

Lab 4**1. List the price and type of all rooms at the Palace Hotel.**

```
SELECT price,type FROM hotel h INNER JOIN room r on h.hotelNo = r.hotelNo
      where h.hotelName='palace';
```

2. List all guests currently staying at the Palace Hotel.

```
select g.guestName,g.guestNo from guest g inner join booking b on
g.guestNo=b.guestNo inner join hotel h on h.hotelNo=b.hotelNo where
h.hotelName='palace' and b.dateTo>=GETDATE() and b.dateFrom<=GETDATE();
```

3. List the details of all occupied rooms at the Palace Hotel today, including the name of the guest staying in the room.

```
SELECT r.*,g.guestName from booking b inner join room r on b.roomNo=r.roomNo
and b.hotelNo=r.hotelNo inner join guest g on b.guestNo=g.guestNo
inner join hotel h on r.hotelNo=h.hotelNo where
b.dateFrom <= GETDATE() AND b.dateTo>=GETDATE() AND h.hotelName='palace';
```

4. List the details of all unoccupied rooms at the Palace Hotel today.

```
(select r.*from room r inner join hotel h on r.hotelNo=h.hotelNo
where h.hotelName='palace') except (SELECT r.*
from booking b inner join room r
on b.roomNo=r.roomNo and b.hotelNo=r.hotelNo inner join hotel h
on r.hotelNo=h.hotelNo where
b.dateFrom <= GETDATE() AND b.dateTo>=GETDATE())
```

5. What is the lost income for each hotel? (use current date)

```
SELECT r.hotelno, SUM(price) as Lost FROM Room r WHERE r.roomno
NOT IN(SELECT b.roomNo FROM Booking b , Hotel h
WHERE (b.datefrom <= GETDATE()
AND b.dateto >= GETDATE())AND b.hotelNo = h.hotelno)
GROUP BY r.hotelno;
```

6. **List the hotel numbers, guest names, and guest cities for which more than one guest having the same address is staying at the same hotel. (Use today's date)**

Note: Here it is required to find out a hotel where 2 or more different guests are staying and these guests are coming from the same city.

Like, 2 guests from Fairfield are staying at a Palace hotel.

```
select b.hotelNo,g.guestName,g.guestAddress from Booking b inner join guest g
on b.guestNo=g.guestNo inner join hotel h on b.hotelNo=h.hotelNo
where g.guestAddress=h.city and (b.datefrom <= GETDATE()
AND b.dateto >= GETDATE()) group by b.hotelNo,g.guestName,g.guestAddress
having count(b.hotelNo)>1
```

7. **What is the most commonly booked room type for each hotel in Fairfield over the whole history of the hotel? Include the number of bookings in the report.**

```
(SELECT b.hotelNo AS hotelNo ,r.type AS type,count(r.type) AS myCOUNT
FROM booking b INNER JOIN room r ON b.hotelNo = r.hotelNo
AND b.roomNo =r.roomNo INNER JOIN hotel h ON h.hotelNo= b.hotelNo
WHERE h.city ='fairfield' GROUP BY r.type,b.hotelNo )
```

8. **Write SQL statements to Insert rows into each of these tables. (It'll suffice if you show only 2 rows insertion)**

```
insert into hotel values (9,'Marriot','DesMoines')
insert into Room values (10,11,'family',106)
insert into guest values (11,'John','Omaha')
```

9. **Update the price of every room in Palace Hotel by 5% more.**

```
update r set r.price=r.price+r.price*0.05 from room
as r inner join hotel h ON h.hotelNo= r.hotelNo
where h.hotelName='palace';
```

10. **Create a separate table with the same structure as the Booking table to hold archive records. Using INSERT statement, copy the records from the Booking table to the**

archive table relating to bookings before 1st January 2019. Delete all bookings before 1st January 2019 from the Booking table.

```
CREATE TABLE Archive( hotelNo nchar(10) ,guestNo nchar(10) ,
dateFrom DATETIME ,dateTo DATETIME, roomNo nchar(10),
PRIMARY KEY(hotelNo,guestNo,dateFrom),
FOREIGN KEY (hotelNo) REFERENCES hotel ,
FOREIGN KEY (guestNo) REFERENCES guest );
```

Command(s) completed successfully.

```
INSERT INTO Archive
SELECT b.hotelNo, b.guestNo, b.dateFrom,b.dateTo,b.roomNo
FROM booking b
WHERE b.dateFrom <= '2015-01-01 ';
```

(1 row(s) affected)

```
DELETE FROM booking WHERE dateFrom <= '2015-01-01';
(1 row(s) affected)
```