# Name of Your Company

**Royal City Housing Complex**

# Project Title

**Recreation Facility Management System in a Housing Complex**

# Team

* **Elson Ricafrente** (MS SQL Server Expert)

You are going to learn MySQL and MS SQL Server. One of you implements the project on MySQL and the other on MS SQL Server. However, you are going to teach the DBMS which you learn to your project partner. At the end both of you know two well-known DBMS’s.

# Weekly Meeting Hours

**Every Wednesday 10am-11:30am and Friday 4pm -5:30pm**

# 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:

* Each building has a name and address: **3 Buildings**
  + **Holland Building – 6174 Parker Avenue, New Westminster V1F 6G3**
  + **Maguire Building – 6175 Parker Avenue, New Westminster V1F 6G3**
  + **Garfield Building – 6176 Parker Avenue, New Westminster V1F 6G3**
* Each facility has a unique name, type( swimming pool, tennis court…), location (address), description
  + **Nile** – Swimming Pool
  + **Iron Will** – Gym
  + **Novak –** Tennis Court
  + **Olympus –** Social Hall
* 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
* The person who reserves the facility can be a resident of the housing complex or a non-resident
* The reservation information includes, date, time interval, deposited amount (only non-residents pay it), the facility where they reserved, Number of guests
* Information about the employee who books the facility is stored
* It is recorded in the database whether the person who reserved the facility showed up

# Assumptions about Cardinality and Participations

* Each building has a name, address, and apartment number
* Each facility has name, type, location, description
* Each reservation requestor has name, date of birth, email, phone number, the apartment where they reside
* The reservation requestor can be a resident or guest
* Each reservation information includes, date, time interval, deposited amount (only non-residents pay it), the facility where they reserved, Number of guests, employee that made the booking
* Each building can have one or more apartments
* One apartment will belong to one and only one building
* Each apartment can have one or more facilities
* One facilities can belong to one or more apartments

Graphical user interface, application

Description automatically generated

# EER Modeling Diagram

Diagram

Description automatically generated

# ER-Model Mapping to Database Relational Schema

The relational Schema is written here

|  |  |
| --- | --- |
| ER MODEL | RELATIONAL SCHEMA |
|  | Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type) |
|  | Building (Bldg\_Name, City, Province, Street, Zip Code, Bldg. #) |
|  | Facility (Facility\_Name, Type, Description)  Facility\_Location (**Facility\_Name**, **Bldg\_Name**, Flr. #) |
|  | Reservation (Reservation #, Date, Start Time, End Time, # of person, Status) |
|  | Employee (Emp\_id, Emp\_Name) |
|  | Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Tenants\_Dependents (**SSN**, Name, DoB, Gender, Relationship) |
|  | Building (Bldg\_Name, City, Province, Street, Zip Code, Bldg. #)  Building\_Apartment (**Bldg\_Name**, Apt #) |
|  | Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Tenants\_Resides\_in\_Apartment (**SSN**, **Bldg\_Name**, Apt #, Occupancy\_Start\_Date, Occupancy\_End\_Date) |
|  | Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **SSN**)  Person(SSN, Email, Name, Dender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type) |
|  | Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, Deposit, **SSN**)  Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type) |
|  | Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **Emp\_id**)  Employee (Emp\_id, Emp\_Name) |
|  | Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **Facility\_Name**)  Facility (Facility\_Name, Type, Description)  Facility\_Location (**Facility\_Name**, **Bldg\_Name**, Flr. #) |
|  | Building (Bldg\_Name, City, Province, Street, Zip Code, Bldg. #)  Facility (Facility\_Name, Type, Description)  Facility\_Location (**Facility\_Name**, **Bldg\_Name**, Flr. #)  Building\_Has\_Facility (**Bldg\_Name, Facility Name**) |

**NORMALIZATION:**

|  |
| --- |
| Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Checking for 1NF  The primary key is a simple key.  The attribute address is composite.  Address has been decomposed; therefore, the relation is in 1NF.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key. |

|  |
| --- |
| Building (Bldg\_Name, City, Province, Street, Zip Code, Bldg. #)  Checking for 1NF  The primary key is a simple key.  The attribute address is composite.  Address has been decomposed; therefore, the relation is in 1NF.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key. |

|  |
| --- |
| Facility (Facility\_Name, Type, Description)  Facility\_Location (**Facility\_Name**, **Bldg\_Name**, Flr. #)  Checking for 1NF  Facility (Facility\_Name, Type, Description, Location)  The primary key is a simple key.  Location has been decomposed into a new relation; therefore, the relation is in 1NF.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key.  Checking for 1NF  Facility\_Location (**Facility\_Name**, **Bldg\_Name**, Flr. #)  The primary key is a simple key.  The nonprime attributes are all simple.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key. |

|  |
| --- |
| Reservation (Reservation #, Date, Start Time, End Time, # of person, Status)  Checking for 1NF  The primary key is a simple key.  All nonprime attributes are simple except for Time\_Slot which is composite.  Time\_Slot has been decomposed; therefore, the relation is in 1NF.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key. |

|  |
| --- |
| Employee (Emp\_id, Emp\_Name)  Checking for 1NF  The primary key is a simple key.  The nonprime attribute Emp\_Name is also simple.  Relation already in 1NF.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  A relation with only 2 attributes cannot have any transitive FD. |

|  |
| --- |
| Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Tenants\_Dependents (**SSN**, Name, DoB, Gender, Relationship)  Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Same as earlier defined in strong entity.  Tenants\_Dependents (**SSN**, Name, DoB, Gender, Relationship)  Checking for 1NF  The primary key is composite key. Name attribute is a partial key.  All nonprime attributes are simple.  No decomposition is required, already in 1st Normal form.  Checking for 2NF  Even though there is a partial key, it cannot be decomposed because it is from the weak entity. To determine Dependents nonprime attributes, it’s owner primary key is required together with the partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key. |

|  |
| --- |
| Building (Bldg\_Name, City, Province, Street, Zip Code, Bldg. #)  Building\_Apartment (**Bldg\_Name**, Apt #)  Building (Bldg\_Name, City, Province, Street, Zip Code, Bldg. #)  Same as earlier defined in strong entity.  Building\_Apartment (**Bldg\_Name**, Apt #)  Checking for 1NF  The primary key is composite key. Apt # attribute is partial key.  There is no nonprime attribute.  No decomposition is required, already in 1st Normal form.  Checking for 2NF  There is no nonprime attribute.  Checking for 3NF  There is no nonprime attribute.  BCNF  There is no nonprime attribute. |

|  |
| --- |
| Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Tenants\_Resides\_in\_Apartment (**SSN**, **Bldg Name**, Apt #, Occupancy\_Start\_Date, Occupancy\_End\_Date)  Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Same as earlier defined in strong entity.  Tenants\_Resides\_in\_Apartment (**SSN**, **Bldg Name**,Apt #, Occupancy\_Start\_Date, Occupancy\_End\_Date)  Checking for 1NF  The primary key is composite key. Apt # attribute is partial key.  The nonprime attribute is simple.  No decomposition is required, already in 1st Normal form.  Checking for 2NF  The partial key cannot determine nonprime attributes.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key. |

|  |
| --- |
| Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **SSN**)  Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **SSN**)  Checking for 1NF  The primary key is a simple key.  The composite nonprime attribute has been decomposed.  Already in 1NF.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Tenants (SSN, Email, Name, Gender, DoB, Phone #)  Same as earlier defined in strong entity.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key.  Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Same as earlier defined in strong entity. |

|  |
| --- |
| Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, Deposit, **SSN**)  Person(SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #, Person Type)  Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **SSN**)  Checking for 1NF  The primary key is a simple key.  The composite nonprime attribute has been decomposed.  Already in 1NF.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key.  Guest (SSN, Email, Name, Gender, DoB, Phone #, City, Province, Street, Zip Code, House #)  Same as earlier defined in strong entity. |

|  |
| --- |
| Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **Emp\_id**)  Employee (Emp\_id, Emp\_Name)  Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **Emp\_id**)  Checking for 1NF  The primary key is a simple key.  The composite nonprime attribute has been decomposed.  Already in 1NF.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key.  Employee (Emp\_id, Emp\_Name)  Same as earlier defined in strong entity. |

|  |
| --- |
| Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **Facility\_Name**)  Facility (Facility\_Name, Type, Description)  Facility\_Location (**Facility\_Name**, **Bldg\_Name**, Flr. #)  Reservation (Reservation #, Date, Start Time, End Time, # of person, Status, **Facility\_Name**)  Checking for 1NF  The primary key is a simple key.  The composite nonprime attribute has been decomposed.  Already in 1NF.  Checking for 2NF  The primary key is a simple key; therefore, there is no partial key.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key.  Facility (Facility\_Name, Type, Description)  Facility\_Location (**Facility\_Name**, **Bldg\_Name**, Flr. #)  Same as earlier defined in strong entity. |

|  |
| --- |
| Building (Bldg\_Name, City, Province, Street, Zip Code, Bldg. #)  Facility (Facility\_Name, Type, Description)  Facility\_Location (**Facility\_Name**, **Bldg\_Name**, Flr. #)  Building\_Has\_Facility (**Bldg\_Name, Facility Name**)  Building (Bldg\_Name, City, Province, Street, Zip Code, Bldg. #)  Checking for 1NF  Same as the earlier defined in strong entity.  Facility (Facility\_Name, Type, Description)  Facility\_Location (**Facility\_Name**, **Bldg\_Name**, Flr. #)  Same as the earlier defined in strong entity.  Building\_Has\_Facility (**Bldg\_Name, Facility Name**)  Checking for 1NF  The primary key is composite key.  There is no nonprime attribute; therefore, already in 1NF.  Checking for 2NF  The primary key is composite key.  There is no nonprime attribute.  Checking for 3NF  No transitive functional dependency on nonprime attributes.  BCNF  No nonprime attribute can determine the key. |