**GYMLIFE**

1. **Requirements:**

* Owner of gym club wants to have a management system to keep track of everything.
* List of staffs with their necessary information. Their work schedule (date and time), role, wage ($/hour)
* List of equipment available in the gym, their serials, brand, manufactures, cost, purchased date.
* List of members of the gym with their necessary information. Their join date, membership types, status, expiration date, contract. Last visit date and time. Transaction history (payment to their membership). Remaining owed balance, checking/saving bank account, routine number, bank name.
* List of trainers with their necessary information. Sign up date, contract, salary
* Member access log every day, monthly, year
* Membership types with price
* New members sign up, date, type.
* ROLE: OWNER, MANAGER, EMPLOYEE, TRAINER, GYMMER

1. **Views:**

* Member: Name, address, phone number, date of birth, gender, role, first day sign up, membership type, bank account info (bank name, account number, routine number), remaining balance, expiration date, last visit date, last visit time, transaction history, active status, trainer
* Employee: Name, address, phone number, date of birth, gender, role, first day of work, wage ($/hour), active status, bank account info, transaction
* Trainer: Name, address, phone number, date of birth, gender, role, first day of work, wage ($/hour), active status, bank account info, transaction, students
* Work schedule: Work day, staff name, time
* Membership: type name, price
* Equipment: machine name, serial numbers, brand, target type, manufacture name, cost
* Manufacture: manufacture name, address, phone number, email, machines imported
* Access log: date, member name, time access in, time log out

1. **Initial entities & attributes:**

* Member: Name, address, phone number, date of birth, gender, email, role, membership type, membership price, first day sign up, expiration date, active status, last visit date, last visit time, trainer, bank name, account number, routine number, remaining balance, transaction history,
* Employee: Name, address, phone number, date of birth, gender, email, role, first day of work, active status, wage ($/hour), bank name, account number, routine number, transaction history,
* Trainer: Name, address, phone number, date of birth, gender, email, role, first day of work, active status, wage ($/hour), bank account number, routine number, transaction history, students
* Work schedule: Work day, staff name, time
* Equipment: machine name, serial numbers, brand, target type, manufacture name, cost
* Manufacture: manufacture name, address, phone number, email, equipment imported from, bank name, account number, routine number, remaining balance owed, transaction history

Since member, employee, and trainer have most of attributes that are similar, we can combine them all together as one entity called Member and distinct them by role including gymmer, employee, trainer. In addition, we will add more role including owner and manager. We also added attribute named password to Member relation, so it would be used for the application security.

The initial entities and attributes are shortened to the following version below. In addition, all repeating groups/attributes are also identified and marked inside parentheses.

* Member: Name, address, phone number, date of birth, username, password, email, gender, (role, wage), (membership type, membership price), first day sign up, expiration date, active status, (last visited date, last visit time), (bank name, account number, routine number, remaining balance owed, transaction history), (trainer, student)
* Work schedule: Work day, staff name, time
* Equipment: machine name, serial numbers, target type, manufacture name, cost
* Manufacture: manufacture name, address, phone number, email, (equipment imported from), (bank name, account number, routine number, remaining balance owed, transaction history)

1. **Normalization:**

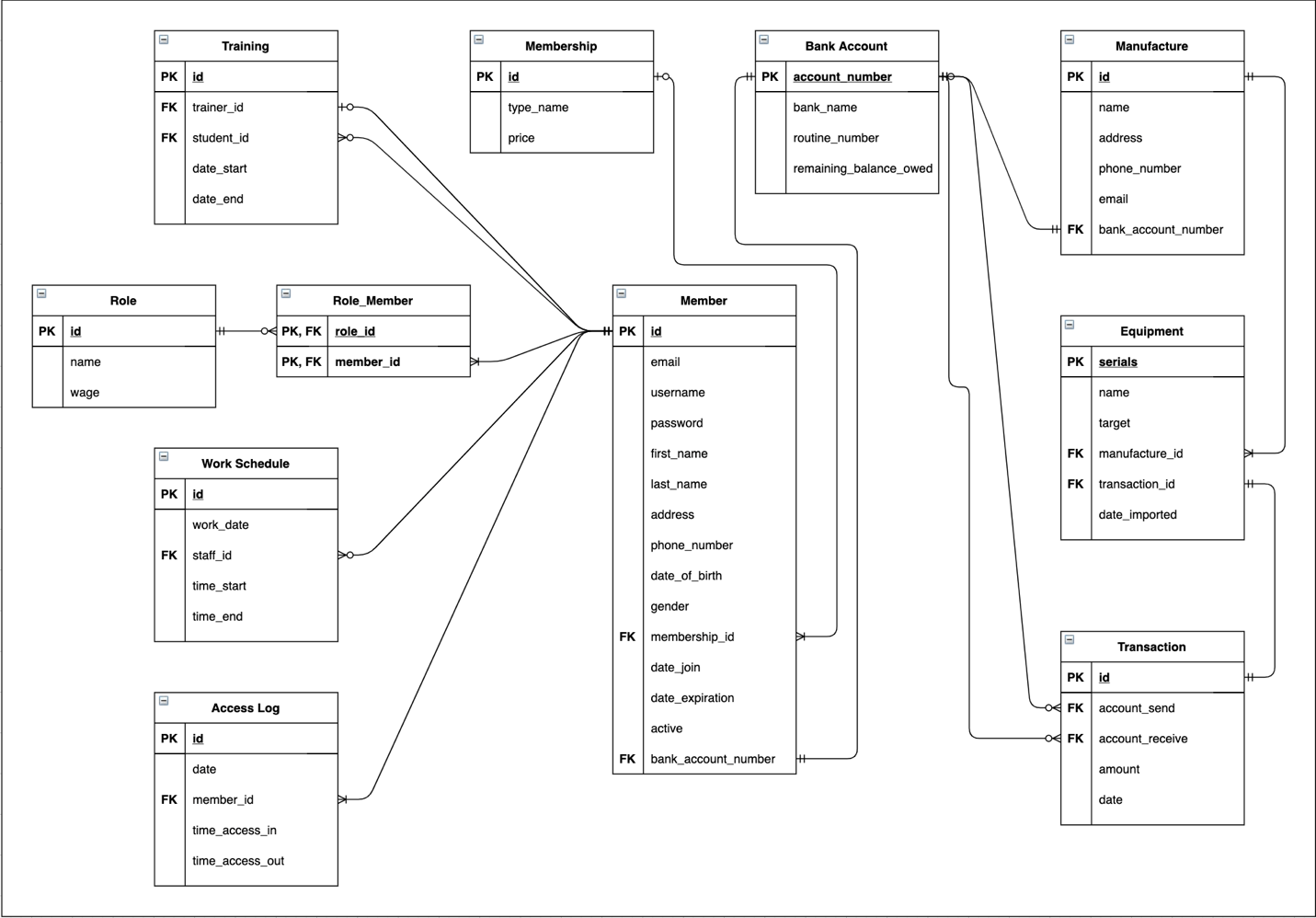
**1NF: Assigning primary key and move repeating groups to new relations.**

**2NF: Move all non-key attributes that are not fully functional dependent on primary key to new relations**

**3NF: Move all non-key attributes that are transitively dependent to new relations**

* Member: id (PK), email, username, password, first\_name, last\_name, address, phone\_number, date\_of\_birth, gender, role\_id (FK), membership\_id (FK), date\_join, date\_expiration, active, bank\_account\_number (FK)
* Bank Account: account\_number (PK), bank\_name, routine\_number, remaining\_balance\_owed
* Transaction: id (PK), account\_send (FK), account\_receive (FK), amount, date
* Role: id (PK), name, wage
* Training: id (PK), trainer\_id (FK), student\_id (FK), date\_start, date\_end
* Work schedule: id (PK), work\_date, staff\_id (FK), time\_start, time\_end
* Membership: id (PK), type\_name, price
* Equipment: serials (PK), name, target, manufacture\_id (FK), transaction\_id (FK), date\_imported
* Manufacture: id (PK), name, address, phone\_number, email, bank\_account\_number (FK)
* Access log: id (PK), date, member\_id (FK), time\_access\_in, time\_access\_out

1. **Entity relationship diagram (ERD)**

****

1. **Business rules:**

* Each member (including owner, manager, employee, trainer, gymmer) has an account stored in Member table with all their basis information
* Each member has a bank account that is stored in Bank Account table. Owner will use his business bank account for receiving money from gymmer, paying wages to staffs, and equipment costs to manufactures.
* Any transaction that happens between 2 parties will be kept tracked in the Transaction table.
* Each member will have a corresponding role with their wages set to a certain amount if the member is a staff. All roles that are tracked in this table including: OWNER, MANAGER, TRAINER, EMPLOYEE, GYMMER

…

1. **Physical Design**