Class test 1  
(Model solutions and comments)

(i) Find the total number of guests currently staying in the hotel.  

15%

SELECT sum(numguests)
FROM booking
WHERE active = true

Note that it is not enough to count the active bookings (or the customers that made those bookings).
It is, however, possible to count the number of guests from the guests table, but we would need to refer to the bookings table to find whether the guests are in active bookings.

SELECT count(cid)
FROM guests
WHERE bookid IN (SELECT bookid
    FROM booking
    WHERE active = true)

One could also use a multiple table query instead of a nested query:

SELECT count(guests.cid)
FROM guests, booking
WHERE guests.bookid = bookings.bookid
    AND bookings.active = true

(ii) Find the names of guests currently staying in the hotel, ordered by the last name.  

20%

This is a minor twist on part (i), asking for the actual guest names, which must be obtained from the customer table.

SELECT customer.firstname, customer.lastname
FROM guests, booking, customer
WHERE guests.bookid = bookings.bookid
    AND bookings.active = true
    AND guests.cid = customer.cid
ORDER BY customer.lastname

An equivalent solution using nested queries requires two levels of nesting:

SELECT firstname, lastname
FROM customer
WHERE cid IN (SELECT cid
    FROM guests
    WHERE bookid IN (SELECT bookid
        FROM booking
        WHERE active = true))

(iii) List all the customers, by their cid, that ever made a booking for other guests, without themselves staying as part of the booking.  

20%

If a customer made a booking, there is a booking record for it. We want to see if the customer is not included among the guests for that booking.
SELECT cid
FROM booking
WHERE cid NOT IN (SELECT guests.cid
FROM guests
WHERE guests.bookid = booking.bookid)

The following attempt to write it as a multiple table query is wrong:

SELECT booking.cid
FROM booking, guests
WHERE guests.bookid = booking.bookid
AND guests.cid <> booking.cid

If the customer stayed with another guest, then there would be a guests record for that guest, which satisfies the condition guests.cid <> booking.cid. But that does not mean that this customer did not stay him/herself along with the guest. It would be wrong to include the booking customer among the results for just that reason.

Some people misinterpreted the question to mean find customers that made a booking but never stayed in the hotel. (This is different from the problem in the question paper, because it includes customers who might have made any one booking in which they were not themselves a guest. It is possible that those customers stayed as guests in other bookings.) The misinterpreted question is much easier to solve, and hence those solutions received only partial credit.

(iv) Find the numbers of rooms booked for today, listed by room type.

This is an easy question, once you realize that the results have to be grouped by rtype.

SELECT rtype, sum(rcount)
FROM booking
WHERE booking.startdate <= current_date
AND booking.startdate + numdays > current_date
GROUP BY rtype

Note that it is not enough to simply count bookings, because a booking might involve multiple rooms. We have to add up the rooms booked.

Most people had difficulty deciding whether a booking is current for today. Sample wrong answers were:

WHERE booking.startdate = current_date

This ignores the case that a booking that started yesterday might last two or more days. Another wrong answer that is “close” to the correct one is

WHERE booking.startdate <= current_date
AND booking.startdate + numdays >= current_date

If a booking started yesterday with numdays equal to 1, then it ends today, which means that those rooms are available for today. But the condition above assumes that it is booked for today.

(v) Find the names of booking customers who have stayed as single guests at some time, but with accompanying guests at another time.

Solution 1: The best solution for this question involves two copies of the booking table corresponding to the “solo” booking and “accompanied” booking, both of which need to be examined. Let us first write the query listing only the customer id’s.

SELECT cid
FROM booking as solo, booking as accompanied
WHERE solo.cid = accompanied.cid
AND solo.numguests = 1
AND accompanied.numguests > 1
This solution is almost right, but it doesn’t check whether the customer is actually staying in the hotel as part of the “solo” and “accompanied” bookings. (The bookings could be for entirely separate guests.) This should be checked by consulting the guests table. An innovative way of writing the conditions for such checking, provided by some students, is

\[
\text{AND (solo.cid, solo.bookid) IN (SELECT (cid, bookid) FROM guests)}
\]

\[
\text{AND (accompanied.cid, accompanied.bookid) IN (SELECT (cid, bookid) FROM guests)}
\]

These clever conditions check that the customer id, booking combination is in guests, which means that the customer stayed as a guest in the particular booking under consideration. A more common way of writing the conditions is:

\[
\text{AND solo.cid IN (SELECT cid FROM guests WHERE guests.bookid = solo.bookid)}
\]

\[
\text{AND accompanied.cid IN (SELECT cid FROM guests WHERE guests.bookid = accompanied.bookid)}
\]

Either way of writing them would do fine.

**Solution 2:** A quite different way of writing the basic solution is to use \text{INTERSECT}:

\[
\text{SELECT cid FROM booking WHERE numguests = 1 INTERSECT SELECT cid FROM booking WHERE numguests > 1}
\]

For this solution, it is quite easy to add the guests check (no need for nested queries):

\[
\text{SELECT cid FROM booking, guests WHERE numguests = 1 AND booking.bookid = guests.bookid INTERSECT SELECT cid FROM booking WHERE numguests > 1 AND booking.bookid = guests.bookid}
\]

So far, we have only found the cid’s of the customers that satisfy the condition in the question. The names of customers can be added just the same way as in question (ii).

**Additional practice problems**

Here are some more problems in this domain for practising SQL.

a. Find the names of guests who have rooms booked for today but are not checked in.

b. Find the names of guests who are expected to check out (depart) today.

c. List all the booking customers in the database along with the total number of different accompanying guests they have stayed with.

d. Find the number of vacant rooms for tomorrow, listed by room type.

e. Find the names of customers who have booked the largest number of “room days” during 2010. (A “room day” is one room booked for one day.)