**ER Diagram Changes:**

- Room is now a  weak entity with branchName. This was changed because we realized the room number should be a partial key that relies on branchName's primary key for identification. Otherwise, multiple hotel branches would not be able to have a room of the same number without causing confusion.
- Removed RoomService's restaurant name and hours
- Added ParkingSpot entity to maintain 7 entities (excluding ISA and Weak entities).
- Removed 'employee helps guest' relationship.
- Removed payment system and its corresponding relationships.
- Changed one-many relationship with guest and room service
- Removed Facilities's hours, agelimit, and capacity

**Schema Derived:**

Facilities(TypeName, Cost)
Uses(**TypeName**, **CustomerID**)
Orders_RoomService(OrderNumber**, CustomerID**, DeliveryTime, Cost)
Requests(**Number**, **BranchName, CustomerID**)
Branch(BranchName, Address, BranchNumber)
Books_Reservation(ConfirmationNumber, **CustomerID**, StartDate, EndDate)
Makes(**ConfirmationNumber, EmployeeID**)
Cleans(**EmployeeID**, **Number, BranchName**)
Employee_WorksAt(EmployeeID, **BranchName**, SIN, Name) (BranchName cannot be NULL)
- Candidate key: SIN

Receptionist(**EmployeeID**)
Housekeeper(**EmployeeID**)
Has_Room(**Number**, **BranchName**, Type, Cost, Cleaned)
Reserves(**Number, BranchName, ConfirmationNumber)**
ParksAt_ParkingSpot(ParkingNumber, Location, Cost, **CustomerID**)
Guest(CustomerID, Address, FirstName, LastName, Email)
- Candidate key: Email

**Functional Dependencies**

(Facilities) TypeName -> Cost
(Room Service) OrderNumber -> DeliveryTime
(Guest) CustomerID -> Address, FirstName, LastName, Email
(Books_Reservation) ConfirmationNumber -> CustomerID
(Books_Reservation) ConfirmationNumber -> StartDate, EndDate
(Has_Room) Number, BranchName -> Type, Cost, Cleaned

(Requests) Number, BranchName -> CustomerID
(Reserves) Number, BranchName -> ConfirmationNumber
(Reserves_Reservation) ConfirmationNumber -> Number
(Branch) BranchName -> Address, branchNumber
(Branch) BranchNumber -> Address
(WorksAt_Employee) EmployeeID -B BranchName
(Employee) EmployeeID -> SIN, Name
(Makes_Reservation) ConfirmationNumber -> EmployeeID
(Has_Room) Type -> Cost

**Normalization:**

Two relations, identified below, are not already in BCNF or 3NF. We must normalize these relations.

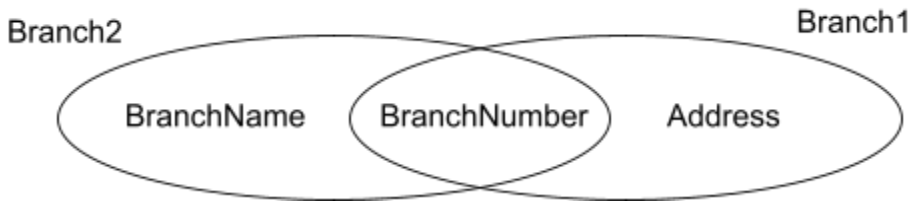
*Branch(BranchName, Address, BranchNumber)*

This relation has the below functional dependencies:
BranchName -> Address, BranchNumber
BranchNumber -> Address

We start by finding closures for these functional dependencies:
BranchName += {BranchName, Address, BranchNumber}
BranchNumber += {BranchNumber, Address}

These closures show that BranchNumber is not a superkey for the Branch relation, so the relation is not in BCNF.

We can decompose on the BranchName -> Address functional dependency.




This gives us relations Branch1 and Branch2.
Branch1(BranchNumber, Address)
Branch2(BranchNumber, BranchName)
Both of these relations have only two attributes so they must be in BCNF and our decomposition is complete.

Two relations, identified below, are not already in BCNF or 3NF. We must normalize these relations.

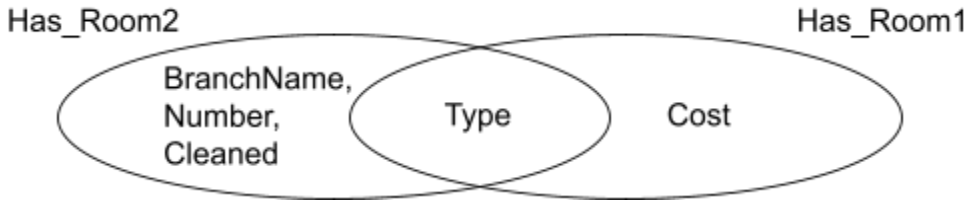
*Has_Room(Number, BranchName, Type, Cost, Cleaned)*

This relation has the below functional dependencies:
Number, BranchName -> Number, BranchName, Type, Cost, Cleaned
Type -> Cost

We start by finding closures for these functional dependencies:
Number, BranchName += {Number, BranchName, Type, Cost, Cleaned}
Type += {Type, Cost}

These closures show that Type is not a superkey for the Has_Room relation, so the relation is not in BCNF.

We can decompose on the Type -> Cost functional dependency.




This gives us relations Has_Room1 and Has_Room2.
Has_Room1(Type, Cost)
Has_Room2(Type, Number, BranchName, Cleaned)
Both of these relations have only two attributes so they must be in BCNF.

**Post-Normalization Schema:**
Relations that were changed are highlighted.

Facilities(TypeName, Cost)
Uses(**TypeName**, **CustomerID**)
Orders_RoomService(OrderNumber**, CustomerID**, DeliveryTime, Cost)
Requests(**Number**, **BranchName, CustomerID**)
Branch1(BranchNumber, Address)
Branch2(BranchName, BranchNumber)
Books_Reservation(ConfirmationNumber, **CustomerID**, StartDate, EndDate)
Makes(**ConfirmationNumber, EmployeeID**)
Cleans(**EmployeeID**, **Number, BranchName**)

Employee_WorksAt(EmployeeID, **BranchName**, SIN, Name) (BranchName cannot be NULL??)

- Candidate key: SIN

Receptionist(**EmployeeID**)

Housekeeper(**EmployeeID**)

Has_Room1(Type, Cost)

Has_Room2(Number, **BranchName**, Type, Cleaned)

Reserves(**Number, BranchName, ConfirmationNumber)**

ParksAt_ParkingSpot(ParkingNumber, Location, Cost, **CustomerID**)

Guest(CustomerID, Address, FirstName, LastName, Email)

- Candidate key: Email

## ***TODO**

**SQL DDL**

```
CREATE TABLE Facilities(
        TypeName char(20) PRIMARY KEY,
        Cost int
);

INSERT INTO Facilities
VALUES
('Swimming Pool', '0'),
('Spa', '30'),
('Gym','0'),
('Sauna', '0'),
('Conference Room', '0');


CREATE TABLE Uses(
        TypeName char(20),
        CustomerID int,
        PRIMARY KEY (TypeName, CustomerID),
        FOREIGN KEY (TypeName) REFERENCES Facilities(TypeName),
        FOREIGN KEY (CustomerID) REFERENCES Guest(CustomerID)
        ON DELETE CASCADE
);

INSERT INTO USES
VALUES
```

```
('Steam Room', '12'),
('Pool locker', '34'),
('Massage', '56'),
('Facial Massage', '78'),
('Equipment Rental', '91');

CREATE TABLE Orders_RoomService (
        OrderNumber int PRIMARY KEY,
        Cost int;
        DeliveryTime char(15)
        CustomerID int,
        FOREIGN KEY (CustomerID) REFERENCES Guest(CustomerID)
);

INSERT INTO Orders_RoomService
VALUES
('1442', '20', '4:00 PM', '12'),
('1844', '15', '1:00 PM', '34'),
('8003', '44', '1:30 AM', '56'),
('9553', '102', '2:30 PM', '78'),
('5773', '85', '5:50 PM', '91');


CREATE TABLE Requests(
        Number int,
        BranchName char(40),
        CustomerID int,
        PRIMARY KEY(Number, BranchName, CustomerID),
        FOREIGN KEY (BranchName) references Branch(BranchName)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
        FOREIGN KEY (Number) references Has_Room(Number)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
        FOREIGN KEY (CustomerID) references Guest(CustomerID)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

INSERT INTO Requests
VALUES
```

```
('100', 'Hilton', '12'),
('300', 'Marriott', '34'),
('100', 'Shelton', '56'),
('500', 'Motel 8', '78'),
('300', 'Best Western', '91');

CREATE TABLE Branch1 (
        BranchNumer int PRIMARY KEY,
        Address char(50)
);

INSERT INTO Branch1
VALUES
('2', '123-944 Best Street'),
('33', '0394 Orange Drive'),
('4', '02937-34 King Street'),
('55', '028 Hunter Drive'),
('3', '0933 Russell Street');

CREATE TABLE Branch2 (
        BranchName char(40) PRIMARY KEY,
        BranchNumber int
);

INSERT INTO Branch2
VALUES
('Hilton', '2'),
('Marriott', '33'),
('Shelton', '4'),
('Motel 8', '55'),
('Best Western', '3');

CREATE TABLE Books_Reservation (
        ConfirmationNumber int PRIMARY KEY,
        CustomerID int,
        StartDate int,
        EndDate int,
        FOREIGN KEY (CustomerID) REFERENCES Guest(CustomerID)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
```

```
INSERT INTO Books_Reservation
VALUES
('551283', '12', 'Sept 8, 2020', 'Sept 9, 2020'),
('488291', '34', 'Aug 23, 2020', 'Aug 25 2020'),
('449103', '56', 'Jan 15, 2020', 'Jan 20, 2020'),
('100482', '78', 'July 3, 2020', 'July 10, 2020');

CREATE TABLE Makes (
        ConfirmationNumber int,
        EmployeeID int,
        PRIMARY KEY(ConfirmationNumber, EmployeeID),
        FOREIGN KEY (ConfirmationNumber) REFERENCES Books_Reservation(
        confirmationNumber)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
        FOREIGN KEY(EmployeeID) REFERENCES Receptionist(EmployeeID)
)

INSERT INTO Makes
VALUES
('551283', '1103'),
('488291', '2774'),
('449103', '1234'),
('100482', '2246'),
('110294', '8736');

CREATE TABLE Cleans (
        EmployeeID int,
        Number int,
        BranchName char(40),
        PRIMARY KEY (EmployeeID, Number, BranchName),
        FOREIGN KEY (EmployeeID) REFERENCES Housekeeper(EmployeeID),
        FOREIGN KEY (Number) REFERENCES Has_Room(Number),
        FOREIGN KEY (BranchName) REFERENCES Branch(BranchName)
);

INSERT INTO Cleans
VALUES
('1103', '100', 'Hilton'),
('5521', '300', 'Marriott'),
```

```
('7328', '300', 'Shelton'),
('5521', '400', 'Motel 8'),
('5521', '100', 'Best Western);

CREATE TABLE Employee_WorksAt (
        EmployeeID int PRIMARY KEY,
        BranchName char (50) NOT NULL,
        SIN int UNIQUE,
        Name char(20) NOT NULL,
        FOREIGN KEY (BranchName) REFERENCES Branch(BranchName)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

INSERT INTO Employee_WorksAt
VALUES
('1103', '123456789', 'Rikki Haynes'),
('2774', '998877665', 'Betty Faulkner'),
('1234', '334652789', 'Steaphanie Philip'),
('2246', '620048245', 'Wilfred Dunn'),
('8736', '960334909', 'Billy Herman');

CREATE TABLE Receptionist (
        EmployeeID int PRIMARY KEY,
        FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);

INSERT INTO Receptionist
VALUES
('1103'),
('2774'),
('1234'),
('2246'),
('8736');

CREATE TABLE Housekeeper (
        EmployeeID int PRIMARY KEY,
        FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);

INSERT INTO Housekeeper
```

```
VALUES
('5521'),
('2774'),
('1234'),
('2246'),
('8736');

CREATE TABLE Has_Room1(
        Type char(40),
        Cost int,
        PRIMARY KEY (Type)
);

INSERT INTO Has_Room1
VALUES
('Standard Double', '400'),
('Standard Suite', '800'),
('Standard Double', '400',),
('Deluxe Suite', '1000',),
('Presidential Suite', '3000');

CREATE TABLE Has_Room2(
        Number int,
        BranchName char(40),
        Type char(40),
        Cleaned char(20),
        PRIMARY KEY (Number, BranchName),
        FOREIGN KEY (BranchName) REFERENCES Branch(BranchName)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

INSERT INTO Has_Room2
VALUES
('100', 'Hilton', 'Yes'),
('200', 'Marriott', 'No'),
('300', 'Shelton', 'Yes'),
('400', 'Motel 8', 'Yes'),
('500', 'Best Western', 'No');

CREATE TABLE Reserves (
```

```
        Number int,
        BranchName char(40),
        ConfirmationNumber int,
        PRIMARY KEY (number, branchName, confirmationNumber)
        FOREIGN KEY (Number) REFERENCES Has_Room(Number),
        FOREIGN KEY (BranchName) REFERENCES Branch(BranchName)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (ConfirmationNumber) REFERENCES
Books_Reservation(ConfirmationNumber)
);

INSERT INTO Reserves
VALUES
('100', 'Hilton', '551283'),
('300', 'Marriott', '488291'),
('100', 'Shelton', '449103'),
('500', 'Motel 8', '100482'),
('300', 'Best Western' , '110294');

CREATE TABLE ParksAt_ParkingSpot(
        ParkingNumber int PRIMARY KEY,
        Location char(40),
        Cost int,
        CustomerID int,
        FOREIGN KEY (CustomerID) REFERENCES Guest(CustomerID)
);

INSERT INTO ParksAt_ParkingSpot
VALUES
('111234', 'Hilton Parkade', '5', '12'),
('323556', 'Marriott Parkade', '3', '34'),
('125523' , 'Shelton Parkade', '15', '56'),
('243356', 'Motel 8 Lot 2', '23', '78'),
('664578', 'Best Western Lot 4', '4', '91');

CREATE TABLE Guest(
        CustomerID int PRIMARY KEY
        Address char(40) NOT NULL,
        FirstName char(40) NOT NULL,
        LastName char(40) NOT NULL,
```

```
        Email char(40) UNIQUE
);

INSERT INTO Guest
VALUES
('12', '7800 Manor Station, Greenville, NC 27834', 'Theia', 'Brown,Theia.Brown@gmail.com'),
('34', '5 Smith Avenue Providence, RI 02904', 'Chase', 'Johnson',
'Chase.Johnson@gmail.com'),
('56', '12 Court Circle Ridgecrest, CA 93555', ' Marcus', 'Oneal' , 'Marcus.Oneal@gmail.com'),
('78', '731 Holly Str Ossining, NY 10562' , 'Rubie' , 'Blackmore', 'Rubie.Blackmore@gmail.com'),
('91' ,'9439 Rock Maple St Anchorage, AK 99504', 'Kaitlin', 'Shaw', ' Kaitlin.Shaw@gmail.com');
```