**GROUP: 82**

## Relational Schema

Room(Room_ID, Room_type, Room_floor, Room_NO)

RoomType(Room_type, Room_basePrice, facilities)

Customer(Customer_ID, fname, SName)

BookingCustomer(BookingCust_ID, Customer_ID, Booking_ID)

Booking(Booking_ID, Room_ID, date_in, date_out)

## Create Tables and Sample Test Data

Create tables and test data for each entity is given below. The sample test data has been created so that queries will work in most conditions.

```
DROP TABLE `Customer`;

CREATE TABLE Customer (
    Customer_ID int NOT NULL,
    fName varchar(255) NOT NULL,
    sName varchar(255) NOT NULL,
    PRIMARY KEY (Customer_ID)
);


INSERT INTO `Customer` (`Customer_ID`,`fName`,`sName`) VALUES
(1,"Kelly","Mendez"),(2,"Evelyn","Chaney"),(3,"Charlotte","Crane"),(4,
"Barbara","Morgan"),(5,"Jaime","Montoya"),(6,"Quyn","Atkinson"),(7,"Fa
tima","Lyons"),(8,"Noel","Tillman"),(9,"Cadman","Wood"),(10,"Derek","M
anning"),(11,"Pamela","Lamb"),(12,"Warren","Finley"),(13,"Wesley","How
ell"),(14,"Jin","Brewer"),(15,"Rowan","William"),(16,"Jelani","Cash"),
(17,"Eleanor","Burgess"),(18,"Reagan","Boone"),(19,"Gareth","Terry"),(
20,"Wayne","Knowles"),(21,"Amir","Becker"),(22,"Beck","Harris"),(23,"S
imon","Schmidt"),(24,"Shelly","Daugherty"),(25,"Regan","Little"),(26,"
Mara","Velasquez"),(27,"Aimee","Mendez"),(28,"Brett","Velasquez"),(29,
"Fuller","Ruiz"),(30,"Lysandra","Swanson"),(31,"Marny","Head"),(32,"Em
erson","Burt"),(33,"Marshall","Combs"),(34,"Nathan","Hayden"),(35,"Col
ton","Rojas"),(36,"Kevin","Hoover"),(37,"Clarke","Faulkner"),(38,"Wang
","Vasquez"),(39,"Dane","Carpenter"),(40,"Lareina","Kirkland"),(41,"Ly
nn","England"),(42,"Hillary","Hill"),(43,"Davis","Dyer"),(44,"Martin",
"Wolf"),(45,"Luke","Love"),(46,"August","Paul"),(47,"Tanek","Chen"),(4
8,"Cassady","Strong"),(49,"John","Knowles"),(50,"Logan","Wright"),(51,
"Isadora","Harvey"),(52,"Shoshana","Payne"),(53,"Julian","Roth"),(54,"
Wayne","Beach"),(55,"Octavius","Ward"),(56,"Britanni","Olsen"),(57,"Ge
raldine","Herman"),(58,"Latifah","Davenport"),(59,"Slade","Farmer"),(6
0,"Lyle","Stone"),(61,"Flynn","Atkinson"),(62,"Adrian","Tyler"),(63,"K
ellie","Spence"),(64,"Ivor","Stewart"),(65,"Kyle","Cohen"),(66,"Brande
```

```
n","Swanson"),(67,"Maggy","Meyer"),(68,"Nero","Patel"),(69,"McKenzie",
"Lindsey"),(70,"Serina","Hughes"),(71,"Zephania","Fields"),(72,"Kylee"
,"Taylor"),(73,"Hamilton","Pugh"),(74,"Josiah","Avila"),(75,"Selma","F
oreman"),(76,"Oleg","Young"),(77,"Cairo","Snider"),(78,"Kareem","Sosa"
),(79,"Jena","Mathis"),(80,"Vivien","Mcfarland"),(81,"Shad","Larsen"),
(82,"Brent","Foreman"),(83,"Nita","Mccullough"),(84,"Deirdre","Boyer")
,(85,"Myra","Ross"),(86,"Sean","Hurley"),(87,"Kyle","Oneal"),(88,"Leig
h","Oliver"),(89,"Eleanor","Waters"),(90,"Coby","Carney"),(91,"Daria",
"Cole"),(92,"Wyatt","Mills"),(93,"Joel","Harper"),(94,"Gil","Cotton"),
(95,"Simon","Woodard"),(96,"Tobias","Savage"),(97,"Garth","Rodriguez")
,(98,"Alika","Odonnell"),(99,"Neville","Prince"),(100,"Amery","Montoya
");

DROP TABLE `Room_Type`;

CREATE TABLE RoomType (
    Room_type varchar(255) NOT NULL,
    Room_basePrice float,
    facilitys text,
    PRIMARY KEY (Room_type)
);

INSERT INTO `RoomType` (`Room_type`,`Room_basePrice`,`facilitys`)
VALUES ("excellent","69.54","dui lectus rutrum urna,
nec"),("deluxe","69.54","dui lectus toilet urna
bath"),("magnificent","69.54","dui shower lily urna nec");

DROP TABLE `Room`;

CREATE TABLE Room (
    Room_ID int NOT NULL,
    Room_type varchar(255) NOT NULL,
    Room_floor int NOT NULL,
    Room_NO int NOT NULL,
    PRIMARY KEY (Room_ID),
    FOREIGN KEY (Room_type) REFERENCES RoomType (Room_type)
);


INSERT INTO `Room` (`Room_ID`,`Room_type`,`Room_floor`,`Room_NO`)
VALUES
("342","deluxe",3,42),("144","magnificent",1,44),("236","excellent",2,
36),("243","excellent",2,43),("218","excellent",2,18),("362","magnific
ent",3,62),("440","excellent",4,40),("350","magnificent",3,50),("196",
"magnificent",1,96),("260","deluxe",2,60),("290","magnificent",2,90),(
"183","excellent",1,83),("349","excellent",3,49),("427","deluxe",4,27)
,("485","magnificent",4,85),("215","excellent",2,15),("138","deluxe",1
,38),("323","magnificent",3,23),("399","deluxe",3,99),("227","deluxe",
2,27);

DROP TABLE `Booking`;
```

```
CREATE TABLE Booking (
    Booking_ID int NOT NULL,
    Room_ID int NOT NULL,
    date_in date NOT NULL,
    date_out date NOT NULL,
    PRIMARY KEY (Booking_ID),
    FOREIGN KEY (Room_ID) REFERENCES Room (Room_ID)
);

INSERT INTO `Booking` (`Booking_ID`,`Room_ID`,`date_in`,`date_out`)
VALUES (1,"196","17-03-03","17-03-27"),(2,"144","17-03-18","17-03-
27"),(3,"399","17-01-11","17-01-28"),(4,"243","17-02-08","17-02-
25"),(5,"227","17-03-18","17-03-30"),(6,"218","17-03-11","17-03-
28"),(7,"399","17-03-17","17-03-31"),(8,"349","17-03-12","17-03-
31"),(9,"342","17-01-19","17-01-31"),(10,"349","17-03-15","17-03-
28"),(11,"399","17-05-05","17-05-16"),(12,"144","17-03-17","17-03-
23"),(13,"342","17-03-06","17-03-29"),(14,"290","17-03-02","17-03-
24"),(15,"427","17-03-10","17-03-25"),(16,"243","17-07-05","17-07-
26"),(17,"485","17-03-08","17-03-24"),(18,"290","17-03-13","17-03-
23"),(19,"362","17-03-10","17-03-28"),(20,"236","17-09-04","17-09-
29"),(21,"236","18-03-22","18-03-29"),(22,"236","18-04-01","18-04-
08");

DROP TABLE `BookingCustomer`;

CREATE TABLE BookingCustomer (
    BookingCust_ID int NOT NULL,
    Customer_ID int NOT NULL,
    Booking_ID int NOT NULL,
    PRIMARY KEY (BookingCust_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID),
    FOREIGN KEY (Booking_ID) REFERENCES Booking(Booking_ID)
);

INSERT INTO `BookingCustomer`
(`BookingCust_ID`,`Customer_ID`,`Booking_ID`) VALUES
(1,98,7),(2,29,18),(3,49,1),(4,9,8),(5,62,1),(6,37,19),(7,24,13),(8,87
,9),(9,89,10),(10,30,19),(11,3,6),(12,37,2),(13,65,16),(14,11,15),(15,
77,19),(16,10,4),(17,95,1),(18,66,9),(19,72,11),(20,13,16),(21,54,13),
(22,91,1),(23,89,3),(24,56,11),(25,86,1),(26,17,4),(27,24,12),(28,25,2
),(29,89,4),(30,87,19),(31,98,21),(32,87,22);
```

## Create View Commands

```
/*shows empty rooms for today*/
CREATE VIEW empty_rooms AS
SELECT Room.Room_type, Room.Room_ID
FROM Room
INNER JOIN Booking ON Room.Room_ID = Booking.Room_ID
```

```
WHERE NOT (DATE(NOW()) BETWEEN Booking.date_in AND Booking.date_out);

/*shows current customers (people using rooms today)*/
CREATE VIEW list_cust AS
SELECT Customer.fName, Customer.sName, Room.Room_ID, Booking.date_in,
Booking.date_out
FROM Customer
INNER JOIN BookingCustomer ON Customer.Customer_ID =
BookingCustomer.Customer_ID
INNER JOIN Booking ON BookingCustomer.Booking_ID = Booking.Booking_ID
INNER JOIN Room ON Booking.Room_ID = Room.Room_ID
WHERE (DATE(NOW()) BETWEEN Booking.date_in AND Booking.date_out);

/*shows bookings for next 30 days*/
CREATE VIEW next_month AS
SELECT Customer.fName, Customer.sName, Booking.Booking_ID,
Booking.Room_ID, Booking.date_in, Booking.date_out
FROM Booking
INNER JOIN BookingCustomer ON Booking.Booking_ID =
BookingCustomer.Booking_ID
INNER JOIN Customer ON BookingCustomer.Customer_ID =
Customer.Customer_ID

WHERE (DATE(NOW()) BETWEEN Booking.date_in AND Booking.date_out) OR
(DATE(NOW() + INTERVAL 1 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 2 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 3 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 4 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 5 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 6 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 7 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 8 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 9 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 10 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 11 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 12 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 13 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 14 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
```

```
(DATE(NOW() + INTERVAL 15 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 16 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 17 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 18 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 19 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 20 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 21 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 22 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 23 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 24 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 25 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 26 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 27 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 28 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 29 DAY) BETWEEN Booking.date_in AND
Booking.date_out)OR
(DATE(NOW() + INTERVAL 30 DAY) BETWEEN Booking.date_in AND
Booking.date_out);
```

Below are the stored procedures needed for the main database.

```
/*check availability for given date of all rooms(procedure)*/
DROP PROCEDURE check_ava_all;
DELIMITER //
CREATE PROCEDURE check_ava_all(start_date DATE, end_date DATE)
BEGIN
select
 Room.Room_ID AS RoomsAvalible
from
 Room
 where not exists
  (select
     1
   from
     Booking
   where
      Room.Room_ID = Booking.Room_ID
```

```
      and (
        start_date between Booking.date_in and Booking.date_out
        or
        end_date between Booking.date_in and Booking.date_out
        ));
END//
DELIMITER ;


/*check avability for a Room_ID and given dates(procedure)*/
DROP PROCEDURE check_ava_room;
DELIMITER //
CREATE PROCEDURE check_ava_room(start_date DATE, end_date DATE,
room_ID int)
BEGIN
select
 Room.Room_ID AS RoomsAvalible
from
 Room
INNER JOIN Booking ON Room.Room_ID = Booking.Room_ID
   where
      Booking.Room_ID = room_ID
      and (
        start_date not between Booking.date_in and Booking.date_out
        or
        end_date not between Booking.date_in and Booking.date_out
        );

END//
DELIMITER ;


/*check if this room type got wanted facility(procedure)*/
DROP PROCEDURE check_fac;
DELIMITER //
CREATE PROCEDURE check_fac(room_type varchar(255), fac varchar(255))
BEGIN
SELECT DISTINCT Room_Type
FROM RoomType
WHERE
RoomType.Room_type = room_type
AND
facilitys LIKE fac;
END//
DELIMITER ;


/*show all the booking with given customer first name and
surname(procedure)*/
DROP PROCEDURE check_cust;
DELIMITER //
```

```
CREATE PROCEDURE check_cust(cust_fName varchar(255), cust_sName
varchar(255))
BEGIN
SELECT DISTINCT Booking.*
From Booking
INNER JOIN Room ON Booking.Room_ID = Room.Room_ID
INNER JOIN BookingCustomer ON Booking.Booking_ID =
BookingCustomer.Booking_ID
INNER JOIN Customer ON BookingCustomer.Customer_ID =
Customer.Customer_ID
WHERE Customer.fName = cust_fName
AND Customer.sName = cust_sName;
END//
DELIMITER ;


/* show all customers for booking with given ID(procedure)*/
DROP PROCEDURE check_custOfBooking;
DELIMITER //
CREATE PROCEDURE check_custOfBooking(book_ID int)
BEGIN
SELECT DISTINCT Customer.*
FROM Customer
INNER JOIN BookingCustomer ON Customer.Customer_ID =
BookingCustomer.Customer_ID
WHERE BookingCustomer.Booking_ID = book_ID;
END//
DELIMITER ;


/*check total price of booking by ID(procedure) */
DROP PROCEDURE check_price;
DELIMITER //
CREATE PROCEDURE check_price(book_ID int)
BEGIN
SET @cust_NO = NULL;
SELECT COUNT(*) INTO @cust_NO
FROM Customer
INNER JOIN BookingCustomer ON Customer.Customer_ID =
BookingCustomer.Customer_ID
WHERE BookingCustomer.Booking_ID = book_ID;
SELECT round(Room_basePrice * @cust_NO,2) AS Total_price
FROM RoomType
INNER JOIN Room ON Room.Room_type = RoomType.Room_type
INNER JOIN Booking ON Booking.Room_ID = Room.Room_ID
WHERE Booking.Booking_ID = book_ID;


END//
DELIMITER ;
```

## SQL Queries

Below are SQL queries of varying complexity, including canned queries.

```
/*what kind of rooms are available for today*/
SELECT DISTINCT Room_Type
FROM empty_rooms;

/*show room types with shower in it*/
SELECT DISTINCT Room_Type
FROM RoomType
WHERE facilitys LIKE "%shower%";

/*check if this room type got wanted facility(querry)*/
CALL check_fac("deluxe", "%urna%");

/*check availability for a Room_ID and given dates(querry)*/
CALL check_ava_room(DATE(NOW()),"2018-04-01",196);

/*check avability for given date of all rooms(querry)*/
CALL check_ava_all("2018-03-24","2018-03-24");

/*show all the booking with given customer first name and
surname(querry)*/
CALL check_cust("Shelly","Daugherty");

/* show all customers for booking with given ID(querry)*/
CALL check_custOfBooking(22);

/*check total price of booking by ID(querry) */
CALL check_price(11);

/*show all the bookings for rooms on second floor*/
SELECT Room.Room_NO, Booking.date_in, Booking.date_out
```

```
FROM Room
INNER JOIN Booking ON Room.Room_ID = Booking.Room_ID
WHERE Room.Room_floor = 2;

/*Show availability of magnificent room for today*/
SELECT *
FROM empty_rooms
WHERE Room_type = "magnificent";

/*Show price for deluxe room for 2 people*/
SELECT round(Room_basePrice * 2,2) AS Total_price
FROM RoomType
WHERE Room_type = "deluxe";

/*Show total price for 3 magnificent rooms with 2 person per each*/
SELECT round((Room_basePrice * 2)*3,2) AS Total_price
FROM RoomType
WHERE Room_type = "magnificent";

/*Shows what kind of facilities booked rooms in a booking ID 22
include*/
SELECT RoomType.facilitys
FROM Booking
INNER JOIN Room ON Booking.Room_ID = Room.Room_ID
INNER JOIN RoomType ON Room.Room_type = RoomType.Room_type
WHERE Booking.Booking_ID = 22;

/*Shows current customer in a Room ID 236*/
SELECT DISTINCT Customer.fName, Customer.sName
FROM Booking
INNER JOIN BookingCustomer ON Booking.Booking_ID =
BookingCustomer.Booking_ID
```

```
INNER JOIN Room ON Booking.Room_ID = 236

INNER JOIN Customer ON BookingCustomer.Customer_ID =
Customer.Customer_ID

WHERE (DATE(NOW()) BETWEEN Booking.date_in AND Booking.date_out);
```