

# **MINI - PROJECT REPORT**

## **HOTEL RESERVATION DATABASE**

**BY**

**AADITYA DHARNE - Roll no : 34**

**MANAV DHRUVE - Roll no : 35**

**SIMRAN DUBEY - Roll no : 37**

**TE EXTC - A**

**COURSE : DBMS**

**AY : 2021-2022**

## **TABLE OF CONTENTS**

| <b>Sr no.</b> | <b>Topic</b>                  |
|---------------|-------------------------------|
| 1             | Storyline                     |
| 2             | Entity Relationship Diagram   |
| 3             | Relational Model              |
| 4             | Components of Database Design |
| 5             | Normalization                 |
| 6             | SQL Queries                   |

# **Storyline**

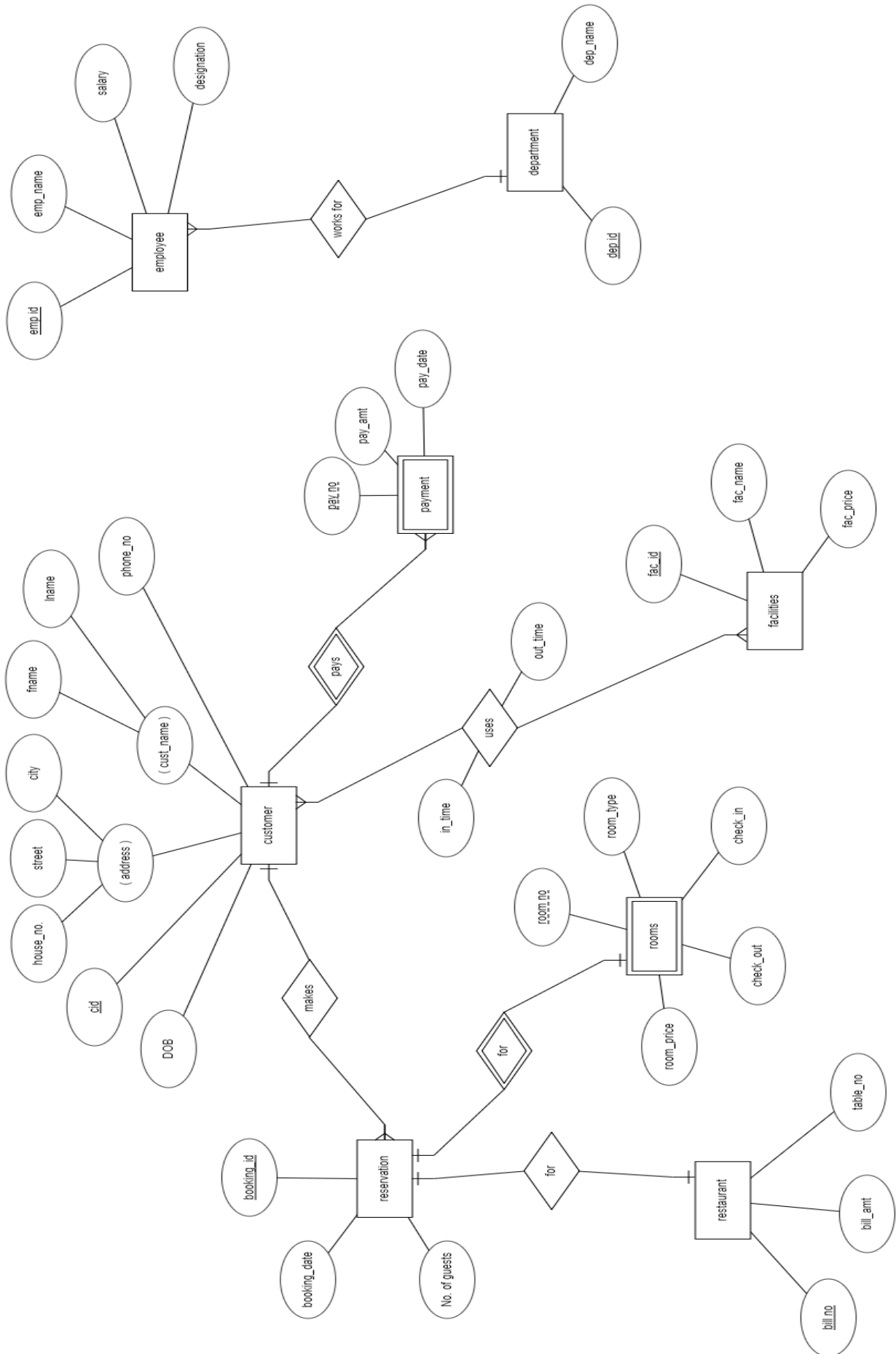
Hotel reservation is an ever-growing business, and it is fundamental in today's life. It automates the booking process for the customers, and hence errors are reduced. It also saves time and cuts costs, and all these points significantly increase the customer's satisfaction.

Basically, a hotel needs to maintain the record of guests and reserve rooms beforehand. Customers should be able to know the availability of the rooms on a particular date. They should be able to reserve the available rooms according to their need in advance. To make their stay comfortable, they should be provided with food and other services. The record of the food taken by each customer and the services availed by the customer should be kept. These records help in generating bills.

These services are provided to the customer through an hotel reservation app on the front-end. But, there are a lot of processes going on at the back-end of the app, and one of them is database management. It helps to build the structure of the application.

In this report, we have created a simple database structure, taking a textbook example of a hotel room reservation system to see how it works in brief.

# Entity Relationship Diagram



## **Relational Model**

- **customer** ( cid, fname, lname, phone\_no, house\_no, street, city, DOB )
- **payment** ( pay\_no, pay\_amt, pay\_date, cid\* )
- **reservation** ( booking\_id, booking\_date, No. of guests, cid\* )
- **rooms** ( room\_no, room\_type, price, check\_in, check\_out, booking\_id\* )
- **restaurant** ( bill\_no, bill\_amt, table\_no, booking\_id\* )
- **facilities** ( fac\_id, fac\_name, fac\_price )
- **cust\_uses\_facilities** ( cid\*, fac\_id\*, in\_time, out\_time )
- **department** ( dep\_id, dep\_name )
- **employee** ( emp\_id, emp\_name, salary, designation, dep\_id\* )

Here is a picture to highlight the underlines better:

- customer ( cid, fname, lname, phone\_no, house\_no, street, city, DOB )
- payment ( pay\_no, pay\_amt, pay\_date, cid\* )
- reservation ( booking\_id, booking\_date, No. of guests, cid\* )
- rooms ( room\_no, room\_type, price, check\_in, check\_out, booking\_id\* )
- restaurant ( bill\_no, bill\_amt, table\_no, booking\_id\* )
- facilities ( fac\_id, fac\_name, fac\_price )
- cust\_uses\_facilities ( cid\*, fac\_id\*, in\_time, out\_time )
- department ( dep\_id, dep\_name )
- employee ( emp\_id, emp\_name, salary, designation, dep\_id\* )

# **Components of Database Design**

## **Tables:**

1. customer
2. cust\_uses\_facilities
3. department
4. employee
5. facilities
6. payment
7. reservation
8. restaurant
9. rooms

## **Relationships:**

1. customer **1:N** payment
2. customer **1:N** reservation
3. customer **M:N** facilities
4. reservation **1:1** rooms
5. reservation **1:1** restaurant
6. employee **N:1** department

## Database:

```
-- Dumping database structure for hotel_reservation
CREATE DATABASE IF NOT EXISTS `hotel_reservation` /*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci */ /*!80016 DEFAULT ENCODING=utf8mb4 */
USE `hotel_reservation`;
```

## For table customer:

```
-- Dumping structure for table hotel_reservation.customer
CREATE TABLE IF NOT EXISTS `customer` (
  `cid` int NOT NULL AUTO_INCREMENT,
  `fname` varchar(50) NOT NULL DEFAULT '0',
  `lname` varchar(50) DEFAULT '0',
  `phone_no` varchar(15) NOT NULL DEFAULT '0',
  `house_no` int NOT NULL DEFAULT '0',
  `street` varchar(50) NOT NULL DEFAULT '0',
  `city` varchar(50) NOT NULL DEFAULT '0',
  `DOB` date NOT NULL,
  PRIMARY KEY (`cid`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Dumping data for table hotel_reservation.customer: ~10 rows (approximately)
/*!40000 ALTER TABLE `customer` DISABLE KEYS */;
INSERT INTO `customer` (`cid`, `fname`, `lname`, `phone_no`, `house_no`, `street`, `city`, `DOB`) VALUES
(1, 'Rahul', 'Seth', '9812567823', 102, 'Bandstand Promenade', 'Mumbai', '1994-04-03'),
(2, 'Kailash', 'Narang', '8445320679', 304, 'Broadway', 'Chennai', '1992-05-21'),
(3, 'Lila', 'Sachdev', '9125678340', 601, 'Lindsay', 'Kolkata', '1980-11-30'),
(4, 'Haimi', 'Nath', '9234561180', 702, 'Kadapa Feeder', 'Amravati', '1973-08-25'),
(5, 'Punit', 'Mehan', '8124508526', 304, 'Mahatma Gandhi', 'Hyderabad', '1985-10-12'),
(6, 'Sima', 'Pawar', '9045628143', 203, 'Kushabhau Thakre', 'Indore', '1998-12-09'),
(7, 'Aishwarya', 'Tripathi', '9734570225', 402, 'Avenue', 'Bangalore', '2001-03-13'),
(8, 'Ranjit', 'Din', '9876543210', 101, 'Ashok Marg', 'Lucknow', '1992-05-19'),
(9, 'Samir', 'Krishna', '9087645632', 504, 'Nai Sarak', 'Delhi', '1987-06-24'),
(10, 'Akshay', 'Oza', '8452067921', 603, 'Marine Drive', 'Mumbai', '2002-09-29');
/*!40000 ALTER TABLE `customer` ENABLE KEYS */;
```

| cid | fname     | lname    | phone_no   | house_no | street              | city      | DOB        |
|-----|-----------|----------|------------|----------|---------------------|-----------|------------|
| 1   | Rahul     | Seth     | 9812567823 | 102      | Bandstand Promenade | Mumbai    | 1994-04-03 |
| 2   | Kailash   | Narang   | 8445320679 | 304      | Broadway            | Chennai   | 1992-05-21 |
| 3   | Lila      | Sachdev  | 9125678340 | 601      | Lindsay             | Kolkata   | 1980-11-30 |
| 4   | Haimi     | Nath     | 9234561180 | 702      | Kadapa Feeder       | Amravati  | 1973-08-25 |
| 5   | Punit     | Mehan    | 8124508526 | 304      | Mahatma Gandhi      | Hyderabad | 1985-10-12 |
| 6   | Sima      | Pawar    | 9045628143 | 203      | Kushabhau Thakre    | Indore    | 1998-12-09 |
| 7   | Aishwarya | Tripathi | 9734570225 | 402      | Avenue              | Bangalore | 2001-03-13 |
| 8   | Ranjit    | Din      | 9876543210 | 101      | Ashok Marg          | Lucknow   | 1992-05-19 |
| 9   | Samir     | Krishna  | 9087645632 | 504      | Nai Sarak           | Delhi     | 1987-06-24 |
| 10  | Akshay    | Oza      | 8452067921 | 603      | Marine Drive        | Mumbai    | 2002-09-29 |

## For table cust\_uses\_facilities:

```
-- Dumping structure for table hotel_reservation.cust_uses_facilities
CREATE TABLE IF NOT EXISTS `cust_uses_facilities` (
  `cid` int NOT NULL,
  `fac_id` varchar(10) NOT NULL DEFAULT '',
  `in_time` datetime NOT NULL,
  `out_time` datetime NOT NULL,
  PRIMARY KEY (`cid`,`fac_id`),
  KEY `FK2_cust-fac` (`fac_id`),
  CONSTRAINT `FK2_cust-fac` FOREIGN KEY (`fac_id`) REFERENCES `facilities` (`fac_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `FK_cust-fac` FOREIGN KEY (`cid`) REFERENCES `customer` (`cid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Dumping data for table hotel_reservation.cust_uses_facilities: ~11 rows (approximately)
/*!40000 ALTER TABLE `cust_uses_facilities` DISABLE KEYS */;
INSERT INTO `cust_uses_facilities` (`cid`, `fac_id`, `in_time`, `out_time`) VALUES
(1, 'SW003', '2022-01-17 20:34:56', '2022-01-17 21:35:43'),
(4, 'SP002', '2022-02-22 16:00:06', '2022-02-22 17:00:10'),
(5, 'GY001', '2022-03-22 19:20:20', '2022-03-22 20:30:10'),
(5, 'SP002', '2022-03-23 15:00:02', '2022-03-23 16:00:16'),
(5, 'SW003', '2022-03-24 17:05:43', '2022-03-24 18:10:44'),
(7, 'SP002', '2022-03-28 14:00:09', '2022-03-28 15:00:56'),
(8, 'GY001', '2022-04-02 07:00:00', '2022-04-02 08:00:08'),
(8, 'SW003', '2022-04-02 18:00:52', '2022-04-02 19:03:25'),
(9, 'SW003', '2022-04-02 10:00:15', '2022-04-02 11:10:23'),
(10, 'GY001', '2022-04-04 08:10:00', '2022-04-04 09:15:31'),
(10, 'SW003', '2022-04-05 15:00:34', '2022-04-05 16:15:19');
/*!40000 ALTER TABLE `cust_uses_facilities` ENABLE KEYS */;
```


| cid | fac_id | in_time             | out_time            |
|-----|--------|---------------------|---------------------|
| 1   | SW003  | 2022-01-17 20:34:56 | 2022-01-17 21:35:43 |
| 4   | SP002  | 2022-02-22 16:00:06 | 2022-02-22 17:00:10 |
| 5   | GY001  | 2022-03-22 19:20:20 | 2022-03-22 20:30:10 |
| 5   | SP002  | 2022-03-23 15:00:02 | 2022-03-23 16:00:16 |
| 5   | SW003  | 2022-03-24 17:05:43 | 2022-03-24 18:10:44 |
| 7   | SP002  | 2022-03-28 14:00:09 | 2022-03-28 15:00:56 |
| 8   | GY001  | 2022-04-02 07:00:00 | 2022-04-02 08:00:08 |
| 8   | SW003  | 2022-04-02 18:00:52 | 2022-04-02 19:03:25 |
| 9   | SW003  | 2022-04-02 10:00:15 | 2022-04-02 11:10:23 |
| 10  | GY001  | 2022-04-04 08:10:00 | 2022-04-04 09:15:31 |
| 10  | SW003  | 2022-04-05 15:00:34 | 2022-04-05 16:15:19 |



### For table department:

```
-- Dumping structure for table hotel_reservation.department
CREATE TABLE IF NOT EXISTS `department` (
  `dep_id` varchar(10) NOT NULL DEFAULT '',
  `dep_name` varchar(50) NOT NULL DEFAULT '',
  PRIMARY KEY (`dep_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Dumping data for table hotel_reservation.department: ~8 rows (approximately)
/*!40000 ALTER TABLE `department` DISABLE KEYS */;
INSERT INTO `department` (`dep_id`, `dep_name`) VALUES
  ('d_01', 'Room_service'),
  ('d_02', 'Front_office'),
  ('d_03', 'Kitchen_dept'),
  ('d_04', 'Maintenance'),
  ('d_05', 'Security'),
  ('d_06', 'Human_resource'),
  ('d_07', 'Sales_marketing'),
  ('d_08', 'IT');
/*!40000 ALTER TABLE `department` ENABLE KEYS */;
```

| dep_id  | dep_name        |
|--|-----------------|
| d_01   | Room_service    |
| d_02   | Front_office    |
| d_03   | Kitchen_dept    |
| d_04   | Maintenance     |
| d_05   | Security        |
| d_06   | Human_resource  |
| d_07   | Sales_marketing |
| d_08   | IT              |

## For table employee:

```
-- Dumping structure for table hotel_reservation.employee
CREATE TABLE IF NOT EXISTS `employee` (
  `emp_id` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT '',
  `emp_name` varchar(50) NOT NULL DEFAULT '',
  `salary` int NOT NULL DEFAULT '0',
  `dep_id` varchar(10) NOT NULL DEFAULT '0',
  `designation` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`emp_id`),
  KEY `FK_employee_department` (`dep_id`),
  CONSTRAINT `FK_employee_department` FOREIGN KEY (`dep_id`) REFERENCES `department` (`dep_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Dumping data for table hotel_reservation.employee: ~15 rows (approximately)
/*!40000 ALTER TABLE `employee` DISABLE KEYS */;
INSERT INTO `employee` (`emp_id`, `emp_name`, `salary`, `dep_id`, `designation`) VALUES
  ('AS191', 'Bhakti Verma', 200000, 'd_03', 'Asschef'),
  ('AS192', 'Parvati Pramod', 200000, 'd_03', 'Asschef'),
  ('CH190', 'Harish Rawal', 500000, 'd_03', 'Headchef'),
  ('HR657', 'Shilpa Rao', 600000, 'd_06', 'HR'),
  ('HR658', 'Dilip Joshi', 650000, 'd_06', 'HR'),
  ('IT406', 'Manav D\'souza', 500000, 'd_08', 'Manager'),
  ('MN508', 'Viral Oza', 100000, 'd_04', 'Electrician'),
  ('MN509', 'Seema Patel', 120000, 'd_04', 'Technician'),
  ('MS801', 'Naman Gawde', 300000, 'd_07', 'Executive'),
  ('RC178', 'Aishwarya Sharma', 100000, 'd_02', 'Receptionist'),
  ('RC179', 'Sahil Sawant', 150000, 'd_02', 'Receptionist'),
  ('RS201', 'Ali Fazal', 150000, 'd_01', 'Attendant'),
  ('RS202', 'Anvi Kashyap', 150000, 'd_01', 'Attendant'),
  ('SC301', 'Bhushan Arora', 80000, 'd_05', 'Watchman'),
  ('SC302', 'Yuvraj Rajput', 90000, 'd_05', 'Watchman');
/*!40000 ALTER TABLE `employee` ENABLE KEYS */;
```

| emp_id | emp_name         | salary  | dep_id | designation  |
|--------|------------------|---------|--------|--------------|
| RS201  | Ali Fazal        | 150,000 | d_01   | Attendant    |
| RS202  | Anvi Kashyap     | 150,000 | d_01   | Attendant    |
| RC179  | Sahil Sawant     | 150,000 | d_02   | Receptionist |
| RC178  | Aishwarya Sharma | 100,000 | d_02   | Receptionist |
| CH190  | Harish Rawal     | 500,000 | d_03   | Headchef     |
| AS191  | Bhakti Verma     | 200,000 | d_03   | Asschef      |
| AS192  | Parvati Pramod   | 200,000 | d_03   | Asschef      |
| MN509  | Seema Patel      | 120,000 | d_04   | Technician   |
| MN508  | Viral Oza        | 100,000 | d_04   | Electrician  |
| SC302  | Yuvraj Rajput    | 90,000  | d_05   | Watchman     |
| SC301  | Bhushan Arora    | 80,000  | d_05   | Watchman     |
| HR658  | Dilip Joshi      | 650,000 | d_06   | HR           |
| HR657  | Shilpa Rao       | 600,000 | d_06   | HR           |
| MS801  | Naman Gawde      | 300,000 | d_07   | Executive    |
| IT406  | Manav D'souza    | 500,000 | d_08   | Manager      |

### For table facilities:

```
-- Dumping structure for table hotel_reservation.facilities
CREATE TABLE IF NOT EXISTS `facilities` (
  `fac_id` varchar(10) NOT NULL,
  `fac_name` varchar(20) NOT NULL,
  `fac_price` int NOT NULL DEFAULT '0',
  PRIMARY KEY (`fac_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```




```
-- Dumping data for table hotel_reservation.facilities: ~3 rows (approximately)
/*!40000 ALTER TABLE `facilities` DISABLE KEYS */;
INSERT INTO `facilities` (`fac_id`, `fac_name`, `fac_price`) VALUES
  ('GY001', 'Gym', 2500),
  ('SP002', 'SPA', 5000),
  ('SW003', 'Swimming Pool', 3000);
/*!40000 ALTER TABLE `facilities` ENABLE KEYS */;
```

| fac_id 🔑 ▲¹ | fac_name      | fac_price |  |
|-------------|---------------|-----------|--|
| GY001       | Gym           | 2,500     |  |
| SP002       | SPA           | 5,000     |  |
| SW003       | Swimming Pool | 3,000     |  |

### For table payment:

```
-- Dumping structure for table hotel_reservation.payment
CREATE TABLE IF NOT EXISTS `payment` (
  `pay_no` int NOT NULL,
  `pay_amt` decimal(10,2) NOT NULL DEFAULT '0.00',
  `pay_date` date NOT NULL,
  `cid` int NOT NULL,
  PRIMARY KEY (`cid`,`pay_no`),
  CONSTRAINT `FK__customer` FOREIGN KEY (`cid`) REFERENCES `customer` (`cid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```



```
-- Dumping data for table hotel_reservation.payment: ~10 rows (approximately)
/*!40000 ALTER TABLE `payment` DISABLE KEYS */;
INSERT INTO `payment` (`pay_no`, `pay_amt`, `pay_date`, `cid`) VALUES
  (1, 14000.00, '2022-01-02', 1),
  (1, 6500.00, '2022-01-15', 2),
  (1, 9500.00, '2022-02-03', 3),
  (1, 17000.00, '2022-02-10', 4),
  (2, 47500.00, '2022-03-09', 5),
  (1, 17000.00, '2022-03-11', 6),
  (2, 23000.00, '2022-03-20', 7),
  (2, 40000.00, '2022-03-24', 8),
  (1, 13500.00, '2022-03-24', 9),
  (3, 27000.00, '2022-03-25', 10);
/*!40000 ALTER TABLE `payment` ENABLE KEYS */;
```

| pay_no |  | pay_amt   | pay_date   | cid |   |
|--------|---|-----------|------------|-----|---|
| 1      |   | 14,000.00 | 2022-01-02 | 1   |   |
| 1      |   | 6,500.00  | 2022-01-15 | 2   |   |
| 1      |   | 9,500.00  | 2022-02-03 | 3   |   |
| 1      |   | 17,000.00 | 2022-02-10 | 4   |   |
| 2      |   | 47,500.00 | 2022-03-09 | 5   |   |
| 1      |   | 17,000.00 | 2022-03-11 | 6   |   |
| 2      |   | 23,000.00 | 2022-03-20 | 7   |   |
| 2      |   | 40,000.00 | 2022-03-24 | 8   |   |
| 1      |   | 13,500.00 | 2022-03-24 | 9   |   |
| 3      |   | 27,000.00 | 2022-03-25 | 10  |   |

## For table reservation:

```
-- Dumping structure for table hotel_reservation.reservation
CREATE TABLE IF NOT EXISTS `reservation` (
  `booking_id` varchar(10) NOT NULL,
  `booking_date` date NOT NULL,
  `No. of guests` int NOT NULL DEFAULT '0',
  `cid` int NOT NULL DEFAULT '0',
  PRIMARY KEY (`booking_id`),
  KEY `FK_cid` (`cid`),
  CONSTRAINT `FK_cid` FOREIGN KEY (`cid`) REFERENCES `customer` (`cid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Dumping data for table hotel_reservation.reservation: ~10 rows (approximately)
/*!40000 ALTER TABLE `reservation` DISABLE KEYS */;
INSERT INTO `reservation` (`booking_id`, `booking_date`, `No. of guests`, `cid`) VALUES
  ('22-0001', '2022-01-02', 4, 1),
  ('22-0002', '2022-01-15', 2, 2),
  ('22-0003', '2022-02-03', 3, 3),
  ('22-0004', '2022-02-10', 4, 4),
  ('22-0005', '2022-03-09', 6, 5),
  ('22-0006', '2022-03-11', 3, 6),
  ('22-0007', '2022-03-20', 5, 7),
  ('22-0008', '2022-03-24', 6, 8),
  ('22-0009', '2022-03-24', 3, 9),
  ('22-0010', '2022-03-25', 5, 10);
/*!40000 ALTER TABLE `reservation` ENABLE KEYS */;
```

| booking_id |  | booking_date | No. of guests | cid |  ▲ <sup>1</sup> |
|------------|---|--------------|---------------|-----|--|
| 22-0001    |   | 2022-01-02   | 4             | 1   |  |
| 22-0002    |   | 2022-01-15   | 2             | 2   |  |
| 22-0003    |   | 2022-02-03   | 3             | 3   |  |
| 22-0004    |   | 2022-02-10   | 4             | 4   |  |
| 22-0005    |   | 2022-03-09   | 6             | 5   |  |
| 22-0006    |   | 2022-03-11   | 3             | 6   |  |
| 22-0007    |   | 2022-03-20   | 5             | 7   |  |
| 22-0008    |   | 2022-03-24   | 6             | 8   |  |
| 22-0009    |   | 2022-03-24   | 3             | 9   |  |
| 22-0010    |   | 2022-03-25   | 5             | 10  |  |

### For table restaurant:

```
-- Dumping structure for table hotel_reservation.restaurant
CREATE TABLE IF NOT EXISTS `restaurant` (
  `bill_no` varchar(20) NOT NULL,
  `bill_amt` int NOT NULL DEFAULT '0',
  `table_no` int NOT NULL DEFAULT '0',
  `booking_id` varchar(10) NOT NULL DEFAULT '0',
  PRIMARY KEY (`bill_no`),
  KEY `FK_book_id2` (`booking_id`),
  CONSTRAINT `FK_book_id2` FOREIGN KEY (`booking_id`) REFERENCES `reservation` (`booking_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Dumping data for table hotel_reservation.restaurant: ~3 rows (approximately)
/*!40000 ALTER TABLE `restaurant` DISABLE KEYS */;
INSERT INTO `restaurant` (`bill_no`, `bill_amt`, `table_no`, `booking_id`) VALUES
  ('67415SAW639K62', 2000, 2, '22-0008'),
  ('78699ADH353TY8', 3000, 7, '22-0005'),
  ('90216FGD431BX9', 1500, 4, '22-0010');
/*!40000 ALTER TABLE `restaurant` ENABLE KEYS */;
```

| bill_no        | bill_amt | table_no | booking_id |
|----------------|----------|----------|------------|
| 67415SAW639K62 | 2,000    | 2        | 22-0008    |
| 78699ADH353TY8 | 3,000    | 7        | 22-0005    |
| 90216FGD431BX9 | 1,500    | 4        | 22-0010    |

## For table rooms:

```
-- Dumping structure for table hotel_reservation.rooms
CREATE TABLE IF NOT EXISTS `rooms` (
  `room_no` int NOT NULL,
  `room_type` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT '',
  `price` int NOT NULL,
  `check_in` datetime NOT NULL,
  `check_out` datetime NOT NULL,
  `booking_id` varchar(10) NOT NULL DEFAULT '',
  PRIMARY KEY (`room_no`,`booking_id`),
  KEY `FK_bookingid1` (`booking_id`),
  CONSTRAINT `FK_bookingid1` FOREIGN KEY (`booking_id`) REFERENCES `reservation` (`booking_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Dumping data for table hotel_reservation.rooms: ~10 rows (approximately)
/*!40000 ALTER TABLE `rooms` DISABLE KEYS */;
INSERT INTO `rooms` (`room_no`, `room_type`, `price`, `check_in`, `check_out`, `booking_id`) VALUES
(201, 'Standard room', 2000, '2022-01-25 11:30:05', '2022-01-27 20:45:17', '22-0002'),
(303, 'Triple room', 3000, '2022-04-01 20:10:19', '2022-04-03 19:28:01', '22-0009'),
(304, 'Triple room', 3000, '2022-02-14 10:45:45', '2022-02-16 16:20:33', '22-0003'),
(401, 'Quad room', 4000, '2022-02-20 12:45:02', '2022-02-23 14:05:48', '22-0004'),
(402, 'Quad room', 4000, '2022-01-15 11:48:22', '2022-01-18 17:27:23', '22-0001'),
(504, 'Deluxe room', 6000, '2022-03-22 12:15:42', '2022-03-24 18:26:20', '22-0006'),
(601, 'Penthouse Suite', 8000, '2022-03-27 15:02:34', '2022-03-29 17:30:51', '22-0007'),
(601, 'Penthouse Suite', 8000, '2022-04-03 09:24:09', '2022-04-06 16:50:49', '22-0010'),
(702, 'Presidential Suite', 10000, '2022-03-21 21:00:34', '2022-03-25 10:04:28', '22-0005'),
(703, 'Presidential Suite', 10000, '2022-04-01 14:40:15', '2022-04-04 17:29:36', '22-0008');
/*!40000 ALTER TABLE `rooms` ENABLE KEYS */;

/*!40101 SET SQL_MODE=IFNULL(@OLD_SQL_MODE, '') */;
/*!40014 SET FOREIGN_KEY_CHECKS=IFNULL(@OLD_FOREIGN_KEY_CHECKS, 1) */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40111 SET SQL_NOTES=IFNULL(@OLD_SQL_NOTES, 1) */;
```

| room_no 🔑 | room_type          | price ▼₢ | check_in            | check_out           | booking_id 🔑 ▲¹ |
|-----------|--------------------|----------|---------------------|---------------------|-----------------|
| 402       | Quad room          | 4,000    | 2022-01-15 11:48:22 | 2022-01-18 17:27:23 | 22-0001         |
| 201       | Standard room      | 2,000    | 2022-01-25 11:30:05 | 2022-01-27 20:45:17 | 22-0002         |
| 304       | Triple room        | 3,000    | 2022-02-14 10:45:45 | 2022-02-16 16:20:33 | 22-0003         |
| 401       | Quad room          | 4,000    | 2022-02-20 12:45:02 | 2022-02-23 14:05:48 | 22-0004         |
| 702       | Presidential Suite | 10,000   | 2022-03-21 21:00:34 | 2022-03-25 10:04:28 | 22-0005         |
| 504       | Deluxe room        | 6,000    | 2022-03-22 12:15:42 | 2022-03-24 18:26:20 | 22-0006         |
| 601       | Penthouse Suite    | 8,000    | 2022-03-27 15:02:34 | 2022-03-29 17:30:51 | 22-0007         |
| 703       | Presidential Suite | 10,000   | 2022-04-01 14:40:15 | 2022-04-04 17:29:36 | 22-0008         |
| 303       | Triple room        | 3,000    | 2022-04-01 20:10:19 | 2022-04-03 19:28:01 | 22-0009         |
| 601       | Penthouse Suite    | 8,000    | 2022-04-03 09:24:09 | 2022-04-06 16:50:49 | 22-0010         |

## Normalization

All the tables satisfy the condition of 4NF. So, there is no need for Normalization to be done.




# SQL QUERIES

1. This query is to find all the details of the customers who come from the city “Mumbai”.

```
1  SELECT * FROM customer WHERE city = 'Mumbai';
2
3  |
```

Output:

| customer (2r x 8c) |   |        |       |            |          |                     |        |            |  |
|--------------------|---|--------|-------|------------|----------|---------------------|--------|------------|--|
| cid                |  | fname  | lname | phone_no   | house_no | street              | city   | DOB        |  |
| 1                  |   | Rahul  | Seth  | 9812567823 | 102      | Bandstand Promenade | Mumbai | 1994-04-03 |  |
| 10                 |   | Akshay | Oza   | 8452067921 | 603      | Marine Drive        | Mumbai | 2002-09-29 |  |

2. This query is to find the age of the customers by subtracting their DOB from the current date.

```
1  SELECT fname, lname, DOB, year(CURDATE())-year(DOB) AS age
2  FROM customer;
3  |
```

Output:

| customer (10r × 4c) |          |            |     |
|---------------------|----------|------------|-----|
| fname               | lname    | DOB        | age |
| Rahul               | Seth     | 1994-04-03 | 28  |
| Kailash             | Narang   | 1992-05-21 | 30  |
| Lila                | Sachdev  | 1980-11-30 | 42  |
| Haimi               | Nath     | 1973-08-25 | 49  |
| Punit               | Mehan    | 1985-10-12 | 37  |
| Sima                | Pawar    | 1998-12-09 | 24  |
| Aishwarya           | Tripathi | 2001-03-13 | 21  |
| Ranjit              | Din      | 1992-05-19 | 30  |
| Samir               | Krishna  | 1987-06-24 | 35  |
| Akshay              | Oza      | 2002-09-29 | 20  |

3. This query is to find the average payment from all the customers.

```

1  SELECT AVG(pay_amt) AS Average_payment
2  FROM payment
3  ;

```

Output:

| payment (1r × 1c) |               |
|-------------------|---------------|
| Average_payment   |               |
|                   | 21,500.000000 |

4. This query is to find the sum of payments of all the customers.

```
1  SELECT SUM(pay_amt) AS Total_sum
2  FROM payment
3  ;
```

Output:

| payment (1r × 1c) |  |
|-------------------|--|
| Average_payment   |  |
| 215,000.00        |  |

5. This query is to find the details of the employees who have salaries equal or over 200000 ordered from highest to lowest salary.

```
1  SELECT emp_name, salary, designation FROM employee
2  WHERE salary >= 200000
3  ORDER BY salary DESC;
4
```

### Output:

| employee (7r x 3c) |         |             |  |
|--------------------|---------|-------------|--|
| emp_name           | salary  | designation |  |
| Dilip Joshi        | 650,000 | HR          |  |
| Shilpa Rao         | 600,000 | HR          |  |
| Harish Rawal       | 500,000 | Headchef    |  |
| Manav D'souza      | 500,000 | Manager     |  |
| Naman Gawde        | 300,000 | Executive   |  |
| Bhakti Verma       | 200,000 | Asschef     |  |
| Parvati Pramod     | 200,000 | Asschef     |  |

6. This query is to find the names of customers and their respective payments, ordered from highest to lowest. This is done by joining the customer and payment table.

```
1  SELECT fname, lname, pay_amt, pay_date FROM customer
2  JOIN payment
3  ON customer.cid = payment.cid
4  ORDER BY pay_amt DESC;
5  |
6
7
```

### Output:

| Result #1 (10r x 4c) |          |         |            |  |
|----------------------|----------|---------|------------|--|
| fname                | lname    | pay_amt | pay_date   |  |
| Punit                | Mehan    | 47,500  | 2022-03-09 |  |
| Ranjit               | Din      | 40,000  | 2022-03-24 |  |
| Akshay               | Oza      | 27,000  | 2022-03-25 |  |
| Aishwarya            | Tripathi | 23,000  | 2022-03-20 |  |
| Haimi                | Nath     | 17,000  | 2022-02-10 |  |
| Sima                 | Pawar    | 17,000  | 2022-03-11 |  |
| Rahul                | Seth     | 14,000  | 2022-01-02 |  |
| Samir                | Krishna  | 13,500  | 2022-03-24 |  |
| Lila                 | Sachdev  | 9,500   | 2022-02-03 |  |
| Kailash              | Narang   | 6,500   | 2022-01-15 |  |

7. This query is to find the customer id, which room they booked, along with the prices. This is done by joining the reservation and rooms table.

```

1  SELECT cid, room_type, price FROM reservation
2  JOIN rooms
3  WHERE reservation.booking_id = rooms.booking_id
4  ORDER BY price ASC;
5

```

Output:

| Result #1 (10r × 3c) |                    |        |
|----------------------|--------------------|--------|
| cid                  | room_type          | price  |
| 2                    | Standard room      | 2,000  |
| 3                    | Triple room        | 3,000  |
| 9                    | Triple room        | 3,000  |
| 1                    | Quad room          | 4,000  |
| 4                    | Quad room          | 4,000  |
| 6                    | Deluxe room        | 6,000  |
| 7                    | Penthouse Suite    | 8,000  |
| 10                   | Penthouse Suite    | 8,000  |
| 5                    | Presidential Suite | 10,000 |
| 8                    | Presidential Suite | 10,000 |

8. This query is to find the employee name, their department, designation and salary ( from highest to lowest). Group By is used to group the employees with the same designation. Department and employee tables are joined.

```

1  SELECT emp_name, dep_name, designation, salary FROM department
2  JOIN employee
3  ON department.dep_id = employee.dep_id
4  GROUP BY designation
5  ORDER BY salary DESC;
6  |

```

**Output:**

| Result #1 (10r × 4c) |                 |              |         |  |
|----------------------|-----------------|--------------|---------|--|
| emp_name             | dep_name        | designation  | salary  |  |
| Shilpa Rao           | Human_resource  | HR           | 600,000 |  |
| Harish Rawal         | Kitchen_dept    | Headchef     | 500,000 |  |
| Manav D'souza        | IT              | Manager      | 500,000 |  |
| Naman Gawde          | Sales_marketing | Executive    | 300,000 |  |
| Bhakti Verma         | Kitchen_dept    | Asschef      | 200,000 |  |
| Ali Fazal            | Room_service    | Attendant    | 150,000 |  |
| Seema Patel          | Maintenance     | Technician   | 120,000 |  |
| Aishwarya Sharma     | Front_office    | Receptionist | 100,000 |  |
| Viral Oza            | Maintenance     | Electrician  | 100,000 |  |
| Bhushan Arora        | Security        | Watchman     | 80,000  |  |


9. This query is to find the name and phone no. of the customers, and which facilities they are using (facility id). Customer and cust\_uses\_facilities tables are joined.

```

1  SELECT fname, lname, phone_no, fac_id FROM customer
2  JOIN cust_uses_facilities
3  ON customer.cid = cust_uses_facilities.cid;
4  |
5

```

**Output:**

| Result #1 (11r × 4c) |          |            |  |
|----------------------|----------|------------|--|
| fname                | lname    | phone_no   | fac_id  |
| Rahul                | Seth     | 9812567823 | SW003  |
| Haimi                | Nath     | 9234561180 | SP002  |
| Punit                | Mehan    | 8124508526 | GY001  |
| Punit                | Mehan    | 8124508526 | SP002  |
| Punit                | Mehan    | 8124508526 | SW003  |
| Aishwarya            | Tripathi | 9734570225 | SP002  |
| Ranjit               | Din      | 9876543210 | GY001  |
| Ranjit               | Din      | 9876543210 | SW003  |
| Samir                | Krishna  | 9087645632 | SW003  |
| Akshay               | Oza      | 8452067921 | GY001  |
| Akshay               | Oza      | 8452067921 | SW003  |

10. This query is to find the type of the customer, whether he/she is a new, familiar or a regular customer by checking their payment no. (1st payment, 2nd payment etc) . This is done by using a case statement and joining the customer and payment tables.

```

1  SELECT fname, lname, pay_no,
2  CASE
3      WHEN pay_no = 1 THEN 'New'
4      WHEN pay_no = 2 THEN 'Familiar'
5      WHEN pay_no > 2 THEN 'Regular'
6  END AS customer_type
7  FROM customer
8  JOIN payment
9  ON customer.cid = payment.cid;
10 |

```



### Output:

| Result #1 (10r x 4c) |          |        |               |
|----------------------|----------|--------|---------------|
| fname                | lname    | pay_no | customer_type |
| Rahul                | Seth     | 1      | New           |
| Kailash              | Narang   | 1      | New           |
| Lila                 | Sachdev  | 1      | New           |
| Haimi                | Nath     | 1      | New           |
| Punit                | Mehan    | 2      | Familiar      |
| Sima                 | Pawar    | 1      | New           |
| Aishwarya            | Tripathi | 2      | Familiar      |
| Ranjit               | Din      | 2      | Familiar      |
| Samir                | Krishna  | 1      | New           |
| Akshay               | Oza      | 3      | Regular       |

11. This query is to find which customers are to be given the “New Year’s Offer” and which to be not, based on whether they booked their reservation between 1st January to 31st January or not. This is done using a case statement and joining the customer and reservation tables.

```
1 SELECT fname, lname, booking_date,
2 CASE
3     WHEN booking_date BETWEEN '2022-01-01' AND '2022-01-31' THEN "New Year's Offer"
4     ELSE 'No Offers'
5 END AS offers
6 FROM customer
7 JOIN reservation
8 ON customer.cid = reservation.cid;
9 |
```

Output:

| Result #1 (10r × 4c) |          |              |                  |  |
|----------------------|----------|--------------|------------------|--|
| fname                | lname    | booking_date | offers           |  |
| Rahul                | Seth     | 2022-01-02   | New Year's Offer |  |
| Kailash              | Narang   | 2022-01-15   | New Year's Offer |  |
| Lila                 | Sachdev  | 2022-02-03   | No Offers        |  |
| Haimi                | Nath     | 2022-02-10   | No Offers        |  |
| Punit                | Mehan    | 2022-03-09   | No Offers        |  |
| Sima                 | Pawar    | 2022-03-11   | No Offers        |  |
| Aishwarya            | Tripathi | 2022-03-20   | No Offers        |  |
| Ranjit               | Din      | 2022-03-24   | No Offers        |  |
| Samir                | Krishna  | 2022-03-24   | No Offers        |  |
| Akshay               | Oza      | 2022-03-25   | No Offers        |  |

12. This query is to find the name of the customer, which facility he/she is using (facility name), along with its price(from highest to lowest). We used CONCAT to get the full name of the customer. And this time, we join 3 tables (customer, cust\_uses\_facilities and facilities).

```
1 SELECT CONCAT(fname,' ',lname) AS cust_name, fac_name, fac_price
2 FROM customer
3 INNER JOIN cust_uses_facilities
4     ON customer.cid = cust_uses_facilities.cid
5 INNER JOIN facilities
6     ON facilities.fac_id = cust_uses_facilities.fac_id
7 ORDER BY fac_price DESC;
8 |
```

Output:

| facilities (11r × 3c) |               |           |
|-----------------------|---------------|-----------|
| cust_name             | fac_name      | fac_price |
| Haimi Nath            | SPA           | 5,000     |
| Punit Mehan           | SPA           | 5,000     |
| Aishwarya Tripathi    | SPA           | 5,000     |
| Rahul Seth            | Swimming Pool | 3,000     |
| Punit Mehan           | Swimming Pool | 3,000     |
| Ranjit Din            | Swimming Pool | 3,000     |
| Samir Krishna         | Swimming Pool | 3,000     |
| Akshay Oza            | Swimming Pool | 3,000     |
| Punit Mehan           | Gym           | 2,500     |
| Ranjit Din            | Gym           | 2,500     |
| Akshay Oza            | Gym           | 2,500     |

13. This query is to find the customers who have made payment more than the average payment. The average payment is 21,500 (from query 3). We have used a sub-query to get the output, along with joining the customer table and the payment table.

```
1  SELECT CONCAT(fname,' ',lname) AS cust_name, pay_amt
2  FROM customer
3  JOIN payment
4  ON customer.cid = payment.cid
5  WHERE pay_amt >
6     (SELECT AVG(pay_amt) FROM payment);
7  |
```

Output:

| payment (4r × 2c)  |           |
|--------------------|-----------|
| cust_name          | pay_amt   |
| Punit Mehan        | 47,500.00 |
| Aishwarya Tripathi | 23,000.00 |
| Ranjit Din         | 40,000.00 |
| Akshay Oza         | 27,000.00 |