**Hotel** (hotel\_id, hotel\_name, address, phone, email, webpage, stars, owner\_id)

F= {hotel\_id → hotel\_name, address, phone, email, stars, owner\_id;

owner\_id → hotel\_id}

(hotel\_id)+ **=** (hotel\_id, hotel\_name, address, phone, email, webpage, stars, owner\_id) = Hotel

(owner\_id)+= (owner\_id, hotel\_id....) = Hotel

hotel\_id and owner\_id are candidate keys, thus minimal superkeys.

We have no BCNF vioaltion, so **Hotel** realtion is in BC normal form

**HotelOwner** (owner\_id, firstName, lastName, phone, salary)

F = {owner\_id → firstName, lastName;

firstName, lastName → owner\_id, phone, salary}

Minimal cover is {owner\_id → firstName,

owner\_id → lastName;

firstName, lastName → owner\_id,

firstName, lastName →phone,

firstName, lastName →salary}

No extrainiousy or redundancy

(firstName, LastName)+= (firstName, lastName , owner\_id, phone, salary) = HotelOwner

(owner\_id)+ = (owner\_id, firstName, LastName.....) = HotelOwner

(firstName, LastName) and owner\_id are candidate keys

No BCNF violation, so Hotelowner is in BCNF form, no decompostion is needed

**Department** (dep\_id, department\_name, staffNumber)

F = {dep\_id → department\_name

dep\_id, department\_name → staffNumber}

Already minimal cover

(dep\_id)+ = (dep\_id, department\_name, staffNumber) = Department

dep\_id is candidate key (minimal superkey)

So, (dep\_id, department\_name) is a superkey too.

We have no BCNF violation

Department is in BCNF form

**Staff** (member\_id, firstName , LastName, phone, memberPosition, salary, hotel\_id, dep\_id)

F= {member\_id → firstName, LastName, hotel\_id, dep\_id;

member\_id, dep\_id → firstName, LastName, phone, memberPosition, salary}

F= {member\_id → firstName,

member\_id →LastName,

member\_id →hotel\_id,

member\_id →dep\_id;

~~member\_id, dep\_id → firstName,~~

~~member\_id, dep\_id → LastName,~~

member\_id, dep\_id →phone,

member\_id, dep\_id →memberPosition,

member\_id, dep\_id →salary}

member\_id, dep\_id → firstName; member\_id →dep\_id; member\_id → firstName,

dep\_id is extrainous, remove it and get member\_id → firstName,

member\_id → firstName becomes redundant, so remve it from F

The same implies for member\_id, dep\_id → LastName

Minimal cover is {member\_id → firstName,

member\_id →LastName,

member\_id →hotel\_id,

member\_id →dep\_id;

member\_id, dep\_id →phone,

member\_id, dep\_id →memberPosition,

member\_id, dep\_id →salary}

(member\_id)+ = (member\_id, firstName, LastName, hotel\_id, dep\_id, phone, memberPosition, salary) = Staff

member\_id is candidate key, thus a superkey

(member\_id, dep\_id) is a superkey too

No BCNF violation

Staff is in BCNF form

**Room** (roomNumber, roomType, price, roomStatus, capacity, hotel\_id)

F = {roomNumber → roomType, roomStatus, capacity, hotel\_id;

roomNumber, roomType, capacity → price}

Already minimal cover {roomNumber → roomType,

roomNumber → roomStatus,

roomNumber → capacity,

roomNumber → hotel\_id;

roomNumber, roomType, capacity → price}

(roomNumber)+ =(roomNumber, roomType, roomStatus, hotel\_id, capacity, price)= Room

Thus, roomNumber is candidate key.

(roomNumber, roomType, capacity) is a superkey

So, we have do not have any BCNF violations.

Room is in BCNF form

**Customer** (customer\_id, passport\_id, firstName, LastName, dob, phone, email)

F = {customer\_id →firstName, LastName, phone, email, passport\_id

customer\_id, passport\_id →firstName, LastName, dob}

Minimal cover is {customer\_id →firstName,

customer\_id →LastName,

customer\_id →phone,

customer\_id → email,

customer\_id →passport\_id,

customer\_id, passport\_id →firstName

customer\_id, passport\_id →LastName

customer\_id, passport\_id → dob}

customer\_id →passport\_id,

customer\_id, passport\_id → firstName and

customer\_id, passport\_id → LastName

Here passpart\_id is extrainous, we remove it and get customer\_id →firstName and customer\_id →LastName. Here they become redundant. So remove them too and get:

Minimal cover {customer\_id →firstName,

customer\_id →LastName,

customer\_id →phone,

customer\_id → email,

customer\_id →passport\_id,

customer\_id, passport\_id → dob}

(customer\_id)+ = (customer\_id, firstName, LastName, phone, email, passport\_id, dob) = =Customer

customer\_id is candidate key

(customer\_id, passport\_id) is superkey

No BCNF violation

Customer is in BCNF form

**RoomReserve** (reserve\_id, customer\_id, roomNumber, checkin, checkout, total\_price)

F = {reserve\_id → customer\_id, roomNumber ;

reserve\_id, roomNumber, customer\_id → checkin, checkout , total\_price}

Already minimal cover

(reserve\_id)+ = (reserve\_id, customer\_id, roomNumber, checkin, checkout, total\_price) = =RoomReserve

reserve\_id is candidate key

(resever\_id, roomNumber, customer\_number) is superkeys

RoomReserve is in BCNF form, there are no BCNF violations

**Payment** (payment\_id, customer\_id, amount, method, payment\_date, payment\_status)

F = {payment\_id → cutomer\_id, amount

payment\_id, customer\_id → amount, method, payment\_date, payment\_status}

Simplify RHS: {payment\_id → cutomer\_id,

payment\_id → amount,

payment\_id, customer\_id → amount,

payment\_id, customer\_id→ method,

payment\_id, customer\_id→ payment\_date,

payment\_id, customer\_id→ payment\_status}

As payment\_id → cutomer\_id; payment\_id → amount and

payment\_id, customer\_id → amount, custmer\_id is extrainious.

Now we get payment\_id → cutomer\_id; payment\_id → amount and payment\_id → amount,

payment\_id→amount is redundant, remove one

Minimal cover is {payment\_id → cutomer\_id,

payment\_id → amount,

payment\_id, customer\_id→ method,

payment\_id, customer\_id→ payment\_date,

payment\_id, customer\_id→ payment\_status}

(payment\_id)+ = (payment\_id, customer\_id, amount, method, payment\_date, payment\_status) = = Payment

payment\_id is candidate key

(payment\_id, customer\_id) is superkey

No BCNF violations, thus Payment is in BCNF form

**Restaurant** (restaurant\_id, restaurant\_tables, hotel\_id)

F = {restaurant\_id → restaurant\_tables, hotel\_id}

Already in minimal cover

(restaurant\_id)+ = (restaurant\_id, restaurant\_tables, hotel\_id) = Restaurant

restaurant\_id is candidate key, thus a superkey

Restaurnat is in BCNF form

**Menutype** (menu\_id, menuType, restaurant\_id)

F = {menu\_id → menuType, restaurant\_id}

Already in minimal cover

(menu\_id)+ = (menu\_id, menuType, restaurant\_id) = Menutype

menu\_id is candidate key

Menutype is in BCNF

**Items** (itemName, usedIn)

F = { itemName → usedIn}

Already in minimal cover

itemName is candidate key

Items is in BCNF

**MenuItem** (menuItem, itemName, menu\_id)

F = { menuItem → itemName, menu\_id}

Already in minimal cover

menuItem is candidate key

MenuItem is in BCNF

**RestaurantTables** (tableNumber, restaurant\_id, tableStatus, bill, ordered)

F = { tableNumber, restaurant\_id → tableStatus, bill, ordered}

Already in minimal cover

(tableNumber, restaurant\_id ) is candidate key

RestaurantTables is in BCNF

**Gym** (gym\_id, equipment, hotel\_id )

F = {gym\_id, → equipment, hotel\_id }

Already in minimal cover

gym\_id is candidate key

Gym in BCNF

**GymEquipment** (equipment\_id, gym\_id, equipmentType, works )

F = {equipment\_id, gym\_id → equipmentType, works\_id}

Already in minimal cover

(equipment\_id, gym\_id)+ = (equipment\_id, gym\_id, equipmentType, works ) = GymEquipment

(equipment\_id, gym\_id) is candidate key

GymEquipment is in BCNF

**Spa** (spa\_id , services, hotel\_id)

F = { spa\_id → services, hotel\_id}

Already in minimal cover

spa\_id is candidate key

Spa is in BCNF

**SpaService** (service\_id, spa\_id , service\_name, price);

F = {service\_id, spa\_id → service\_name, price}

Already in minimal cover

(service\_id, spa\_id) is candidate key

SpaService is in BCNF