DBMS - Mini Project
(HOTEL MANAGEMENT SYSTEM)

Submitted By:
Name Setti Durga Poojitha

SRN :PES2UG20CS458
Semester-V
Section -G

Submitted to:
Dr.Geeta D
Associate Professor

Table of Contents

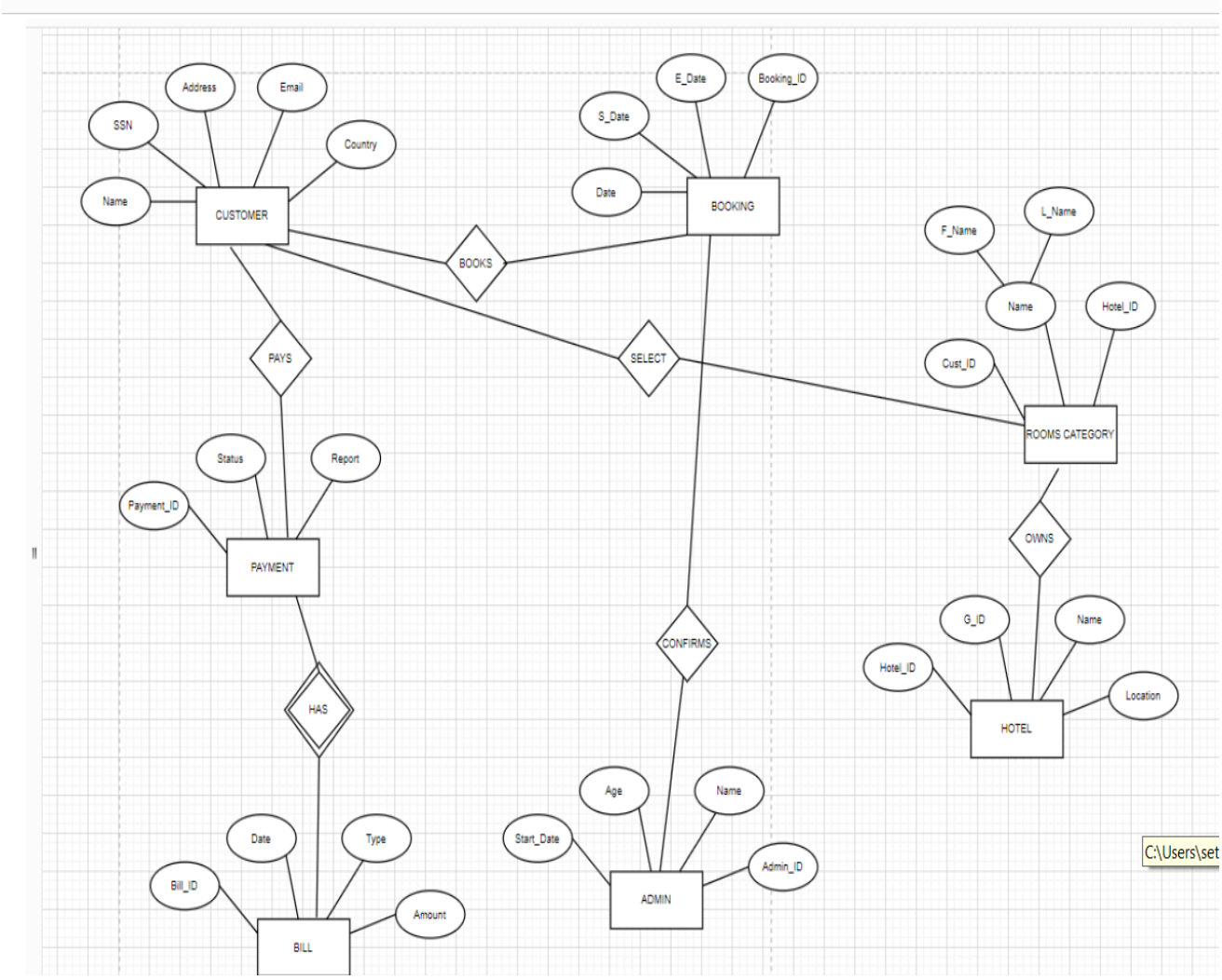
CONTENT	PAGE NUMBER
1.DESCRIPTION AND SCOPE OF THE PROJECT	
2.ER-DIAGRAM	
3.RELATIONAL SCHEMA	
4.JOIN OPERATIONS	
5.AGGREGATE FUNCTIONS	
6.SET OPERATIONS	
7.FUNCTIONS AND STORED PROCEDURES	
8.TRIGGERS AND CURSORS	
9.FRONT-END	
10.CONCLUSION	

Short Description and Scope of the Project

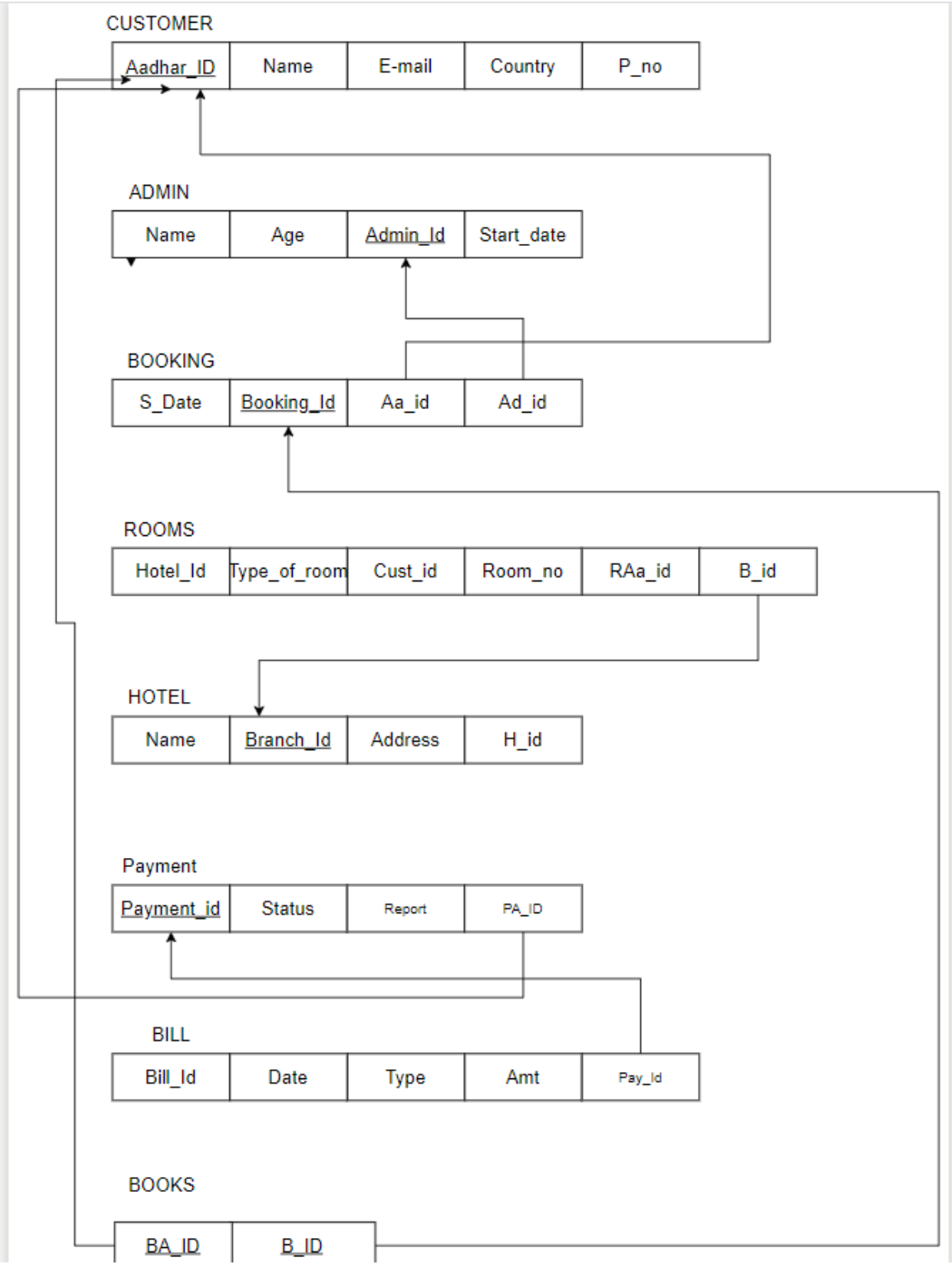
This project consists of the implementation of a hotel management system.
First it consists of the ER-diagram and the relation schema of the hotel management system.This is done to define the skeleton of the database system.
The relational schema defines the outline of the tables and the relationships among them.
Then DDL and DML statements were used to create the tables and to populate the database.
It consists the details of customers,the admin employees,the details of the hotel rooms,the information about the different branches of the hotel franchise.It also consists of details of the payment and bills including the status of the payment.All of this information is stored in the tables and the relationships among the tables have been established through a primary-foreign key constraints.
And a number of operations have been performed on the data using different sql queries like join,aggregate,set.
Functions,stored procedures ,triggers have also been implemented.

Tools Used:
Mysql workbench,streamlit.
Languages used:python(for front-end),sql

ER Diagram


Address
Email
SSN
Country
Name
CUSTOMER
E_Date
Booking_ID
S_Date
Date
BOOKING
BOOKS
PAYS
SELECT
F_Name
L_Name
Name
Hotel_ID
Cust_ID
ROOMS CATEGORY
Status
Report
Payment_ID
PAYMENT
OWNS
CONFIRMS
G_ID
Name
Hotel_ID
Location
HOTEL
HAS
Age
Name
Start_Date
Admin_ID
ADMIN
Date
Type
Bill_ID
Amount
BILL
C:\Users\set

Relational Schema



DDL statements - Building the database

Database creation

```
CREATE SCHEMA `hotel_db` ;
```

CREATION OF TABLES:

```
CREATE TABLE `hotel_db`.`customer` (
  `Aadhar_id` INT NOT NULL,
  `Name` VARCHAR(45) NOT NULL,
  `E-mail` VARCHAR(45) NOT NULL,
  `P_no` INT NULL,
  PRIMARY KEY (`Aadhar_id`));

CREATE TABLE `hotel_db`.`admin` (
  `Admin_Id` INT NOT NULL,
  `name` VARCHAR(45) NOT NULL,
  `Age` INT NOT NULL,
  `Start_date` DATE NOT NULL,
  PRIMARY KEY (`Admin_Id`));


CREATE TABLE `hotel_db`.`booking` (
  `Booking_Id` INT NOT NULL,
  `S_date` DATE NULL,
  `Aadhar_id` INT NULL,
  `Admin_Id` INT NULL,
  PRIMARY KEY (`Booking_Id`),
  INDEX `Aadhar_Id_idx` (`Aadhar_id` ASC) VISIBLE,
  INDEX `Admin_Id_idx` (`Admin_Id` ASC) VISIBLE,
  CONSTRAINT `Aadhar_Id`
    FOREIGN KEY (`Aadhar_id`)
    REFERENCES `hotel_db`.`customer` (`Aadhar_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Admin_Id`
    FOREIGN KEY (`Admin_Id`)
    REFERENCES `hotel_db`.`admin` (`Admin_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE `hotel_db`.`hotel` (
  `Branch_id` INT NOT NULL,
  `Name` VARCHAR(45) NOT NULL,
  `Address` VARCHAR(45) NULL,
  `H_id` INT NOT NULL,
```

```
  PRIMARY KEY (`Branch_id`));


CREATE TABLE `hotel_db`.`rooms` (
  `Room_no` INT NOT NULL,
  `Hotel_id` INT NOT NULL,
  `TOR` VARCHAR(45) NOT NULL,
  `Cus_id` INT NOT NULL,
  `Branch_id` INT NULL,
  PRIMARY KEY (`Room_no`),
  INDEX `Branch_id_idx` (`Branch_id` ASC) VISIBLE,
  CONSTRAINT `Branch_id`
    FOREIGN KEY (`Branch_id`)
    REFERENCES `hotel_db`.`hotel` (`Branch_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);


CREATE TABLE `hotel_db`.`payment` (
  `payment_id` INT NOT NULL,
  `Status` VARCHAR(45) NOT NULL,
  `Aadhar_id` INT NOT NULL,
  PRIMARY KEY (`payment_id`),
  INDEX `Aadhar_id_idx` (`Aadhar_id` ASC) VISIBLE,
  CONSTRAINT `Aadhar_id`
    FOREIGN KEY (`Aadhar_id`)
    REFERENCES `hotel_db`.`customer` (`Aadhar_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE `hotel_db`.`bill` (
  `bill_id` INT NOT NULL,
  `Amt` INT NOT NULL,
  `payment_id` INT NULL,
  PRIMARY KEY (`bill_id`),
  INDEX `payment_id_idx` (`payment_id` ASC) VISIBLE,
  CONSTRAINT `payment_id`
    FOREIGN KEY (`payment_id`)
    REFERENCES `hotel_db`.`payment` (`payment_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);


CREATE TABLE `hotel_db`.`books` (
  `BA_Id` INT NOT NULL,
  `B_id` INT NOT NULL,
```

```
  PRIMARY KEY (`BA_Id`, `B_id`),
  CONSTRAINT `BA_id`
    FOREIGN KEY (`BA_Id`)
    REFERENCES `hotel_db`.`customer` (`Aadhar_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `B_id`
    FOREIGN KEY (`BA_Id` , `B_id`)
    REFERENCES `hotel_db`.`booking` (`Aadhar_id` , `Booking_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

Population of the database:

```
INSERT INTO `hotel_db`.`customer` (`Aadhar_id`, `Name`, `E-
mail`, `P_no`) VALUES ('30645502', 'Joshua',
'joshua@gmail.com', '23456');
INSERT INTO `hotel_db`.`customer` (`Aadhar_id`, `Name`, `E-
mail`, `P_no`) VALUES ('30397150', 'Vernon',
'vernon@gmail.com', '98805');
INSERT INTO `hotel_db`.`customer` (`Aadhar_id`, `Name`, `E-
mail`, `P_no`) VALUES ('30918049', 'Arjun',
'arjun@gmail.com', '23456');
INSERT INTO `hotel_db`.`customer` (`Aadhar_id`, `Name`, `E-
mail`, `P_no`) VALUES ('40349981', 'Gyu', 'gyu@gmail.com',
'67895');


INSERT INTO `hotel_db`.`admin` (`Admin_Id`, `name`, `Age`,
`Start_date`) VALUES ('481', 'Jun', '35', '2002-05-22');
INSERT INTO `hotel_db`.`admin` (`Admin_Id`, `name`, `Age`,
`Start_date`) VALUES ('281', 'RM', '38', '2005-08-16');
INSERT INTO `hotel_db`.`admin` (`Admin_Id`, `name`, `Age`,
`Start_date`) VALUES ('381', 'Woozi', '40', '2002-02-24');
INSERT INTO `hotel_db`.`admin` (`Admin_Id`, `name`, `Age`,
`Start_date`) VALUES ('560', 'Harry', '34', '2003-06-12');


UPDATE `hotel_db`.`booking` SET `Admin_Id` = '481'
WHERE (`Booking_Id` = '35');
```

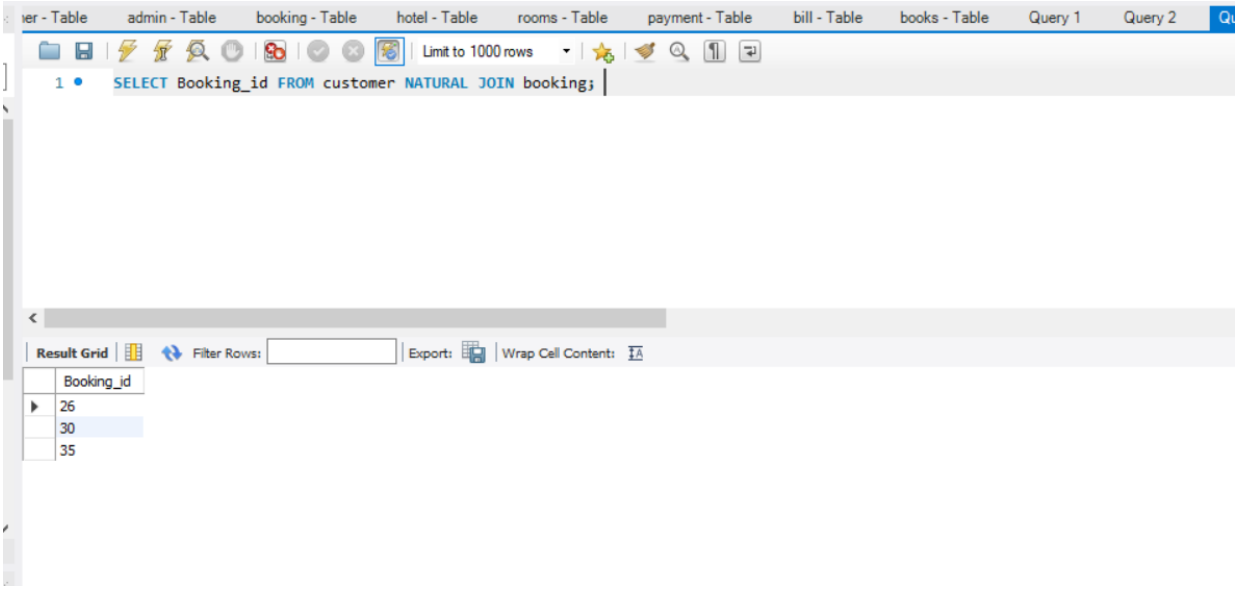
```
INSERT INTO `hotel_db`.`hotel` (`Branch_id`, `Name`,
`Address`, `H_id`) VALUES ('560036', 'Marina', 'Airportroad',
'20');
INSERT INTO `hotel_db`.`hotel` (`Branch_id`, `Name`,
`Address`, `H_id`) VALUES ('560043', 'Shoyo', 'Jayanagar',
'30');
INSERT INTO `hotel_db`.`hotel` (`Branch_id`, `Name`,
`Address`, `H_id`) VALUES ('560047', 'fukima',
'Malleshwaram', '35');
INSERT INTO `hotel_db`.`hotel` (`Branch_id`, `Name`,
`Address`, `H_id`) VALUES ('560053', 'Tanjiro', 'Electroniccity',
'46');

INSERT INTO `hotel_db`.`rooms` (`Room_no`, `Hotel_id`,
`TOR`, `Cus_id`, `Branch_id`) VALUES ('203', '100', 'single',
'67', '560036');
INSERT INTO `hotel_db`.`rooms` (`Room_no`, `Hotel_id`,
`TOR`, `Cus_id`, `Branch_id`) VALUES ('303', '105', 'double',
'56', '560043');
INSERT INTO `hotel_db`.`rooms` (`Room_no`, `Hotel_id`,
`TOR`, `Cus_id`, `Branch_id`) VALUES ('456', '222', 'triple',
'41', '560047');

INSERT INTO `hotel_db`.`payment` (`payment_id`, `Status`,
`PA_id`) VALUES ('24', 'paid', '30397150');
INSERT INTO `hotel_db`.`payment` (`payment_id`, `Status`,
`PA_id`) VALUES ('25', 'notpaid', '30918049');
INSERT INTO `hotel_db`.`payment` (`payment_id`, `Status`,
`PA_id`) VALUES ('26', 'pending', '40349981');


INSERT INTO `hotel_db`.`bill` (`bill_id`, `Amt`, `payment_id`)
VALUES ('2202', '2000', '24');
INSERT INTO `hotel_db`.`bill` (`bill_id`, `Amt`, `payment_id`)
VALUES ('2454', '3000', '26');
INSERT INTO `hotel_db`.`bill` (`bill_id`, `Amt`) VALUES
('2593', '2300');

INSERT INTO `hotel_db`.`books` (`BA_Id`, `B_id`) VALUES
('30645502', '30');
INSERT INTO `hotel_db`.`books` (`BA_Id`, `B_id`) VALUES
('30397150', '26');
```

Join Queries

Showcase at least 4 join queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1.Retrieve booking_ids of people who booked the hotel rooms

SELECT Booking_id FROM customer NATURAL JOIN booking;



2.Retrieve aadhar ids and names of people who have booked the rooms.

```
SELECT Aadhar_id,Name FROM customer NATURAL JOIN booking;
```

Result Grid

Aadhar_id	Name
30397150	Vernon
30645502	Joshua
40349981	Gyu

3.Retrieve name,age of the admins who confirmed the bookings

SELECT Name,age FROM admin NATURAL JOIN booking;

```
SELECT Name,age FROM admin NATURAL JOIN booking;
```

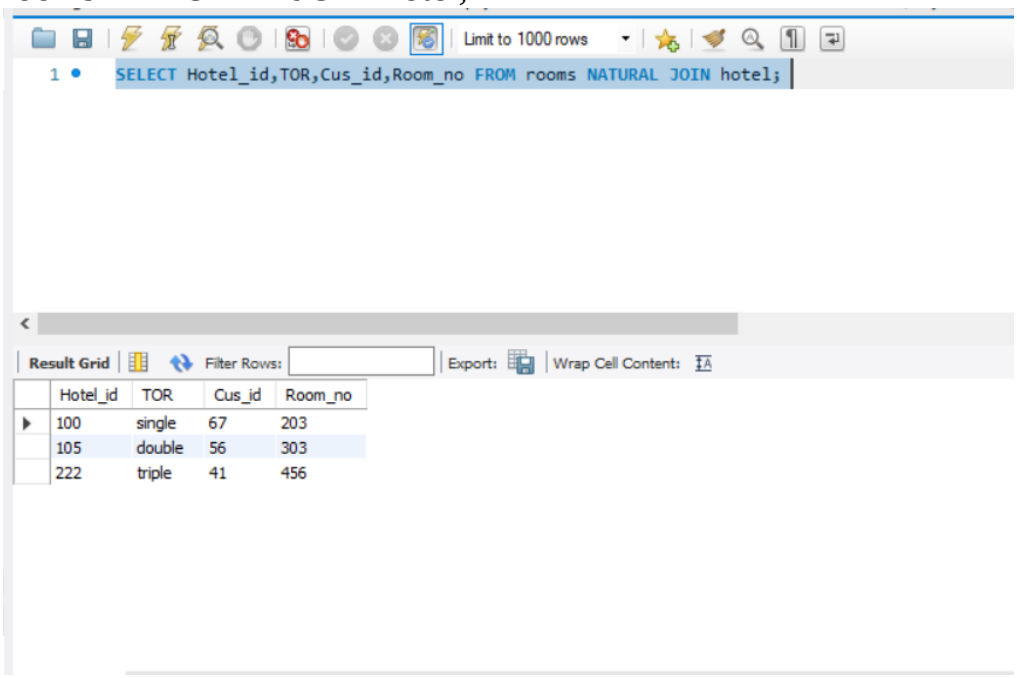
Result Grid

Name	age
RM	38
Woozi	40
Jun	35

4.Retrieve details of the room in a hotel situated in a particular branch

SELECT Hotel_id,TOR,Cus_id,Room_no FROM

rooms NATURAL JOIN hotel;



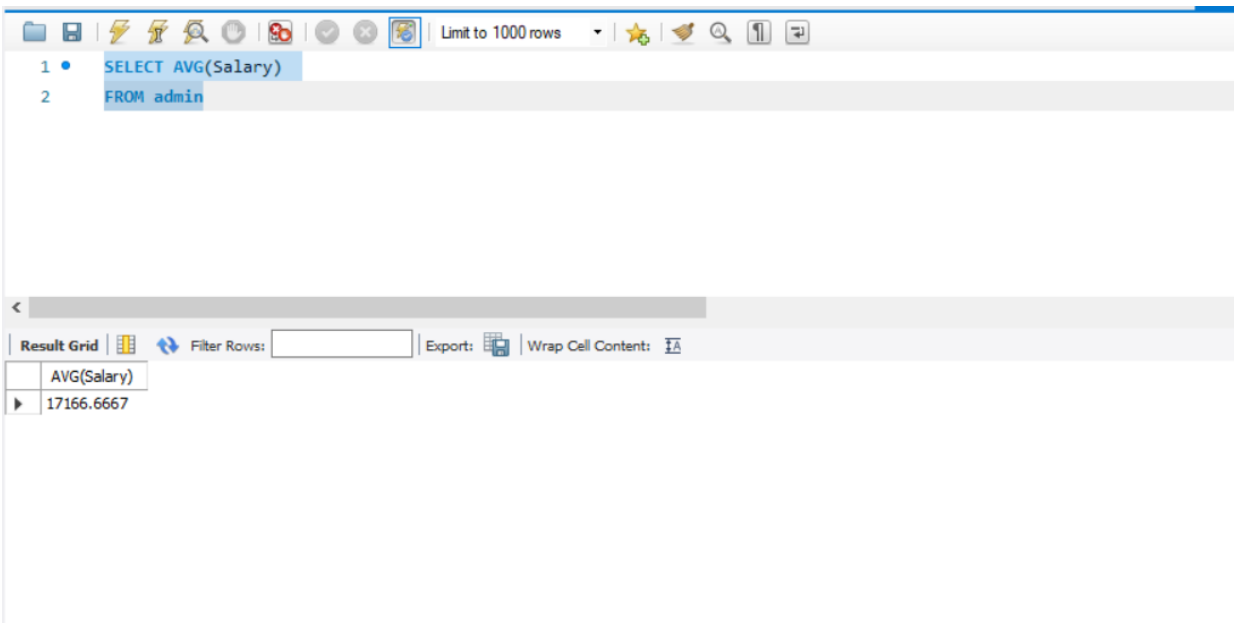
Hotel_id	TOR	Cus_id	Room_no
100	single	67	203
105	double	56	303
222	triple	41	456

Aggregate Functions

Showcase at least 4 Aggregate function queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1.Retrieve the average salary of the admin employees working.

SELECT AVG(Salary)
FROM admin



2.Display average working hours of the admin employees

SELECT AVG(Working_hours)
FROM admin

a customer - Table admin - Table booking - Table hotel - Table rooms - Table payment - Table bill - Table books - Table

Limit to 1000 rows

```
1 • SELECT AVG(Working_hours)
2   FROM admin
```

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

	AVG(Working_hours)
▶	6.8333

3.Display the minimum and maximum price of rooms

```
1 • SELECT MAX(price)
2   FROM rooms
```

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

	MAX(price)
▶	7000

```
SELECT MIN(price)
FROM rooms
```

MIN(price)
2000

4.find the average price of rooms

```
SELECT AVG(price)
FROM rooms
```

AVG(price)
4200.0000

Set Operations

Showcase at least 4 Set Operations queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1.Retrieve all the details of the customers whose room bookings have been confirmed.

SELECT * FROM customer
WHERE Aadhar_id IN (SELECT Aadhar_id
FROM booking);

```
SELECT * FROM customer
WHERE Aadhar_id IN (SELECT Aadhar_id FROM booking);
```

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

Aadhar_id	Name	E-mail	P_no
30397150	Vernon	vernon@gmail.com	98805
30645502	Joshua	joshua@gmail.com	23456
40349981	Gyu	gyu@gmail.com	67895
NULL	NULL	NULL	NULL

2.Display payment_id and status in the bill using union operation

```
SELECT payment_id,Status FROM payment
UNION
SELECT payment_id,status FROM bill;
```

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

payment_id	Status
24	paid
25	notpaid
26	pending

3.Retrieve the aadhar id's of customers who have booked the rooms

SELECT customer.Aadhar_id
FROM customer
WHERE customer.Aadhar_id IN (SELECT booking.Aadhar_id FROM booking);

```
SELECT customer.Aadhar_id
FROM customer
WHERE customer.Aadhar_id IN (SELECT booking.Aadhar_id FROM booking);
```

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

Aadhar_id
30397150
30645502
40349981
NULL

Functions and Procedures

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

Function:

Find out how many years has the admin employees been working at the hotel and display it as separate column attribute ‘years;. Use the Start_date values of the employee

```
DELIMITER //

CREATE FUNCTION no_of_years(date1 date)
RETURNS int DETERMINISTIC
BEGIN
 DECLARE date2 DATE;
  Select current_date()into date2;
  RETURN year(date2)-year(date1);
END

//

DELIMITER
Select name, age, Admin_id,
no_of_years(Start_date) as 'years' from admin;
```

```
DELIMITER //

CREATE FUNCTION no_of_years(date1 date) RETURNS int DETERMINISTIC
BEGIN
 DECLARE date2 DATE;
  Select current_date()into date2;
  RETURN year(date2)-year(date1);
END

//

DELIMITER
```

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

name	age	Admin_id	years
RM	38	281	17
Woozi	40	381	20
Jun	35	481	20
Harry	34	560	19
Ron	40	681	17
Percy	50	783	17

```
Select name, age, Admin_id, no_of_years(Start_date) as 'years' from admin;
```

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

name	age	Admin_id	years
RM	38	281	17
Woozi	40	381	20
Jun	35	481	20
Harry	34	560	19
Ron	40	681	17
Percy	50	783	17

2.Stored procedure

Create a stored procedure to retrieve the details of all the customers

```
DELIMITER //
CREATE PROCEDURE sp_GetCustomers()
BEGIN
   select Aadhar_id,Name,P_no from customer;
END //

DELIMITER ;
CALL sp_GetCustomers()
```

```
DELIMITER //
CREATE PROCEDURE sp_GetCustomers()
BEGIN
    select Aadhar_id,Name,P_no from customer;
END //

DELIMITER ;
CALL sp_GetCustomers()
```

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

Aadhar_id	Name	P_no
30397150	Vernon	98805
30645502	Joshua	23456
30918049	Arjun	23456
40349981	Gyu	67895

Triggers and Cursors

Create a Trigger and a Cursor. State the objective. Run and display the results.

A trigger is created which will display a message that a person should be more than 18 years old to book a room in the hotel.If the age of the person being inserted into the table is less than 18,an error message is displayed.

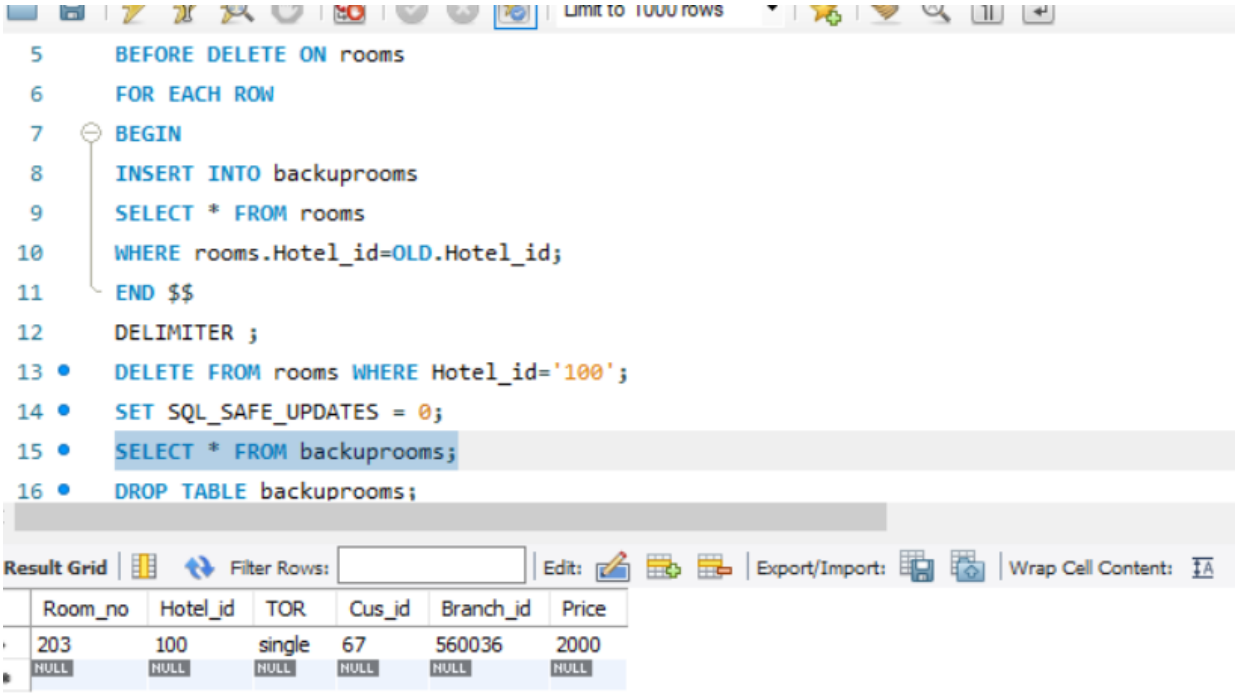
```
delimiter //
CREATE TRIGGER person_bi BEFORE INSERT
ON customer
FOR EACH ROW
IF NEW.Age < 18 THEN
SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'Person must be older than 18.';
END IF; //
delimiter ;
```

```
delimiter //
CREATE TRIGGER person_bi BEFORE INSERT
ON customer
FOR EACH ROW
IF NEW.Age < 18 THEN
SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'Person must be older than 18.';
END IF; //
delimiter ;
INSERT INTO `hotel_db`.`customer` VALUES ('20573145', 'Bhanu', 'bhanu@gmail.com', '23456','17');
```

123	19:14:37	SELECT * FROM customer LIMIT 0, 1000	5 row(s) returned
124	19:18:14	INSERT INTO 'hotel_db`.`customer` VALUES ('20573145', 'Bhanu', 'bhanu@gmail.com', '23456','17')	Error Code: 1644. Person must be older than 18.

Cursor:
A cursor for backing up data if a record is being deleted from the rooms table
Code:
CREATE TABLE backuprooms LIKE rooms;

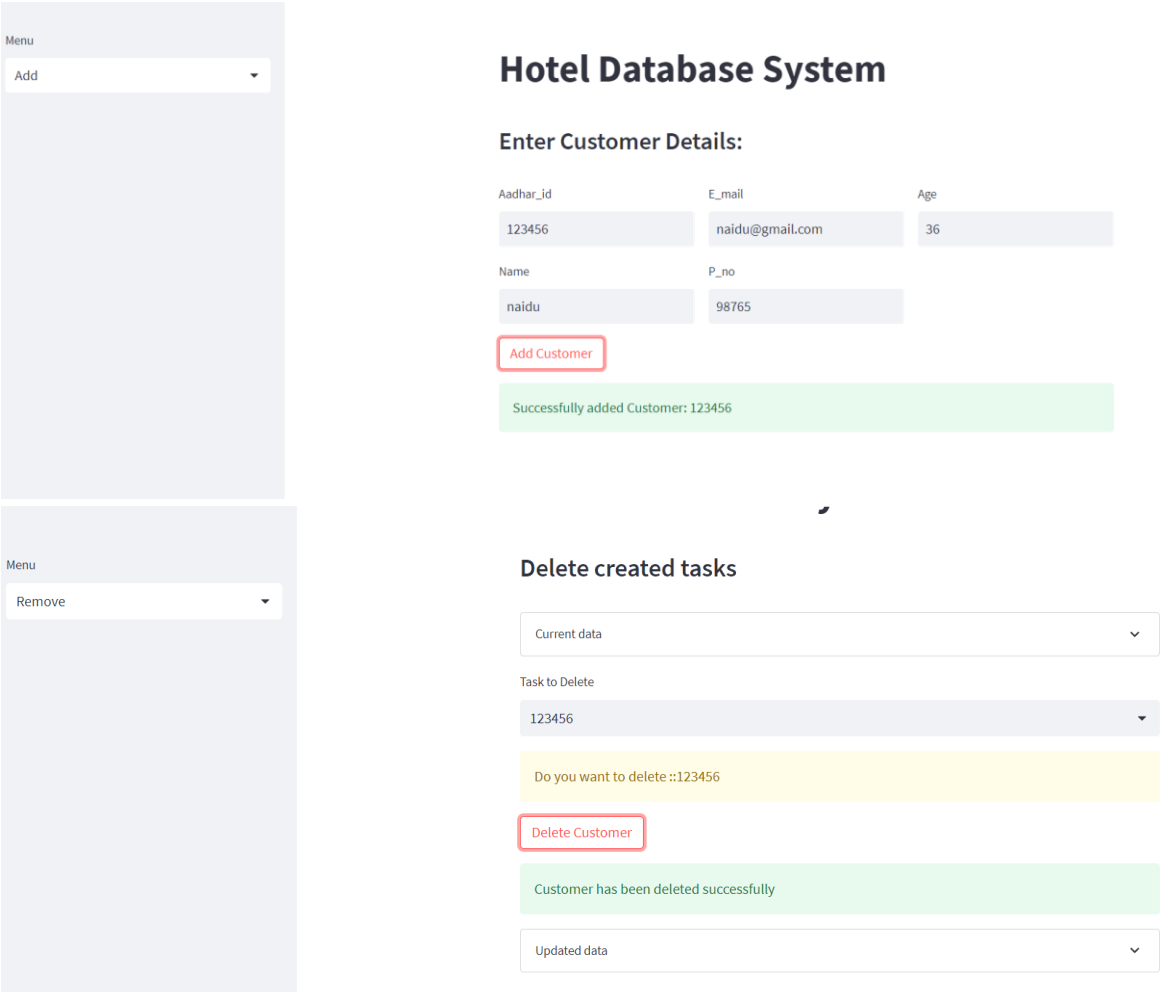
```
DELIMITER $$
CREATE TRIGGER backupr
BEFORE DELETE ON rooms
FOR EACH ROW
BEGIN
INSERT INTO backuprooms
SELECT * FROM rooms
WHERE rooms.Hotel_id=OLD.Hotel_id;
END $$
DELIMITER ;
```



Developing a Frontend

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table 2. There should be an window to accept and run any SQL statement and display the result



Menu

Remove

	Aadhar_id	Name	E_mail	P_no	Age
0	123456	naidu	naidu@gmail.com	98765	36
1	23456789	Poojitha	poo@gmail.com	45326	21
2	30397150	Vernon	vernon@gmail.com	98805	20
3	30645502	Joshua	joshua@gmail.com	23456	25
4	30918049	Arjun	arjun@gmail.com	23456	26
5	40349981	Gyu	gyu@gmail.com	67895	35

Task to Delete

123456

Do you want to delete ::123456

Delete Customer

Menu

Edit

Hotel Database System

Update created tasks

Current Customers

Customers to Edit

23456789

Aadhar_id

23476789

E_mail

poo@gmail.com

Age

21

Name

Poojitha

P_no

45326

Update Customer

Conclusion:

This project consists of the implementation of a hotel management system.
First it consists of the ER-diagram and the relation schema of the hotel management system.This is done to define the skeleton of the database system.
The relational schema defines the outline of the tables and the relationships among them.
Then DDL and DML statements were used to create the tables and to populate the database.
It consists the details of customers,the admin employees,the details of the hotel rooms,the information about the different branches of the hotel franchise.It also consists of details of the payment and bills including the status of the payment.All of this information is stored in the tables and the relationships among the tables have been established through a primary-foreign key constraints.
And a number of operations have been performed on the data using different sql queries like join,aggregate,set. Functions,stored procedures ,triggers have also been implemented.
A simple front-end using streamlit was also developed to perform simple CRUD operations .