Hostel Management

{ Hostelio-Manageoma}

Date : 28 - 04-2022

Members	Our TA
Ankur Kumar Shukla [202051029]	Haripriya
Anisha Katiyar　　[202051027]	Goswami
Jayant Asudhani　　[202052319]	Ma'am
Abhishek Singh　　[202051006]	
Garv Chouhan　　[202051070]	

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them. We are highly indebted to Prof. Novarun Deb for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express my gratitude towards my parents & member of IIITV for their kind co-operation and encouragement which help me in completion of this project.

We would like to express my special gratitude and thanks to TA Haripriya Gunjanlalgi Goswami for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

******************   ******************

Export the database from the database management system. Combine it with your source code file. Write down the steps to run your project in word file and submit all three components in a zip file. It should be submitted by 29 April 2021.

IDEA

Main aim of the project was to develop a full fledged and robust database system to track the details regarding students and their hostel stay along with tracking and storing the normal in and out movement of students and maintaining a database that can store the transactions of students related to the hostel payments and other activities common between hostel and students that can be captured in the database which can capture the miniworld.we tried to develop the database such that it can handle complex SQL queries and produce the result with accurate data corresponding to the queries.

Exported data base

```
-- MySQL dump 10.13  Distrib 8.0.28, for Win64 (x86_64)
--
-- Host: localhost    Database: hostelio_manageoma
-- -----------------------------------------------------
-- Server version     8.0.28


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
```

```
/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;

/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;

/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;


--

-- Table structure for table `flat`

--


DROP TABLE IF EXISTS `flat`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `flat` (
  `FLAT_NO` int DEFAULT NULL,
  `BUILD_NAME` varchar(10) DEFAULT NULL,
  `ELECTRIC_CONSUME` int DEFAULT NULL,
  `FINE` int DEFAULT NULL,
  `F_STATUS` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;


--

-- Dumping data for table `flat`

--


LOCK TABLES `flat` WRITE;

/*!40000 ALTER TABLE `flat` DISABLE KEYS */;
```

```sql
INSERT INTO `flat` VALUES (402,'F',NULL,NULL,'1/5
Full'),(203,'A',123,12340,'4/7 FULL');

/*!40000 ALTER TABLE `flat` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Table structure for table `hostel`

--


DROP TABLE IF EXISTS `hostel`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `hostel` (

  `H_Name` char(10) DEFAULT NULL,

  `No_of_Room` int DEFAULT NULL,

  `No_of_Students` int DEFAULT NULL,

  `F_STATUS` varchar(10) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;


--

-- Dumping data for table `hostel`

--


LOCK TABLES `hostel` WRITE;

/*!40000 ALTER TABLE `hostel` DISABLE KEYS */;

INSERT INTO `hostel` VALUES ('Raj Labdhi',102,650,'Not Full');

/*!40000 ALTER TABLE `hostel` ENABLE KEYS */;

UNLOCK TABLES;
```

```
--
-- Table structure for table `hostel_staff`
--

DROP TABLE IF EXISTS `hostel_staff`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `hostel_staff` (
  `First_name` varchar(25) NOT NULL,
  `Second_name` varchar(25) DEFAULT NULL,
  `Flat_Number` int NOT NULL,
  `Attendence` int DEFAULT NULL,
  `Role` varchar(25) NOT NULL,
  `Contact_Number` bigint DEFAULT NULL,
  `Staff_ID` varchar(15) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `hostel_staff`
--

LOCK TABLES `hostel_staff` WRITE;
/*!40000 ALTER TABLE `hostel_staff` DISABLE KEYS */;
INSERT INTO `hostel_staff` VALUES
('Vijay','Singh',101,30,'Manager',9856245686,'AD605'),('Sanjay','Khan'
,102,29,'Manager',9856226208,'AD705'),('Akshat','Seth',303,30,'F-Block
Warden',7392999208,'AD805'),('Abhijit','Patel',103,25,'Manager',992023
5686,'AD905'),('Aditya',NULL,102,5,'Ambulance
Driver',9856925686,'AD205'),('Jennet',NULL,403,30,'Girls
Warden',9856206586,'DD105');
```

```
/*!40000 ALTER TABLE `hostel_staff` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Table structure for table `student`

--


DROP TABLE IF EXISTS `student`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `student` (

  `RECORD_NO` int DEFAULT NULL,

  `STUDENT_ID` varchar(15) NOT NULL,

  `FIRST_NAME` varchar(25) NOT NULL,

  `LAST_NAME` varchar(25) DEFAULT NULL,

  `YEAR_OF_STUDY` int DEFAULT NULL,

  `PROGRAM` varchar(50) DEFAULT NULL,

  `FLAT_NO` int DEFAULT NULL,

  `CONTACT_NO` bigint DEFAULT NULL,

  `GUARDIAN_NO` bigint DEFAULT NULL,

  PRIMARY KEY (`STUDENT_ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;


--

-- Dumping data for table `student`

--


LOCK TABLES `student` WRITE;
```

```
/*!40000 ALTER TABLE `student` DISABLE KEYS */;

INSERT INTO `student` VALUES
(9,'201851071','Dev','Kumar',4,'CSE',203,8235997171,6752737261),(10,'2
01952009','Shruti','Gupta',3,'IT',303,8233627381,6780092101),(8,'20195
2211','Karan','Singh',3,'IT',202,8235262571,6781999101),(5,'202051001'
,'Manan','Shukla',2,'CSE',501,9012123666,8200188201),(2,'202051006','A
bhishek','Singh',2,'CSE',704,7899292200,8673299101),(1,'202051027','An
isha','Katiyar',3,'CSE',402,8104199999,9999900000),(17,'202051032','Br
ijesh','Agal',2,'CSE',702,9994654800,9551158985),(14,'202051055','Isha
n','Pandey',2,'CSE',203,8104195569,9213579620),(15,'202051056','Aman',
'Kothari',2,'CSE',501,9569874983,9945579852),(13,'202051063','Akshat',
'Khandelwal',2,'CSE',704,9945987090,9578412562),(1,'202051070','Ankur'
,'Shukla',2,'CSE',402,7823992200,8299299101),(18,'202051081','Tirth','
Joshi',2,'CSE',601,9991811050,9595959820),(19,'202051082','Ankur','Shu
kla',2,'CSE',402,7823992200,8299299101),(6,'202051089','Garv','Chauhan
',2,'CSE',603,9101012731,7817811101),(16,'202051090','Tejas','Joshi',2
,'CSE',604,9959873260,994565239),(4,'202051112','Anisha','Katiyar',2,'
CSE',402,7823992200,8299299101),(12,'202051232','Naman','Singh',4,'CSE
',204,8233435571,6703929101),(3,'202052303','Jayant','Asudhani',2,'IT'
,401,39881937811,9910176321),(7,'202151067','Rahul','Shukla',1,'CSE',3
02,7823765400,8921299101),(11,'202151123','Dhruv','Mishra',3,'CSE',701
,8239937820,6389343401);

/*!40000 ALTER TABLE `student` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Table structure for table `student_in_out`

--


DROP TABLE IF EXISTS `student_in_out`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `student_in_out` (
  `Record_No` bigint NOT NULL AUTO_INCREMENT,
  `STUDENT_ID` varchar(15) NOT NULL,
```

```sql
  `STD_DEPARTURE` timestamp NULL DEFAULT NULL,

  `STD_ARRIVAL` timestamp NULL DEFAULT NULL,

  `REASON` varchar(255) DEFAULT NULL,

  PRIMARY KEY (`Record_No`)

) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;


--

-- Dumping data for table `student_in_out`

--


LOCK TABLES `student_in_out` WRITE;

/*!40000 ALTER TABLE `student_in_out` DISABLE KEYS */;

INSERT INTO `student_in_out` VALUES (1,'202051027','2022-02-26
18:30:00','2022-03-01 18:30:00',' Going Home
'),(2,'202051029','2022-03-01 18:30:00','2022-03-01 18:30:00',' Temple
Visit '),(3,'20205020','2022-03-01 18:30:00',NULL,'WLK');

/*!40000 ALTER TABLE `student_in_out` ENABLE KEYS */;

UNLOCK TABLES;


--

-- Table structure for table `transact`

--


DROP TABLE IF EXISTS `transact`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `transact` (

  `TRANS_ID` int DEFAULT NULL,
```

```
  `STUDENT_ID` varchar(15) NOT NULL,

  `HOSTEL_AMT` int DEFAULT NULL,

  `MESS_AMT` int DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;


--

-- Dumping data for table `transact`

--


LOCK TABLES `transact` WRITE;

/*!40000 ALTER TABLE `transact` DISABLE KEYS */;

INSERT INTO `transact` VALUES
(12345,'202051006',30000,15000),(12346,'202051111',32000,14000),(12347
,'202051112',28000,15000),(14416,'202051001',30000,12000),(35622,'2020
52189',15000,NULL),(25628,'20205111',20000,8000),(18226,'202051091',NU
LL,12000),(NULL,'202052089',NULL,NULL),(25118,'202052051',20000,8000),
(10006,'202051001',10000,6000),(78822,'202052100',32000,NULL),(65662,'
202052041',18000,5000),(16616,'202051005',40000,12000);

/*!40000 ALTER TABLE `transact` ENABLE KEYS */;

UNLOCK TABLES;

/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;


/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;

/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

```
-- Dump completed on 2022-04-28 22:28:50
```

Source code

-- QUERY TO CREATE AND USE THE HOSTEL DATABASE :-

CREATE DATABASE HOSTELIO_MANAGEOMA;

USE HOSTELIO_MANAGEOMA;

SHOW DATABASES;

-- QUERIES FOR THE CREATION OF TABLES:-

-- (1) Table STUDENT

CREATE TABLE STUDENT(

RECORD_NO INT(5),

STUDENT_ID VARCHAR(15) NOT NULL,

FIRST_NAME VARCHAR(25) NOT NULL,

```
        LAST_NAME VARCHAR(25),

        YEAR_OF_STUDY INT(8),

        PROGRAM VARCHAR(50),

        FLAT_NO INT(4),

        CONTACT_NO BIGINT(13),          -- BIGINT takes (-2^-63 TO 2^63)

    GUARDIAN_NO BIGINT(13),

            PRIMARY KEY(STUDENT_ID));


DESCRIBE STUDENT;


-- (2) Table STUDENT_IN_OUT
 CREATE TABLE STUDENT_IN_OUT(

        RECORD_NO BIGINT(10),

    STUDENT_ID VARCHAR(15) NOT NULL,

        STD_DEPARTURE TIMESTAMP,

        STD_ARRIVAL TIMESTAMP,

        REASON varchar(255),

            PRIMARY KEY(RECORD_NO));


DESCRIBE STUDENT_IN_OUT;


-- (3) Table TRANSACT
CREATE TABLE TRANSACT(

        TRANS_ID INT(5),

        STUDENT_ID VARCHAR(15) NOT NULL,

        HOSTEL_AMT INT(20),

        MESS_AMT INT(20),

        REASONS_IF_ANY varchar(255));
```

```
DESCRIBE TRANSACT;


-- (4) Table FLAT
CREATE TABLE FLAT(
        FLAT_NO INT(50),
        BUILD_NAME varchar(10),
        ELECTRIC_CONSUME INT(10),
        FINE INT(10),
        F_STATUS VARCHAR(10));


DESCRIBE FLAT;


-- (5) Table HOSTEL STAFF
CREATE TABLE Hostel_staff (
        First_name VARCHAR(25) NOT NULL,
        Second_name VARCHAR(25),
        Flat_Number INT(8) NOT NULL,
        Attendence INT(8),
        Role VARCHAR(25) NOT NULL,
        Contact_Number BIGINT(13),
        Staff_ID VARCHAR(15) NOT NULL);


 DESCRIBE Hostel_Staff;


-- (6) Table HOSTEL
CREATE TABLE hostel(
        H_Name char(10),
        No_of_Room INT(10),
        No_of_Students INT(10),
```

```
        F_STATUS VARCHAR(10));
DESCRIBE Hostel;
show tables;



  insert into STUDENT
values(1,202051027,"Anisha","Katiyar",2,"CSE",402,8104199999,9999900000);
  insert into STUDENT
values(2,202051029,"Ankur","Shukla",NULL,"CSE",703,9999999900,22222222);


  SELECT * FROM STUDENT;


  insert into STUDENT_IN_OUT values(1,202051027,"2022-02-27","2022-03-02"," Going
Home ");
  insert into STUDENT_IN_OUT values(2,202051029,"2022-03-02","2022-03-02"," Temple
Visit ");
  insert into FLAT(flat_no,build_name,f_status) values (402,"F","1/5 Full");


  SELECT * FROM STUDENT_IN_OUT;
  SELECT * FROM FLAT;


  ALTER TABLE TRANSACT DROP REASONS_IF_ANY;   -- unnecessary field in Transact
Table
  SELECT * FROM TRANSACT;


  ALTER TABLE STUDENT_IN_OUT MODIFY Record_No BIGINT(10) NOT NULL
AUTO_INCREMENT;
  SELECT * FROM STUDENT_IN_OUT;            -- Feature of Auto Incrementation added


  INSERT INTO FLAT(FLAT_NO,BUILD_NAME,ELECTRIC_CONSUME,FINE,F_STATUS)
VALUES(203,"A",123,12340,"4/7 FULL");
```

```sql
insert into student_in_out(STUDENT_ID,STD_DEPARTURE,REASON)
values(20205020,"2022-03-2","WLK");


Insert into transact values
        ('12345','202051006',30000,15000),
    ('12346','202051111',32000,14000),
    ('12347','202051112',28000,15000);


DELETE FROM TRANSACT WHERE STUDENT_ID='202051006';

SELECT * FROM TRANSACT;


DELETE FROM STUDENT_IN_OUT WHERE RECORD_NO=2;

SELECT * FROM STUDENT_IN_OUT;


UPDATE FLAT SET FLAT_NO=303 WHERE FLAT_NO=203 ;

SELECT * FROM FLAT;


UPDATE STUDENT SET YEAR_OF_STUDY=3 WHERE STUDENT_ID=202051029 ;

SELECT * FROM STUDENT;


SHOW FULL TABLES;


insert into FLAT values (604,"A",72,NULL,"3/7");

INSERT INTO FLAT(FLAT_NO,BUILD_NAME,ELECTRIC_CONSUME,FINE,F_STATUS)
VALUES(702,"C",57,2095,"1/5");

select * from Flat;


UPDATE FLAT SET F_STATUS="0/5 VACANT" WHERE F_STATUS=" 0/5";

SELECT * FROM FLAT;
```

```
insert into FLAT values (601,"A",107,NULL,"0/7 VACANT");



insert into STUDENT
values(1,202051027,"Anisha","Katiyar",2,"CSE",402,8104199999,9999900000);
insert into STUDENT
values(2,202051029,"Ankur","Shukla",2,"CSE",303,9998899900,9945621457);
insert into STUDENT
values(3,202051025,"Foram","Patel",2,"CSE",402,8234199999,9999562170);
insert into STUDENT
values(4,202051030,"Jayant","Asudhani",2,"IT",801,8554165321,9994598700);
insert into STUDENT
values(5,202051031,"Garv","Chauhan",2,"CSE",303,9945939900,9548587026);
insert into STUDENT
values(6,202051036,"Abhishek","Singh",2,"CSE",101,8554195699,9945987450);
insert into STUDENT
values(7,202051063,"Akshat","Khandelwal,2,"CSE",704,9945987090,9578412562);
insert into STUDENT
values(8,202051055,"Ishan","Pandey",2,"CSE",203,8104195569,9213579620);
insert into STUDENT
values(9,202051056,"Aman","Kothari",2,"CSE",501,9569874983,9945579852);
insert into STUDENT
values(10,202051090,"Tejas","Joshi",2,"CSE",604,9959873260,994565239);
insert into STUDENT
values(11,202051032,"Brijesh","Agal",2,"CSE",702,9994654800,9551158985);
insert into STUDENT
values(12,202051081,"Tirth","Joshi",2,"CSE",601,9991811050,9595959820);
SELECT * FROM STUDENT;


UPDATE Student SET FLAT_NO=303 WHERE FLAT_NO=703;
SELECT * FROM STUDENT;
```

```
INSERT INTO Hostel_staff
values("Vijay","Singh","101",30,"Manager",9856245686,"AD605");


INSERT INTO Hostel_staff
values("Sanjay","Khan","102",29,"Manager",9856226208,"AD705");

INSERT INTO Hostel_staff values("Akshat","Seth","303",30,"F-Block
Warden",7392999208,"AD805");

INSERT INTO Hostel_staff
values("Abhijit","Patel","103",25,"Manager",9920235686,"AD905");

INSERT INTO Hostel_staff values("Aditya",NULL,"102",5,"Ambulance
Driver",9856925686,"AD205");

INSERT INTO Hostel_staff values("Jennet",NULL,"403",30,"Girls
Warden",9856206586,"DD105");


 Select * from hostel_staff;



insert into student_in_out(STUDENT_ID,STD_DEPARTURE,REASON)
values(202051070,"2022-03-02","WLK"),
(202051070,"2022-03-02","SHOP"),
(202051027,"2022-03-04","PARTY"),
(202051029,"2022-03-04","PARTY"),
(202051025,"2022-03-04","PARTY"),
(202051030,"2022-03-04","PARTY"),
(202051031,"2022-03-04","WLK"),
(202051036,"2022-03-05","COMP"),
(202051063,"2022-03-05","COMP"),
(202051055,"2022-03-05","COMP"),
(202051056,"2022-03-05","WLK"),
(202051083,"2022-03-05","WLK"),
(202051090,"2022-03-06","SHOP"),
```

```
(202051032,"2022-03-06","WLK"),

(202051081,"2022-03-06","PARTY");


UPDATE student_in_out SET STD_DEPARTURE ="2022-03-05, 22:22:22",STD_ARRIVAL
="2022-03-05, 23:00:17" WHERE record_no=14;

UPDATE student_in_out SET STD_DEPARTURE ="2022-03-05, 22:27:41",STD_ARRIVAL
="2022-03-05, 23:01:45" WHERE record_no=15;

UPDATE student_in_out SET STD_DEPARTURE ="2022-03-05, 19:19:33",STD_ARRIVAL
="2022-03-05, 20:41:03" WHERE record_no=17;

Select * from student_in_out;


Insert into TRANSACT values
 (14416,'202051027',30000,12000),
 (35622,'202051029',15000,NULL),
 (25628,'202051083',20000,8000),
 (18226,'202051091',NULL,12000),
 (NULL,'202051063',NULL,NULL),
 (25118,'202052051',20000,8000),
 (10006,'202051001',10000,6000),
 (78822,'202051090',32000,NULL),
 (65662,'202052041',18000,5000),
 (16616,'202051081',40000,12000);


alter table hostel drop f_status;
insert into hostel values("A",21,105);
insert into hostel values("B",19,95);
insert into hostel values("C",20,80);
insert into hostel values("D",21,70);
insert into hostel values("F",21,100);
select * from hostel;
```

```sql
-- [1] Join Operation ( Inner Join ) to depict attributes from 2 Individual Tables
 SELECT DISTINCT STUDENT_ID,FIRST_NAME,BUILD_NAME,STUDENT.FLAT_NO,F_STATUS
FROM STUDENT INNER JOIN FLAT ON STUDENT.FLAT_NO = FLAT.FLAT_NO;


-- [2] Cardinality Check
SELECT * from Student;
SELECT * from Student Group By Student_ID having count(Student_ID)>1;
SELECT * from Student_In_Out;
SELECT * from Student_In_Out having count(Student_ID)>1;


-- [3] Using Count to Print Duplicate Entries
 Select FLAT_NO from Flat Group By FLAT_NO Having Count(FLAT_NO)>1;


-- [4] With Clause for Indicating Flats with Fine higher than the average
 WITH temporaryTable(averageFine) as (SELECT avg(FINE) from FLAT)
        SELECT FLAT_NO,BUILD_NAME,FINE
        FROM FLAT, temporaryTable
        WHERE FINE > temporaryTable.averageFine;


-- [5] Like Command to display Vacant Flats
 SELECT Flat_No, BUILD_NAME
 FROM FLAT
 WHERE F_Status NOT LIKE '% FULL';




 Select * from student;
```

Select * from student_in_out;

Select * from flat;

Select * from hostel;

Select * from hostel_staff;

Select * from transact;

```
1 •   use hostelio_manageoma;
2 •   show tables;
3
4 •   describe student ;
5 •   describe  flat;
6 •   describe hostel_staff ;
7 •   describe student_in_out ;
8 •   describe transact ;
9
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:

Tables_in_hostelio_manageoma
flat
hostel
hostel_staff
student
student_in_out
transact

```
3
4 •    describe student ;
5 •    describe  flat;
6 •    describe hostel_staff ;
7 •    describe student_in_out ;
8 •    describe transact ;
9
```

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

Field	Type	Null	Key	Default	Extra
First_name	varchar(25)	NO		NULL	
Second_name	varchar(25)	YES		NULL	
Flat_Number	int	NO		NULL	
Attendence	int	YES		NULL	
Role	varchar(25)	NO		NULL	
Contact_Number	bigint	YES		NULL	
Staff_ID	varchar(15)	NO		NULL	

```
2 •    show tables;
3
4 •    describe student ;
5 •    describe  flat;
6 •    describe hostel_staff ;
7 •    describe student_in_out ;
8 •    describe transact ;
9
```

Set limit for number of rows
Workbench will automaticall

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

Field	Type	Null	Key	Default	Extra
Record_No	bigint	NO	PRI	NULL	auto_increment
STUDENT_ID	varchar(15)	NO		NULL	
STD_DEPARTURE	timestamp	YES		NULL	
STD_ARRIVAL	timestamp	YES		NULL	
REASON	varchar(255)	YES		NULL	

```
3
4 ●    describe student ;
5 ●    describe  flat;
6 ●    describe hostel_staff ;
7 ●    describe student_in_out ;
8 ●    describe transact ;
9
```

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

Field	Type	Null	Key	Default	Extra
TRANS_ID	int	YES		NULL	
STUDENT_ID	varchar(15)	NO		NULL	
HOSTEL_AMT	int	YES		NULL	
MESS_AMT	int	YES		NULL	

Result 11   Result 12   Result 13   Result 14   Result 15   Result 16 ✕

```
2 ●    show tables;

3

4 ●    describe student ;
5 ●    describe  flat;
6 ●    describe hostel_staff ;
7 ●    describe student_in_out ;
8 ●    describe transact ;
9
```

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

Field	Type	Null	Key	Default	Extra
FLAT_NO	int	YES		NULL	
BUILD_NAME	varchar(10)	YES		NULL	
ELECTRIC_CONSUME	int	YES		NULL	
FINE	int	YES		NULL	
F_STATUS	varchar(10)	YES		NULL	

Python script code

main.py

```python
from dbhelp1 import dbhelp1




# main work



help= dbhelp1()
# to show the table
help.show_table()
```

```
# working python command to interact with the database


# help.insert(1230,"sai","1234567890")

# help.insert(1234,"suresh","1234567891")

# help.insert(1235,"suresah","1234567892")

# help.fetchid(1234)

# help.fetchall()

# help.deleteuser(1234)

# help.fetchall()

# help.updateuser(1235,"ankur kumar shukla","979201")

# help.fetchall()

# help.general_querry_fetch()



# help.delete_row()



# help.update_row()
```

dbhelp1.py

```
# created and managed by : Ankur Kumar Shukla

# Date: 28-04-2022   (last update)

# objective : To interact with the database




from click import *
```

```python
import mysql.connector as connector




class dbhelp1:

    def __init__(self):


        # constructor for connection

        self.con =
connector.connect(host='localhost',port='3306',user='root',password='root'
,database='hostelio_manageoma', auth_plugin='mysql_native_password')

        print(" Welcome Sir , at your service \n CONNECTED FROM DATABSE");


    def createtable(self,tname2,attribute=2):


        querry= 'create table if not exists {}(userid int primary key,
username varchar(200))'.format(tname2)

        cur=self.con.cursor()

        cur.execute(querry)

        self.con.commit()

        print("table created")


    def createtable(self,tname3,attribute=3):


        querry= 'create table if not exists {}(userid int primary key,
username varchar(200),phone(12))'.format(tname3)

        cur=self.con.cursor()

        cur.execute(querry)

        self.con.commit()
```

```python
        print("table created")



    def show_table(self):

        querry="show tables"

        cur=self.con.cursor()

        # cur=confirm.cursor()

        cur.execute(querry)

        # self.con.commit()

        for row in cur:

            print(row)



        print("table shown")



    # show_table()




        # querry='create table if not exists tablename(userid int primary
key, username varchar(200),phone varchar(12))'



        # cur=self.con.cursor()

        # cur.execute(querry)

        # print("Created")

        #  insert the element in table
    def insert(self,userid,username,phone):

        querry='insert into tablename
values({},"{}","{}")'.format(userid,username,phone)
```

```python
        cur=self.con.cursor()

        cur.execute(querry)

        self.con.commit()

        print("Inserted")



    #print all content of table

    def fetchall(self):

        querry='select * from tablename'

        cur=self.con.cursor()

        cur.execute(querry)

        for row in cur:


            print("userid: ",row[0])

            print("username: ",row[1])

            print("phone: ",row[2])

print("=====================================================")


    # print particular id

    def fetchid(self,userid):

        querry='select * from tablename where userid= {}'.format(userid)

        cur = self.con.cursor()

        cur.execute(querry)

        for row in cur:

            print("userid: ",row[0])

            print("username: ",row[1])

            print("phone: ",row[2])
```

```python
    #  delete user with user id

    def deleteuser(self,userid):

        querry= 'delete from tablename where userid={}'.format(userid)

        cur= self.con.cursor()

        cur.execute(querry)

        self.con.commit()

        print("deleted")


    # update user with user id

    def updateuser(self,userid,newname,newphone):

        querry = 'update tablename set username="{}",phone="{}" where
userid="{}"'.format(newname,newphone,userid)

        cur = self.con.cursor()

        cur.execute(querry)

        self.con.commit()

        print("updated")


    def general_querry_fetch(self):
        # input querry from user

        querry = input("Enter your querry to fetch: ")

        cur = self.con.cursor()

        cur.execute(querry)

        data= cur.fetchall()

        for row in data:

            for i in range(len(row)):

                print(row[i])


        print("\n")
```

```python
        print("========data fetched========")

        print("querry executed")



    def delete_row(self):

        querry = input("Enter the querry  to  delete: ")

        cur = self.con.cursor()

        cur.execute(querry)

        # show the table


        self.con.commit()

        print("deleted")



    def update_row(self):

        querry = input("Enter the querry  to  update: ")

        cur = self.con.cursor()

        cur.execute(querry)

        # show the table


        self.con.commit()

        print("updated")
```

```python
# class godmode:

#     def __init__(self):

#         self.con =
connector.connect(host='localhost',port='3306',user='root',password='root'
,database='hostelio_manageoma', auth_plugin='mysql_native_password')

#         print(" Welcome Sir , at your service \n CONNECTED FROM
DATABSE");


#     def general_querry(self):

#         # input querry from user

#         querry = input("Enter your querry: ")

#         cur = self.con.cursor()

#         cur.execute(querry)

#         self.con.commit()

#         print("querry executed")
```
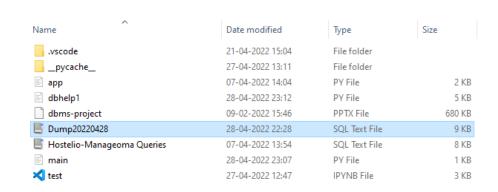
```
('flat',)
('hostel',)
('hostel_staff',)
('student',)
('student_in_out',)
('transact',)
table shown
Enter the querry  to  delete: delete from student where userid is 202051029
Traceback (most recent call last):
    self._cmysql.query(query,
_mysql_connector.MySQLInterfaceError: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '202051029' at line 1

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "c:\Users\Ankur Shukla\Desktop\dbms project\main.py", line 22, in <module>
    help.delete_row()
  File "c:\Users\Ankur Shukla\Desktop\dbms project\dbhelp1.py", line 113, in delete_row
    cur.execute(querry)
  File "C:\Users\Ankur Shukla\AppData\Local\Programs\Python\Python310\lib\site-packages\mysql\connector\cursor_cext.py", line 269, in execute
    result = self._cnx.cmd_query(stmt, raw=self._raw,
  File "C:\Users\Ankur Shukla\AppData\Local\Programs\Python\Python310\lib\site-packages\mysql\connector\connection_cext.py", line 528, in cmd_query
    raise errors.get_mysql_exception(exc.errno, msg=exc.msg,
mysql.connector.errors.ProgrammingError: 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '202051029'
 at line 1
PS C:\Users\Ankur Shukla\Desktop\dbms project> & "C:/Users/Ankur Shukla/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Ankur Shukla/Desktop/dbms project/main.py"
 Welcome Sir , at your service
 CONNECTED FROM DATABSE
('flat',)
('hostel',)
('hostel_staff',)
('student',)
('student_in_out',)
('transact',)
table shown
Enter the querry  to  delete: delete from student where userid = 202051029
Traceback (most recent call last):
  File "C:\Users\Ankur Shukla\AppData\Local\Programs\Python\Python310\lib\site-packages\mysql\connector\connection_cext.py", line 523, in cmd_query
    self._cmysql.query(query,
```

```
PS C:\Users\Ankur Shukla\Desktop\dbms project> & "C:/Users/Ankur Shukla/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Ankur Shukla/Desktop/dbms project/main.py"
 Welcome Sir , at your service
 CONNECTED FROM DATABSE
('flat',)
('hostel',)
('hostel_staff',)
('student',)
('student_in_out',)
('transact',)
table shown
Enter the querry  to  delete: delete from student where userid = 202051029
Traceback (most recent call last):
  File "C:\Users\Ankur Shukla\AppData\Local\Programs\Python\Python310\lib\site-packages\mysql\connector\connection_cext.py", line 523, in cmd_query
    self._cmysql.query(query,
_mysql_connector.MySQLInterfaceError: Unknown column 'userid' in 'where clause'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "c:\Users\Ankur Shukla\Desktop\dbms project\main.py", lin    help.delete_row()
  File "c:\Users\Ankur Shukla\Desktop\dbms project\dbhelp1.py",
line 113, in delete_row                                             cur.execute(querry)
  File "C:\Users\Ankur Shukla\AppData\Local\Programs\Python\Pyth
on310\lib\site-packages\mysql\connector\cursor_cext.py", line 269, in execute                                                  result = self._cnx.cmd_query(stmt, raw=self._raw,
  File "C:\Users\Ankur Shukla\AppData\Local\Programs\Python\Pyth
mysql.connector.errors.ProgrammingError: 1054 (42S22): Unknown column 'userid' in 'where clause'
PS C:\Users\Ankur Shukla\Desktop\dbms project> & "C:/Users/Ankur Shukla/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Ankur Shukla/Desktop/dbms project/main.py"
 Welcome Sir , at your service
 CONNECTED FROM DATABSE
('flat',)
('hostel',)
('hostel_staff',)
('student',)
('student_in_out',)
('transact',)
table shown
Enter the querry  to  delete: delete from student where student_id= 202051029
deleted
```

```
  Enter the querry  to  update: update student set year_of_study=3 where year_of_study=3
  updated
  PS C:\Users\Ankur Shukla\Desktop\dbms project> & "C:/Users/Ankur Shukla/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Ankur Shukla/Desktop/dbms
  PS C:\Users\Ankur Shukla\Desktop\dbms project> & "C:/Users/Ankur Shukla/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Ankur Shukla/Desktop/dbms
   Welcome Sir , at your service
   CONNECTED FROM DATABSE
  ('flat',)
  ('hostel',)
  ('hostel_staff',)
  ('student',)
  ('student_in_out',)
  ('transact',)
  table shown
  Enter the querry  to  update: update student set year_of_study=3 where year_of_study=2
  updated
  PS C:\Users\Ankur Shukla\Desktop\dbms project>
```

Steps to initialize

First of all we need to create database by running *.sql file in mysql workbench or shell

Name	Date modified	Type	Size
.vscode	21-04-2022 15:04	File folder	
__pycache__	27-04-2022 13:11	File folder	
app	07-04-2022 14:04	PY File	2 KB
dbhelp1	28-04-2022 23:12	PY File	5 KB
dbms-project	09-02-2022 15:46	PPTX File	680 KB
Dump20220428	28-04-2022 22:28	SQL Text File	9 KB
Hostelio-Manageoma Queries	07-04-2022 13:54	SQL Text File	8 KB
main	28-04-2022 23:07	PY File	1 KB
test	27-04-2022 12:47	IPYNB File	3 KB

Then we have to connect with your local database by altering the username and password

```
class dbhelp1:
    def __init__(self):

        # constructor for connection
        self.con = connector.connect(host='localhost',port='3306',user='root',password='root',database='hostelio_manageoma', auth_plugin='mysql_native_password')
        print(" Welcome Sir , at your service \n CONNECTED FROM DATABSE");

    def createtable(self,tname2,attribute=2):
```

Once this ! we are now ready to use all function (C R U D operation) on the database from python application . We just need to import dbhelp and all function can be use seamlessly like any other standard library

```
main.py > ...
  1    from dbhelp1 import dbhelp1
  2
  3
  4
  5    # main work
  6
  7
  8    help= dbhelp1()
  9    # to show the table
 10    help.show_table()
```

We had also added function which can perform general task , all complex operation as we do on workbench .

****

Hence we had successfully achieved the goal of our project .

** ****