DB – Assignment – 4

[solution]

Question 1:

1. This will produce a relation with a single attribute (hotelNo) giving the number of those hotels with a room price greater than £50.
2. This will produce a join of the Hotel and Room relations containing all the attributes of both Hotel and Room (there will be two copies of the hotelNo attribute). Essentially this will produce a relation containing all rooms at all hotels.
3. This will produce a join of Hotel and those tuples of Room with a price greater than £50. Essentially this will produce a relation containing all hotel names with a room price above £50.
4. This will produce a relation containing the names of all guests who have booked all hotels in London.

Question 2:

1. Hotel
2. σtype=‘S’ ∧ price < 20(Room)
3. ΠguestName, guestAddress(Guest)
4. Πprice, type(Room hotelNo (σhotelName = ‘Grosvenor Hotel’(Hotel)))
5. Guest guestNo (σdateFrom ≤ ‘01-01-15’ ∧ dateTo ≥ ‘01-01-15’ (

Booking hotelNo (σhotelName = ‘Grosvenor Hotel’(Hotel))))

(substitute ‘01-01-15’ for today’s date).

1. (Room hotelNo (σhotelName = ‘Grosvenor Hotel’(Hotel))) // Outer Join

ΠguestName, hotelNo, roomNo(

(Guest guestNo (σdateFrom ≤ ‘01-01-15’ ∧ dateTo ≥  ‘01-01-15’ (

Booking hotelNo (σhotelName=‘Grosvenor Hotel’(Hotel))))

(substitute ‘01-01-15’ for today’s date).

1. ΠguestNo, guestName, guestAddress(Guest guestNo (σdateFrom ≤ ‘01-01-15’ ∧ dateTo ≥ ‘01-01-15’ (

Booking hotelNo (σhotelName=‘Grosvenor Hotel’(Hotel)))))

(substitute ‘01-01-15’ for today’s date).

1. ΠroomNo, hotelNo, type(Room hotelNo (σhotelName=‘Grosvenor Hotel’ (Hotel)))

Security - hides the price details from people who should not see it.

Reduced complexity - a query against this view is simpler than a query against the two underlying base relations.