

Schema:

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 \rightarrow Address, BranchNumber

BranchNumber \rightarrow Address

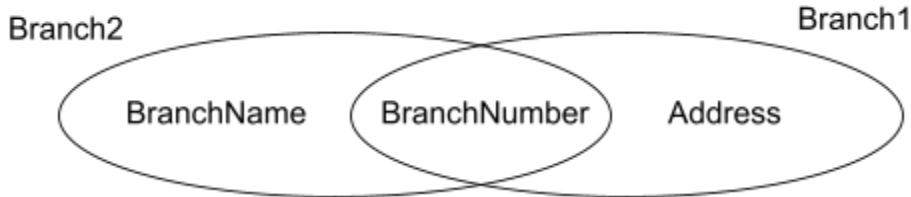
We start by finding closures for these functional dependencies:

BranchName \rightarrow {BranchName, Address, BranchNumber}

BranchNumber \rightarrow {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 \rightarrow 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 \rightarrow Number, BranchName, Type, Cost, Cleaned

Type \rightarrow Cost

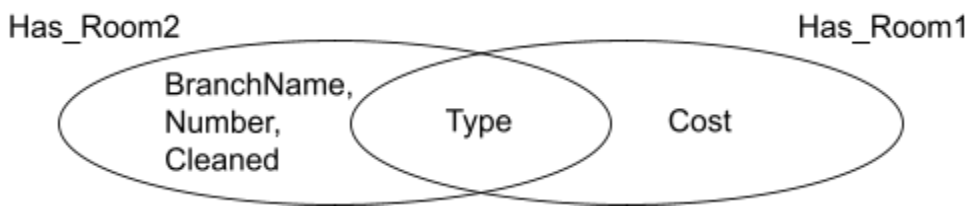
We start by finding closures for these functional dependencies:

Number, BranchName \rightarrow {Number, BranchName, Type, Cost, Cleaned}

Type \rightarrow {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 \rightarrow 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