

Hotel Database System

BDM Final project

**Part 1:**

a)

We created five database table which are as follow:

1)Customers (Constraints: PRIMARY KEY, NOT NULL)

2)Room (Constraints: PRIMARY KEY, NOT NULL)

3)Reservation (Constraints: PRIMARY KEY, FOREIGN KEY, DEFAULT, NOT NULL)

4)Offers (Constraints: NOT NULL)

5)Events (Constraints: FOREIGN KEY, NULL, DEFAULT)

Schema for Customers Table:

CREATE TABLE bdmproject.`customers` (

`name` varchar(100) NOT NULL,

`email` varchar(100) NOT NULL,

`cell\_no` varchar(100) NOT NULL,

PRIMARY KEY (`cell\_no`)

Instead of putting Room and Reservation in Customers table, we created separate schema for Room and Reservation table

Schema for Room Table:

CREATE TABLE bdmproject.`room` (

`roomNum` varchar(100) NOT NULL,

`room\_type` varchar(100) NOT NULL,

PRIMARY KEY (`roomNum`)

)

Schema for Reservation Table:

CREATE TABLE bdmproject.`reservation` (

`arrivaldate` date NOT NULL,

`checkoutdate` date NOT NULL,

`noofperson` int(20) NOT NULL,

`room\_type` varchar(100) NOT NULL,

`roomnumber` varchar(100) DEFAULT NULL,

`cell\_no` varchar(100) NOT NULL,

`price` varchar(45) DEFAULT '200',

KEY `cell\_no` (`cell\_no`),

CONSTRAINT `reservation\_ibfk\_1` FOREIGN KEY (`cell\_no`) REFERENCES `customers` (`cell\_no`)

ON DELETE RESTRICT ON UPDATE RESTRICT )

Schema for Offers table:

CREATE TABLE bdmproject.`offers` (

`offer\_id` int(11) NOT NULL,

`promotion` varchar(100) NOT NULL,

`offer\_type` varchar(100) NOT NULL)

Schema for Events table:

CREATE TABLE bdmproject.`events` (

`event\_type` varchar(100) DEFAULT NULL,

`RoomNum` varchar(100) DEFAULT NULL,

`cell\_no` varchar(100) DEFAULT NULL,

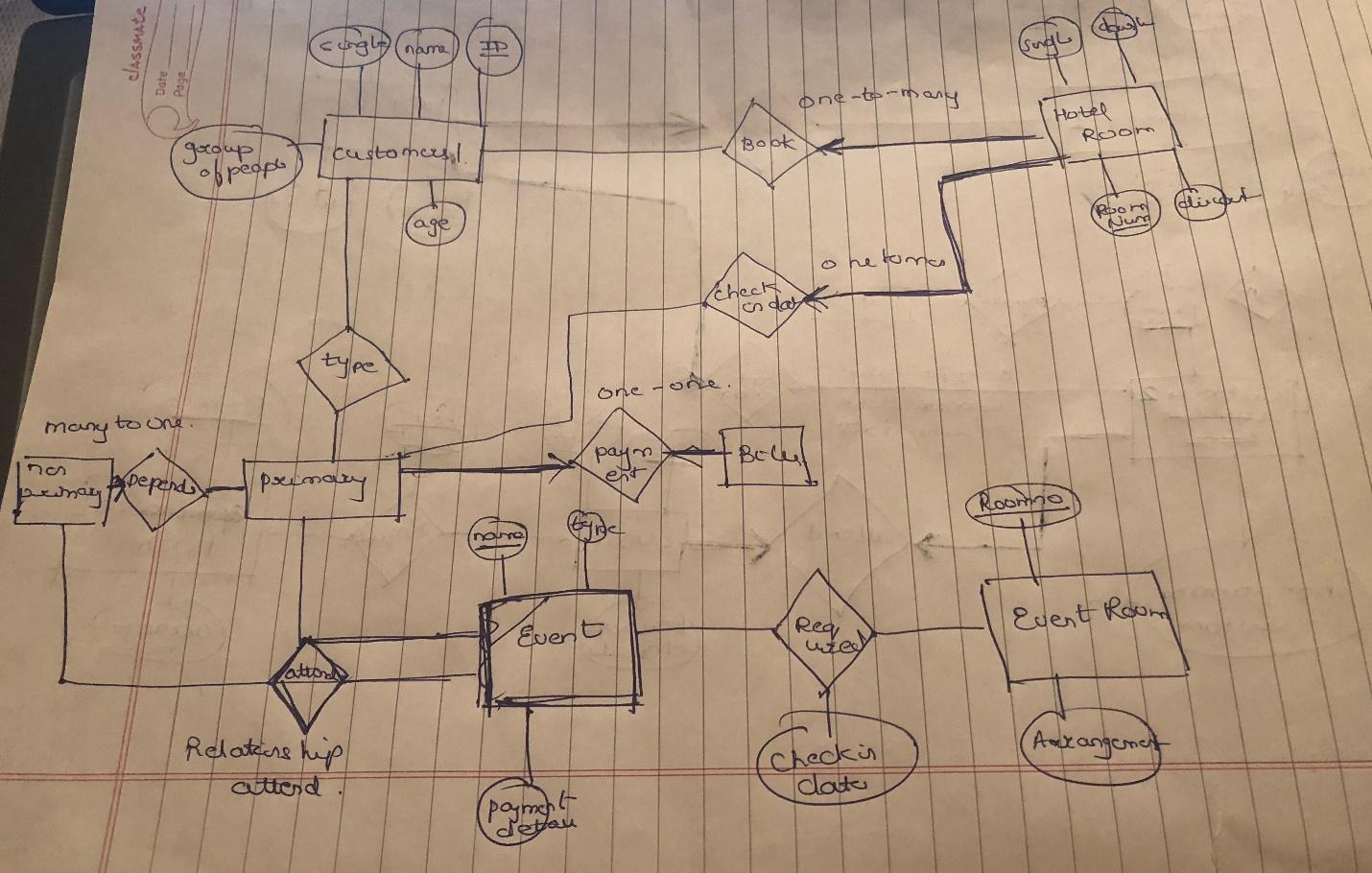
KEY `cell\_no` (`cell\_no`),

KEY `RoomNum` (`RoomNum`),

CONSTRAINT `events\_ibfk\_1` FOREIGN KEY (`cell\_no`) REFERENCES `customers` (`cell\_no`),

CONSTRAINT `events\_ibfk\_2` FOREIGN KEY (`RoomNum`) REFERENCES `room` (`roomnum`))

b) ER Diagram



In the ER diagram, we’ve considered hotel room as an entity which has attributes- room no. as primary key, room floor and room type- single, double, queen or king sized, and the discounts associated with it. The customers are identified by their name, ID, age, and general preferences about room types. Further the customer is divided into a primary customer and non-primary customer. Here, the primary customer is responsible for bill payment of the room which includes payment details and dates of occupancy. While the non-primary customers are the one’s those who are occupying the room together with the primary one’s. In the diagram we’ve shown a relationship between them as non-primary depends on primary.

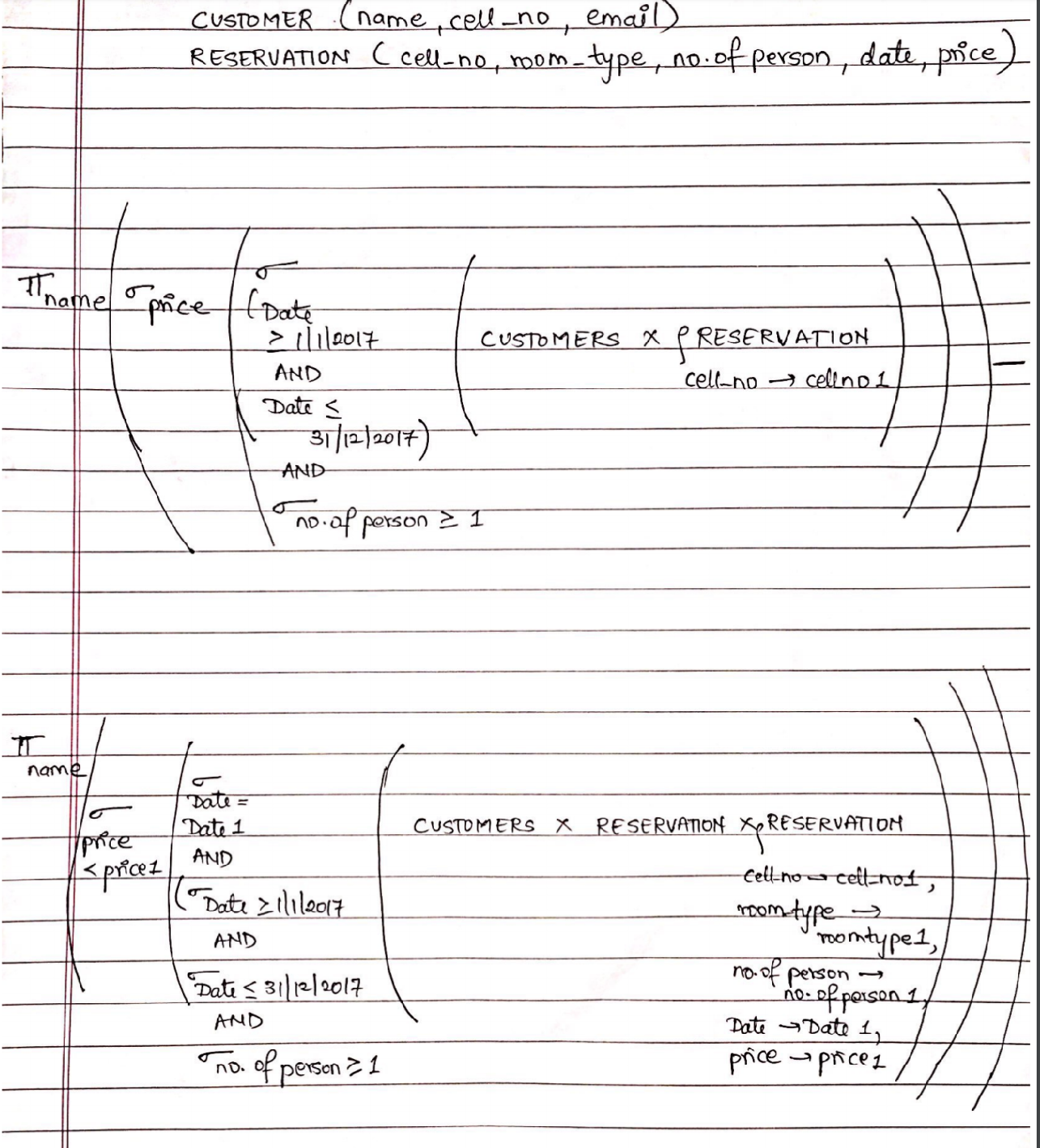
Event is considered as another entity which has attributes- event name as primary key, event type and payment details. The room for the event is booked by a customer that is billed for it. Event room is taken as another entity which has room no. as the primary key and the arrangements required and every person that stays in the room appears as a customer.

Since one room cannot be occupied by two different primary customers, we’ve taken check- in date to have a record of each customer that so no room is occupied on the same date by two different customers.

Part 2:

Relational Algebra

Find the customer (or customers) who paid the highest room rate in 2017 and is also related to at least one more non-primary customer.



b) SQL Query

Find the customer (or customers) who paid the highest room rate in 2017 and is also related to at least one more non-primary customer.

Query:

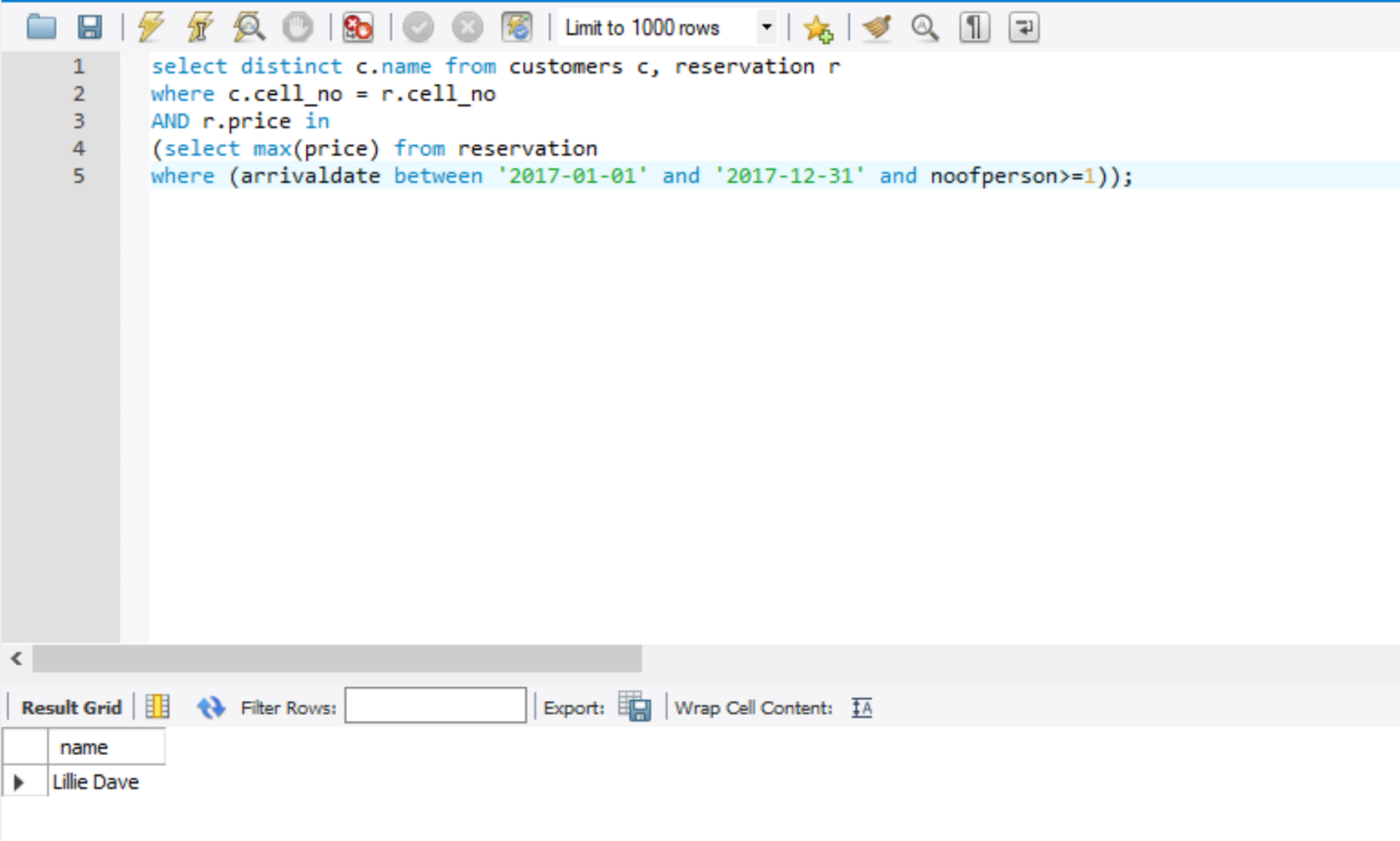
select distinct c.name from customers c, reservation r

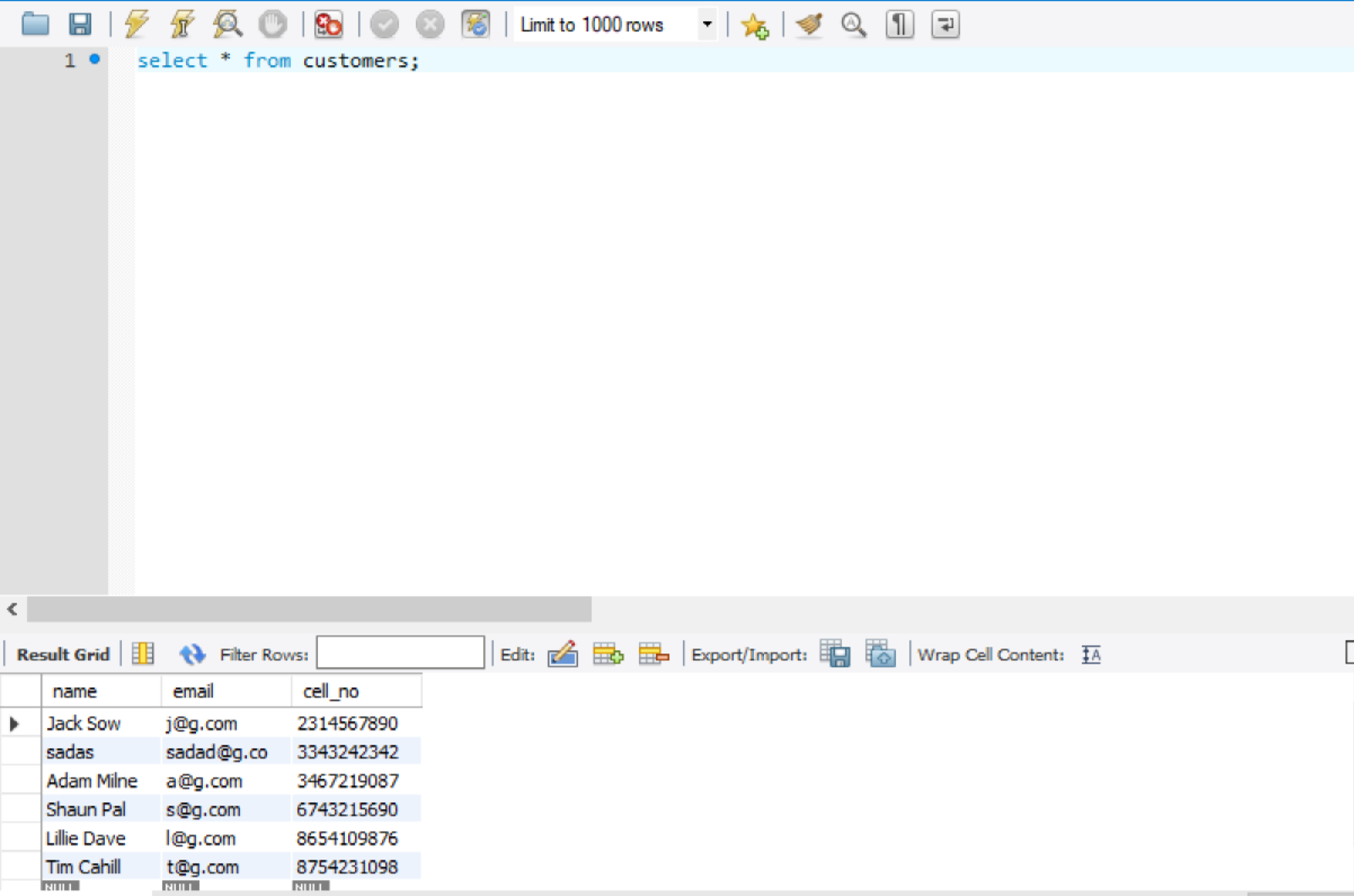
where c.cell\_no = r.cell\_no

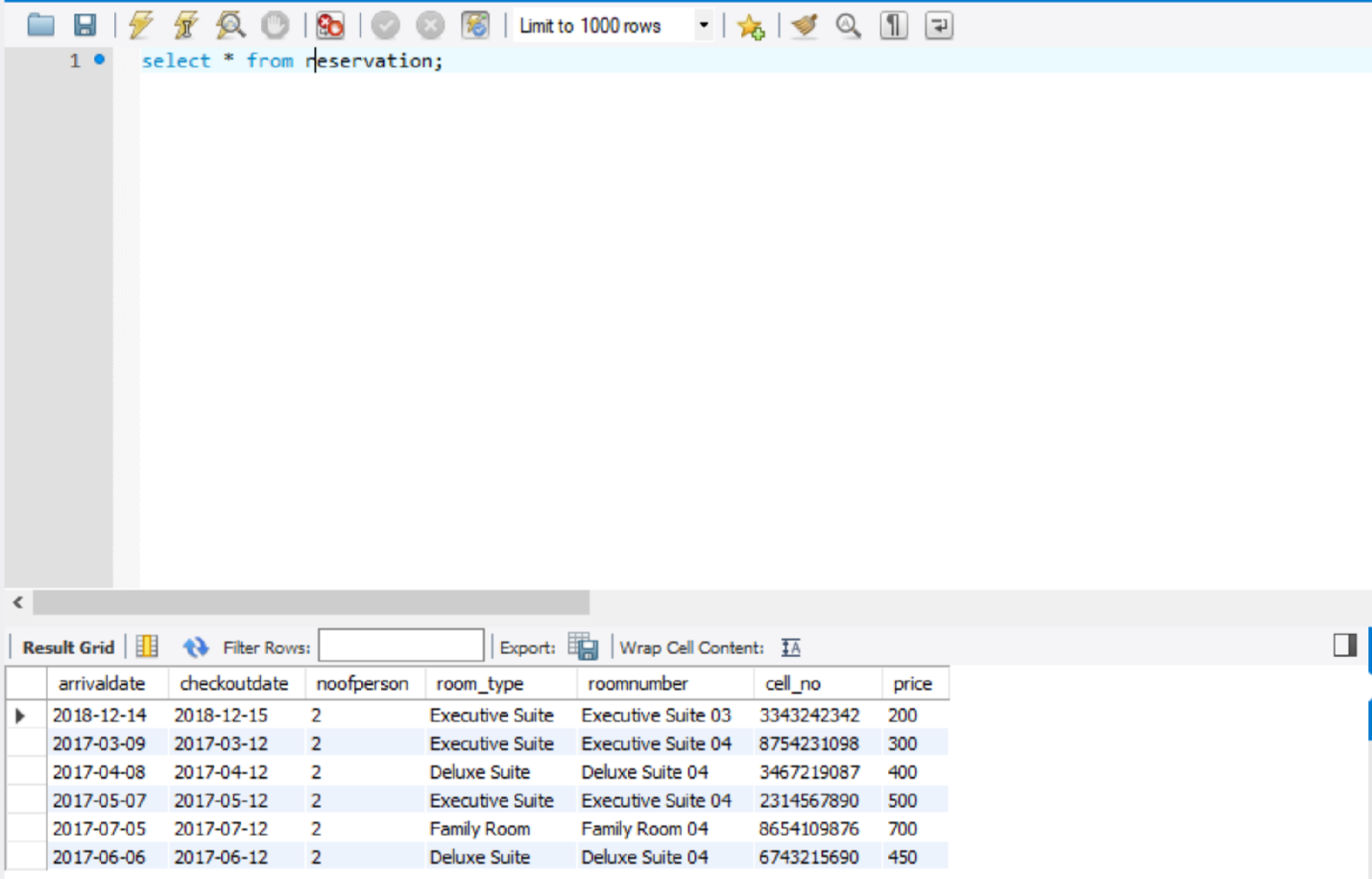
AND r.price in

(select max(price) from reservation

where (arrivaldate between '2017-01-01' and '2017-12-31' and noofperson>=1));



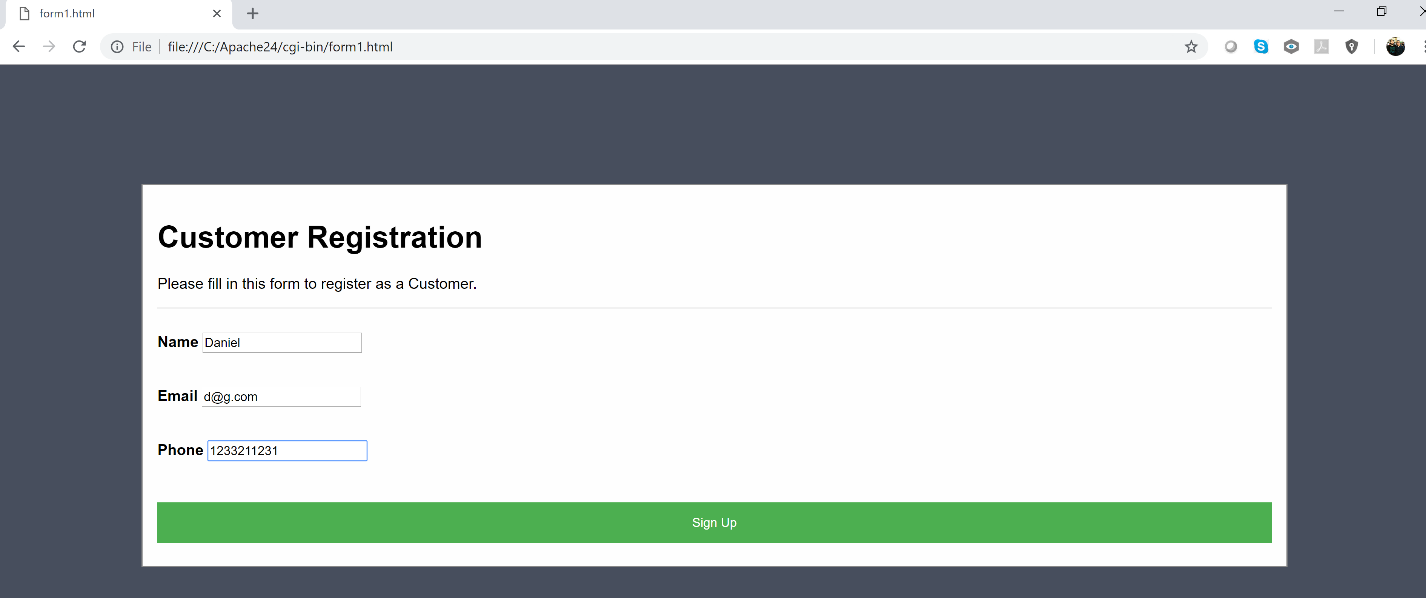


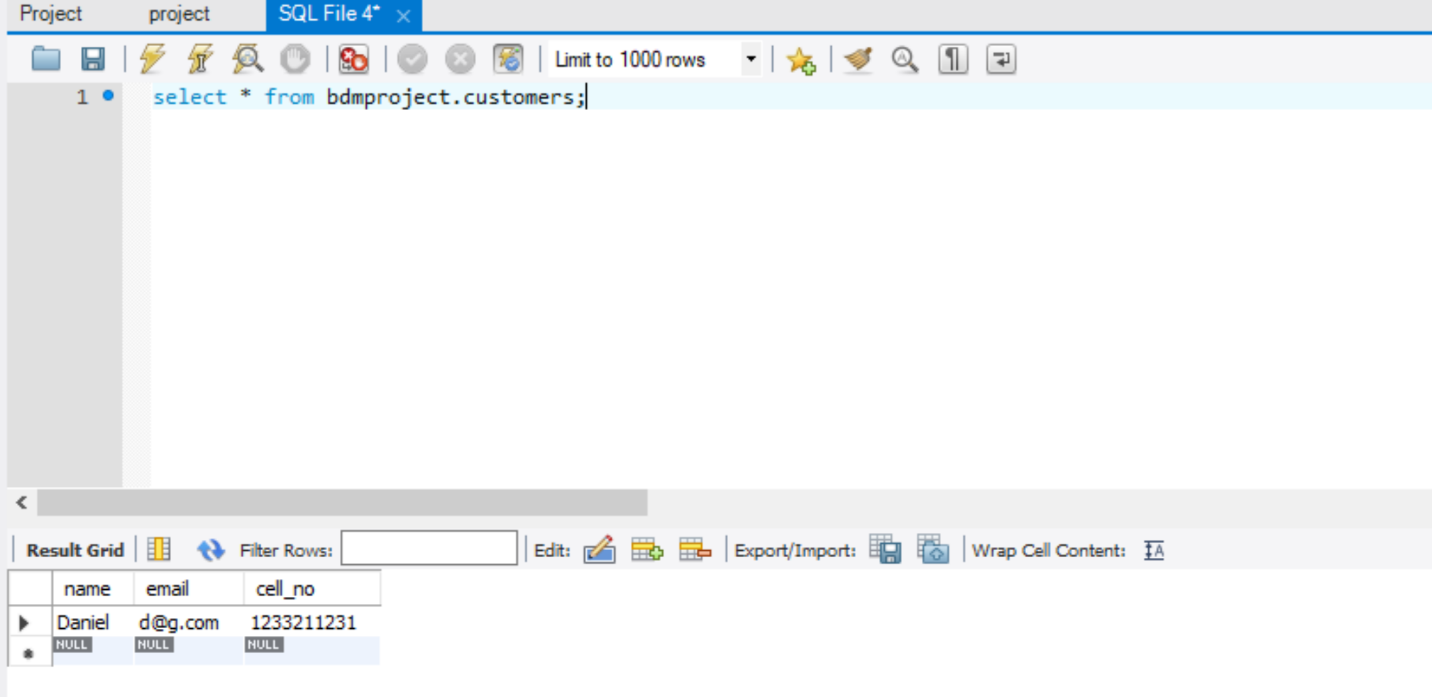


Part 3:

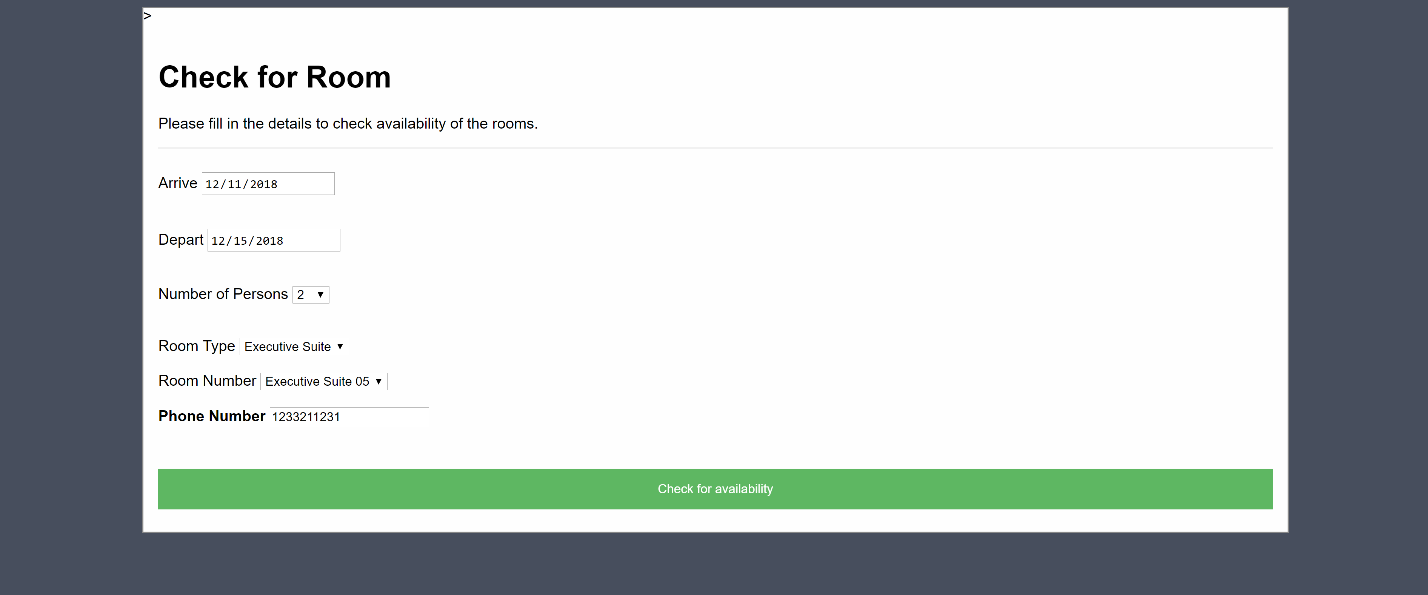
HTML Page:1

Adding Customer:

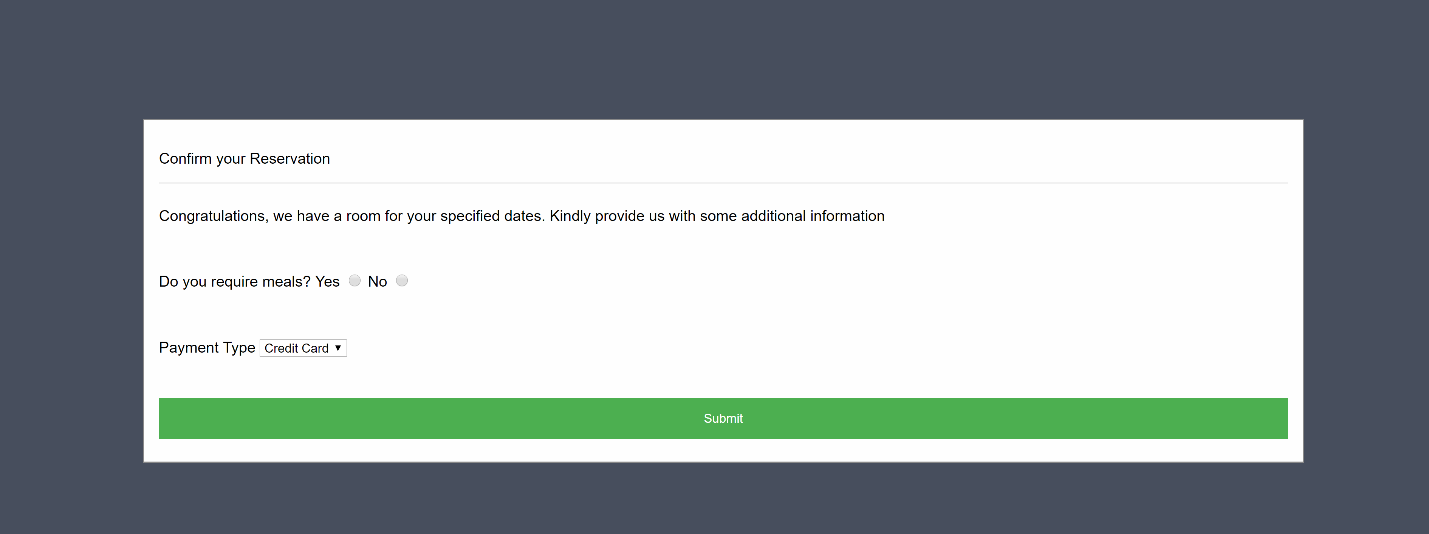


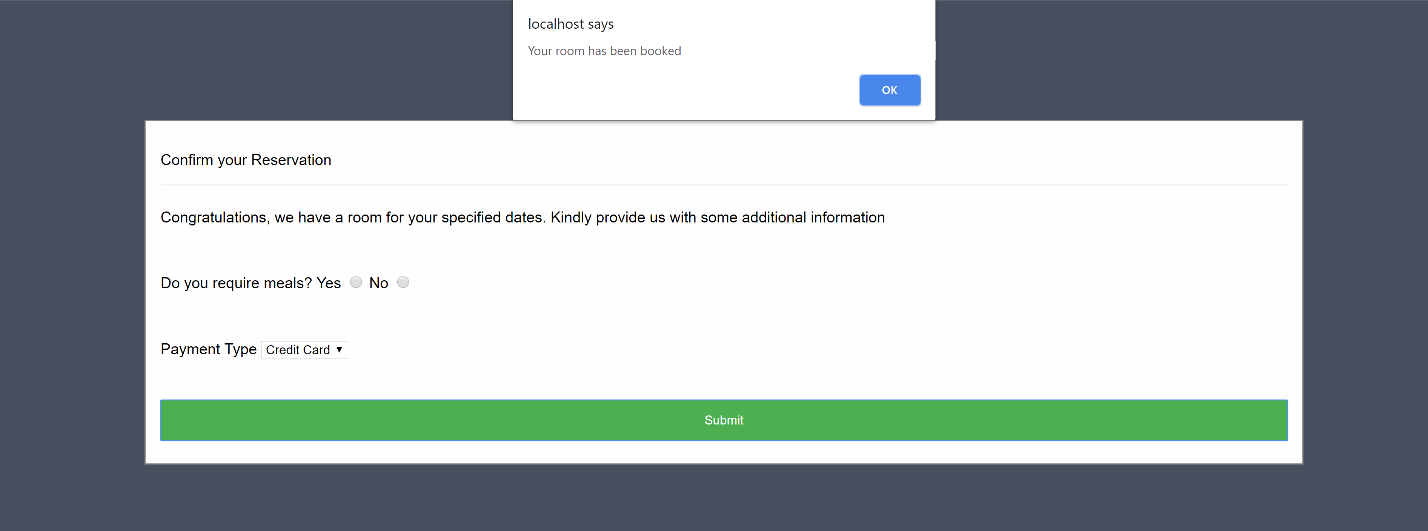


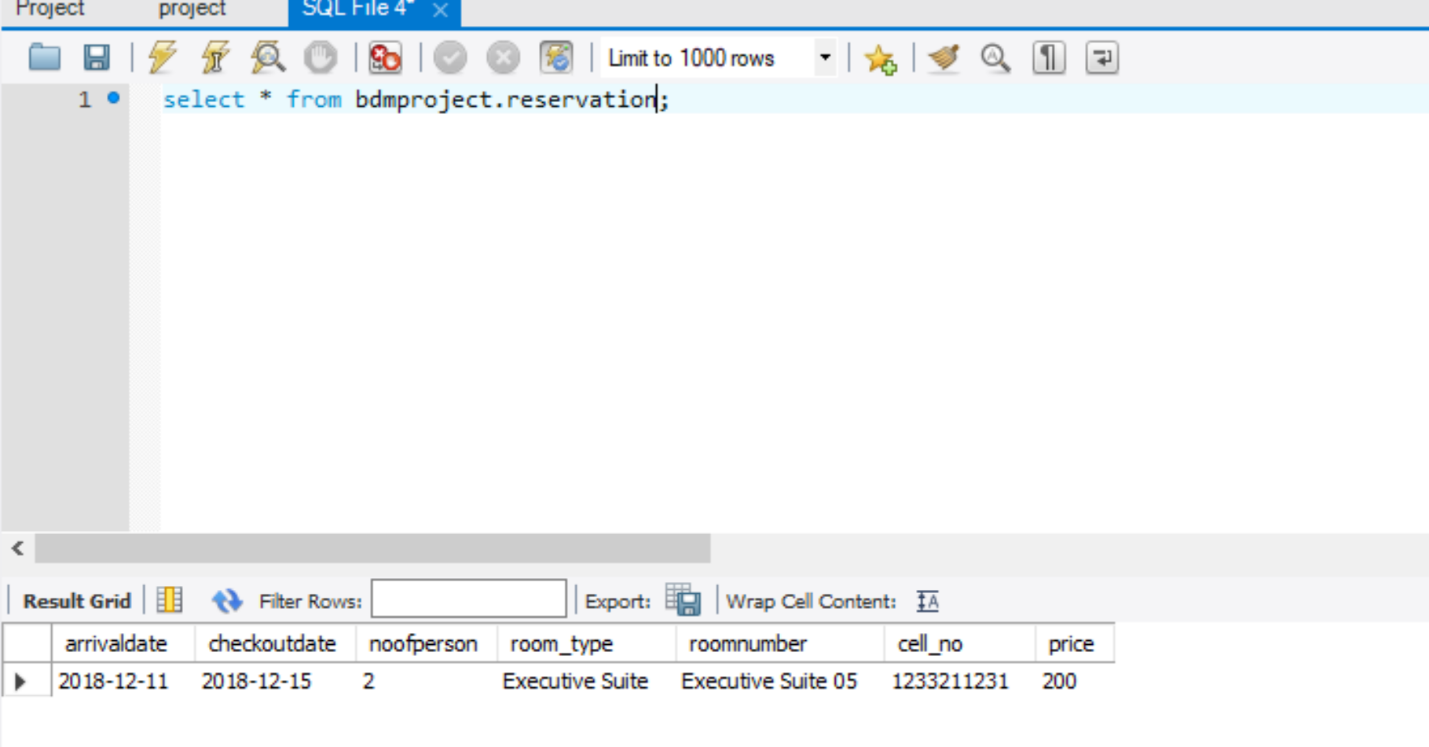
Page2: Checking Room availability



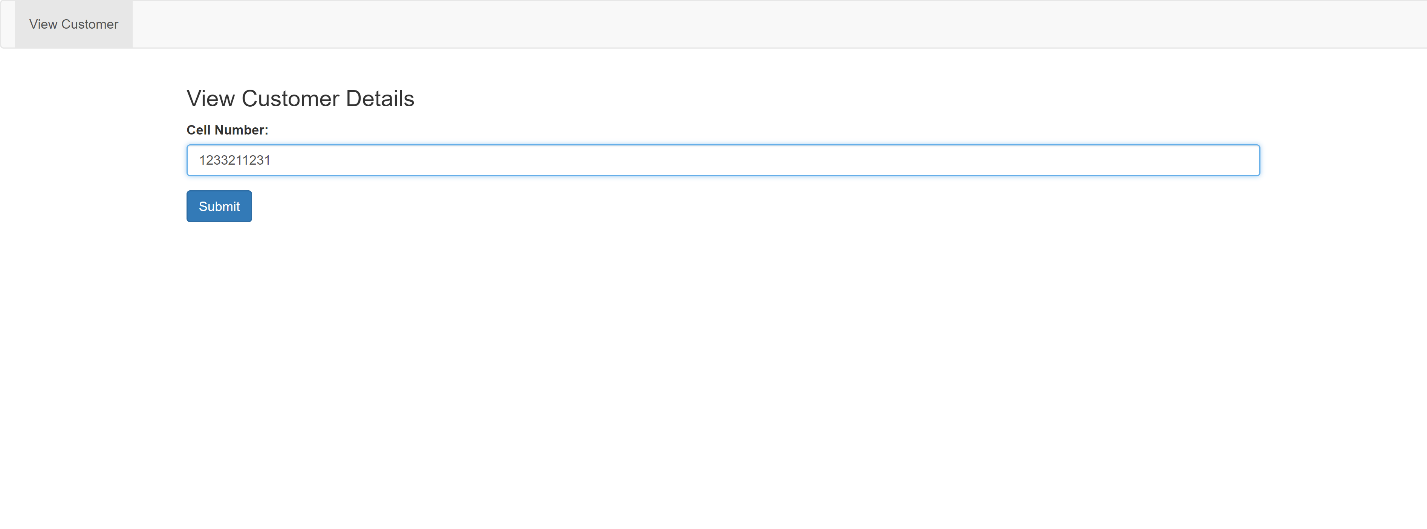
Page3: Making Room reservation







Page4: Checking Reservation with Contact detail



Output: Reservation detail of customer fetched by contact number

