# Project Title

Recreation Facility Management System in a Housing Complex

# Team

* Tho Quyen Ngo (MySQL Expert)
* Dai-Dung Nguyen (MS SQL Server Expert)

# Weekly Meeting Hours

We will meet and work on the project every Thursday from 7PM to 9PM.

# Project Description

Consider a housing complex with several high-rise buildings. This housing complex has several recreation facilities (Swimming Pool, Gym, Tennis Course). Tenants living in these buildings can reserve these facilities and this database system is created for the management of these facilities.

The data requirements are summarized as follows:

i. Each building has a name and address

ii. Each facility has a unique name, type(swimming pool, tennis court…), location (address), description

iii. Each tenant or the person who reserves the facility, has name, date of birth, email, phone number, the apartment where they reside, the name of all residents in the same apartment(house…)…start time and end time they lived in that apartment

iv. The person who reserves the facility can be a resident of the housing complex or a non-resident

v. The reservation information includes, date, time interval, deposited amount (only non-residents pay it), the facility where they reserved, Number of guests

vi. Information about the employee who books the facility is stored

vii. It is recorded in the database whether the person who reserved the facility showed up.

viii. Each tenant only have ONE StartDate and EndDate

ix. Facilities might be on the same floor (have the same floor number) but different building

x. Apartments from different buildings can have the same Appartment Number

# Assumptions about Cardinality and Participations

Tenant (1,M) lives in (1,1) Building

A Tenant must live in a building, a building has at least 1 tenant

A building could have many tenants, and a tenant just lives in one building

Employee (0,1) registers (1,N) Booking

Booking(1,M) book for (0,1) Facility

Building (1,1) has (1,M) Facility

Building (1,1) has (1,M) Apartment

Tenant(1,M) lives in (0,1) apartment

Tenant (0,1) books (0,M) bookings

Non-tenant (1,1) books (0,M) bookings

Person includes tenants and non-residents. Employee could be a tenant or non-resident. Person includes tenants and non-residents. Employee could be a tenant or non-resident.

# EER Modeling Diagram

# ER-Model Mapping to Database Relational Schema

Person(SinNumber, Name, DOB, IsEmployee, IsTenant)

Tenant\_Apartment(SinNumber, **AppNumber, BuildingNo**, StartDate, EndDate)

Booking(BookingNo, BookingDate, NoOfGuests, TimeInterval, ShowedUp, **Booker\_SIN, FacilityName, Emp\_SIN**)

Non-tenant\_Booking(**BookingNo,** Deposit)

Facility(**FacilityName**, FacilityType, FacilityLocation, FacilityDescription, **BuildingNo**)

Building(BuildingNo**,** BuildingName, BuildingAddress)

Apartment(AppNumber, **BuildingNo**)

PersonPhoneNo(**PersonSIN**, PhoneNo)

PersonEmailAddress(**PersonSIN**, EmailAdd)

# Normalization

Person(SinNumber, Name, DOB, IsEmployee, IsTenant)

* SinNumber can define Name, DOB, IsEmployee and IsTenant. All of the attributes are also atomics => 1NF
* Only one single key: SIN => 2NF
* No transitive => 3NF

Tenant\_Apartment(SinNumber, **AppNumber, BuildingNo**, StartDate, EndDate):

* SinNumber, **AppNumber, BuildingNo**: composite key that can define all other attributes. StartDate and EndDate might be multi-valued when a tenant leaves the apartment and then come back, but for the sake of simplicity for this project, we will assume each tenant only have ONE StartDate and EndDate => 1NF
* StartDate and EndDate depend on the whole key which is including SIN, AppNumber, BuildingNo. => 2NF
* StartDate and EndDate depend only on the key, there’s no way to tell what they would be without the key =>3NF

Booking(BookingNo, BookingDate, NoOfGuests, TimeInterval, ShowedUp**, Booker\_SIN, FacilityName, Emp\_SIN**)

* BookingNo can determine Date, NoOfGuests, TimeIntervaland. All of the attributes are also atomics => 1NF
* ShowedUp, Date, NoOfGuests, TimeInterval depend on the key which is BookingNo. => 2NF
* ShowedUp, Date, NoOfGuests, TimeInterval depend only on the BookingNo =>3NF

Non-tenant\_Booking(**BookingNo,** Deposit)

* BookingNo can determine the deposit because it can track whether that person is a tenant or not and then determine deposit.

Facility(FacilityName, FacilityType, FacilityLocation, FacilityDescription, **BuildingNo**)

* It’s stated in the beginning that FacilityName is unique. FacilityName can determine FacilityType, FacilityLocation (for example: 3rd floor or 4th floor), FacilityDescription. All of the attributes are also atomics. => 1NF
* All attributes depend on the key and only the facilityName (A building might have several facility of the same type, facilities might have the same location (both on 2nd floor), but there are multiple building…) Facility Desctiption can be considered a candidate key, but it’s not suitable to be a primary key due to its length and form.

=> 2NF 3NF

Building(BuildingNumber**,** BuildingName, BuildingAddress)

* BuildingNumber can determine its name and address. All of the attributes are also atomics => 1NF
* All attributes depend on the key and only the BuildingName => 2NF 3NF

Apartment(ApartmentNumber, **BuildingNo**)

* All attributes are part of the primary key.

Person\_PhoneNo(**Person\_SIN**, PhoneNo)

* All attributes are part of the primary key.

Person\_EmailAddress(**Person\_SIN**, EmailAddress)

* All attributes are part of the primary key.

# Determining Data Types (Domain) and Constraints

Data types and constraints are set based on common sense. Here are some of the notables settings:

Person: name is limited to 100 letters, SinNumer is a positive Int with at most 9 digits

PersonEmail: has a limit of 50 letters

BuildingAddress: can have 255 letters for longer address

The Date Type is also used where applicable (Date of birth, bookingdate, start date, end date)

NoOfGuest and TimeInterval must be positive and larger than 0

NoOfGuest should also be less than 50 due to facility size

TimeInterval caps at 72, no one should be able to book for more than 3 days

# Creating Database and Tables - SQL DDL

SQL commands are saved in script files named createTable, deleteTable and insertData. The script files are stored beside this document within the same folder. The scripts are also saved in the APEX workspace.

# Inserting Values in Tables

SQL commands are saved in script files named createTable, deleteTable and insertData. The script files are stored beside this document within the same folder. The scripts are also saved in the APEX workspace.

# SQL Queries

SQL commands are saved in script files named Queries. The script files are stored beside this document within the same folder. The scripts are also saved in the APEX workspace.

# Views

SQL commands are saved in script files named Queries. The script files are stored beside this document within the same folder. The scripts are also saved in the APEX workspace.