DATABASE MANAGEMENT SYSTEM

MINI PROJECT

RESORT MANAGEMENT SYSTEM

Submitted By:

LAXMIKANT B GURAV

PES1UG20CS658

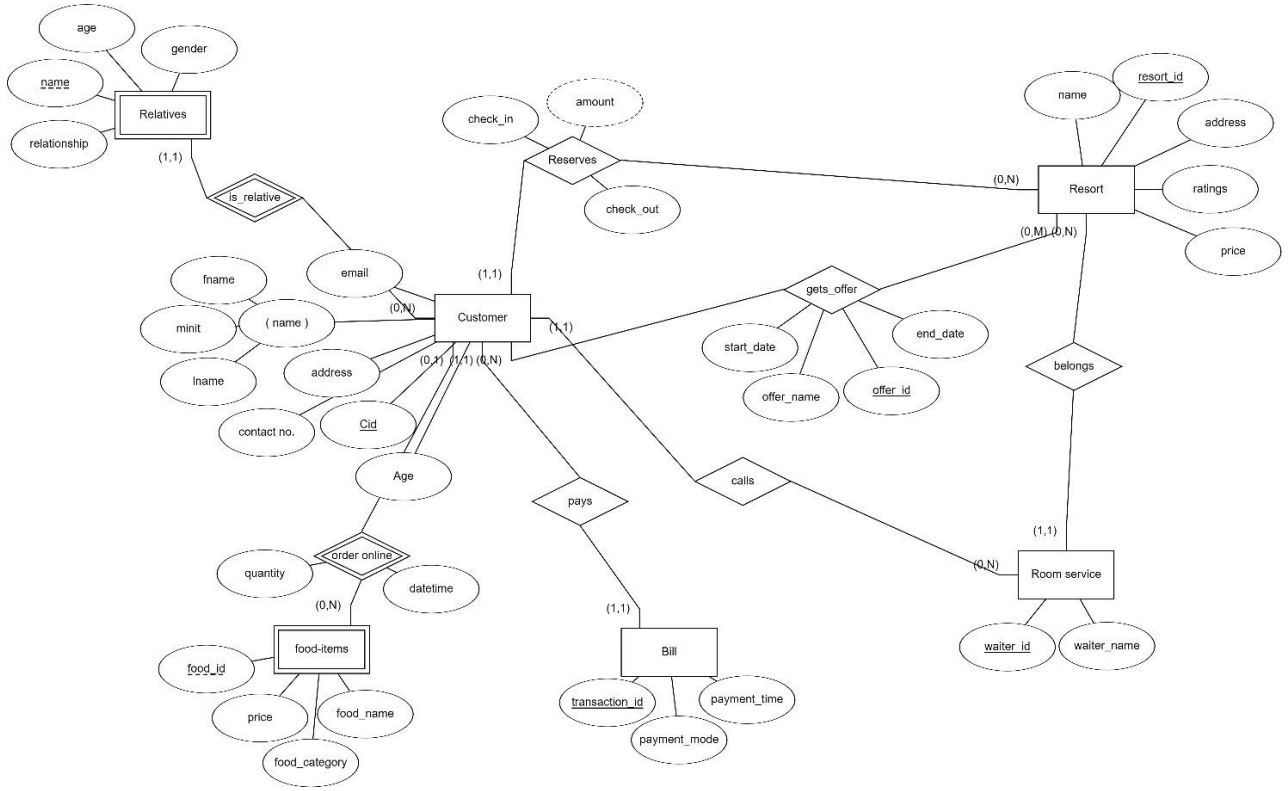
V Semester Section K

ABSTRACT

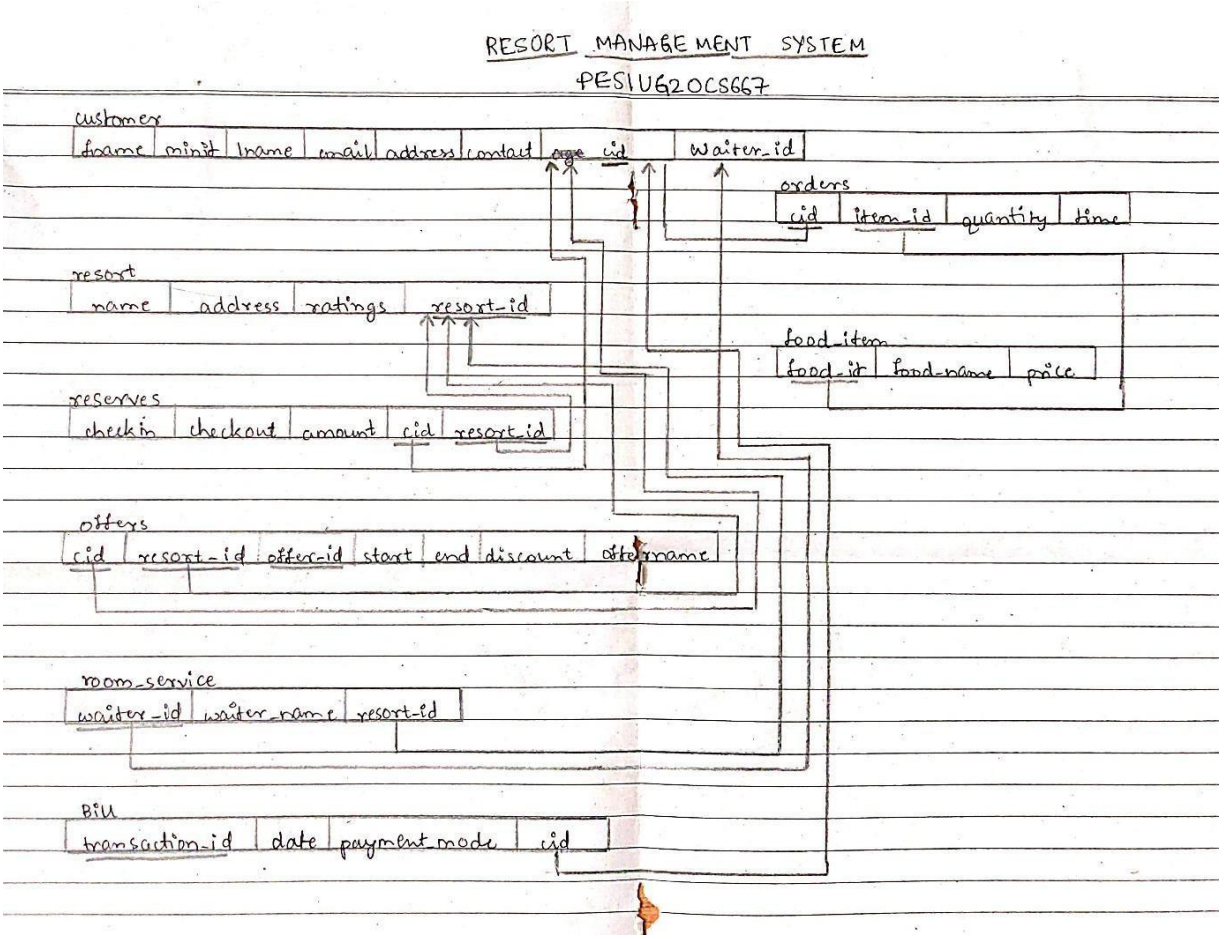
Traditionally, all the information of the customers who have booked a particular resort is stored in books/ledgers. This is very inefficient and labour intensive. This problem can be solved using SQL database, which is stored digitally and information is safe.

The database consists of several entities like customer, resort, offers, food_item, reservation, bill,. Etc which contain all the necessary information.

ER Diagram



Relational Schema



DDL statements - Building the database

```
CREATE TABLE resort (
    resort_id DECIMAL(4, 0) PRIMARY KEY CHECK (resort_id > 0),
    resort_name varchar(50) NOT NULL,
    address varchar(40) NOT NULL,
    rating DECIMAL(3,2),
    price_per_day float
);
CREATE TABLE room_service (
    waiter_id DECIMAL(3, 0) PRIMARY KEY,
    waiter_name varchar(20),
    resort_id DECIMAL(4, 0),
    FOREIGN KEY (resort_id) REFERENCES resort(resort_id) ON DELETE CASCADE
);
CREATE TABLE customer (
    cid Decimal(4, 0) PRIMARY KEY CHECK (cid > 0),
    fname VARCHAR(20),
```

```
    minit CHAR(1),
    lname VARCHAR(20),
    address varchar(30),
    email varchar(30),
    contactNo DECIMAL(10, 0),
    waiter_id DECIMAL(3, 0),
    FOREIGN KEY (waiter_id) REFERENCES room_service(waiter_id) ON DELETE
CASCADE
);
CREATE TABLE offers (
    offer_id DECIMAL(2, 0),
    offer_name varchar(20),
    cid DECIMAL(4, 0) CHECK (cid > 0),
    resort_id DECIMAL(4, 0) CHECK (resort_id > 0),
    discount int,
    startdate DATE,
    enddate DATE,
    FOREIGN KEY (resort_id) REFERENCES resort(resort_id) ON DELETE CASCADE,
    FOREIGN KEY (cid) REFERENCES customer(cid) ON DELETE CASCADE,
    PRIMARY KEY(resort_id, cid, offer_id)
);
CREATE TABLE reservation(
    cid DECIMAL(4, 0) CHECK (cid > 0),
    resort_id DECIMAL(4, 0) CHECK (resort_id > 0),
    checkin DATE NOT NULL,
    checkout DATE NOT NULL,
    amount float,
    CHECK (checkout > checkin),
    FOREIGN KEY (cid) REFERENCES customer(cid) ON DELETE CASCADE,
    FOREIGN KEY (resort_id) REFERENCES resort(resort_id) ON DELETE CASCADE,
    PRIMARY KEY(resort_id, cid)
);
CREATE TABLE relatives(
    cid DECIMAL(4, 0),
    relative_name VARCHAR(20),
    gender char(1),
    relationship VARCHAR(20),
    FOREIGN KEY (cid) REFERENCES customer(cid) ON DELETE CASCADE ON UPDATE
CASCADE,
    PRIMARY KEY (cid, relative_name)
);
CREATE TABLE food_item (
    food_id decimal(2, 0),
    food_name varchar(20),
    price numeric CHECK (
        price BETWEEN 0.00 AND 500.00
```

```
    ),
    PRIMARY KEY(food_id)
);
 = customer orders food via some food delivery app
CREATE TABLE orders (
    cid DECIMAL(4, 0),
    item_id DECIMAL(3, 0),
    quantity INT,
    time DATETIME,
    FOREIGN KEY(cid) REFERENCES customer(cid) ON DELETE RESTRICT,
    FOREIGN KEY(item_id) REFERENCES food_item(food_id) ON DELETE RESTRICT,
    PRIMARY KEY (cid, item_id)
);
 = number of digits in transaction Id varies, 12 is most common
CREATE TABLE bill (
    transaction_id DECIMAL(12, 0) PRIMARY KEY,
    date DATE,
    cid DECIMAL(4, 0),
    paymentmode varchar(10),
    FOREIGN KEY(cid) REFERENCES customer(cid) ON DELETE RESTRICT
);
```

Populating the Database

```
insert into food_item Values('01','thaali','400');
insert into food_item Values('02','pizza','250');
insert into food_item Values('03','ghee rice','120');
insert into food_item Values('04','schezwan fried rice','120');
insert into food_item Values('05','chicken biryani','180');
insert into food_item Values('06','mutton thaali','320');
insert into food_item Values('07','surma fish','240');


insert into offers Values('01','winter vaction',
'1001','1006','20','2022-12-20','2022-12-30');
insert into offers Values('02','diwali offer',
'1005','1001','15','2022-10-01','2022-12-31');
insert into offers Values('03','special offer',
'1007','1007','25','2022-11-01','2022-12-31');


insert into resort Values ('1001',"The Dukes Retreat",'Lonavala',5,1999);
insert into resort Values ('1002',"Ferreira Resort",'Lonavala',4,1799);
```

```
insert into resort Values ('1003',"Villa San Lorentz",'Lonavala',5,1699);
insert into resort Values ('1004',"Misty Meadows",'Lonavala',4,1499);
insert into resort Values ('1005',"Sunshine Resort",'Lonavala',5,1699);
insert into resort Values ('1006',"Dandeli Jungle Resort",'Dandeli',5,1499);
insert into resort Values ('1007',"Wild Planet Jungle Resort",'Dandeli',5,1299);
insert into resort Values ('1008',"Swast-Mast Resort",'Lonavala',4,1399);
insert into resort Values ('1009',"Alurkar Resort",'Belgaum',5,1499);
insert into resort Values ('1010',"Gavkari",'Belgaum',5,999);


insert into room_service Values('101','chotu','1001');
insert into room_service Values('102','bhola','1002');
insert into room_service Values('103','brijesh','1003');
insert into room_service Values('104','ajay','1004');
insert into room_service Values('105','shukh','1005');
insert into room_service Values('106','vikalp','1006');
insert into room_service Values('107','alam','1007');
insert into room_service Values('108','suresh','1008');
insert into room_service Values('109','keshav','1006');
insert into room_service Values('110','sharad','1009');
insert into room_service Values('111','munna','1010');

insert into customer Values('1001','narendra','','modi','gujarat'
,'modi@gmail.com' ,'1234567890','106');
insert into customer Values('1002','amit','','shah','gujarat' ,'shah@gmail.com'
,'1234567890','102');
insert into customer Values('1003','atal','b','vajpayee','bihar'
,'vajpayee@gmail.com' ,'1234567890','103');
insert into customer Values('1004','abdul','','kalam','patna' ,'aniket@gmail.com'
,'1234567890','104');
insert into customer Values('1005','yogi','','adityanath','uttar pradesh'
,'yogi@gmail.com' ,'1234567890','102');
insert into customer Values('1006','balasaheb','','thakre','mumbai'
,'thakre@gmail.com' ,'1234567890','101');
insert into customer Values('1007','basavraj','','bommai','karnataka'
,'bommai@gmail.com' ,'1234567890','102');


insert into bill  Values('250707244234','2022-11-21','1001','upi');
insert into bill Values('202622637838','2022-11-22','1002','debit card');
insert into bill Values('644161913172','2022-11-15','1003','credit card');
insert into bill Values('988602103725','2022-11-21','1004','upi');
insert into bill  Values('859741505883','2022-11-21','1005','cash');
insert into bill Values('346488919858','2022-11-26','1006','credit card');


insert into orders Values('1001','1','3','2022-11-21 10:55:05');
insert into orders Values('1002','4','4','2022-11-21 13:56:55');
insert into orders Values('1003','5','2','2022-11-10 12:35:05');
insert into orders Values('1004','7','8','2022-11-20 21:55:05');
insert into orders Values('1002','3','1','2022-11-18  07:24:05');
```

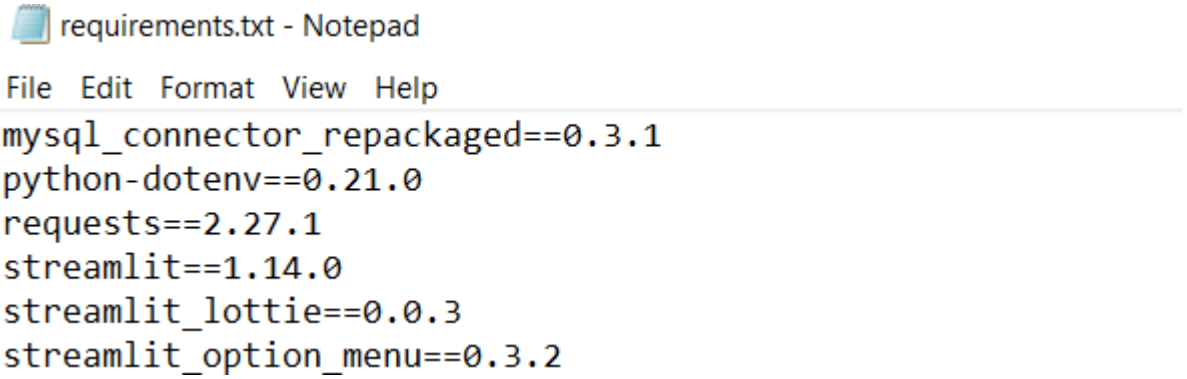
```
insert into orders Values('1006','4','4','2022-11-25  09:18:05');


insert into relatives Values('1002','bhaskar bhat','m','friend');
insert into relatives Values('1003','vijay verma','m','friend');
insert into relatives Values('1003','sujay patil','m','friend');
insert into relatives Values('1004','sharad shukla','m','friend');
insert into relatives Values('1004','anil desai','m','son');
insert into relatives Values('1005','manish gupta','m','colleague');
insert into relatives Values('1005','kartik singh','m','colleague');
insert into relatives Values('1005','shilpa trivedi','f','colleague');
insert into relatives Values('1006','manthan patil','m','friend');


insert into reservation values('1001','1001','2022-11-11','2022-11-21',15592);
insert into reservation values('1002','1002','2022-11-15','2022-11-22',12593);
insert into reservation values('1003','1003','2022-11-15','2022-11-227',11893);
insert into reservation values('1004','1004','2022-11-16','2022-11-21',7495);
insert into reservation values('1005','1005','2022-11-18','2022-11-21',4077.6);
insert into reservation values('1006','1006','2022-11-20','2022-11-26',8994);
```

Tools Used

- UI for database operations – streamlit
- Database connection – mysql-connector-python
- Xampp



Queries

Join queries (at least 6)

Write the query in English Language, Show the equivalent SQL statement and also screenshot of the query and the results.

Include 2 regular join, 2 co-related and 2 nested queries

Regular join

Display first name, last name and offer name for all customers who booked a resort under special offer.

select fname, lname, offer_name from customer c join offers o on c.cid=o.cid;

```
MariaDB [mohite_resorts]> select fname, lname, offer_name from customer c join offers o on c.cid=o.cid;
+----------+------------+-----------------+
| fname    | lname      | offer_name      |
+----------+------------+-----------------+
| yogi     | adityanath | diwali offer    |
| narendra | modi       | winter vaction  |
| basavraj | bommai     | special offer   |
+----------+------------+-----------------+
3 rows in set (0.001 sec)

MariaDB [mohite_resorts]>
```

Display food_name, price and quantity of all food orders made by customer with cid=1001

```
select f.food_name, f.price, o.quantity from food_item f join orders o on
f.food_id=o.item_id where o.cid=1001;
```

```
MariaDB [mohite_resorts]> select f.food_name, f.price, o.quantity from food_item f join orders o on f.food_id=o.item_id where o.cid=1001;
+-----------+-------+----------+
| food_name | price | quantity |
+-----------+-------+----------+
| thaali    |   400 |        3 |
+-----------+-------+----------+
1 row in set (0.001 sec)

MariaDB [mohite_resorts]>
```

Correlated queries

Display resort_id, resort name and address of all those resorts where the number of bookings is greater than 1.

```
select r.resort_id, r.resort_name,r.address from resort r where r.resort_id in
(
select re.resort_id from reservation re group by re.resort_id having count(*)>1
);
```

```
MariaDB [mohite_resorts]> select r.resort_id, r.resort_name,r.address from resort r where r.resort_id in (
    -> select re.resort_id from reservation re group by re.resort_id having count(*)>1);
+-----------+------------------+-----------+
| resort_id | resort_name      | address   |
+-----------+------------------+-----------+
|      1001 | The Dukes Retreat | Lonavala |
+-----------+------------------+-----------+
1 row in set (0.001 sec)

MariaDB [mohite_resorts]>
```

Display information of resort which are available for booking

```
select r.resort_id, r.resort_name, r.address from resort r where not exists
(select * from reservation re where re.resort_id=r.resort_id);
```

```
MariaDB [mohite_resorts]> select r.resort_id, r.resort_name, r.address from resort r where not exists (select *
from reservation re where re.resort_id=r.resort_id);
+-----------+---------------------------+----------+
| resort_id | resort_name               | address  |
+-----------+---------------------------+----------+
|      1003 | Villa San Lorentz         | Lonavala |
|      1007 | Wild Planet Jungle Resort | Dandeli  |
|      1008 | Swast-Mast Resort         | Lonavala |
|      1009 | Alurkar Resort            | Belgaum  |
|      1010 | Gavkari                   | Belgaum  |
|      5365 | test                      | test loc |
|      6659 | Rose Garden's             | hubli    |
+-----------+---------------------------+----------+
7 rows in set (0.001 sec)

MariaDB [mohite_resorts]>
```

Nested queries

List all customers who did not order any food during their stay at a resort.

```
select c.fname, c.lname from customer c where cid not in (select cid from customer
natural join reservation);
```

```
MariaDB [mohite_resorts]> select c.fname, c.lname from customer c where cid not in (select cid from customer nat
ural join reservation);
+----------+----------+
| fname    | lname    |
+----------+----------+
| atal     | vajpayee |
| basavraj | bommai   |
+----------+----------+
2 rows in set (0.017 sec)

MariaDB [mohite_resorts]>
```

Display customer information whose booking amount is greater than average amount of all reservation.

```
select c.fname,
    c.lname,
    c.address,
    r.amount
from customer c
    join reservation r on c.cid = r.cid
where r.amount >(
        select avg(amount)
        from reservation
    );
```

```
MariaDB [mohite_resorts]> select c.fname,
    ->     c.lname,
    ->     c.address,
    ->     r.amount
    -> from customer c
    ->     join reservation r on c.cid = r.cid
    -> where r.amount >(
    ->         select avg(amount)
    ->         from reservation
    ->     );
+----------+-------+---------+--------+
| fname    | lname | address | amount |
+----------+-------+---------+--------+
| narendra | modi  | gujarat |  15592 |
| amit     | shah  | gujarat |  12593 |
+----------+-------+---------+--------+
2 rows in set (0.001 sec)

MariaDB [mohite_resorts]>
```

Aggregate Functions (at least 2)

Showcase at least 2 Aggregate function queries. Write the query in English Language, Show the equivalent SQL statement and also screenshot of the query and the results

1. Display total bookings for each resort

```
select r.resort_id, r.resort_name, count(*) as total_bookings from resort r join reservation rs on r.resort_id=rs.resort_id group by resort_id;
```

```
MariaDB [mohite_resorts]> select r.resort_id, r.resort_name, count(*) as total_bookings from resort r
join reservation rs on r.resort_id=rs.resort_id group by resort_id;
+-----------+----------------------+----------------+
| resort_id | resort_name          | total_bookings |
+-----------+----------------------+----------------+
|      1001 | The Dukes Retreat    |              2 |
|      1002 | Ferreira Resort      |              1 |
|      1004 | Misty Meadows        |              1 |
|      1005 | Sunshine Resort      |              1 |
|      1006 | Dandeli Jungle Resort |              1 |
+-----------+----------------------+----------------+
5 rows in set (0.003 sec)

MariaDB [mohite_resorts]>
```

2. List all resorts city wise

```
select r.address as location, count(*) total_resorts from resort r group by r.address;
```

```
MariaDB [mohite_resorts]> select r.address as location, count(*) total_resorts from resort r group by
r.address;
+----------+---------------+
| location | total_resorts |
+----------+---------------+
| Belgaum  |             2 |
| Dandeli  |             2 |
| hubli    |             1 |
| Lonavala |             6 |
+----------+---------------+
4 rows in set (0.001 sec)

MariaDB [mohite_resorts]>
```

Set Operations (at least 2)

1. List all customers who have booked a resort with “Diwali offer” in Lonavala.

```
(select re.cid from reservation re natural join resort r where address="lonavala")
 Union
(select c.cid from customer c natural join offers o where offer_name="diwali
offer");
```

```
MariaDB [mohite_resorts]> (select re.cid from reservation re natural join resort r where address="lonavala")
    -> union
    -> select c.cid from customer c natural join offers o where offer_name="diwali offer";
+------+
| cid  |
+------+
| 1001 |
| 1002 |
| 1004 |
| 1005 |
| 6853 |
| 9257 |
+------+
```

2. List all customers who have NOT paid the bill but have ordered food item

```
(select c.cid,c.fname,c.lname from customer c where c.cid not in (select b.cid
from bill b))
intersect
(select c.cid, c.fname, c.lname from customer c where c.cid in (select o.cid from
orders o));
```

```
MariaDB [mohite_resorts]> (select c.cid,c.fname,c.lname from customer c where c.cid not in (select b.cid from bill b))
    -> intersect
    -> (select c.cid, c.fname, c.lname from customer c where c.cid in (select o.cid from orders o));
+------+---------+-------+
| cid  | fname   | lname |
+------+---------+-------+
| 3221 | Hithesh | Nayak |
+------+---------+-------+
1 row in set (0.001 sec)

MariaDB [mohite_resorts]>
```

View (at least 1)

Demonstrate creation and querying one view

Creating a view –

```
create view cust_reservation as (select * from customer c natural join reservation
re);
```

```
MariaDB [mohite_resorts]> create view cust_reservation as (select * from customer c natural join reservation re);
Query OK, 0 rows affected (0.011 sec)

MariaDB [mohite_resorts]>
```

List all customers along with resort information who have booked resort in Lonavala. Sort the output by first name of customer.

Without view we would have to join 3 tables(customer, reservation, resort)

```
select cr.fname,cr.lname,cr.address as cust_address, r.address as resort_address,
cr.checkin, cr.checkout from cust_reservation cr join resort r on
cr.resort_id=r.resort_id where r.address="lonavala" order by fname;
```

```
MariaDB [mohite_resorts]> select cr.fname,cr.lname,cr.address as cust_address, r.address as resort_address, cr.checkin, cr.checkout from cust_reservation cr join resort r
 on cr.resort_id=r.resort_id where r.address="lonavala" order by fname;
+---------+-----------+--------------+---------------+-----------+------------+
| fname   | lname     | cust_address | resort_address | checkin   | checkout   |
+---------+-----------+--------------+---------------+-----------+------------+
| abdul   | kalam     | patna        | Lonavala      | 2022-11-16 | 2022-11-21 |
| amit    | shah      | gujarat      | Lonavala      | 2022-11-15 | 2022-11-22 |
| narendra | modi     | gujarat      | Lonavala      | 2022-11-11 | 2022-11-21 |
| Nikhil  | Mohite    | Karnataka    | Lonavala      | 2022-11-17 | 2022-11-18 |
| yogi    | adityanath | uttar pradesh | Lonavala     | 2022-11-18 | 2022-11-21 |
+---------+-----------+--------------+---------------+-----------+------------+
5 rows in set (0.001 sec)

MariaDB [mohite_resorts]>
```

Triggers (Functions or Procedures)

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

Procedure - display information of all resorts in a given city

@input: city

@output: total resort count in the given city

```
DELIMITER $$
CREATE OR REPLACE PROCEDURE get_resort_count(IN city varchar(15), OUT r_count
integer)
BEGIN
    SELECT COUNT(*) into r_count FROM resort where address=city;
END $$
DELIMITER ;

SET @r_count=0;
CALL get_resort_count(“hubli”, @r_count);
SELECT @r_count;
```

```
MariaDB [mohite_resorts]> DELIMITER $$
MariaDB [mohite_resorts]> CREATE OR REPLACE PROCEDURE get_resort_count(IN city varchar(15), OUT r_count integer)
    -> BEGIN
    ->     SELECT COUNT(*) into r_count FROM resort where address=city;
    -> END $$
Query OK, 0 rows affected (0.027 sec)

MariaDB [mohite_resorts]> DELIMITER ;
MariaDB [mohite_resorts]> set @c=0;
Query OK, 0 rows affected (0.000 sec)

MariaDB [mohite_resorts]> call get_resort_count("hubli",@c);
Query OK, 1 row affected (0.001 sec)

MariaDB [mohite_resorts]> select @c;
+------+
| @c   |
+------+
|    1 |
+------+
1 row in set (0.001 sec)

MariaDB [mohite_resorts]> select * from resort where address="hubli";
+-----------+---------------+---------+--------+---------------+
| resort_id | resort_name   | address | rating | price_per_day |
+-----------+---------------+---------+--------+---------------+
|      6659 | Rose Garden's | hubli   |   4.50 |           799 |
+-----------+---------------+---------+--------+---------------+
1 row in set (0.001 sec)

MariaDB [mohite_resorts]>
```

Trigger - display error message when a new entry is added to a resort which is already booked.

```
DELIMITER $$
CREATE OR REPLACE TRIGGER valid_reservation_on_insert
BEFORE INSERT
ON reservation FOR EACH ROW
BEGIN
    DECLARE error_msg VARCHAR(255);
    SET error_msg = ("Resort is already booked!");
    -- if checkin for new entry is before checkout for that resort
    IF NEW.checkin < (select checkout from reservation where
resort_id=NEW.resort_id) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = error_msg;
    END IF;
END $$

CREATE OR REPLACE TRIGGER valid_reservation_on_update
BEFORE UPDATE
ON reservation FOR EACH ROW
BEGIN
    DECLARE error_msg VARCHAR(255);
    SET error_msg = ("Resort is already booked!");
    -- if checkin for new entry is before checkout for that resort
    IF NEW.checkin < (select checkout from reservation where
resort_id=NEW.resort_id) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = error_msg;
    END IF;
END $$

DELIMITER ;
```

```
MariaDB [mohite_resorts]> DELIMITER $$
MariaDB [mohite_resorts]> CREATE OR REPLACE TRIGGER valid_reservation_on_update
    -> BEFORE INSERT
    -> ON reservation FOR EACH ROW
    -> BEGIN
    ->     DECLARE error_msg VARCHAR(255);
    ->     SET error_msg = ("Resort is already booked!");
    ->     -- if checkin for new entry is before checkout for that resort
    ->     IF NEW.checkin < (select checkout from reservation where resort_id=NEW.resort_id)  THEN
    ->         SIGNAL SQLSTATE '45000'
    ->         SET MESSAGE_TEXT = error_msg;
    ->     END IF;
    -> END $$
Query OK, 0 rows affected (0.012 sec)

MariaDB [mohite_resorts]> DELIMITER ;
MariaDB [mohite_resorts]>
```

```
MariaDB [mohite_resorts]> CREATE OR REPLACE TRIGGER valid_reservation_on_update
    -> BEFORE UPDATE
    -> ON reservation FOR EACH ROW
    -> BEGIN
    ->     DECLARE error_msg VARCHAR(255);
    ->     SET error_msg = ("Resort is already booked!");
    ->     -- if checkin for new entry is before checkout for that resort
    ->     IF NEW.checkin < (select checkout from reservation where resort_id=NEW.resort_id)  THEN
    ->         SIGNAL SQLSTATE '45000'
    ->         SET MESSAGE_TEXT = error_msg;
    ->     END IF;
    -> END $$
Query OK, 0 rows affected (0.020 sec)

MariaDB [mohite_resorts]> DELIMITER ;
MariaDB [mohite_resorts]>
```

Example

Successful insertion

```
MariaDB [mohite_resorts]> insert into customer values (9257,"bhaskar","","sharma","Karnataka","nikhilmohitelhs@gmail.com",09663147739,NULL);
Query OK, 1 row affected (0.008 sec)

MariaDB [mohite_resorts]> select * from reservation;
+------+-----------+------------+------------+--------+
| cid  | resort_id | checkin    | checkout   | amount |
+------+-----------+------------+------------+--------+
| 1001 |      1001 | 2022-11-11 | 2022-11-21 |  15592 |
| 1002 |      1002 | 2022-11-15 | 2022-11-22 |  12593 |
| 1004 |      1004 | 2022-11-16 | 2022-11-21 |   7495 |
| 1005 |      1005 | 2022-11-18 | 2022-11-21 | 4077.6 |
| 1006 |      1006 | 2022-11-20 | 2022-11-26 |   8994 |
| 3221 |      9439 | 2023-01-01 | 2023-01-07 |   7794 |
+------+-----------+------------+------------+--------+
6 rows in set (0.000 sec)

MariaDB [mohite_resorts]> insert into reservation values(9257,1003,"2022-11-17","2022-11-18",1699.000000);
Query OK, 1 row affected (0.010 sec)

MariaDB [mohite_resorts]>
```

Case when error is triggered

```
MariaDB [mohite_resorts]> insert into customer values (3000,"hitesh","","patel","barmer","hitesh@gmail.com",6363997532,NULL);
Query OK, 1 row affected (0.007 sec)

MariaDB [mohite_resorts]> insert into reservation values(3000,1003,"2022-11-17","2022-11-18",1699.000000);
ERROR 1644 (45000): Resort is already booked!
MariaDB [mohite_resorts]> select * from reservation where resort_id=1003;
+------+-----------+------------+------------+--------+
| cid  | resort_id | checkin    | checkout   | amount |
+------+-----------+------------+------------+--------+
| 9257 |      1003 | 2022-11-17 | 2022-11-18 |   1699 |
+------+-----------+------------+------------+--------+
1 row in set (0.000 sec)

MariaDB [mohite_resorts]>
```

Developing a Frontend

The frontend should support

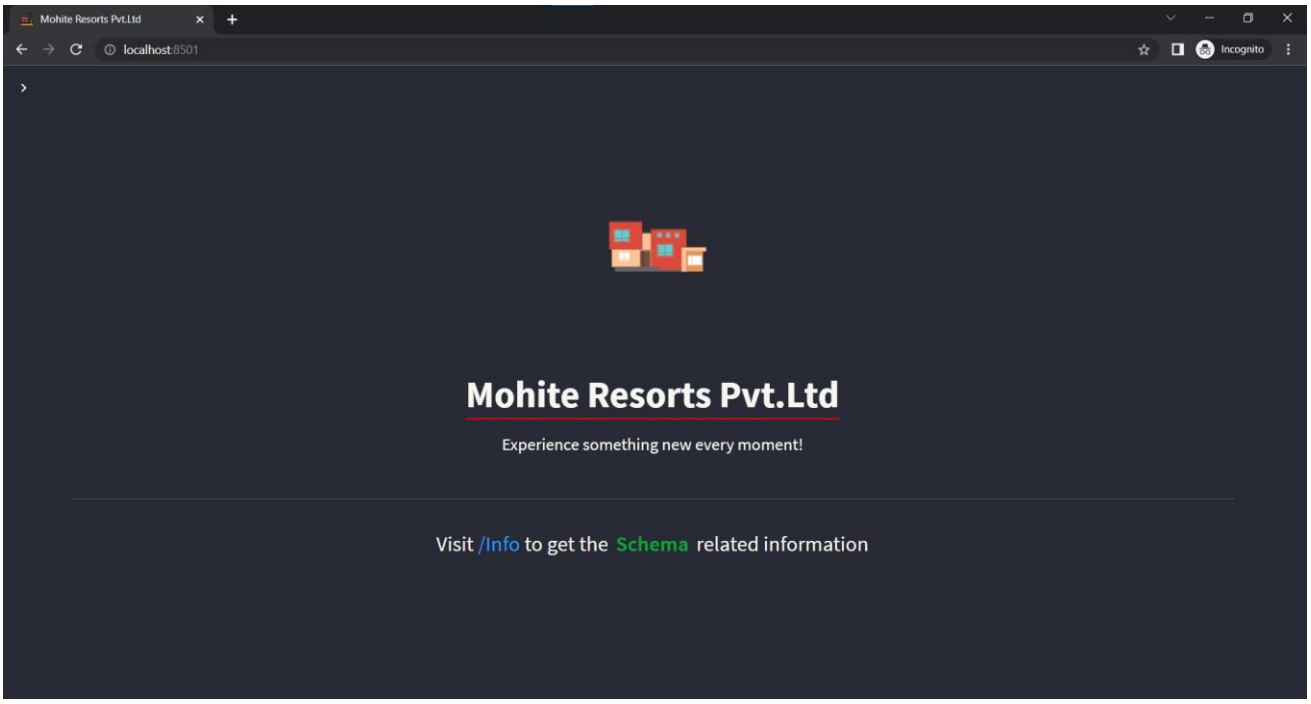
1. Addition, Modification and Deletion of records from any chosen table – Done ✅
2. There should be a window to accept and run any SQL statement and display the result – Done ✅

Frontend made using **Streamlit**

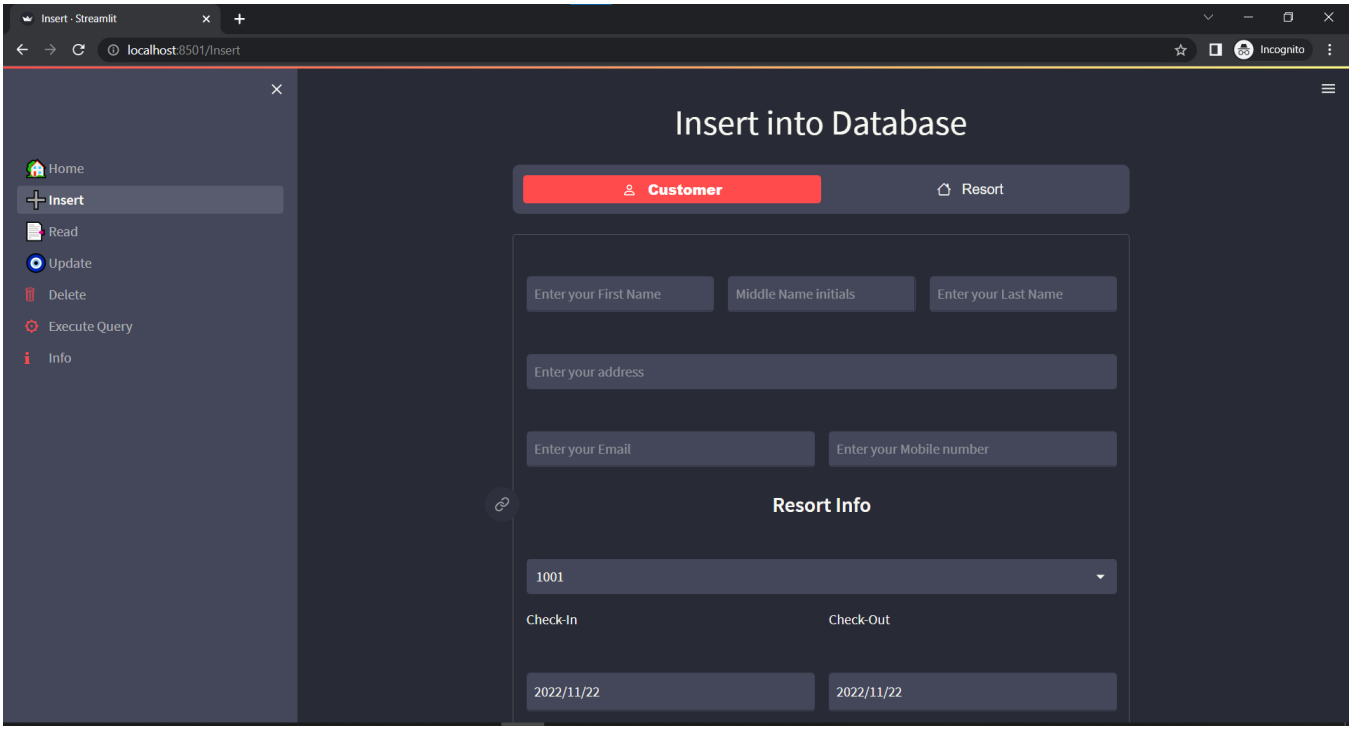
Project code + materials

GitHub – https:/ www.github.com/nkilm/db-for-a-resort

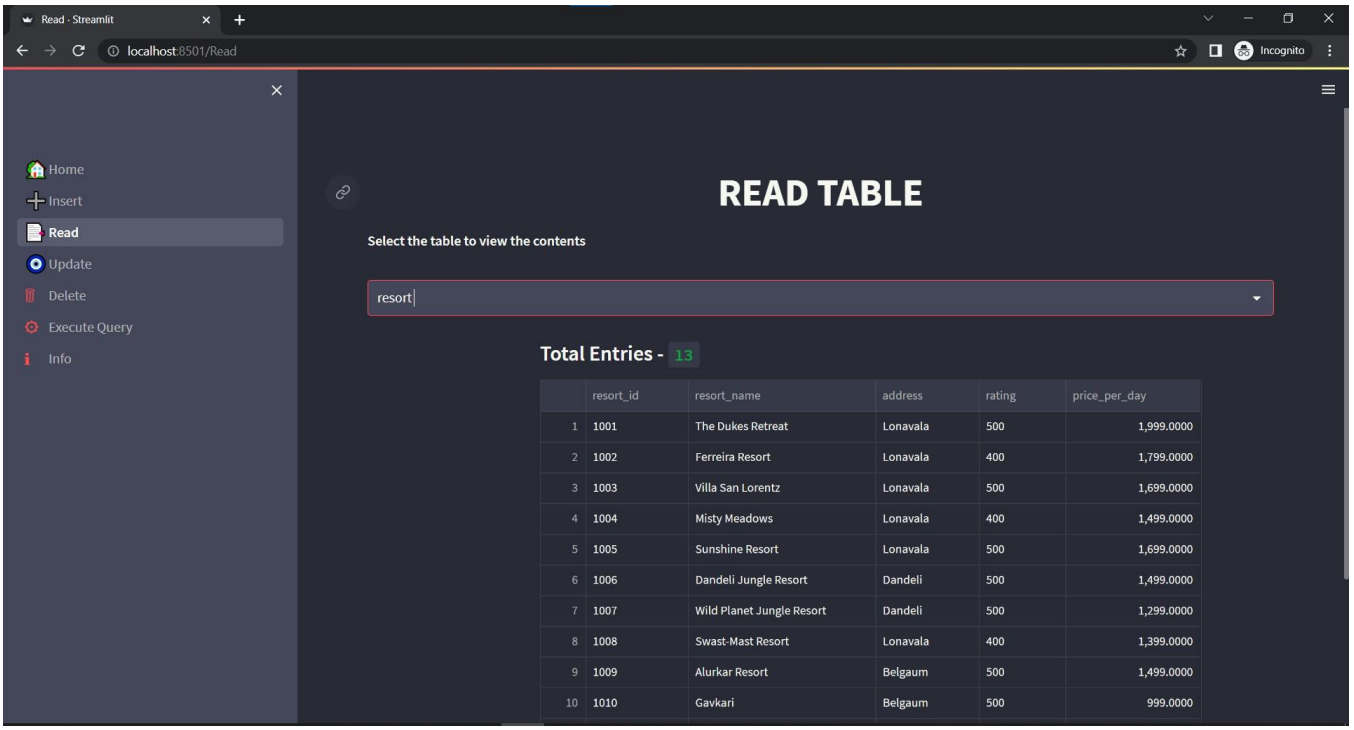
Frontend Visuals



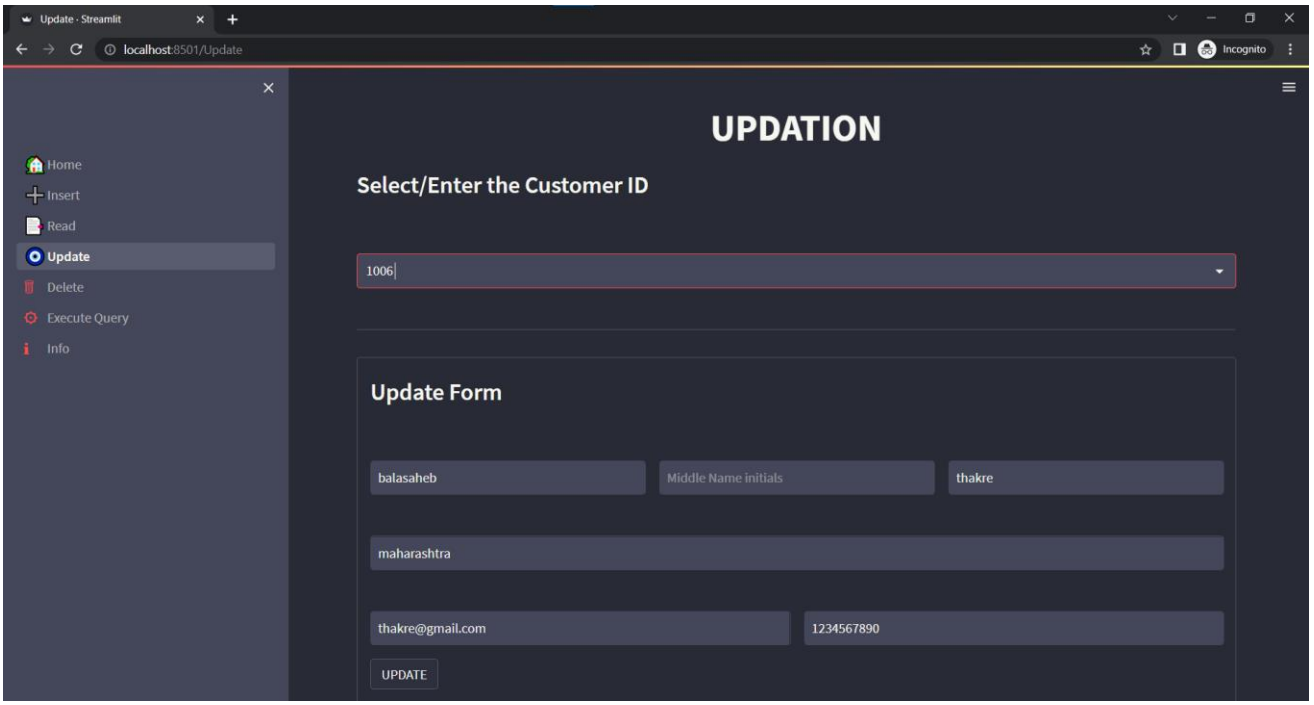
INSERT - OPERATION



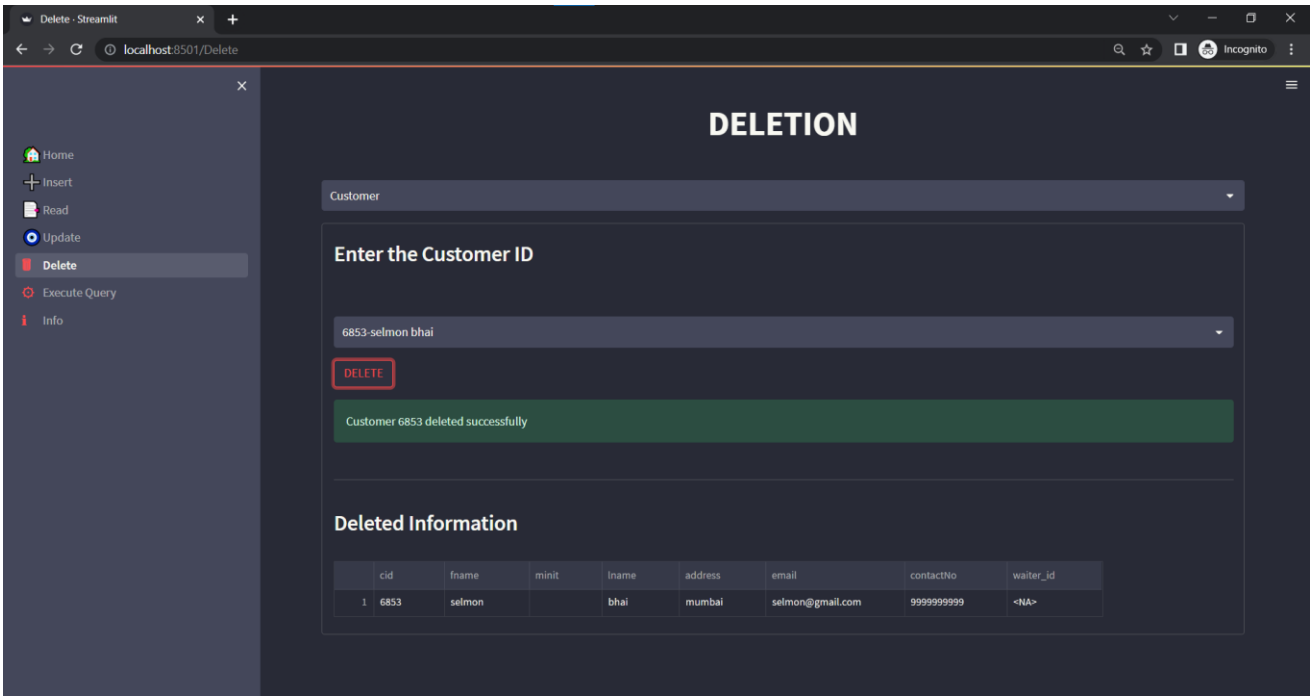
READ-OPERATION



UPDATE-OPERATION



DELETE-OPERATION



QUERY-EXECUTION