Host

We used artificial key hostID for this table’s primary key, as no other attribute about a host could uniquely identify a tuple.

Because of this, hostID has the constraint that it can’t be null.

House

Since this is a weak entity set, we used artificial key hostID and houseID jointly for this table’s primary key, as no other attribute about a house could uniquely identify a tuple.

Because of this, hostID and houseID have the constraint that it can’t be null.

Amenities

Since this is a weak entity set, we used artificial key hostID and houseID for this table’s primary key, as no other attribute about amenities could uniquely identify a tuple.

Because of this, hostID and houseID have the constraint that it can’t be null.

Guest

We used artificial key guestID for this table’s primary key, as no other attribute about a guest could uniquely identify a tuple.

Because of this, guestID has the constraint that it can’t be null.

Reservation

We used artificial key reservationID for this table’s primary key, as no other attribute about a reservation could uniquely identify a tuple.

Because of this, reservationID has the constraint that it can’t be null.

Reviews

We used artificial key reviewID for this table’s primary key, as no other attribute about a review could uniquely identify a tuple.

Because of this, reviewID has the constraint that it can’t be null.

madeNpaidBy

We used artificial key paymentID for this table’s primary key, as no other attribute about a payment could uniquely identify a tuple.

Because of this, paymentID has the constraint that it can’t be null.

views

This table uses guestID and hostID plus houseID as its primary keys. There is a foreign key/RI constraint: each guestID and hostID and houseID in views must correspond to an existing guestID and host ID and houseID in Guest and Host and House, respectively. If any one of these is deleted in the Guest or Host or House tables, they will be deleted in the views table as well.

hostedBy

This table uses hostID and houseID as its primary keys. There is a foreign key/RI constraint: each hostID and houseID in hostedBy must correspond to an existing hostID and houseID in Host and House, respectively. If either is deleted in the Host or House tables, they will be deleted in the hostedBy table as well.

receives

This table uses reservationID and hostID as its primary keys. There is a foreign key/RI constraint: each reservationID and hostID in madeFor must correspond to an existing reservationID and hostID in Reservation and Host, respectively. If either is deleted in the Reservation or House tables, they will be deleted in the receives table as well.

givenTo

This table uses hostID and houseID and reviewID as its primary keys. There is a foreign key/RI constraint: each hostID and houseID and reviewID in givenTo must correspond to an existing hostID and houseID and reviewID in Host and House and Review, respectively. If any is deleted in the Host or House or Review tables, they will be deleted in the givenTo table as well.

givenBy

This table uses guestID and reviewID as its primary keys. There is a foreign key/RI constraint: each guestID and reviewID in givenBy must correspond to an existing guestID and reviewID in Guest and Review, respectively. If either is deleted in the Guest or Review tables, they will be deleted in the givenBy table as well.