

DATABASE ASSIGNMENT #04

(a) $\pi_{\text{hotelNo}} (\sigma_{\text{price} > 50} (\text{Room}))$

This will produce a relation with single attribute hotelNo giving the number of those hotels where price is > 50 .

(b) $\sigma_{\text{hotel.hotelNo} = \text{Room.hotelNo}} (\text{Hotel} \bowtie \text{Room})$

It will produce a relation containing all the attributes of both the table because it's a join between these 2 tables.

(c) $\pi_{\text{hotelName}} (\text{Hotel} \bowtie_{\text{hotel.hotelNo} = \text{Room.hotelNo}} (\sigma_{\text{price} > 50} (\text{Room})))$

It will produce a relation with single attribute hotelName having rows from both the tables after joins and room will only join with those rows which after filtering where price of room in hotel > 50 .

(d) $\pi_{\text{guestName, hotelNo}} (\text{Booking} \bowtie_{\text{Booking.guestNo} = \text{Guest.guestNo}} (\text{Guest})) \div \pi_{\text{hotelNo}} (\sigma_{\text{City} = \text{'London'}} (\text{Hotel}))$

This will produce a relation with two attributes guestName and hotelNo and will keep a record of all guest names who have booked a hotel in London.

Question #2

1) List all hotels.

π hotel (Hotel) or π hotelNo, hotelName, city (Hotel)

2) List all single rooms with a price below £20 per night
O type = 'S' and price < 20 (Room)

3) List the names and cities of all guests.

π guestName, guestAddress (Guest)

4) List the price and type of all rooms at the Grosvenor hotel

~~π price, type (Room \int hotelNo (σ hotelName = 'Grosvenor' (Hotel)))~~

π price, type (Room \int hotelNo (σ hotelName = 'Grosvenor' (Hotel)))

5) List all guests currently staying at Grosvenor hotel

π guestNo, ~~guestName~~ (Booking \int hotelNo (σ hotelName = 'Grosvenor' (Hotel)))

6) List the details of all rooms at Grosvenor hotel including the name of the guest staying in the room, if the room is occupied

Rooms \int hotelNo (σ hotelName = 'Grosvenor' (Hotel)) \bowtie π hotelNo = hotelNo and roomNo = roomNo
 π guestName, hotelNo, roomNo (Guest \int guestNo (Booking \int hotelNo (σ hotelName = 'Grosvenor' (Hotel))))

g) List the details (guestNo, guestName, and guestAddress) of all guests staying at gussvenor hotel.

Π guestNo, guestName, guestAddress ($\sigma_{\text{hotelName} = \text{'Gussvenor'}}$ (Hotel \bowtie Booking \bowtie guestNo = guestNo (Guest)))

h) create view of all rooms in Gussvenor hotel, excluding price details.

Π roomNo, hotelNo, type ($\sigma_{\text{hotelName} = \text{'Gussvenor'}}$ (Hotel \bowtie Booking \bowtie roomNo = roomNo (Rooms)))

Advantages = security: hides price details from people who should not see it.

Reduced complexity: a query against view is simpler than query against 2 underlying relations.