**Kathmandu University**

**Department of Computer Science and Engineering**

**Dhulikhel, Kavre**



**A Project Report**

**On**

**“CampusHousing\_DB”**

**[Code No: COMP 231]**

**(For partial fulfillment of 2nd year/2nd Semester in Computer Science)**

**Submitted by**

**Pragyan Shrestha (38)**

**Submitted to:**

**Sanjog Sigdel**

**Department of Computer Science and Engineering**

**Submission Date: 31-12-2024**

Contents

[Introduction 2](#_Toc186058201)

[ER Diagram 2](#_Toc186058202)

[Normalization of table (1NF -> 2NF -> 3NF) : 3](#_Toc186058203)

[3](#_Toc186058204)

[Building Table : 6](#_Toc186058205)

[Student Table : 6](#_Toc186058206)

[Room table : 6](#_Toc186058207)

[Roommate table : 7](#_Toc186058208)

[Projection 8](#_Toc186058209)

[Projection on Room : 8](#_Toc186058210)

[Cartesian Product 8](#_Toc186058211)

[Union operation : 8](#_Toc186058212)

[Exception Operation : 9](#_Toc186058213)

[Inner join : 9](#_Toc186058214)

[Left join : 10](#_Toc186058215)

[Right join : 10](#_Toc186058216)

[Transaction Management : 11](#_Toc186058217)

[Transaction with error scenario : 11](#_Toc186058218)

[Trasnaction without Rollback scenario : 11](#_Toc186058219)

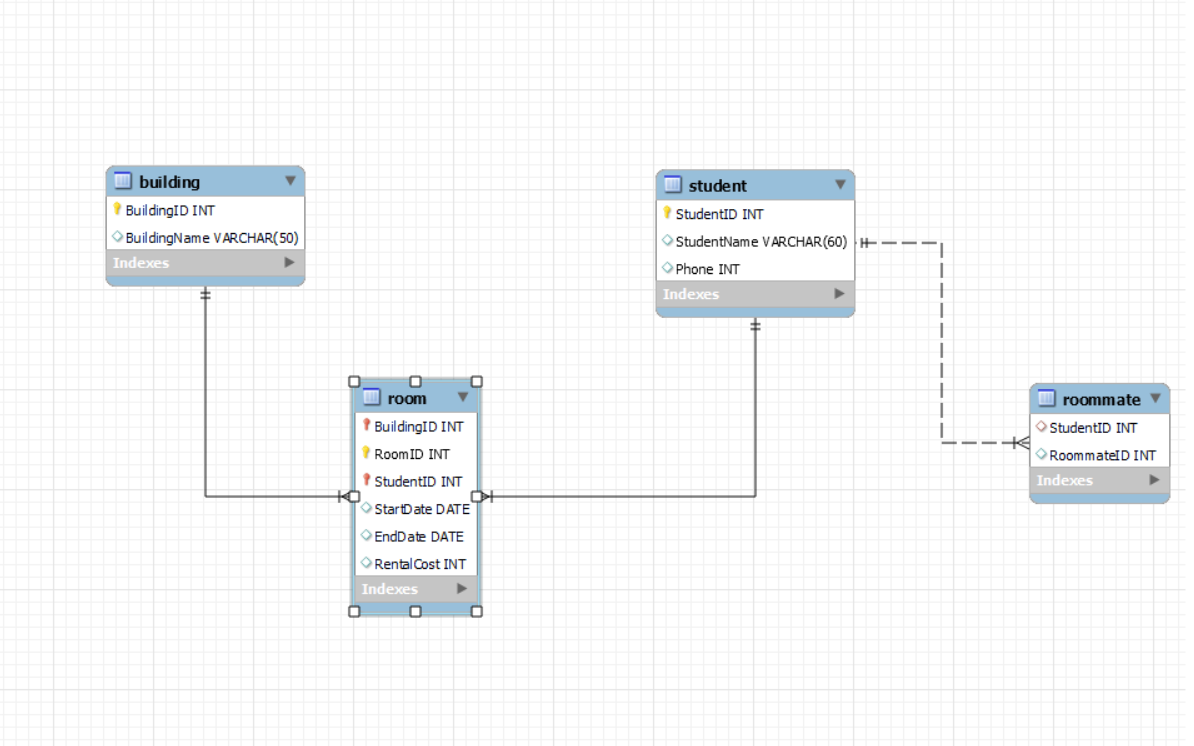
[Transaction with Rollback : 12](#_Toc186058220)

[Conclusion 13](#_Toc186058221)

# Introduction

The following project demonstrates SQL concepts like normalization, transaction, procedure and various SQL operation in MySQL Workbench. The database based on a campus housing has tables for Building, Room, Student and Roommate. All of the SQL operation have been executed as shown and taught in the classroom environment.

# ER Diagram



# Normalization of table (1NF -> 2NF -> 3NF) :

**1NF Table** :

**Creation :** create table campusHousing (

BuildingID int,

BuildingName varchar (50),

RoomNo int,

StudentID int,

StudentName varchar (60),

Phone int,

StartDate date,

EndDate date,

RentalCost int,

RoomateID int );

**Insertion :**

insert into campusHousing (BuildingID,

BuildingName,

RoomNo,

StudentID,

StudentName,

Phone,

StartDate,

EndDate,

RentalCost,

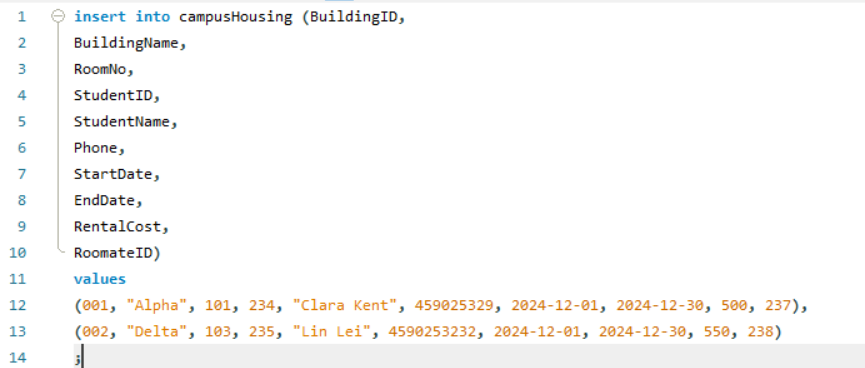
RoomateID)

values

(001, "Alpha", 101, 234, "Clara Kent", 252452, "2024-12-01", "2024-12-30", 500, 237),

(002, "Delta", 103, 235, "Lin Lei", 34144, "2024-12-01", "2024-12-30", 550, 238);

# A screenshot of a computer Description automatically generatedA screenshot of a phone call Description automatically generated

Insertion of data : 

A screenshot of a computer

Description automatically generated

**Creation :**

create table Building (

BuildingID int,

BuildingName varchar (50),

primary key (BuildingID)

);

create table Student (

StudentID int,

StudentName varchar (60),

Phone int,

primary key (StudentID)

);

create table Room (

BuildingID int,

RoomID int,

StudentID int,

StartDate date,

EndDate date,

RentalCost int,

RoommateID int,

primary key (BuildingID, StudentID, RoomID),

FOREIGN KEY (BuildingID) REFERENCES Building(BuildingID),

FOREIGN KEY (StudentID) REFERENCES Student(StudentID)

);

**Insertion** :

insert into Building(BuildingID, BuildingName)

values

(001, "Alpha"),

(002, "Delta");

insert into Student(StudentID, StudentName, Phone)

values

(01, "Clara", 452523),

(02, "Lin", 13241);

insert into Room(BuildingID, RoomID, StudentID, StartDate, EndDate, RentalCost, RoommateID)

values

(001, 1, 01, "2024-12-01", "2024-12-30", 500, 03),

(002, 2, 02, "2024-12-01", "2024-12-30", 550, 04);

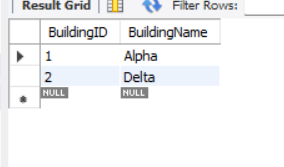
**2NF Table** : For conversion to 2NF table the table should be in 1NF and all the non primary values should be wholly dependent on primary keys. We do that here by breaking down the table into multiple smaller tables : Building, Room, Student each with their own respective values. A screenshot of a computer program

Description automatically generated

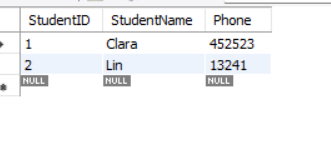
A screenshot of a computer

Description automatically generated

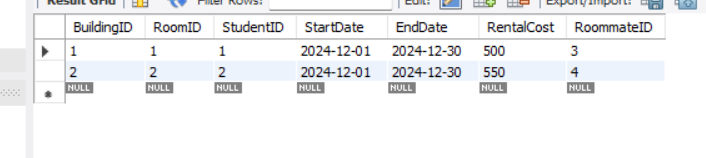
## Building Table :



## Student Table :



## Room table :



3NF table : A table should be in 2NF and transitive dependency must be eliminated. It is done here by breaking the Room table into Room assignment and Roomate assignment table. As BuildingID, RoomID, StudentID -> (determines) StudentID -> (determines) RommateID which implies BuildingID, RoomID, StudentID -> RommateID so this dependency has to be removed by creating another table.

**Convesion to 3NF :**

alter table Room drop column RoommateId;

create table Roommate (

StudentID int,

RoommateID int,

foreign key (StudentID) references Student (StudentID));

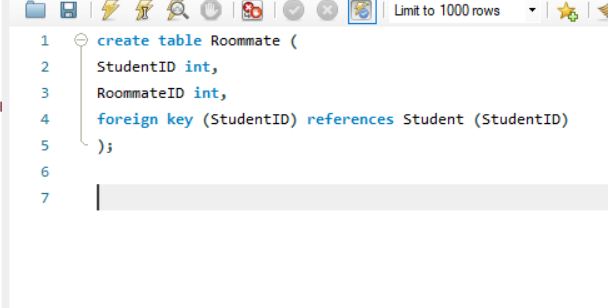
**Inserting values :**

insert into roommate (StudentID, RoommateID)

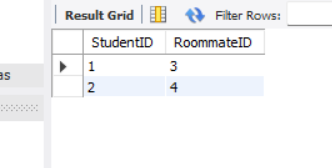
values

(1, 3),

(2, 4);

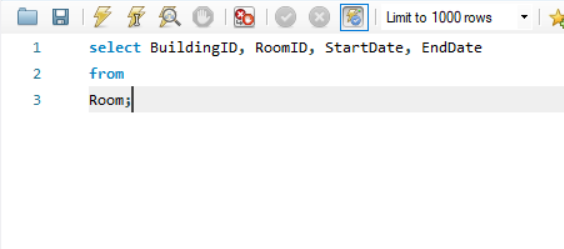


## Roommate table :



# Projection

## Projection on Room :

A screenshot of a computer

Description automatically generated

# Cartesian Product

## Union operation :

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## Exception Operation :

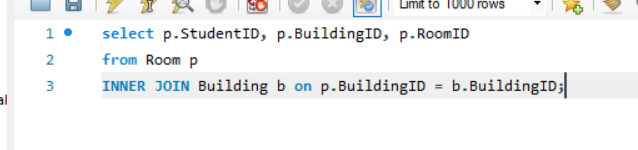
A screenshot of a computer

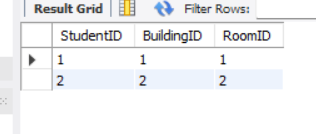
Description automatically generated

A screenshot of a computer

Description automatically generated

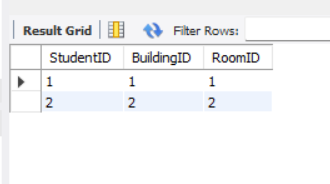
## Inner join :





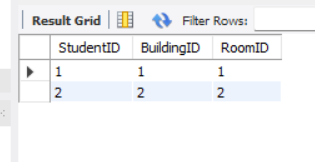
## Left join :

A screenshot of a computer

Description automatically generated

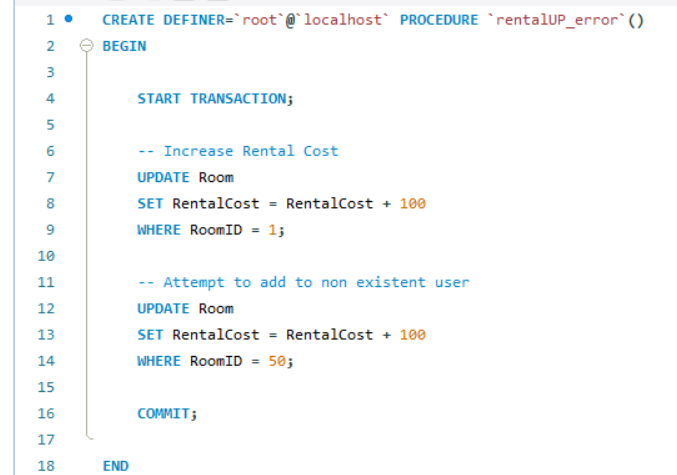
## Right join :

A screenshot of a computer

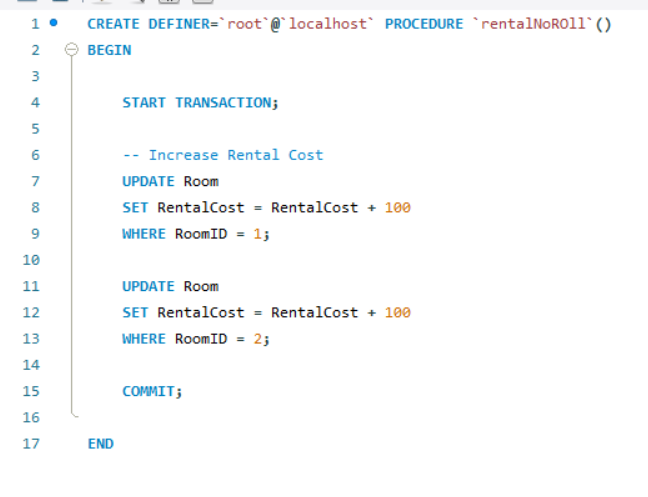
Description automatically generated

# Transaction Management :

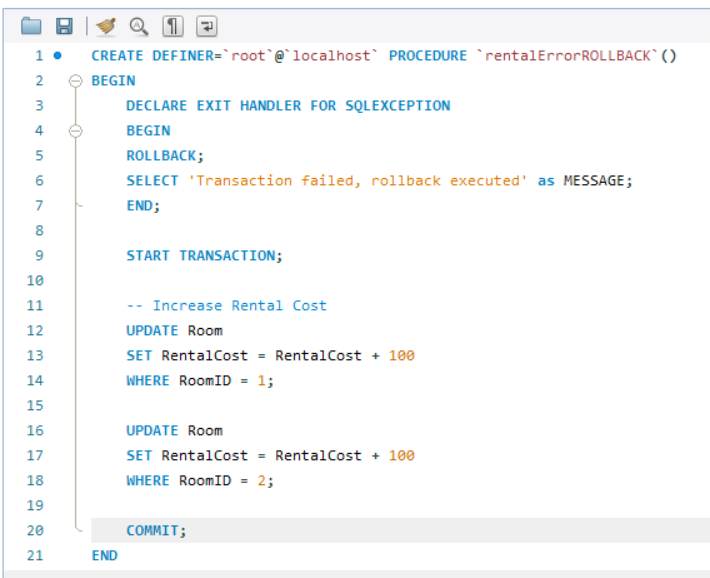
## Transaction with error scenario :



## Trasnaction without Rollback scenario :



## Transaction with Rollback :



# Conclusion

The above queries and operation successfully demonstrate SQL knowledge like normalization, insertion, updating, projection up to creating procedures for different scenarios. The implementation in a campusHousing\_DB ensures data integrity and efficiency as well as proper error handling.