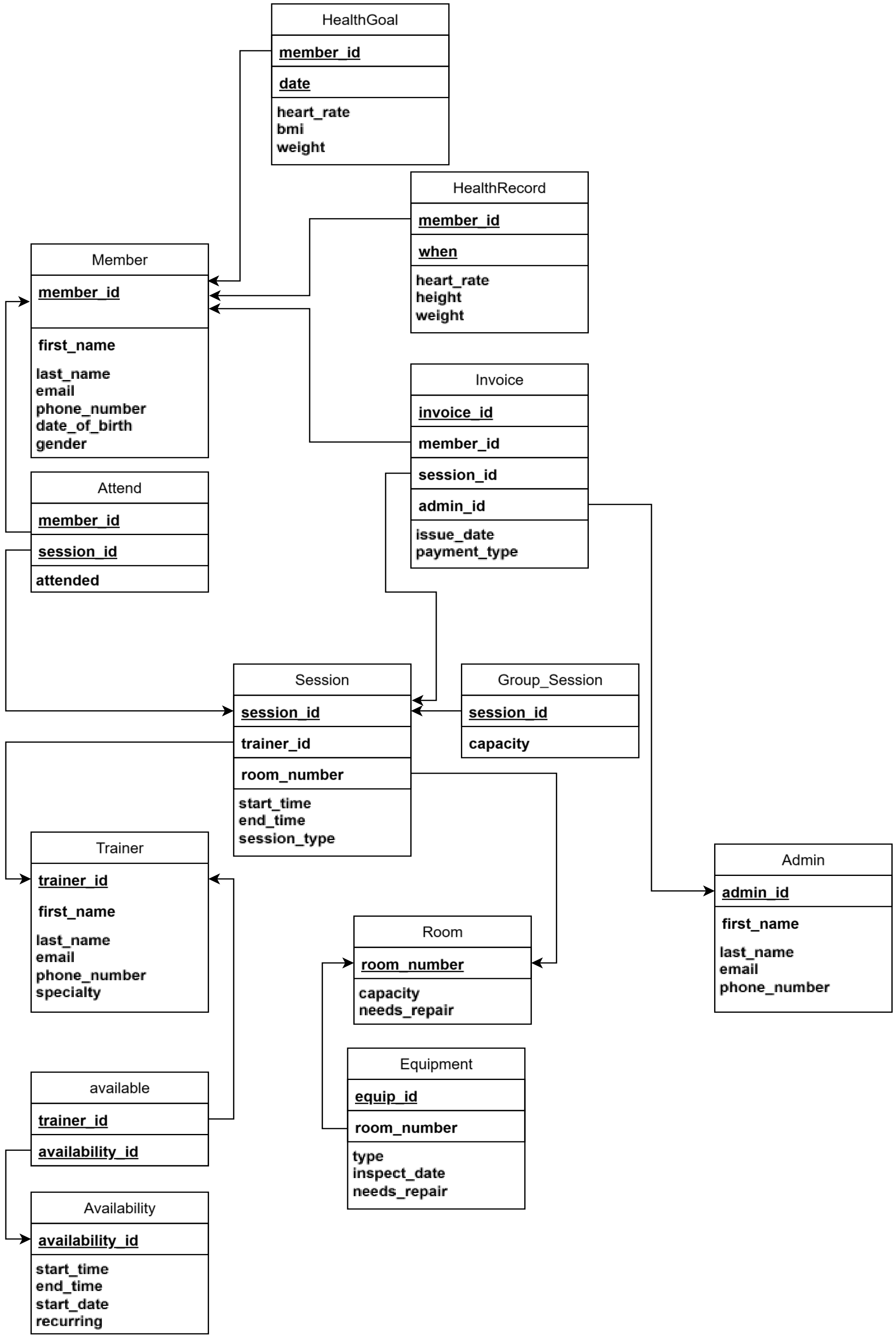

Figure 2: Entity Relationship Mapping

Assumptions:

Entity	Assumption
Member	>Only one Phone Number linked at a time. >Member cannot use same email if they are also a trainer or admin. (Email unique to one user). >Member putting signing up with correct information (Only basic error checking invalid date/ Phone Number not a number) >Member doesn't have access to invoices right away. >Can't have different goals for the same day
Trainer	>Can train outside there specialty >No overriding availability (single can't overlap with reccuring)
Admin	>Assigned Invoices to review at random
Health Goal	>Can't have different goals for the same day (best, next best). One day one goal >Height can be derived (Bmi and weight)
Health Record	>Accurate stats entered. No check for realistic inputs >BMI is derived between (height and Weight)
Invoice	>Assigned to an admin to review at random. >Stay even if Session is canceled or member misses (Up to admin review how to handle)
Availability	>Is contained within one day. (Can't be available over night) >Multiple trainers can have the same schedule
Room	>Sessions can be assigned even if broken and needing repairs. (Assumed someone will repair before actual date)
Equipment	>No effect on operational status of a room.
Session	>Is a personal session unless it is in the group session. >Trainer specialty doesn't have to match the type of session. >Start time and end time include dates >No overnight sessions >Length is derived from start and end times
Group Session	>Capacity can be greater than any room. (Still can't assign a room over capacity) >Number of participants derived from attended table

Normalizations:

FD1: member_id -> first_name, last_name, email, phone_number, date_of_birth, gender

FD2: (member_id, date) -> heart_rate, bmi, weight

Members can have multiple health/fitness goals

Different members can have health goals for the same date

(Height is derived)

FD3: (member_id, when(timestamp)) -> heart_rate, weight, height

Member can multiple health records which often look the same. Drastic health changes are rarer.

(BMI is derived)

FD4: invoice_id -> member_id, session_id, admin_id, issue_date, payment_type

Members can have multiple invoices (different class, sessions (if group) will have multiple invoices aswell, admin will mange multiple invoices, if people sign up the same day invoices will have same dates. Payment_type changes (Sometimes user will pay with credit card, debit or cash).

(Collect/payed is derived from payment type)

FD5: (admin_id) -> first_name, last_name, email, phone_number

FD6: (session_id) -> trainer_id, room_number, start_time, end_time, session_type

FD7: (session_id) -> Capacity

(Can be derived as 1 if not in the Group Session table)

(If not its own table session would have many repeated 1s)

FD8: (room_number) -> capacity, needs_repair

(Many rooms can have the same capacity)

(Broken room Window, floor, etc)

FD9: (equip_id) -> room_number, type, inspect_date, needs_repair

(Multiple of same type, not all of same type are inspected on the same day)

(needs_repair is not for room) (Broken machine)

(Can change rooms)

FD10: (session_id, member_id) -> Attended

(Some member may show up some not)

FD11: (availability_id) -> Start_time, end_time, start_date, recurring

(Many trainers will have the same availability especially recurring)

Index, View, Trigger

Index is on emails of trainers, members and admins.

See User in Model

View is represented by GroupSessionSummary a view of all the group sessions with how many people signed up vs capacity. When called for the first time it runs an HQL (Hibernate Querry Language) query then caches the results only updating when a group session is added or entry added to attend table.

Trigger is represented by SessionJoinTableListener (attend) when someone signs up for a class hibernate triggers the listener creating a row in the invoice table and assigning it a random admin for review.