



**GLA**  
**UNIVERSITY**  
**MATHURA**  
Established vide U.P. Act 21 of 2010.

## ASSIGNMENT: CASE STUDY

**Name of the Student** = **RITIK GUPTA**

**Course and Branch** = **B.TECH(Computer Science)**

**University Roll no.** = **191500659**

**Section (Roll no.)** = **C (48)**

**Year** = **2nd**

**Semester** = **III**

**Faculty** = **Dr. Neeraj Gupta (Assistant Professor)**

**Subject** = **Applied Database Management System  
(BCSC 0014)**

**TOPIC :: HOSTEL**

**MANAGEMENT**

**SYSTEM**



# **ACKNOWLEDGEMENT**

I would like to express my special thanks pf gratitude towards my teacher Dr.Neeraj Gupta, who gave me the golden opportunity to this wonderful project of Case Study , who also guided me in completing my project.

I have tried all my best efforts and do a lot what that project needed from my side for its better representation. This is my III Sem. Applied DBMS project is mainly based on "Hostel Management System" ( in a real life scenario). And so I am able to make this project report successfully.

Thanks to all.

Regrads,

*Ritik Gupta*

Date :- 20/Dec/2020

## **Table Of Contents**

- **Introduction**
  - ⇒ *Purpose*
  - ⇒ *Objective*
  - ⇒ *Scope of the Project*
  - ⇒ *Overview of Project*
- **Overall Description**
  - ⇒ *System Requirement*
- **User Requirements Definition**
- **Data Design**
  - ⇒ *Entity Relationship Diagram*
  - ⇒ *Conceptual Schema*
- **Entities and Attributes**
  - ⇒ *Hostel*
  - ⇒ *Administrator*
  - ⇒ *Student*
  - ⇒ *Room*
  - ⇒ *Visitors*

⇒ *Furniture*

- **Connectivity between MySQL and Python3**
- **Creation of Database using Python**
- **Implementation of tables in MySQL**
- **Implementation of queries in Python3**
- **Software Requirements**
  - ⇒ Tools
    - ⇒ MySQL Server
    - ⇒ Python3
    - ⇒ Draw.io
    - ⇒ Operating System :: Microsoft Windows
  - ⇒ Drivers
    - ⇒ MySQL Connector / Python
- **Conclusion**

- **INTRODUCTION**

- **Purpose :-**

The Software Requirements Specification (SRS) will provide a detailed description of requirements for the Hostel Management System (HMS). This SRS will be helpful for complete understanding what is to be expected from the newly introduced system which is to be constructed. From this SRS, the Hostel Management System can be designed, constructed and finally tested.

- **Objective :-**

- i).To deal with Hostel Management System in an easy and an efficient manner.
- ii).Create strong and secrete database that allow for any connection in a secret way, to prevent any outside or inside attacks.

- **Scope of the Project :-**

- i).Hostel Management System is designed for Hostel (like schools, Universities).
- ii).There will be predefined criteria for the Reservation to the hostels.
- iii).He/She checks the attested application forms of the students obtained from the internet and verify it with the student database.

iv).If the students are found eligible then they are allotted to the hostel Room.

➤ **Overview of Project** :-

Hostel Room Allocation System is a web application which aims at computerization of current procedure of allocating hostel rooms. Currently the process involves students filling up the forms and submitting them in respective hostel offices which involves a lot of paperwork, hence less efficient.

● **OVERALL DESCRIPTION**

➤ **System Requirement** :-

The Web Application has two main parts:

- 1) Hostel Administrators
- 2) Students

The student can select among the allocated hostel to a specified batch and the Hostel Administrator can assign the room number in the specific hostel that the student has selected upon the availability.

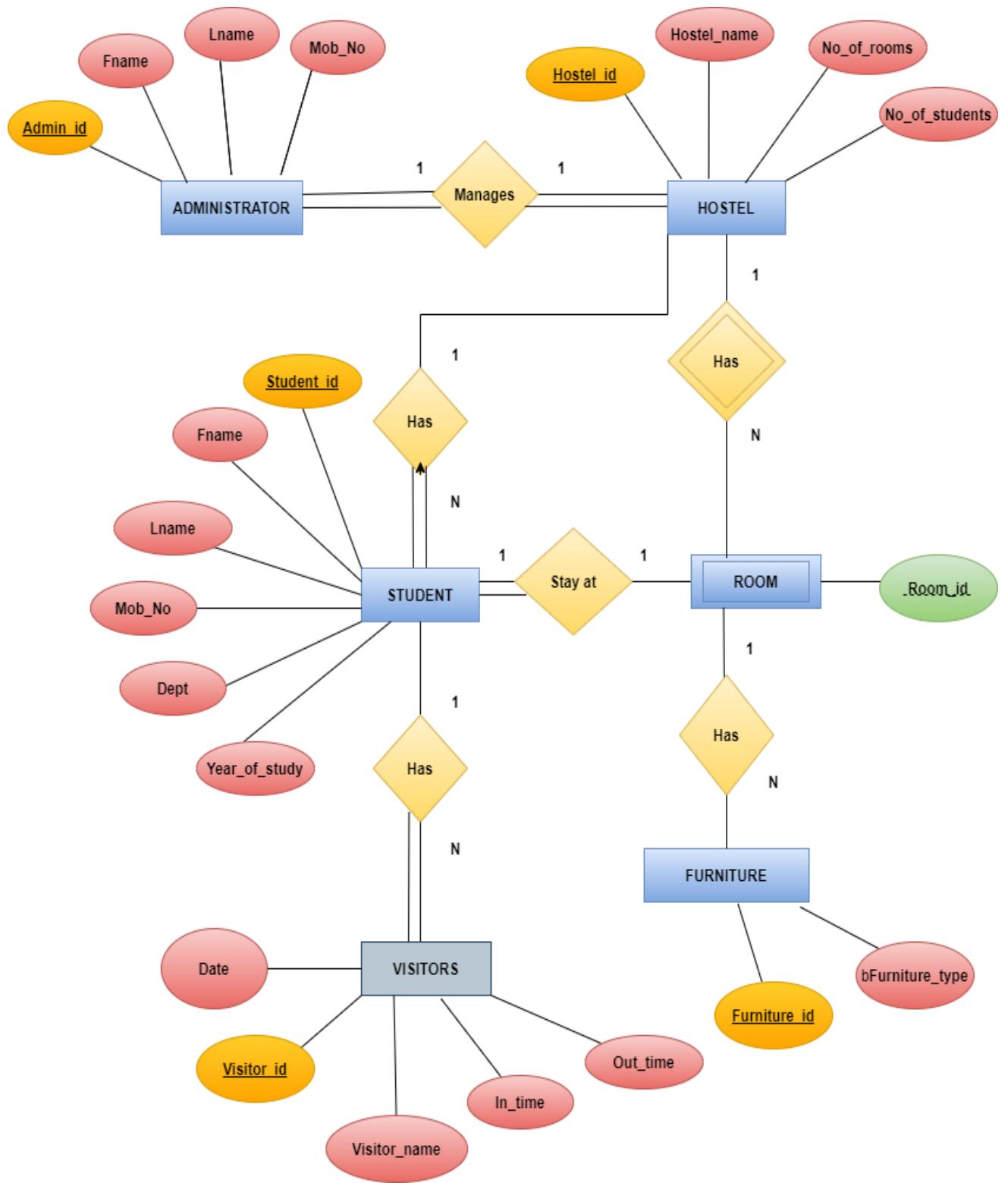
- **USER REQUIREMENTS**  
**DEFINITION**

The user requirement for this system is to make the system fast, flexible, less prone to error, reduce expenses and save the time.

1. Less human error
2. Strength and strain of manual labor can be reduced
3. High security
4. Data redundancy can be avoided to some extent
5. Data consistency
6. Easy to handle
7. Easy data updating
8. Easy record keeping
9. Backup data can be easily generated.

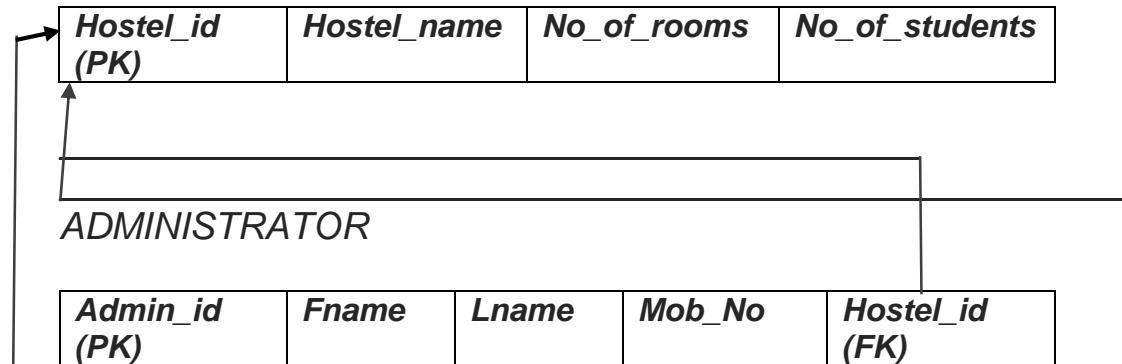
- **DATA DESIGN**  
➤ **Entity Relationship Diagram** :-

An entity relationship diagram is a pictorial representation of the entities that are interest to some enterprise and the relationship that hold among these entities.

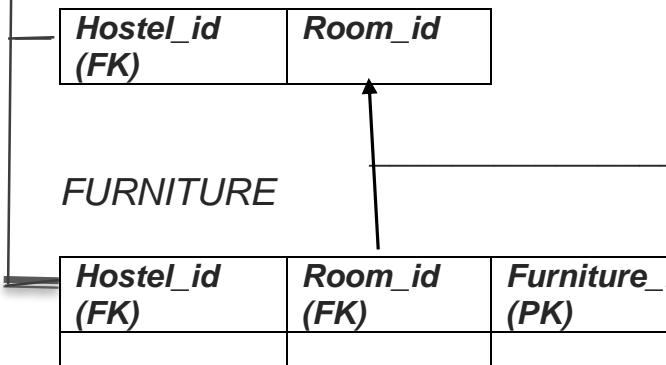


➤ **Conceptual Schema** :-

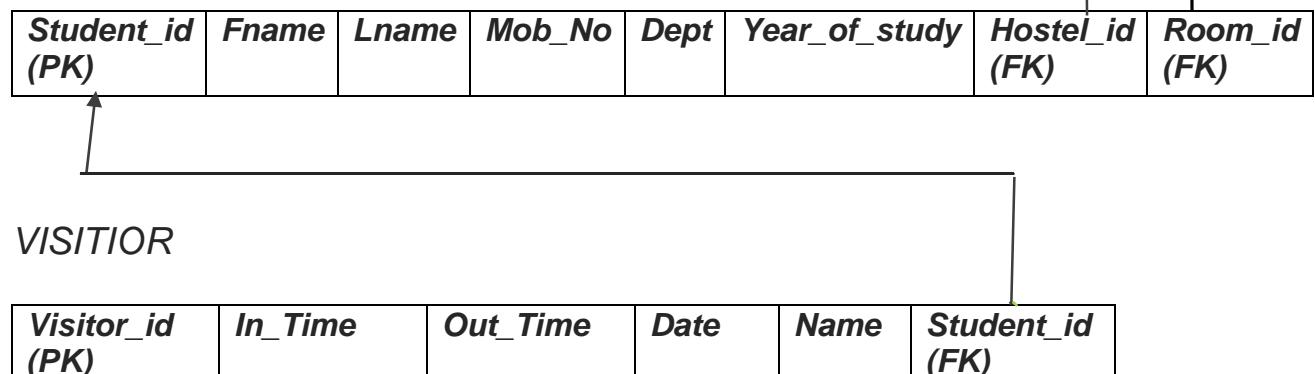
*HOSTEL*



*ROOM*



*STUDENT*



- **ENTITIES AND ATTRIBUTES**

This section of the document explains the entities used in the project, their attributes and how they will work together. Basically, this is intended to make the design more easy and understandable for everyone.

➤ **Entities :-**

1. Hostel
2. Administrator
3. Student
4. Room
5. Visitor
6. Furniture

➤ **Attributes :-**

**#Hostel**

An Institution has many hostels and each hostel is represented using this ‘Hostel’ entity. Hostel model takes part in the following relationships.

1. Administrator manages Hostel .
2. Hostel has Students.
3. Hostel has Rooms.

Name	Data Type	Type
Hostel_id	Integer	Primary key attribute
Hostel_name	String	Non_key attribute
No_of_rooms	Integer	Non_key attribute
No_of_students	Integer	Non_key attribute

## **#Administrator**

**Every hostel has an administrator and is represented using the ‘administrator’ entity. Administrator entity takes part in following relationships.**

### **1. Administrator manages Hostel.**

Name	Data Type	Type
Admin_id	Integer	Primary key attribute
Fname	String	Non_key_attribute
Lname	String	Non_key_attribute
Mob_No	String	Non_key_attribute
Hostel_id	Integer	Foreign Key attribute

## **#Student**

**Every hostel has students and they are represented by the ‘student’ entity.**

**Student entity participates in the following relationships.**

- 1. Hostel has Students .**
- 2. Student has visitor.**
- 3. Students stay at room.**

Name	Data Type	Type
Student_id	Integer	Primary Key attribute
Fname	String	Non_key_attribute
Lname	String	Non_key_attribute
Mob_No	String	Non_key_attribute
Dept	String	Non_key_attribute
Year_of_study	Integer	Non_key_attribute
Hostel_id	Integer	Foreign Key attribute
Room_id	Integer	Foreign Key attribute

## **#Room**

**Every Hostel has rooms and they are represented using ‘room’ entity. Room**

entity participates in the following relationships.

1. Hostel has Rooms .
2. Student stays at room .
3. Room has Furniture.

Name	Data Type	Type
Hostel_id	Integer	Foreign Key attribute
Room_id	Integer	Foreign Key attribute

## #Visitors

Every student has visitors and they are represented using ‘Visitor’ entity. Visitor

entity participates in the following relationships.

1. Student has visitors .

Name	Data Type	Type
Visitor_id	Integer	Primary Key attribute
In_time	Date-time-field	Non_key attribute
Out_time	Date-time-field	Non_key attribute
Date	Date-time-field	Non_key attribute
Name	String	Non_key attribute
Student_id	Integer	Foreign Key attribute

## #Furniture

Every room has furniture and they are represented using ‘furniture’ entity.

Furniture participated in following relationships.

1. Room has Furniture .

Name	Data Type	Type
Furniture_id	Integer	Primary Key attribute
Room_id	Integer	Foreign Key attribute
Hostel_id	Integer	Foreign Key attribute
Furniture_type	String	Non_key attribute

- **CONNECTIVITY BETWEEN  
MYSQL AND PYTHON**

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with icons for File Explorer, Task View, Edge browser, Mail, Google Chrome, and File Explorer. The main area displays two windows: a code editor and a terminal window.

**Code Editor:** The title bar says "python c.py - C:/Users/dell/python c.py (3.9.0)". The content of the script is:

```
import mysql.connector
conn = mysql.connector.connect(host="localhost", user="Ritik",
                                password="9837546958", port = 3307)
if conn.is_connected():
    print("Successfully Connected")
else:
    print("Unable to Connect")
```

**Terminal Window:** The title bar says "Python 3.9.0 Shell". The output shows the Python interpreter running the script and connecting to MySQL:

```
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/dell/python c.py =====
Successfully Connected
>>>
```

The status bar at the bottom of the terminal window indicates "Ln: 6 Col: 4".

- **CREATION OF DATABASE  
USING PYTHON**

```

HostelManagementSystem.db.py - C:/Users/dell/Documents/Applied DBMS lab/HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
try:
    db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                  password = "9837546958", port = 3307)
    print("Connected Successfully")
except Exception as e:
    print("Unable to Connect")
    print(e)

try:
    mycursor = db.cursor()
    mycursor.execute("CREATE DATABASE hms")
    print("Database Created Successfully")

except Exception as e:
    print("Database Not Created")
    print(e)

```

Python 3.9.0 Shell

```

File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/dell/Documents/Applied DBMS lab/HostelManagementSystem_db.py
Connected Successfully
Database Created Successfully
>>>

```

MySQL Workbench

Schemas

- hms
- sys
- test
- test1
- test2

Query 1

```

SHOW databases;

```

Result Grid | Filter Rows | Export | Wrap Cell Contents | Result 1 x

Database
hms
information_schema
mysql
performance_schema
sys
test
test1
test2

No object selected

Object Info Session

- **IMPLEMENTATION OF TABLES  
IN MYSQL**

# #Hostel

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (hms, sys), Tables, Views, Stored Procedures, Functions.
- Query Editor:** Title: Query 1. Contains the following SQL code:

```
1 • use hms
2 • Create table Hostel( Hostel_id Integer, Hostel_Name Varchar(20) not null, No_of_Rooms Integer not null,
3 •                                     No_of_Students Integer not null, Primary Key (Hostel_id));
4 • Insert into Hostel (Hostel_id, Hostel_Name, No_of_Rooms, No_of_Students) values ( 1001, "Pt.Lokmani Sharma", 200, 600);
5 • Insert into Hostel (Hostel_id, Hostel_Name, No_of_Rooms, No_of_Students) values ( 1009, "Dr.S.Radhakrishna", 350, 800);
6 • Insert into Hostel (Hostel_id, Hostel_Name, No_of_Rooms, No_of_Students) values ( 1004, "Dr.Rajendra Prasad", 100, 300);
7 • Insert into Hostel (Hostel_id, Hostel_Name, No_of_Rooms, No_of_Students) values ( 1003, "Dr C.V. Raman", 250, 720);
8 • Insert into Hostel (Hostel_id, Hostel_Name, No_of_Rooms, No_of_Students) values ( 1008, "Kalpana Chawala", 200, 580);
9 • Insert into Hostel (Hostel_id, Hostel_Name, No_of_Rooms, No_of_Students) values ( 1010, "Sarojini Naidu", 200, 600);
10 • Insert into Hostel (Hostel_id, Hostel_Name, No_of_Rooms, No_of_Students) values ( 1006, "Sir Visvesvaraya", 150, 450);
11 • Select * from hostel
```
- Result Grid:** Shows the data inserted into the Hostel table.

Hostel_Id	Hostel_Name	No_of_Rooms	No_of_Students
1001	Pt.Lokmani Sharma	200	600
1003	Dr.C.V. Raman	250	720
1004	Dr.Rajendra Prasad	100	300
1006	Sir Visvesvaraya	150	450
1008	Kalpana Chawala	200	580
1009	Dr.S.Radhakrishna	350	800
1010	Sarojini Naidu	200	600
NULL	NULL	NULL	NULL
- Object Info:** Session: hostel 5 x. Contains a search bar and system status indicators (ENG 01:19 IN 12/14/2020).

# #Administrator

MySQL Workbench

Ritik x

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS

hms

- Tables
- Views
- Stored Procedures
- Functions

sys

- Tables
- sys\_config

  - Columns
  - variable
  - value
  - set\_time
  - set\_by

- Indexes
- PRIMARY
- ForeignKeys
- Triggers
- sys\_config\_ir
- sys\_config\_u
- Views
- Stored Procedures
- Functions

test

Administration Schemas

No object selected

Object Info Session Administrator 12 x

Type here to search

Result Grid | Filter Rows: [ ] | Edit: [ ] | Export/Import: [ ] | Wrap Cell Content: [ ]

Admin_id	Fname	Lname	Mob_No	Hostel_id
10	Amit	Tarkar	9484765004	1004
12	Kuldeep	Sharma	774455980	1003
15	Sara	Khan	7654094200	1010
20	Kalpana	Mishra	9087653450	1008
22	Sooraj	Gupta	6754300150	1006
25	Dharmendra	Singh	8765409332	1001
32	Rajesh	Sharma	7765432350	1009
HULL	HULL	HULL	HULL	HULL

Result Grid Form Editor Field Types Query Stats Execution Plan

```

1 • use hms
2 • Create table Administrator( Admin_id Integer, Fname Varchar(20) not null, Lname Varchar(20) not null, Mob_No Varchar(20) not null, Hostel_id Integer,
3 • Primary Key (Admin_id), Foreign Key (Hostel_id) references Hostel(Hostel_id) ON DELETE CASCADE);
4 • Insert into Administrator(Admin_id , Fname, Lname, Mob_No, Hostel_id ) values ( 25, "Dharmendra", "Singh", "8765409332", 1001);
5 • Insert into Administrator(Admin_id , Fname, Lname, Mob_No, Hostel_id ) values ( 32, "Rajesh", "Sharma", "7765432350", 1009);
6 • Insert into Administrator(Admin_id , Fname, Lname, Mob_No, Hostel_id ) values ( 10, "Amit", "Tarkar", "9484765004", 1004);
7 • Insert into Administrator(Admin_id , Fname, Lname, Mob_No, Hostel_id ) values ( 12, "Kuldeep", "Sharma", "774455980", 1003);
8 • Insert into Administrator(Admin_id , Fname, Lname, Mob_No, Hostel_id ) values ( 20, "Kalpana", "Mishra", "9087653450",1008);
9 • Insert into Administrator(Admin_id , Fname, Lname, Mob_No, Hostel_id ) values ( 15, "Sara", "Khan", "7654094200", 1010);
10 • Insert into Administrator(Admin_id , Fname, Lname, Mob_No, Hostel_id ) values ( 22, "Sooraj", "Gupta", "6754300150",1006);
11 • Select * from Administrator;

```

# #Student

MySQL Workbench

Ritik x

File Edit View Query Database Server Tools Scripting Help

Navigator Query 1

SCHEMAS

hms

- Tables
  - administrator
  - Columns
    - Admin\_Id
    - Fname
    - Lname
    - Mob\_No
    - Hostel\_Id
  - Indexes
    - PRIMARY
    - Hostel\_Id
  - Foreign Keys
  - Triggers
- hostel
- room

Administration Schemas

Information

No object selected

Object Info Session Student 15 x Read Only

Type here to search

Result Grid Form Editor Field Types Query Stats Execution Plan

```
2 • CREATE TABLE Student( Student_id Integer, Fname Varchar(20) not null, Lname Varchar(20) not null, Mob_No Varchar(20) not null,
3 •                                         Dept Varchar(10) not null, Year_of_Study Integer not null, Hostel_id Integer, room_id Integer,
4 •                                         Foreign key (Hostel_id) references Hostel(Hostel_id));
5 • Insert into Student(Student_id, Fname, Lname, Mob_No, Dept, Year_of_Study, Hostel_id, Room_id) values ( 48, "Ritik", "Gupta", "8630765432", "B.tech", 2, 1001, 133);
6 • Insert into Student(Student_id, Fname, Lname, Mob_No, Dept, Year_of_Study, Hostel_id, Room_id) values ( 23, "Prashant", "Choudhary", "8887765098", "B.tech", 3, 1009, 205);
7 • Insert into Student(Student_id, Fname, Lname, Mob_No, Dept, Year_of_Study, Hostel_id, Room_id) values ( 39, "Raunaq", "Arora", "7609880009", "BBA", 3, 1004, 104 );
8 • Insert into Student(Student_id, Fname, Lname, Mob_No, Dept, Year_of_Study, Hostel_id, Room_id) values ( 44, "Rishabh", "Singh", "7695099434", "BCA", 1, 1003, 310);
9 • Insert into Student(Student_id, Fname, Lname, Mob_No, Dept, Year_of_Study, Hostel_id, Room_id)values ( 12, "Suhana", "Verma", "8904567688", "B.Sc", 3, 1008, 205);
10 • Insert into Student(Student_id, Fname, Lname, Mob_No, Dept, Year_of_Study, Hostel_id, Room_id) values ( 16, "Payal", "Tiwari", "8907600987", "B.tech", 1, 1010, 016);
11 • Insert into Student(Student_id, Fname, Lname, Mob_No, Dept, Year_of_Study, Hostel_id, Room_id)values ( 28,"Prem", "Chopra", "7654320987", "BBA", 2, 1006, 2020);
12 • Select * from Student;
```

Student_id	Fname	Lname	Mob_No	Dept	Year_of_Study	Hostel_id	room_id
48	Ritik	Gupta	8630765432	B.tech	2	1001	133
23	Prashant	Choudhary	8887765098	B.tech	3	1009	205
39	Raunaq	Arora	7609880009	BBA	3	1004	104
44	Rishabh	Singh	7695099434	BCA	1	1003	310
12	Suhana	Verma	8904567688	B.Sc	3	1008	205
16	Payal	Tiwari	8907600987	B.tech	1	1010	016
28	Prem	Chopra	7654320987	BBA	2	1006	2020

## #Visitors

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for Home, Refresh, Undo, Redo, Save, Print, Find, Replace, Copy, Paste, and others.
- Navigator:** Shows the database schema for the 'hms' database, including the 'Visitors' table with columns: Visitor\_id, In\_time, Out\_time, Date, Visitor\_Name, and Student\_id.
- Query Editor:** Contains the SQL code for creating the 'Visitors' table and inserting 10 rows of data. The table is created with an auto-incrementing primary key 'Visitor\_id'. The inserted data includes visitors like Sunil, Brajesh, Rakesh, Manoj, Beena, Charu, and Jay, along with their respective visit times and dates.
- Result Grid:** Displays the 10 rows of data from the 'Visitors' table in a grid format.
- Information:** Shows the status message "No object selected".
- Bottom Bar:** Includes tabs for Object Info, Session, and Visitors 19, along with Apply and Revert buttons. A search bar and a set of small icons are also present.

## #Room

MySQL Workbench

Ritik x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

SCHEMAS: hms

Tables: administrator, room, student, visitors

Columns: Admin\_Id, Fname, Lname, Mob\_No, Hostel\_Id

Indexes: PRIMARY, Hostel\_Id

Foreign Keys:

Triggers:

hostel

Columns: Hostel\_Id, Hostel\_Name, No\_of\_Room, No\_of\_Stude

Indexes:

Foreign Keys:

Triggers:

Views:

Information: No object selected

Object Info Session Room 14 x

Type here to search

Result Grid | Filter Rows: Export: Wrap Cell Content:

Hostel_Id	Room_Id
1001	133
1009	205
1004	104
1003	310
1008	205
1010	16
1006	2020

Result Grid | Filter Rows: Export: Wrap Cell Content:

Form Editor

Field Types

Query Stats

Execution Plan

Read Only

ENG IN 03:21 12/14/2020

```

1 • use hms;
2 • CREATE TABLE ROOM( Hostel_id Integer, Room_id Integer, Foreign key (Hostel_id) references Hostel(Hostel_id));
3 • Insert into Room(Room_id, Hostel_id ) values ( 133, 1001);
4 • Insert into Room(Room_id, Hostel_id ) values ( 205, 1009);
5 • Insert into Room(Room_id, Hostel_id ) values ( 104, 1004);
6 • Insert into Room(Room_id, Hostel_id ) values ( 310, 1003);
7 • Insert into Room(Room_id, Hostel_id ) values ( 205,1008);
8 • Insert into Room(Room_id, Hostel_id ) values ( 016, 1010);
9 • Insert into Room(Room_id, Hostel_id ) values ( 2020,1006);
10 • Select * from Room;
11

```

## #Furniture

MySQL Workbench

Ritik x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

SCHEMAS: hms

Tables: administrator, room, student, visitors

Stored Procedures: sys

Functions: sys

test

test1

test2

Information: Schema: hms

Object Info Session Furniture 20 x

Type here to search

Result Grid | Filter Rows: Export/Import: Wrap Cell Content:

Furniture_id	Room_id	Hostel_Id	Furniture_type
29012	310	1003	Bed
30005	16	1010	Bed
38765	104	1004	Study Table
48453	205	1008	study Table
48675	2020	1006	Char
50552	133	1001	Char
65438	205	1009	Char
65438	205	1009	Char

Result Grid | Filter Rows: Export: Wrap Cell Content:

Form Editor

Field Types

Query Stats

Execution Plan

Apply

Revert

ENG IN 06:01 12/14/2020

```

1 • use hms;
2 • CREATE TABLE Furniture( Furniture_id Integer(10), Room_id Integer (10), Hostel_id Integer, Furniture_type Varchar(20) not null,
   primary key (FURNITURE_ID));
3
4 • Insert into Furniture(Furniture_id, Room_id, Hostel_id, Furniture_type) values (50552, 133, 1001, "Chair");
5 • Insert into Furniture(Furniture_id, Room_id, Hostel_id, Furniture_type) values (65438, 205,"1009", "Chair");
6 • Insert into Furniture(Furniture_id, Room_id, Hostel_id, Furniture_type) values (38765, 104, 1004, "Study Table");
7 • Insert into Furniture(Furniture_id, Room_id, Hostel_id, Furniture_type) values (29012, 310, 1003, "Bed");
8 • Insert into Furniture(Furniture_id, Room_id, Hostel_id, Furniture_type) values (48453, 205, 1008, "study Table");
9 • Insert into Furniture(Furniture_id, Room_id, Hostel_id, Furniture_type) values (30005, 016, 1010, "Bed");
10 • Insert into Furniture(Furniture_id, Room_id, Hostel_id, Furniture_type) values (48675, 2020, 1006,"Chair");
11 • Select * from Furniture;

```

- **IMPLEMENTATION OF QUERIES IN PYTHON3**

**1). Fetch all the values from Student table using various methods and attributes.**

=> Methods

i). fetchall() :- *return all the result in the form of tuples.*

The screenshot shows a Windows desktop environment. At the bottom, there's a taskbar with icons for File Explorer, Edge browser, Mail, Photos, and Task View. A search bar says "Type here to search". The system tray shows the date as 12/14/2020 and the time as 09:26. The desktop background is white.

**Code Editor Window:**

```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
mycursor.execute("Select * from Student")
a = mycursor.fetchall()
for i in a:
    print(i)
print(a)
db.close()
```

**Python Shell Window:**

```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133)
(23, 'Prashant', 'Choudhary', '8887765098', 'B.tech', 3, 1009, 205)
(39, 'Raunag', 'Arora', '7609880009', 'BBA', 3, 1004, 104)
(44, 'Rishabh', 'Singh', '7695099434', 'BCA', 1, 1003, 310)
(12, 'Suhana', 'Verma', '8904567688', 'B.Sc', 3, 1008, 205)
(16, 'Payal', 'Tiwari', '8907600987', 'B.tech', 1, 1010, 16)
(28, 'Prem', 'Chopra', '7654320987', 'BBA', 2, 1006, 2020)
[(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133), (23, 'Prashant',
'Choudhary', '8887765098', 'B.tech', 3, 1009, 205), (39, 'Raunag', 'Arora', '76
09880009', 'BBA', 3, 1004, 104), (44, 'Rishabh', 'Singh', '7695099434', 'BCA', 1
, 1003, 310), (12, 'Suhana', 'Verma', '8904567688', 'B.Sc', 3, 1008, 205), (16,
'Payal', 'Tiwari', '8907600987', 'B.tech', 1, 1010, 16), (28, 'Prem', 'Chopra',
'7654320987', 'BBA', 2, 1006, 2020)]
>>>
```

## **ii). fetchone():- return one row at a time as tuple.**

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with icons for File Explorer, Edge browser, Mail, Task View, and others. The system tray shows the date and time as 12/15/2020 10:12 AM, and the language as ENG IN. Below the taskbar, there are two open windows. The left window is a code editor titled "HostelManagementSystem\_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem\_db.py (3.9.0)". It contains Python code for connecting to a MySQL database and fetching one row at a time using the fetchone() method. The right window is a "Python 3.9.0 Shell" window. It shows the Python interpreter running the script and outputting the fetched data as tuples. The output shows four rows of student information: (48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133), (48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133), (23, 'Prashant', 'Choudhary', '8887765098', 'B.tech', 3, 1009, 205), and (23, 'Prashant', 'Choudhary', '8887765098', 'B.tech', 3, 1009, 205).

```
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")

mycursor = db.cursor()

mycursor.execute("Select * from Student")
a = mycursor.fetchone()
print(a)

b = mycursor.fetchone()
print(b)

db.close()
```

```
>>> 
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133)
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133)
(23, 'Prashant', 'Choudhary', '8887765098', 'B.tech', 3, 1009, 205)
>>>
```

## **iii). Fetchmany(<n>):- fetch <n> rows in the form of list of tuples.**

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with icons for File Explorer, Edge browser, Mail, Task View, and others. The system tray shows the date and time as 12/15/2020 10:15 AM, and the language as ENG IN. Below the taskbar, there are two open windows. The left window is a code editor titled "HostelManagementSystem\_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem\_db.py (3.9.0)". It contains Python code for connecting to a MySQL database and fetching four rows at once using the fetchmany(4) method. The right window is a "Python 3.9.0 Shell" window. It shows the Python interpreter running the script and outputting the fetched data as a list of tuples. The output shows four rows of student information: [(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133), (23, 'Prashant', 'Choudhary', '8887765098', 'B.tech', 3, 1009, 205), (39, 'Raunaq', 'Arora', '7609880009', 'BBA', 3, 1004, 104), (44, 'Rishabh', 'Singh', '7695099434', 'BCA', 1, 1003, 310)].

```
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")

mycursor = db.cursor()

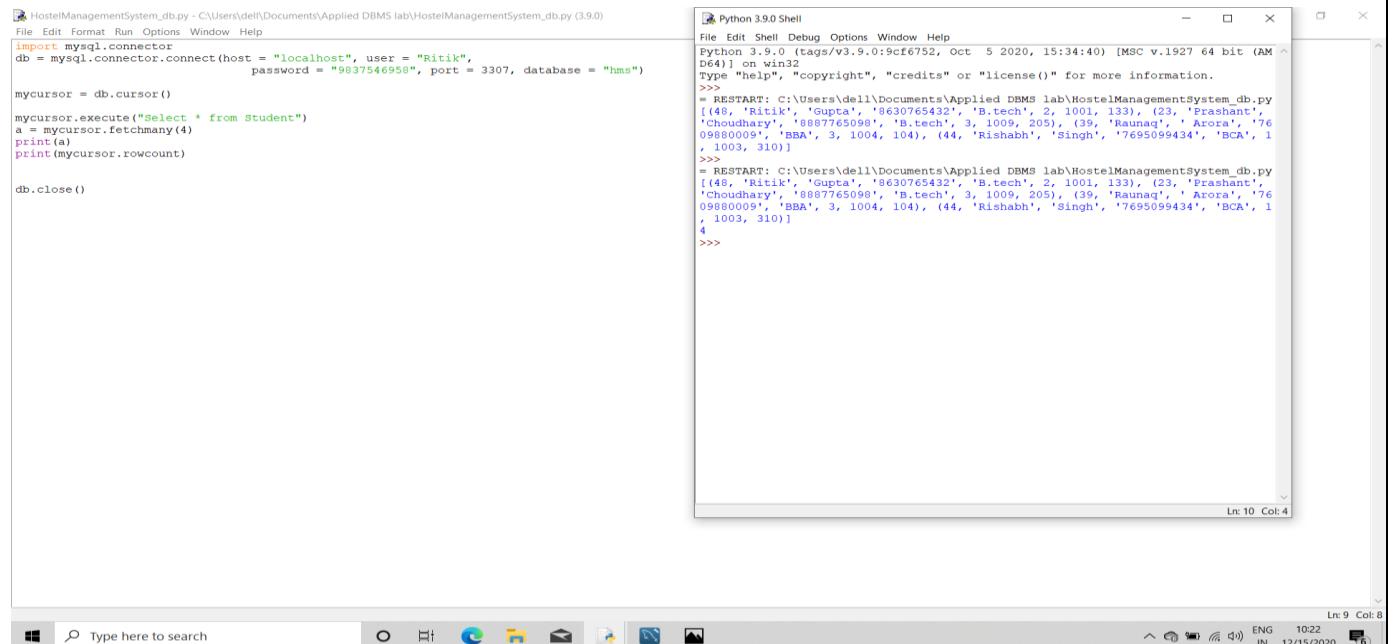
mycursor.execute("Select * from Student")
a = mycursor.fetchmany(4)
print(a)

db.close()
```

```
>>> 
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133), (23, 'Prashant', 'Choudhary', '8887765098', 'B.tech', 3, 1009, 205), (39, 'Raunaq', 'Arora', '7609880009', 'BBA', 3, 1004, 104), (44, 'Rishabh', 'Singh', '7695099434', 'BCA', 1, 1003, 310)]
>>> |
```

## =>Attributes

i). rowcount :- *returns the number of rows in the result set.*

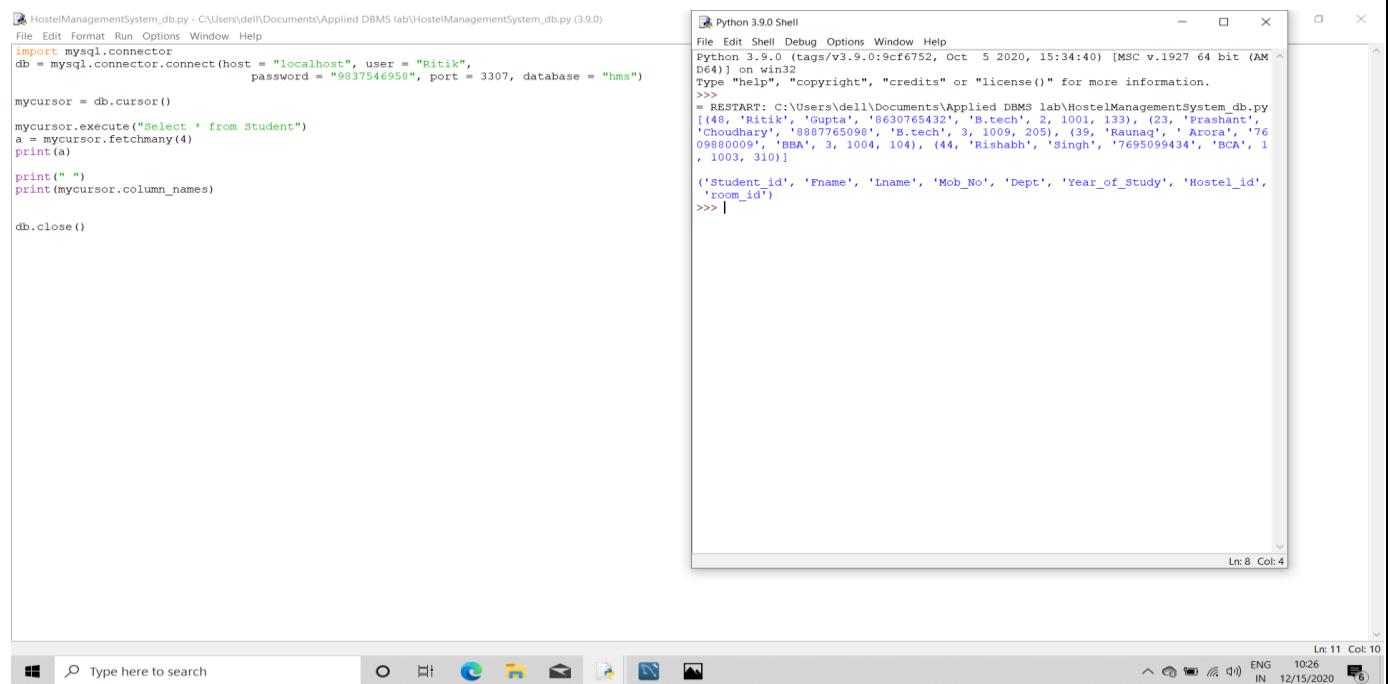


```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
mycursor.execute("Select * from Student")
a = mycursor.fetchmany(4)
print(a)
print(mycursor.rowcount)

db.close()

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133), (23, 'Prashant',
'Choudhary', '8887765098', 'B.tech', 3, 1009, 205), (39, 'Raunaq', 'Arora', '76
09880009', 'BBA', 3, 1004, 104), (44, 'Rishabh', 'Singh', '7695099434', 'BCA', 1
, 1003, 310)]
4
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133), (23, 'Prashant',
'Choudhary', '8887765098', 'B.tech', 3, 1009, 205), (39, 'Raunaq', 'Arora', '76
09880009', 'BBA', 3, 1004, 104), (44, 'Rishabh', 'Singh', '7695099434', 'BCA', 1
, 1003, 310)]
4
>>>
```

ii).column\_names :- *returns all the columns of table in the form of tuple.*



```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
mycursor.execute("Select * from Student")
a = mycursor.fetchmany(4)
print(a)
print(mycursor.column_names)

db.close()

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133), (23, 'Prashant',
'Choudhary', '8887765098', 'B.tech', 3, 1009, 205), (39, 'Raunaq', 'Arora', '76
09880009', 'BBA', 3, 1004, 104), (44, 'Rishabh', 'Singh', '7695099434', 'BCA', 1
, 1003, 310)]
('Student_id', 'Fname', 'Lname', 'Mob_No', 'Dept', 'Year_of_Study', 'Hostel_id',
'room_id')
>>> |
```

**iii). statement :- returns the SQL statement of the result set.**

The screenshot shows a Windows desktop environment. In the top-left corner, there is a code editor window titled "HostelManagementSystem\_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem\_db.py (3.9.0)". The code in the editor is:

```

import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                             password = "9837546958", port = 3307, database = "hms")

mycursor = db.cursor()

mycursor.execute("Select * from Student")
a = mycursor.fetchmany(4)
print(a)

print(" ")
print(mycursor.statement)

db.close()

```

In the top-right corner, there is a "Python 3.9.0 Shell" window. The shell output shows the results of the SQL query and the printed statements:

```

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[(48, 'Ritik', 'Gupta', '8630765432', 'B.tech', 2, 1001, 133), (23, 'Prashant',
'Choudhary', '8887765098', 'B.tech', 3, 1009, 205), (39, 'Raunaq', 'Arora', '76
09880009', 'BBA', 3, 1004, 104), (44, 'Rishabh', 'Singh', '7695099434', 'BCA', 1
, 1003, 310)]
Select * from Student
>>> |

```

## 2). Display the structure of visitors table.

The screenshot shows a Windows desktop environment. In the top-left corner, there is a code editor window titled "HostelManagementSystem\_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem\_db.py (3.9.0)". The code in the editor is:

```

import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                             password = "9837546958", port = 3307, database = "hms")

mycursor = db.cursor()

sql = "DESCRIBE VISITORS";
mycursor.execute(sql)
result = mycursor.fetchall()

print(result)

```

In the top-right corner, there is a "Python 3.9.0 Shell" window. The shell output shows the structure of the "VISITORS" table:

```

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[('Visitor_id', 'b'int', 'NO', 'PRI', 'None', 'auto_increment'), ('In_Time', 'b'varc
har(40), 'NO', 'None', ''), ('Out_time', 'b'varchar(40)', 'NO', 'None', ''), ('Date',
'b'varchar(20)', 'NO', 'None', ''), ('Visitor_Name', 'b'varchar(10)'), ('NO', 'None', ''),
('Student_id', 'b'int', 'YES', 'None', '')]
>>> |

```

### 3). Update mobile no. of the Admin named “Sooraj”.

The screenshot shows a Windows desktop environment. On the left, there is a code editor window titled "HostelManagementSystem\_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem\_db.py (3.9.0)". The code contains Python code for updating an administrator's mobile number in a MySQL database. On the right, there is a terminal window titled "Python 3.9.0 Shell" showing the output of running the script. The terminal output indicates that 1 record was affected.

```

HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")

mycursor = db.cursor()

sql = "Update Administrator set Mob_no = '7859409932' where Fname = 'Sooraj'";
mycursor.execute(sql)
db.commit()

print(mycursor.rowcount,"records affected")

```

```

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
1 records affected
>>>

```

The screenshot shows the MySQL Workbench interface. In the center, there is a query editor window titled "HMS tables queries\*". The query "select \* from Administrator;" has been run. Below the query editor, the "Result Grid" shows the data from the "administrator" table. The table has columns: Admin\_Id, Fname, Lname, Mob\_No, and Hostel\_Id. The data grid displays 10 rows of administrator information.

Admin_Id	Fname	Lname	Mob_No	Hostel_Id
10	Amit	Tarkar	9484765004	1004
12	Kuldeep	Sharma	774455980	1003
15	Sara	Khan	7654094200	1010
20	Kalpana	Mishra	9087653450	1008
22	Sooraj	Gupta	7859409932	1006
25	Dharmendra	Singh	8765409332	1001
32	Rajesh	Sharma	7765432350	1009
HULL	HULL	HULL	HULL	HULL
HULL	HULL	HULL	HULL	HULL
HULL	HULL	HULL	HULL	HULL

## 4). Display hostel details of student whose year\_of\_study is less than 2.

The screenshot shows a Windows desktop environment. On the left is a code editor window titled "HostelManagementSystem\_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem\_db.py (3.9.0)". It contains Python code for querying a MySQL database to select room details for students whose year of study is less than 2. On the right is a Python 3.9.0 Shell window, also titled "Python 3.9.0 Shell". The shell displays the command-line interface and the results of the executed query, which are empty lists: `[]` and `[]`.

```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
sql = "Select room.hostel_id, room.room_id from room inner join student on room.room_id = Student.room_id where student.year_of_study < 2";
mycursor.execute(sql)

result = mycursor.fetchall()

print(result)

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[], []
>>> |
```

## 5). Display the name of the student along with the name of the visitor whose student\_id = 48.

The screenshot shows a Windows desktop environment. On the left is a code editor window titled "HostelManagementSystem\_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem\_db.py (3.9.0)". It contains Python code for querying a MySQL database to select student names and visitor names for a specific student ID (48). On the right is a Python 3.9.0 Shell window, also titled "Python 3.9.0 Shell". The shell displays the command-line interface and the results of the executed query, which are tuples: `('Ritik', 'Gupta', 4831, 'Sunil')` and `['Ritik', 'Gupta', 4831, 'Sunil']`.

```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
s="Select Student.Fname, Student.Lname, Visitors.visitor_id, Visitors.visitor_name from student inner join Visitors on student.student_id = Visitors.student_id where Visitors.student_id=48"
mycursor.execute(s)

result = mycursor.fetchall()

print(result)

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[('Ritik', 'Gupta', 4831, 'Sunil')]
['Ritik', 'Gupta', 4831, 'Sunil']
>>> |
```

## 6). Find average no. of rooms in each hostel.

The screenshot shows a Windows desktop environment. At the top is the taskbar with icons for Start, Search, Task View, File Explorer, Mail, Photos, and Python. Below the taskbar are two windows: a code editor titled "HostelManagementSystem\_db.py" and a Python 3.9.0 Shell window. The code editor contains the following Python script:

```
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                             password = "9837546958", port = 3307, database = "hms")

mycursor = db.cursor()
sql = "Select round(avg(no_of_rooms),2) from hostel"
mycursor.execute(sql)

result = mycursor.fetchall()

print(result)
```

The Python shell window shows the output of the script:

```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[Decimal('207.14'),)
```

At the bottom right of the desktop, there is a system tray with icons for battery, signal, and date/time (12/14/2020).

## 7). Display hostel names where no. of students is above average.

The screenshot shows a Windows desktop environment. At the top is the taskbar with icons for Start, Search, Task View, File Explorer, Mail, Photos, and Python. Below the taskbar are two windows: a code editor titled "HostelManagementSystem\_db.py" and a Python 3.9.0 Shell window. The code editor contains the following Python script:

```
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                             password = "9837546958", port = 3307, database = "hms")

mycursor = db.cursor()

sql="Select hostel.hostel_name from hostel where no_of_students >(Select round(avg(no_of_students),2)from hostel)"

mycursor.execute(sql)
result = mycursor.fetchall()

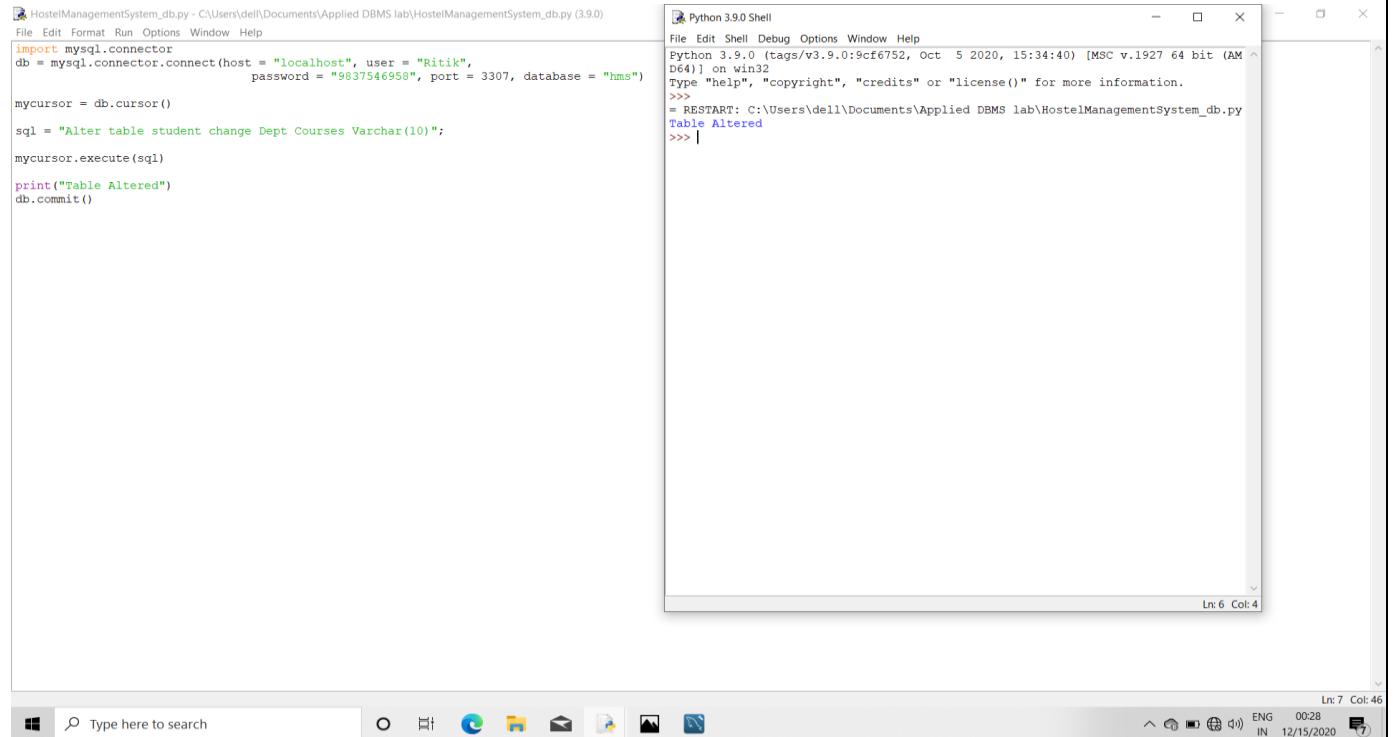
print(result)
```

The Python shell window shows the output of the script:

```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
[("Pt.Lokmani Sharma"), ('Dr C.V. Raman'), ('Kalpana Chawala'), ('Dr.S.Radhak
rishna'), ('Sarojini Naidu',)]
```

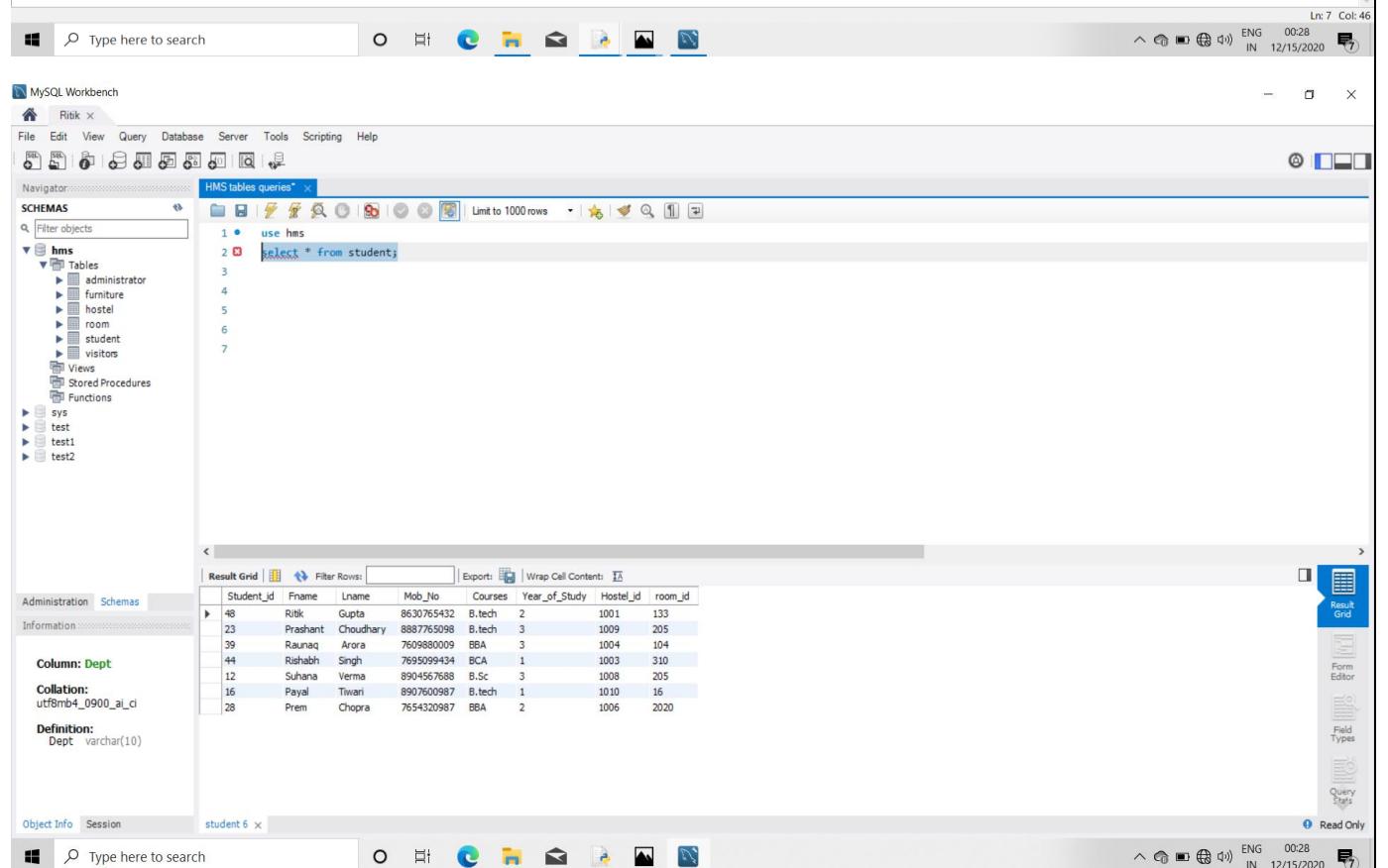
At the bottom right of the desktop, there is a system tray with icons for battery, signal, and date/time (12/15/2020).

## 8). Change column name from Dept to courses in student table.



```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
sql = "Alter table student change Dept Courses Varchar(10);"
mycursor.execute(sql)
print("Table Altered")
db.commit()

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
Table Altered
>>> |
```



```
MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Ritik x
Navigator: HMS tables queries*
SCHEMAS
Filter objects
Tables
    administrator
    furniture
    hostel
    room
    student
    visitors
    Views
    Stored Procedures
    Functions
sys
test
test1
test2

1 use hms
2 select * from student;
3
4
5
6
7

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
Administration Schemas
Information: Column: Dept
Definition: Dept varchar(10)

Object Info Session student 6 x
Type here to search
```

Student_id	Fname	Lname	Mob_No	Courses	Year_of_Study	Hostel_id	room_id
48	Ritik	Gupta	8630765432	B.tech	2	1001	133
23	Prashant	Choudhary	8887765098	B.tech	3	1009	205
39	Raunaq	Arora	7609880009	BBA	3	1004	104
44	Rishabh	Singh	7695099434	BCA	1	1003	310
12	Suhana	Verma	8904567688	B.Sc	3	1008	205
16	Payal	Tiwari	8907600987	B.tech	1	1010	16
28	Prem	Chopra	7654320987	BBA	2	1006	2020

## 9). Display Information of students alphabetically by their year of study.

The screenshot shows a Windows desktop environment. In the center is a Python 3.9.0 Shell window titled "Python 3.9.0 Shell". The code in the shell window is as follows:

```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                             password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
sql = "Select * from student order by year_of_study;"
mycursor.execute(sql)

result = mycursor.fetchall()
for r in result:
    print(r)
```

The output of the code is displayed in the shell window, listing student names and their years of study. The taskbar at the bottom of the screen shows various application icons and the system clock.

## 10). Display all unique type of furnitures.

The screenshot shows a Windows desktop environment. In the center is a Python 3.9.0 Shell window titled "Python 3.9.0 Shell". The code in the shell window is as follows:

```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                             password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
sql = "Select furniture_type from furniture group by furniture_type";
mycursor.execute(sql)
result = mycursor.fetchall()

for r in result:
    print(r)
```

The taskbar at the bottom of the screen shows various application icons and the system clock.

## 11). Display furniture of hostel with id = 1006.

The screenshot shows a Windows desktop environment with two application windows open:

- Python 3.9.0 Shell**: A terminal window showing Python code execution. The code is a script named `HostelManagementSystem_db.py` that connects to a MySQL database and deletes a record from the `furniture` table where `hostel_id = 1006`. The output shows the command was successful with 0 records affected.
- MySQL Workbench**: A database management tool window titled "HMS tables queries". It contains a query editor with the following SQL code:

```
1 use hms
2 select * from furniture;
```

The results grid below shows a list of furniture items, including their `Furniture_Id`, `Room_Id`, `Hostel_Id`, and `Furniture_Type`. The data is as follows:

Furniture_Id	Room_Id	Hostel_Id	Furniture_Type
29012	310	1003	Bed
30005	16	1010	Bed
38765	104	1004	Study Table
48453	205	1008	study Table
50552	133	1001	Chair
65438	205	1009	Chair

## 12). Find count of no. of different courses in a student table.

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with icons for Start, Search, Task View, File Explorer, Edge browser, Mail, File Explorer, and Python. The system tray shows the date and time as 02/11 12/15/2020. In the center, there are two windows: a code editor titled "HostelManagementSystem\_db.py" and a Python Shell window titled "Python 3.9.0 Shell".

**Code Editor Content:**

```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
sql = "Select count(Distinct courses) from student";
mycursor.execute(sql)
result = mycursor.fetchall()

for r in result:
    print(r)
```

**Python Shell Content:**

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
(4,)
>>>
```

## 13). Find max and min no. of students in a hostel

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with icons for Start, Search, Task View, File Explorer, Edge browser, Mail, File Explorer, and Python. The system tray shows the date and time as 02/18 12/15/2020. In the center, there are two windows: a code editor titled "HostelManagementSystem\_db.py" and a Python Shell window titled "Python 3.9.0 Shell".

**Code Editor Content:**

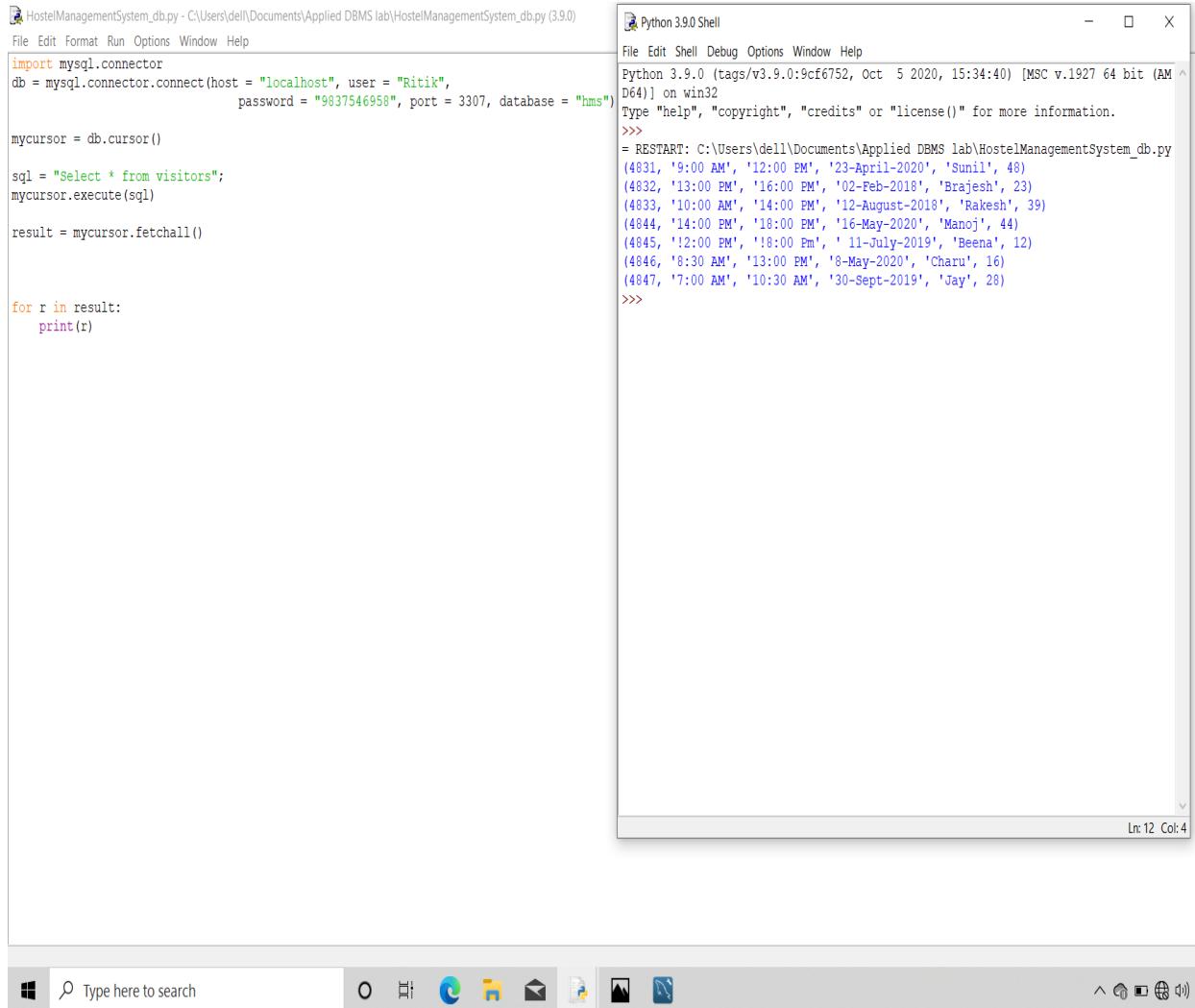
```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
sql = "Select max(no_of_students), min(no_of_Students) from hostel";
mycursor.execute(sql)
result = mycursor.fetchall()

for r in result:
    print(r)
```

**Python Shell Content:**

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
(800, 300)
>>>
```

## 14). Display all the records stored in visitors table.



```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
sql = "Select * from visitors";
mycursor.execute(sql)
result = mycursor.fetchall()
for r in result:
    print(r)
```

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
(4831, '9:00 AM', '12:00 PM', '23-April-2020', 'Sunil', 48)
(4832, '13:00 PM', '16:00 PM', '02-Feb-2018', 'Brajesh', 23)
(4833, '10:00 AM', '14:00 PM', '12-August-2018', 'Rakesh', 39)
(4844, '14:00 PM', '18:00 PM', '16-May-2020', 'Manoj', 44)
(4845, '12:00 PM', '18:00 PM', '11-July-2019', 'Beena', 12)
(4846, '8:30 AM', '13:00 PM', '8-May-2020', 'Charu', 16)
(4847, '7:00 AM', '10:30 AM', '30-Sept-2019', 'Jay', 28)
>>>
```



## 15). Truncate table furniture.

The screenshot shows a Windows desktop environment with two application windows open. The top window is the 'Python 3.9.0 Shell' showing a command-line interface for Python 3.9.0. The bottom window is 'MySQL Workbench' showing a database management interface for the 'hms' schema.

**Python 3.9.0 Shell:**

```
HostelManagementSystem_db.py - C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py (3.9.0)
File Edit Format Run Options Window Help
import mysql.connector
db = mysql.connector.connect(host = "localhost", user = "Ritik",
                                password = "9837546958", port = 3307, database = "hms")
mycursor = db.cursor()
sql = "Truncate table Furniture";
mycursor.execute(sql)
db.commit()

Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Documents\Applied DBMS lab\HostelManagementSystem_db.py
>>> |
```

**MySQL Workbench:**

The MySQL Workbench interface shows the 'hms' schema in the Navigator pane. In the central query editor, the following SQL query is visible:

```
1 ● use hms
2
3 ☐ Select * from furniture
4
5
6
```

The screenshot shows the MySQL Workbench interface after the 'Truncate table Furniture' command has been executed. The 'Result Grid' pane displays the contents of the 'furniture' table, which is currently empty.

Furniture_id	Room_id	Hostel_id	Furniture_type
NULL	NULL	NULL	NULL

## 16). Delete table furniture.

The screenshot shows a Windows desktop environment with several open windows:

- Python 3.9.0 Shell**: A terminal window showing the output of a script that drops a table named "Furniture". The script connects to a MySQL database named "hms" and executes the SQL command "Drop table Furniture". The response from the shell indicates the table was successfully dropped.
- MySQL Workbench**: A database management tool window. The "Navigator" pane shows the "hms" schema with tables: administrator, hostel, room, student, visitors. The "HMS tables queries" pane contains the following SQL session:

```
1 use hms
2
3 show tables
```

The results grid below shows the same five tables listed.
- Taskbar**: Shows the system tray with icons for battery, signal, and volume, and a clock displaying "02:41 12/15/2020".

- **CONCLUSION**

As this is the end of a project of case study – Hostel Management System . I made this conclusion to reiterate my points regarding this project. As of all I gathered some information related to this topic. As the output of this topic report, now I am capable to doing such projects which last I think that it was difficult. The support and guidance of my teacher Dr.Neeraj Gupta Sir helped me a lot.

As we that in todays scenario database of every system is necessary and take care that there would be consistency and no- data redundancy all over my project. In brief, this project is created successfully with no errors.



