Guest House Assessment

2.When do they get here? List the arrival time and the first and last names for all guests due to arrive on 2016-11-05, order the output by time of arrival.

```
SELECT first_name,last_name,arrival_time
FROM booking JOIN guest ON guest_id=guest.id
WHERE booking_date='2016-11-05'ORDER by arrival_time
```

3.Look up daily rates. Give the daily rate that should be paid for bookings with ids 5152, 5165, 5154 and 5295. Include booking id, room type, number of occupants and the amount

```
Select booking.booking_id,booking.room_type_requested,booking.occupants,rate.amount from
booking
Join rate on (booking.room_type_requested=rate.room_type and
booking.occupants=rate.occupancy)
where booking.booking_id=5152 or booking.booking_id=5154 or booking.booking_id=5295
```

4.Who’s in 101? Find who is staying in room 101 on 2016-12-03, include first name, last name and address.

```
Select
guest.first_name,
guest.last_name,
guest.address
from guest
join booking
on (booking.guest_id = guest.id)where booking.room_no = 101 AND booking.booking_date =
'2016-12-03'
```

5.How many bookings, how many nights? For guests 1185 and 1270 show the number of bookings made and the total number of nights. Your output should include the guest id and the total number of bookings and the total number of nights.

```
select
guest_id ,COUNT(nights),SUM(nights)
from booking
```

```
where guest_id=1185 or guest_id=1270
group by guest_id
```

6.Ruth Cadbury. Show the total amount payable by guest Ruth Cadbury for her room bookings. You should JOIN to the rate table using room_type_requested and occupants.

```
Select sum(booking.nights*rate.amount)
From  booking JOIN rate
On (booking.occupants = rate.occupancy and
booking.room_type_requested=rate.room_type)
join guest
on(booking.guest_id=guest.id)
where
guest.first_name = 'Ruth'
and guest.last_name = 'Cadbury';
```

7.Including Extras. Calculate the total bill for booking 5346 including extras

```
SELECT
sum(booking.nights * rate.amount)+sum(e.amount)
FROM booking
join rate
on(booking.occupants = rate.occupancy
and booking.room_type_requested = rate.room_type)
JOIN
(SELECT booking_id,sum(amount)
FROM extra
group by booking_id)
                    AS e
                    ON (e.booking_id = booking.booking_id)
WHERE
        booking.booking_id = 5346;
```

8.Edinburgh Residents. For every guest who has the word “Edinburgh” in their address show the total number of nights booked. Be sure to include 0 for those guests who have never had a booking. Show last name, first name, address and number of nights. Order by last name then first name.

```
SELECT
guest.last_name,
guest.first_name,
guest.address,
```

```
CASE
WHEN SUM(booking.nights) IS NULL
THE     0
ELSE    SUM(booking.nights)
        END
        AS nights
from booking
        right join
                guest
                ON (guest.id = booking.guest_id)
where
        guest.address LIKE '%Edinburgh%'
Group by
        guest.last_name, guest.first_name, guest.address
Order by
        guest.last_name, guest.first_name
```

9.How busy are we? For each day of the week beginning 2016-11-25 show the number of bookings starting that day. Be sure to show all the days of the week in the correct order.

```
select booking_date as i ,count(booking_id) as arrivals
from booking
where booking_date between '2016-11-25' and  '2016-12-01'
group by
booking_date
```

10.How many guests? Show the number of guests in the hotel on the night of 2016-11-21. Include all occupants who checked in that day but not those who checked out.

```
select SUM(occupants)
from booking
where booking_date <= '2016-11-21'and DATE_ADD(booking_date, INTERVAL nights DAY) >
'2016-11-21'
```

11.Coincidence. Have two guests with the same surname ever stayed in the hotel on the evening? Show the last name and both first names. Do not include duplicates.

```
select distinct
        a.last_name,
        a.first_name,
        b.first_name
From(Select * From booking  JOIN guest
ON (booking.guest_id = guest.id))
        as a
        JOIN(Select     * From booking JOIN guest on(booking.guest_id = guest.id))
```

```
        As b
        on(a.last_name = b.last_name)
        and a.booking_date <= b.booking_date
        and DATE_ADD(a.booking_date, INTERVAL (a.nights - 1) DAY) >= b.booking_date
        and a.first_name > b.first_name
ORDER BY
b.last_name;
```

12.Check out per floor. The first digit of the room number indicates the floor – e.g. room 201 is on the 2nd floor. For each day of the week beginning 2016-11-14 show how many rooms are being vacated that day by floor number. Show all days in the correct order.

```
SELECT
  DATE_ADD(booking_date, INTERVAL nights DAY) AS i,
  SUM(CASE WHEN room_no LIKE '1%' THEN 1 ELSE 0 END) AS 1st,
  SUM(CASE WHEN room_no LIKE '2%' THEN 1 ELSE 0 END) AS 2nd,
  SUM(CASE WHEN room_no LIKE '3%' THEN 1 ELSE 0 END) AS 3rd
FROM booking WHERE

DATE_ADD(booking_date, INTERVAL nights DAY) BETWEEN '2016-11-14' AND '2016-11-20'

GROUP BY
```

13.Free rooms? List the rooms that are free on the day 25th Nov 2016

```
SELECT room.id
FROM room
EXCEPT
SELECT room.id
FROM room
JOIN booking
ON(booking.room_no = room.id)
WHERE booking.booking_date <= '2016-11-25' AND
DATE_ADD(booking.booking_date,INTERVAL booking.nights DAY)
   > '2016-11-25'
```

14.Single room for three nights required. A customer wants a single room for three consecutive nights. Find the first available date in December 2016.

```
SELECT ttt.free_room as id, ttt.checkout as MIN
FROM
(
SELECT tt.id, tt.room_type, tt.booking_date, tt.closest_booking, tt.checkout
,(CASE WHEN TIMESTAMPDIFF(DAY,tt.checkout,tt.closest_booking) > 3 OR tt.closest_booking IS
NULL THEN tt.id END)free_room
FROM
```

```
(
SELECT r.id, r.room_type, t.booking_date, t.closest_booking, t.checkout
FROM room r
LEFT JOIN
(
SELECT b.room_no, b.room_type_requested, b.booking_date, DATE_ADD(b.booking_date,
INTERVAL b.nights DAY) checkout,
	LEAD(b.booking_date, 1) OVER (
	PARTITION BY b.room_no
	ORDER BY b.booking_date
	) closest_booking
FROM room r JOIN booking b
ON (r.id = b.room_no)
WHERE YEAR(b.booking_date) = '2016' AND MONTH(b.booking_date) = '12' AND
b.room_type_requested = 'single'
ORDER BY b.booking_date
)t
ON (r.id = t.room_no)
WHERE t.room_no IS NOT NULL
ORDER BY t.booking_date
)tt
)ttt
WHERE ttt.free_room IS NOT NULL
ORDER BY ttt.booking_date ASC
LIMIT 1
```

15.Gross income by week. Money is collected from guests when they leave. For each Thursday in November and December 2016, show the total amount of money collected from the previous Friday to that day, inclusive.

```
SELECT
	DATE_ADD(MAKEDATE(2016, 7), INTERVAL WEEK(DATE_ADD(booking.booking_date,
INTERVAL booking.nights - 5 DAY), 0) WEEK) AS Thursday,
	SUM(booking.nights * rate.amount) + SUM(e.amount) AS Weekly_income
From booking JOIN rate
	on(booking.occupants = rate.occupancy
	and booking.room_type_requested = rate.room_type)
	Left join (Select booking_id,SUM(amount) as amount
	From extra
	group by booking_id)
		AS e
		ON (e.booking_id = booking.booking_id)
GROUP BY
	Thursday;
```