**3 Schema:**

**Manager**(salary:integer, phone:integer, manager\_ID: integer, address: string, hotel\_address:string, hotel\_name:string)

Primary key:{hotel\_address, hotel\_name, work\_number}

Candidate key:{hotel\_address, hotel\_name, work\_number}

Foreign key:hotel\_address, hotel\_name

**Worker**(salary:integer, phone:integer, work number: integer, address: string, hotel\_address:string, hotel\_name:string, service\_description: integer, cleaning\_kit:integer)

Primary key:{hotel\_address, hotel\_name, work\_number}

Candidate key:{hotel\_address, hotel\_name, work\_number}

Foreign key:hotel\_address, hotel\_name

**Room**(number:integer, price:integer, type: string, hotel\_address: string, hotel\_name:string)

Primary key:{hotel\_address, hotel\_name, number}

Candidate key:{hotel\_address, hotel\_name, number}

Foreign key:hotel\_address, hotel\_name

**Meeting** (meeting ID:integer, topic:string, location: string, time: integer)

Primary key:meeting ID

Candidate key:meeting ID,{time, location}

Foreign key:none

**Hotel**(address:string, name:string, ZIP code: integer)

Primary key:{address, name}

Candidate key:{address, name}

Foreign key:none

**Bonus**(bonus\_number:integer, work\_number:integer, amount: integer, hotel\_address:string, hotel\_name:string)

Primary key:{bonus\_number, hotel\_address, hotel\_name}

Candidate key:{bonus\_number, hotel\_address, hotel\_name}

Foreign key:hotel\_address, hotel\_name

**Financial summery**(expenditure:integer, income:integer, summary\_number: integer, FD\_address : string, FD\_name : string)

Primary key:{summary\_number, FD\_address, FD\_name}

Candidate key:{summary\_number, FD\_address,FD\_name}

Foreign key: FD\_address, FD\_name

**Financial department**(ZIP\_code:integer, address:string, name:string)

Primary key:{address, name}

Candidate key:{address, name}

Foreign key:none

**Organize(**hotel\_address:string, hotel\_name:string, manager\_ID:integer, time: integer, meeting\_ID: integer)

Primary key:{hotel\_address, hotel\_name, time, meeting\_ID, manager\_ID:integer}

Candidate key:{hotel\_address, hotel\_name, time, meeting\_ID, manager\_ID:integer}

Foreign key:hotel\_address, hotel\_name, time, meeting\_ID, manager\_ID

**Attend(**hotel\_address:string, hotel\_name:string, time: integer, meeting\_ID: integer, worker\_work\_number:integer)

Primary key:{hotel\_address, hotel\_name, time, meeting\_ID, worker\_work\_number:integer}

Candidate key:{hotel\_address, hotel\_name, time, meeting\_ID, worker\_work\_number:integer}

Foreign key:hotel\_address, hotel\_name, time, meeting\_ID, worker\_work\_number:integer

**Contact(**hotel\_address:string, hotel\_name:string, FD\_address:string,FD\_name:string)

Primary key:{hotel\_address, hotel\_name, FD\_address, FD\_name}

Candidate key:{hotel\_address, hotel\_name, FD\_address, FD\_name}

Foreign key:hotel\_address, hotel\_name,FD\_address, FD\_name

**Decide(**hotel\_address:string, hotel\_name:string, bonus\_number:integer, FD\_address:string, dep\_name:string, FS\_number:integer, )

Primary key:{hotel\_address, hotel\_name, bonus\_number, FD\_address, FD\_name, FS\_number}

Candidate key:{hotel\_address, hotel\_name, bonus\_number, FD\_address, FD\_name, FS\_number}

Foreign key:hotel\_address, hotel\_name, bonus\_number, FD\_address, FD\_name, FS\_number

**Assign(**hotel\_address:string, hotel\_name:string, worker\_work\_number:integer, room\_number:integer)

Primary key:{hotel\_address, hotel\_name, worker\_work\_number, room\_number}

Candidate key:{hotel\_address, hotel\_name, worker\_work\_number, room\_number}

Foreign key:hotel\_address, hotel\_name, worker\_work\_number, room\_number

**4 Functional Dependency:**

These are the relations that have non-trivial functional dependency

**Manager:** hotel\_address, hotel\_name, manager\_ID -> phone, address, salary

**Worker:**  hotel\_address, hotel\_name, work number -> phone, address, salary, cleaning\_kit, service\_desciprtion

**Room:** hotel\_address, hotel\_name, number-> type

type -> price

**Hotel:** address -> ZIP code

address, name -> phone

**Bonus:** hotel\_address, hotel\_name, bonus number -> work\_number, amount

**Financial summary:** FD\_address, FD\_name,FS\_number -> income, expenditure

**Financial department:** address -> ZIP code

address, name -> phone

**Meeting:** meetingID -> topic, location, time

**5:Normalization**

**Hotel**(address,ZIP\_code, name, phone)

Functional dependencies: address -> ZIP\_code

address, name -> phone

Decomposition:

**Hotel1**(address,ZIP\_code)

Primary key: address

Candidate key:address

Foreign key:none

**Hotel2**(address, name, phone)

Primary key: {address, name}

Candidate key: {address, name}

Foreign key:none

Then no function functional dependency violates BCNF

**Room**(hotel\_address, hotel\_name, number,type,price)

Functional dependencies: hotel\_address, hotel\_name, number-> type

type -> price

Decomposition:

**Room1**(type,price)

Primary key: type

Candidate key: type

Foreign key:none

**Room2**(hotel\_address, hotel\_name, number, type)

Primary key: {hotel\_address, hotel\_name, number}

Candidate key: {hotel\_address, hotel\_name, number}

Foreign key:hotel\_address, hotel\_name

**6 SQL DDL tabel creation:**

CREATE TABLE Hotel1

(

address CHAR(30),

ZIP\_code CHAR(6),

PRIMARY KEY(address))

CREATE TABLE Hotel2

(

address CHAR(30),

name CHAR(20),

phone integer,

PRIMARY KEY(address, name))

CREATE TABLE Manager

(salary INTEGER,

phone INTEGER,

manager\_id INTEGER,

address CHAR(30),

hotel\_address CHAR(30),

hotel\_name CHAR(30)

PRIMARY KEY(hotel\_address, hotel\_name, manager\_id),

FOREIGN KEY(hotel\_address, hotel\_name) REFERENCES Hotel2(address, name),

ON DELETE CASCADE)

CREATE TABLE Worker

(salary INTEGER,

phone INTEGER,

work\_number INTEGER,

address CHAR(30),

hotel\_address CHAR(30),

hotel\_name CHAR(30),

service\_description CHAR(100),

cleaning\_kit INTEGER,

PRIMARY KEY(hotel\_address, hotel\_name, work\_number),

FOREIGN KEY(hotel\_address, hotel\_name) REFERENCES Hotel2(address, name),ON DELETE CASCADE)

CREATE TABLE Room1

(number INTEGER,

type CHAR(20),

hotel\_address CHAR(30),

hotel\_name CHAR(30),

PRIMARY KEY(hotel\_address, hotel\_name, number),

FOREIGN KEY(hotel\_address, hotel\_name) REFERENCES Hotel2(address, name)ON DELETE CASCADE)

CREATE TABLE Room2

(price INTEGER,

type CHAR(20),

PRIMARY KEY(type))

CREATE TABLE Meeting

(meeting ID INTEGER,

time INTEGER,

topic CHAR(20),

location CHAR(30),

PRIMARY KEY(meeting ID))

CREATE TABLE Bonus

(bonus\_number INTEGER,

work\_numberINTEGER,

amount INTEGER,

hotel\_address CHAR(30),

hotel\_name CHAR(30),

PRIMARY KEY(hotel\_address, hotel\_name, bonus\_number)，

FOREIGN KEY(hotel\_address, hotel\_name) REFERENCES Hotel2(address, name),ON DELETE CASCADE))

CREATE TABLE Financial\_department

(ZIP\_code INTEGER,

address CHAR(30),

phone INTEGER,

name CHAR(30),

PRIMARY KEY(address,name))

CREATE TABLE Financial\_summary

(expenditure INTEGER,

Income INTEGER,

summary\_number INTEGER,

FD\_address CHAR(20),

FD\_name CHAR(20),

PRIMARY KEY(FD\_name, FD\_address, summary\_number)，

FOREIGN KEY(FD\_address, FD\_name) REFERENCES Financial\_department(address, name),

ON DELETE CASCADE))

**Financial summery**(expenditure:integer, income:integer, summary\_number: integer, FD\_address : string, FD\_name : string)

Primary key:{summary\_number, FD\_address, FD\_name}

Candidate key:{summary\_number, FD\_address,FD\_name}

Foreign key: FD\_address, FD\_name