Tim Capehart

CST324

Assignment 1

Exercises:

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

1. Foreign Keys: Room(hotelNo), Booking(hotelNo), Booking(guestNo, roomNo)

For entity integrity, the attributes of the hotels and bookings with hotelNo and the bookings with guestNo must not be null.

Referential integrity is maintained by insuring that the foreign keys used in tables have a candidate key to match with in the home relation or that the foreign key is null.

|  |  |  |  |
| --- | --- | --- | --- |
| Room |  |  |  |
| roomNo | hotelNo | Type | Price |
| 101 | 1 | Single | 100.00 |
| 102 | 1 | Suite | 200.00 |
| 201 | 2 | Single | 130.00 |
| 202 | 2 | Single | 140.00 |
| 203 | 2 | Suite | 300.00 |

2.

|  |  |  |
| --- | --- | --- |
| Hotel |  |  |
| hotelNo | hotelName | City |
| 1 | Hilton | New York |
| 2 | Trump | Las Vegas |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Booking |  |  |  |  |
| hotelNo | guestNo | dateFrom | dateTo | roomNo |
| 1 | 234 | 11-12-18 | 11-14-18 | 101 |
| 1 | 235 | 11-12-18 | 11-18-18 | 102 |
| 2 | 987 | 10-18-18 | 10-22-18 | 201 |
| 2 | 456 | 10-02-18 | 10-12-18 | 202 |
| 2 | 346 | 11-20-18 | 11-22-18 | 203 |

|  |  |  |
| --- | --- | --- |
| Guest |  |  |
| guestNo | guestName | guestAddress |
| 234 | Tim | 123 Main St |
| 235 | Joe | 345 Wall St |
| 987 | Fred | 484 Park Av |
| 456 | Calvin | 984 Park St |
| 346 | Hobbes | 38 Main Av |

General Constraints: Date format 00-00-00, Max # of chars for names and addresses.

3.

5. a) All hotel numbers that have rooms that cost more than 50.

b) All hotel numbers that are on both hotel table and room table

c) All hotel names that have rooms that cost more than 50.

d) Complete details of guests that are leaving before Jan 1st 2007.

e) Complete details of hotels that have rooms that cost more than 50.

f) Names of guests and hotel numbers that are booked in London.

6. a) TRC: { R.hotelNo | Room(R) ^ price < 50 ) }

DRC: { rn | R(Room(rn, hn, t, p) ^ (p > 50)) }

b) TRC: { h, r | (Hotel(h) ^  r (Room(r)) ^ h.hotelNo = r.hotelNo) }

DRC: { N | N(Hotel(hn1, hname, c) ^ N(Room(rn, hn2, t, p) ^ (hn1 = hn2) }

c) TRC: { n | (Hotel(n) ^ (Room(n) ^ n.price > 50)) }

DRC: { N |  N(Hotel(hn, hname, c) ^ N(Room(rn, hn, t, p) ^ p > 50)) }

d) TRC: { G | Guest(G) ^ (B)(Booking(B) ^ (G.guestNo = B.guestNo) ^ B.dateTo < 1-Jan-2007 ) }

DRC: { G |  G(Guest(gno, gna, gad)^( B | B(Booking(hno, gno, df, dt, rno) ^

dt < ‘1-Jan-2007) }

e) TRC: { H | Hotel(H) ^ (R)(Room(R) ^( (H.hotelNo = R.hotelNo) ^ R.price > 50) }

DRC: { H | H(Hotel(hno, hna, c) ^ (R | R(Room(hno, rno, t, p) ^ p > 50)) }

f) TRC: { G.guestName | (Guest(G) ^ Hotel(H) ^ (H.ciry = ‘London’)) ^ (B | Booking(B)) ^ (B ^ (G.guestNo = B.guestNo))) }

DRC: { guestName | (gname, gno1, hotelno1, df, dt, gno2, gadd, hotelno2, hotelname, city)(Hotel (hotelno2, hotelname, city) ^ (city = ‘London’)) ^ (Guest(gno1, gname, gadd) ^ (Booking(hotelno2, gno2, df, dt, rno) ^ (gno1 = gno2) ^ (hotelno1 = hotelno2))) }

7. a) Hotel names of all hotels in London

b) Hotel names that have a room that costs more than 50.

c) Hotel names of hotels where John Smith has a booking.

d) Hotel name, guest name, from dates for guest that has multiple bookings with different start dates at the same hotel.

8. a) RA: PI HotelName (sigma city = ‘London’)

DRC: { hotelName | Hotel(hotelName, hotelNumber, city) ^ (city = ‘London’)) }

b) RA: PI hotelName (Hotel >< Hotel.hotelNo = Room.hotelNo (sigma price > 50(Room))) }

DRC: { hotelName | hotelName, hotelNo1, city, roomNo, hotelNo2, type, price)(Hotel(hotelName, hotelNo1, city)^(Room(roomNo, hotelNo2, type, price)^(hotelNo1 = hotelNo2) ^ (price > 50) ) }

c) RA: PI hotelName (Hotel >< Hotel.hotelNo = Booking.hotelNo (Booking) ^ (Booking ><  Booking.guestNo = Guest.guestNo (sigma guestName = ‘John Smith’(Guest)) }

9. a) RA: Hotel

TRC: { H | Hotel(H) }

DRC: { H | (H(Hotel (hotelNo, hotelName, city)) }

b) RA: type = ‘Single’ ^ price < 20(Room)

TRC: { R | Room(R) ^ R.type = ‘Single’ ^ R.price < 20 }

DRC: { R | ((R(Room(roomNo, hotelNo, type, price)) ^ type = ‘Single’ ^ price < 20 }

c) RA: guestName, city(GuestGuest.guestNo = Booking.guestNo(Booking Booking.hotelNo = Hotel.hotelNo(otel))

TRC: { G.guestName, H.city | Guest(G) ^ (H) Hotel(H) ^ (B) Booking(B) ^ G.guestNo = B.guestNo ^ B.hotelNo = H.hotelNo }

DRC: { guestName, city | (Guest(guestNo1, guestName, guestAddress) ^ (Hotel(hotelNo1, hotelName, city)) ^ (Booking(hotelNo2, guestNo2, dateFrom, dateTo, roomNo) ^ guestNo1 = guestNo2 ^ hotelNo1 = hotelNo2 ) }

d) RA: price, type(RoomRoom.hotelNo = Hotel.hotelNo(hoteName = ‘Grosvenor Hotel’(Hotel))

TRC: { R.price, R.type | Room(R) ^ (H) Hotel(H) ^ R.hotelNo = H.hotelNo ^ H.hotelName = ‘Grosvenor Hotel’ )}

DRC: { price, type | Room(roomNo, hotelNo1, type, price) ^ Hotel(hotelNo2, hotelName, city) ^ hotelNo1 = hotelNo2 ^ hotelName = ‘Grosvenor Hotel’ }

e) RA: guestName(GuestGuest.guestNo = Booking.guestNo(BookingBooking.hotelNo = Hotel.hotelNo(hotelName = ‘Grosvenor Hotel’(Hotel))

TRC: { G.guestName | Guest(G) ^ (H) (Hotel(H)) ^ (B) (Booking(B)) ^ G.guestNo = B.guestNo ^ B.hotelNo = H.hotelNo ^ H.hotelName = ‘Grosvenor Hotel’)) }

DRC: { guestName | Guest(guestNo1, guestName, guestAddress) ^ (Hotel(hotelNo1, hotelName, city) ^ (Booking(hotelNo2, guestNo2, dateFrom, dateTo, roomNo) ^ guestNo1 = guestNo2 ^ hotelNo1 = hotelNo2 ^ hotelName = ‘Grosvenor Hotel’) }

f) RA: roomNo, hotelNo, type, price, guestName(Room Room.hotelNo = Booking.hotelNo(Booking.roomNo = Room.roomNo ^ Room.roomNo != NULL (BookingBooking.hotelNo = Hotel.hotelNo (hotelName = ‘Grosvenor Hotel’ (HotelBooking.guestNo = Guest.guestNo(Guest))) }

TRC: { R.roomNo, R.hotelNo, R.price, R.type G.guestName | Room(R) ^ Guest(G) ^ (H) (Hotel(H)) ^ (B) (Booking(B)) ^ R.hotelNo = B.hotelNo ^ B.roomNo = R.roomNo ^ R.roomNo != NULL ^ B.hotelNo = H.hotelNo ^ H.hotelName = ‘Grosvenor Hotel’ ^ B.guestNo = G.guestNo) }

DRC: { roomNo, hotelNo, price, type, guestName | (R(Room(hotelNo1, roomNo1, type, price)ooking(hotelNo2, guestNo1, dateFrom, dateTo, roomNo2)) (H(Hotel(hotelNo3, hotelName, city) (G(Guest(guestNo2, guestName, guestAddress)) ^ hotelNo1 =hotelNo2 ^ roomNo1 = roomNo2 ^ hotelNo2 = hotelNo3 ^ roomNo2 != NULL ^ hotelName = ‘Grosvenor Hotel’ ^ guestNo1 = guestNo2) }

g) RA: guestNo, guestName, guestAddress(Guest Guest.guestNo Booking.guestNo  Booking Booking.hotelNo = Hotel.hotelNo (hotelName = ‘Grosvenor Hotel’ (Hotel)))

TRC: { G.guestNo, G.guestName, G.guestAddress | (Guest(G) ^ (B) (Booking(B)) ^ (H) (Hotel(H)) ^ G.guestNo = B.guestNo ^ B.hotelNo = H.hotelNo ^ H.hotelNo = ‘Grosvenor Hotel’ )) }

DRC: { guestNo, guestName, guestAddress | (G(Guest(guestNo1, guestName, guestAddress) ^ B(Booking(hotelNo1, guestNo2, dateFrom, dateTo, roomNo)) ^ H(Hotel(hotelNo2, hotelName, city) ^ guestNo1 = guestNo2 ^ hotelNo1 = hotelNo2 ^ hotelName = ‘Grosvenor Hotel’ }