University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: March 1st, 2023

Group Number: 64

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Henry Kim	32523722	q1y6v	henryshkim@hotmail.com
Noel Illing	65046468	b1o2w	illing.noel@gmail.com
Babak Hadady	77085647	y9m6h	babakhad@student.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

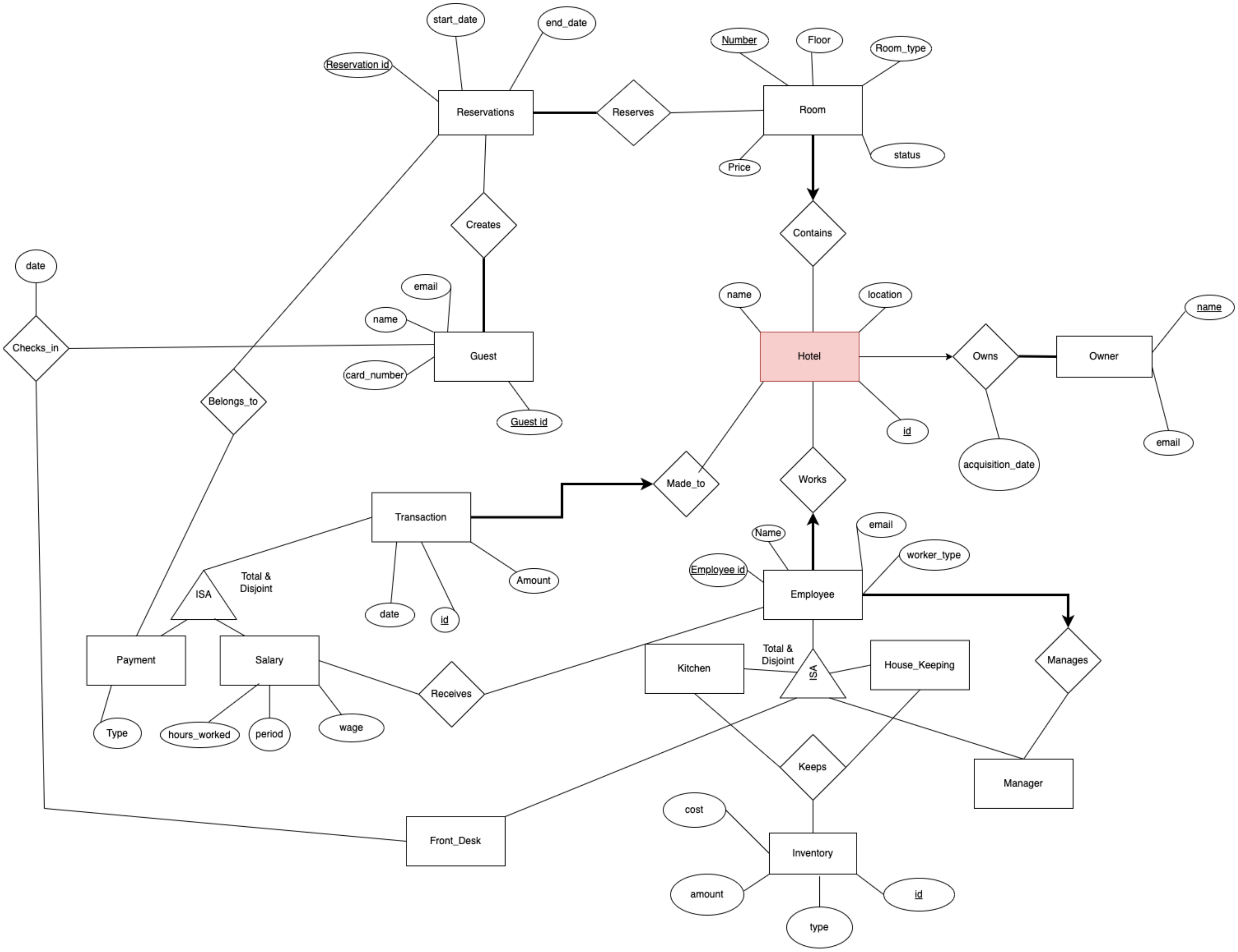
Brief Summary:

The domain of this application will be a hospitality management service. It will be helping manage room states/reservations, payment transactions, and employees for multiple hotels.

ER Diagram: (diagram on last page)

ChangeLog:

- Added ISA constraints for both ISA → Makes the ISA contain specialization constraints so that each superclass/subclass has more specific attributes for its role
- Added participation constraint → Rooms must belong to a Hotel
- Added participation constraint → Guest must make a Reservation
- Added Many-to-one relation between Hotel and Owner → Every hotel has an owner
- Added participation constraint → Hotel must have an Owner
- Added Hotel attribute number → belongs to the primary key, to allow for multiple hotels to exist with same name
- Updated Key Constraint: Changed Owner Has Hotel to Owner Owns Hotel → makes relation more specific
- Added key constraint to Room → A room can only belong to a single hotel
- Updated Key Constraint: Changed Hotel Has Rooms to Hotel Contains Rooms → makes relation more specific
- Updated Key Constraint: Payment Makes Reservation to Payment Belongs_to Reservation → makes relation more specific
- Updated Key Constraint: Guest Makes Reservation to Guest Creates Reservation → makes relation more specific
- Updated Key Constraint: Hotel makes Transaction to Transaction made to Hotel → more specific
- Added cost attribute to Inventory → creates more detailed information for Inventory
- Removed Books → unnecessary relationship
- Added attribute wage to Salary → adds more context to what the base wage was for the salary
- Added participation constraint to TransactionMadeTo → must always be made to a hotel
- Added participation constraint to EmployeeWorks → must work in a hotel
- Added worker_type attribute to Employee → required attribute for determining type of employee
- Added participation constraint to Reservation → must be made to a room
- Added participation to Employee → an employee must have a manager
- Added attribute hours_worked to Salary → key information that must be a part of a salary


start_date
end_date
Reservation id
Reservations
Reserves
Number
Floor
Room_type
Room
Price
status
Creates
Contains
date
email
name
Checks_in
Guest
card_number
Guest id
Belongs_to
name
location
Hotel
Owns
Owner
name
email
id
acquisition_date
Transaction
Made_to
Works
Name
email
worker_type
Employee id
Employee
Total &
Disjoint
ISA
Amount
date
id
Payment
Salary
Receives
Kitchen
Total &
Disjoint
ISA
House_Keeping
Manages
Type
hours_worked
period
wage
Keeps
Manager
cost
Inventory
Front_Desk
amount
type
id

Relational Schema:

HotelOwns(name: string, id: integer, location: string, **owner_name**: string, acquisition_date: string)

PK: id
FK: owner_name
NOT NULL: owner_name, location, acquisition_date

Guest(guest_id: integer, name: string, card_number: integer, email: string)

PK: guest_id
CK: email, card_number
UNIQUE: email, card_number
NOT NULL: name, card_number, email

Owner(name: string, email: string)

PK: name
CK: email
UNIQUE: email
NOT NULL: email

Reserves(**reservation_id**: integer, **number**: integer)

PK: reservation_id, number
FK: reservation_id, number

RoomContains(number: integer, floor: integer, room_type: string, price: integer, status: string, **id**: integer)

PK: number
FK: id
NOT NULL: id, status, price, room_type, floor

Reservations(reservation_id: integer, start_date: string, end_date: string)

PK: reservation_id
NOT NULL: start_date, end_date

Creates(**guest_id**: integer, **reservation_id**: integer)

PK: guest_id, reservation_id
FK: guest_id, reservation_id

TransactionMadeTo(id: integer, amount: integer, date: string, type: string, period: string, wage: integer, hours_worked: integer, **hotel_id**: integer)

PK: id
FK: hotel_id
NOT NULL: hotel_id, amount

Inventory(id: integer, amount: integer, type: string, cost: integer)

PK: id

BelongsTo(**id**: integer, **reservation_id**: integer)

PK: id, reservation_id
FK: id, reservation_id

Keeps(**id**: integer, **employee_id**: integer)

PK: id, employee_id
FK: id, employee_id

Checks_in(**employee_id**: integer, **guest_id**: integer, date: string)

PK: employee_id, guest_id
FK: employee_id, guest_id
NOT NULL: date

Receives(**id**: integer, **employee_id**: integer)

PK: id, employee_id
FK: id, employee_id

EmployeeManagesWorks(employee_id: integer, name: string, email: string, worker_type: string, **manager_id**: integer, **id:** integer)

PK: employee_id
FK: manager_id, id
CK: email
UNIQUE: email
NOT NULL: id, worker_type, manager_id

Functional Dependencies:

Guest:

Guest_id → card_number, name, email
email → card_number, name, guest_id
Card_number → email, name, guest_id

EmployeeManagesWorks:

Employee_id → name, email, manager_id, id, worker_type
email → name, manager_id, employee_id, id, worker_type

RoomContains:

Room_type, Floor → Price
Number → Room_type, Floor, Price, status, id

Reservation:

Reservation_id → start_date, end_date

Creates:

Guest_id, reservation_id → Guest_id, reservation_id

BelongsTo:

Id, reservation_id → id, reservation_id

Owner:

Name → email
Email → Name

TransactionMadeTo:

Id → date, amount, type, period, hotel_id, wage, hours_worked
Hours_worked, amount → wage

Reserves:

reservation_id, id → reservation_id, id

HotelOwns:

id → location, name, owner_name, acquisition_date
location → name

Inventory:
Id → amount, type, cost
Type, amount → cost

Keeps:
Employee_id, id → Employee_id, id

Receives:
Employee_id, id → Employee_id, id

Checks_in:
Employee_id, guest_id → date
date → employee_id

Normalization:

Every relation is in BCNF except for the following.

RoomContains:

Room_type, Floor → Price
Number → Room_type, Floor, Price, status, id

{Room_type, Floor}+ = Room_type, Floor, Price
{Number}+ = Number, Room_type, Floor, Price, status, id

The first FD (Room_type, Floor → Price) violates BCNF since Room_type, Floor is not a superkey of the RoomContains relation.

RoomContains1(Room_type, Floor, Price), RoomContains2(Room_type, Floor, Number, status, id)
RoomContains1 and RoomContains2 are in BCNF

Inventory:

Id → amount, type, cost
Type, amount → cost

{Id} + = Id, amount, type, cost
{Type, amount} + = type, amount, cost

The second FD (type, amount → cost) violates BCNF since type, amount is not a superkey of the Inventory relation.

Inventory1(type, amount, cost), Inventory2(type, amount, id)
Inventory1 and Inventory2 are in BCNF

HotelOwns:

id → location, name, owner_name, acquisition_date
location → name

{Id} + = Id, location, name, owner_name, acquisition_date
{Location} + = location, name

The second FD (location → name) violates BCNF since location is not a superkey of the HotelOwns relation.

HotelOwns1(location, name), HotelOwns2(Id, location, owner_name, acquisition_date)
HotelOwns1 and HotelOwns2 are in BCNF

TransactionMadeTo:

Id → date, amount, type, period, hotel_id, wage
Hours_worked, amount → wage

{Id} + = Id, date, amount, type, period, hotel_id, wage, hours_worked
{hours_worked, amount} + = wage, hours_worked, amount
The second FD (Hours_worked, amount → wage) violates BCNF since (hours_worked, amount) is not a superkey of the TransactionMadeTo relation.

TransactionMadeTo1(hours_worked, wage, amount), TransactionMadeTo2(Id, period, date, type, hotel_id, hours_worked, amount)
TransactionMadeTo1 and TransactionMadeTo2 are in BCNF

Checks_in:

employee_id, guest_id → date
date → employee_id

{employee_id, guest_id} + = Employee_id, guest_id, date
{date} + = date, employee_id

The second FD (date → employee_id) violates BCNF since date is not a superkey of the Checks_in relation.

Checks_in1(date, employee_id), Checks_in2(date, guest_id)
Checks_in1 and Checks_in2 are in BCNF

SQL DDL statements:

```
CREATE TABLE HotelOwns1 (
        location  CHAR(40) NOT NULL,
        name      CHAR(20),
        PRIMARY KEY (location)
)

CREATE TABLE HotelOwns2 (
        location  CHAR(40) NOT NULL,
        owner_name CHAR(20) NOT NULL,
        id INTEGER,
        acquisition_date CHAR(20) NOT NULL,
        PRIMARY KEY (id)
        FOREIGN KEY (location) REFERENCES HotelOwns1
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (owner_name) REFERENCES Owner(name)
                ON DELETE SET NULL
                ON UPDATE CASCADE
)

CREATE TABLE GUEST (
        guest_id INTEGER PRIMARY KEY,
        name CHAR(20) NOT NULL,
        card_number   INTEGER NOT NULL,
        email CHAR(30) NOT NULL,
        UNIQUE (email, card_number)
)



CREATE TABLE Owner (
        name CHAR(20) PRIMARY KEY,
        email CHAR(20) NOT NULL,
        UNIQUE (email)
)

CREATE TABLE Reserves (
        reservation_id INTEGER,
        number INTEGER,
        PRIMARY KEY (reservation_id, number)
        FOREIGN KEY (reservation_id) REFERENCES Reservation
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (number) REFERENCES RoomContains2
                ON DELETE SET NULL
                ON UPDATE CASCADE
)

CREATE TABLE RoomContains1 (
        floor INTEGER,
        room_type CHAR(20),
        price INTEGER NOT NULL,
        PRIMARY KEY (room_type, floor)
)

CREATE TABLE RoomContains2 (
        id INTEGER NOT NULL,
        floor INTEGER NOT NULL,
        room_type CHAR(20) NOT NULL,
        status CHAR(20) NOT NULL,
        number CHAR(20) PRIMARY KEY
        FOREIGN KEY (id) REFERENCES HotelOwns2
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (floor, room_type) REFERENCES RoomContains1
                ON DELETE SET NULL
                ON UPDATE CASCADE
)
```

```
CREATE TABLE Reservations (
        start_date CHAR(20) NOT NULL,
        end_date CHAR(20) NOT NULL,
        reservation_id INTEGER PRIMARY KEY,
)


CREATE TABLE Creates(
        Guest_id INTEGER,
        Reservation_id INTEGER,
        PRIMARY KEY(Guest_id, Reservation_id)
        FOREIGN KEY (Guest_id) REFERENCES Guest
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (Reservation_id) REFERENCES Reservations
                ON DELETE SET NULL
                ON UPDATE CASCADE
)

CREATE TABLE Inventory1 (
        amount INTEGER,
        cost INTEGER,
        type    CHAR(20),
        PRIMARY KEY(type, amount)
)

CREATE TABLE Inventory2 (
        amount INTEGER,
        type    CHAR(20),
        id      INTEGER PRIMARY KEY
        FOREIGN KEY (type, amount) REFERENCES Inventory1
                ON DELETE SET NULL
                ON UPDATE CASCADE
)

CREATE TABLE BelongsTo (
        id INTEGER,
        reservation_id INTEGER,
        PRIMARY KEY(id, reservation_id)
        FOREIGN KEY (id) REFERENCES TransactionMadeTo2
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (reservation_id) REFERENCES Reservations
                ON DELETE SET NULL
                ON UPDATE CASCADE
)

CREATE TABLE Keeps (
        id INTEGER,
        employee_id INTEGER,
        PRIMARY KEY(id, employee_id)
        FOREIGN KEY (id) REFERENCES Inventory2
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (employee_id) REFERENCES EmployeeManagerWorks
                ON DELETE SET NULL
                ON UPDATE CASCADE
)




CREATE TABLE Checks_in1(
        date CHAR(30) NOT NULL,
        employee_id INTEGER,
        PRIMARY KEY (date)
        FOREIGN KEY (employee_id) REFERENCES EmployeeManagerWorks
                ON DELETE SET NULL
                ON UPDATE CASCADE
)

CREATE TABLE Checks_in2(
        date CHAR(30) NOT NULL,
        guest_id INTEGER,
        PRIMARY KEY (guest_id)
        FOREIGN KEY (guest_id) REFERENCES Guest
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (date) REFERENCES Checks_in1
                ON DELETE SET NULL
                ON UPDATE CASCADE
)

CREATE TABLE Receives (
        id INTEGER,
        employee_id INTEGER,
        PRIMARY KEY(id, employee_id)
        FOREIGN KEY (id) REFERENCES TransactionMadeTo2
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (employee_id) REFERENCES EmployeeManagerWorks
                ON DELETE SET NULL
                ON UPDATE CASCADE
)

CREATE TABLE EmployeeManagesWorks (
        employee_id INTEGER PRIMARY KEY,
        id INTEGER NOT NULL,
        name CHAR(20),
        email CHAR(30),
        manager_id INTEGER NOT NULL,
        worker_type CHAR (20) NOT NULL,
        FOREIGN KEY (manager_id) REFERENCES EmployeeManagerWorks(employee_id)
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (id) REFERENCES HotelOwns2
                ON DELETE SET NULL
                ON UPDATE CASCADE
        UNIQUE (email)
)

CREATE TABLE TransactionMadeTo1 (
        amount INTEGER NOT NULL,
        hours_worked INTEGER,
        wage INTEGER ,
        PRIMARY KEY(wage, period)
)



CREATE TABLE TransactionMadeTo2 (
        id INTEGER PRIMARY KEY,
        date CHAR(30),
        type CHAR(20),
        period CHAR(60),
        hotel_id INTEGER NOT NULL,
        hours_worked INTEGER,
        amount INTEGER NOT NULL,
        FOREIGN KEY(hotel_id) REFERENCES HotelOwns2(id)
```

```
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (hours_worked, amount) REFERENCES TransactionMadeTo1
                ON DELETE SET NULL
                ON UPDATE CASCADE
)
```

INSERT statements:

HotelOwns1:

```
INSERT
INTO HotelOwns1(location, name)
VALUES (‘900 W Georgia St, Vancouver’, ‘Fairmont’)

INSERT
INTO HotelOwns1(location, name)
VALUES (‘5959 Student Union Blvd, Vancouver’, ‘Gage’)

INSERT
INTO HotelOwns1(location, name)
VALUES (‘783 Imagination Rd, Vancouver’, ‘Imagine Hotel’)

INSERT
INTO HotelOwns1(location, name)
VALUES (‘696 Lover Drive, Vancouver’, ‘Love Hotel’)

INSERT
INTO HotelOwns1(location, name)
VALUES (‘3204 Database St, China’, ‘Data Hotel’)
```

HotelOwns2:

```
INSERT
INTO HotelOwns2(location, owner_name, id, acquisition_date)
VALUES (‘900 W Georgia St, Vancouver’, ‘Elon Musk’, 1234, ‘October 10, 2004’)

INSERT
INTO HotelOwns2(location, owner_name, id, acquisition_date)
VALUES (‘5959 Student Union Blvd’, ‘Tarzan Man’, 3243, ‘November 17, 1999’)

INSERT
INTO HotelOwns2(location, owner_name, id, acquisition_date)
```

```
VALUES (‘783 Imagination Rd, Vancouver’, ‘John Smith’, 2342, ‘January 1, 2002’)

INSERT
INTO HotelOwns2(location, owner_name, id, acquisition_date)
VALUES (‘696 Lover Drive, Vancouver’, ’Heimerdinger Smith’, 4532, ‘July 23, 2005’’)

INSERT
INTO HotelOwns2(location, owner_name, id, acquisition_date)
VALUES (‘3204 Database St, China’, ‘Garen Darius’, 8978, ‘February 11, 1987’)
```

Guest:

```
INSERT
INTO Guest(guest_id, name, card_number, email)
VALUES (‘123456’, ‘Henry Kim’, ‘24429988’, ‘walkingbuddies2002@gmail.com’)

INSERT
INTO Guest(guest_id, name, card_number, email)
VALUES (‘222222’, ‘Benry Bim, ‘48603847’, ‘zedandshen@gmail.com’)

INSERT
INTO Guest(guest_id, name, card_number, email)
VALUES (‘333333’, ‘Jenry Jim’, ‘66739853’, ‘yuumicarry@gmail.com’)

INSERT
INTO Guest(guest_id, name, card_number, email)
VALUES (‘444444’, ‘Tenry Tim’, ‘12546434’, ‘thisisnotanemail@gmail.com’)

INSERT
INTO Guest(guest_id, name, card_number, email)
VALUES (‘666666’, ‘Lenry Lim’, ‘89745676’, ‘impostersussy@gmail.com’)
```

Owner:

```
INSERT
INTO Owner(name, email)
VALUES (‘Henry Kim’, ‘henry@gmail.com’)


INSERT
INTO Owner(name, email)
VALUES (‘Noel Illing’, ‘noel@gmail.com’)

INSERT
```

```
INTO Owner(name, email)
VALUES (‘Babak Bob’, ‘babak@gmail.com’)

INSERT
INTO Owner(name, email)
VALUES (‘Henry Joe’, ‘henryJoe@gmail.com’)

INSERT
INTO Owner(name, email)
VALUES (‘Henry Cam’, ‘henryCam@gmail.com’)
```

Reserves:

```
INSERT
INTO Reserves(reservation_id, number)
VALUES (101234, 401)

INSERT
INTO Reserves(reservation_id, number)
VALUES (101235, 501)

INSERT
INTO Reserves(reservation_id, number)
VALUES (101236, 601)

INSERT
INTO Reserves(reservation_id, number)
VALUES (101237, 701)

INSERT
INTO Reserves(reservation_id, number)
VALUES (101238, 801)
```

RoomContains1:

```
INSERT
INTO RoomContains1(floor, room_type, price)
VALUES (‘4’, ‘Double’, ‘200’)
```

```
INSERT
INTO RoomContains1(floor, room_type, price)
VALUES (‘5’, ‘Master’, ‘300’)

INSERT
INTO RoomContains1(floor, room_type, price)
VALUES (‘6’, ‘Single’, ‘200’)

INSERT
INTO RoomContains1(floor, room_type, price)
VALUES (‘7’, ‘Queen’, ‘250’)

INSERT
INTO RoomContains1(floor, room_type, price)
VALUES (‘8’, ‘King’, ‘300’)
```

RoomContains2:

```
INSERT
INTO RoomContains2(id, floor, room_type, status, number)
VALUES (‘1234’, ‘4’, ‘Double’, ‘occupied’, ‘401’)

INSERT
INTO RoomContains2(id, floor, room_type, status, number)
VALUES (‘3243’, ‘5’, ‘Master’, ‘vacant’, ‘501’)

INSERT
INTO RoomContains2(id, floor, room_type, status, number)
VALUES (‘3243’, ‘6’, ‘Single’, ‘occupied’, ‘601’)

INSERT
INTO RoomContains2(id, floor, room_type, status, number)
VALUES (‘1234’, ‘7’, ‘Queen’, ‘occupied’, ‘701’)

INSERT
INTO RoomContains2(id, floor, room_type, status, number)
VALUES (‘3243’, ‘8’, ‘King’, ‘occupied’, ‘801’)
```

Reservations:

```
INSERT
INTO Reservations(start_date, end_date, reservation_id)
VALUES (‘October 1, 2012’, ‘October 5, 2012’, ‘101234’)
```

```
INSERT
INTO Reservations(start_date, end_date, reservation_id)
VALUES (‘January 1, 2012’, ‘January 5, 2012’, ‘101235’)

INSERT
INTO Reservations(start_date, end_date, reservation_id)
VALUES (‘February 1, 2012’, ‘February 5, 2012’, ‘101236’)

INSERT
INTO Reservations(start_date, end_date, reservation_id)
VALUES (‘March 1, 2012’, ‘March 5, 2012’, ‘101237’)

INSERT
INTO Reservations(start_date, end_date, reservation_id)
VALUES (‘April 1, 2012’, ‘April 5, 2012’, ‘101238’)
```

Creates:

```
INSERT
INTO Creates(Guest_id, Reservation_id)
VALUES (‘123456’, ‘101234’)

INSERT
INTO Creates(Guest_id, Reservation_id)
VALUES (‘222222’, ‘101235’)

INSERT
INTO Creates(Guest_id, Reservation_id)
VALUES (‘333333’, ‘101236’)

INSERT
INTO Creates(Guest_id, Reservation_id)
VALUES (‘444444’, ‘101237’)

INSERT
INTO Creates(Guest_id, Reservation_id)
VALUES (‘666666’, ‘101238’)
```

Inventory1:

```
INSERT
INTO Inventory1(amount, cost, type)
VALUES (‘100’, ‘1000’, ‘chairs’)
```

```
INSERT
INTO Inventory1(amount, cost, type)
VALUES (‘50’, ‘100’, ‘spoons’)

INSERT
INTO Inventory1(amount, cost, type)
VALUES (‘100’, ‘80’, ‘forks’)

INSERT
INTO Inventory1(amount, cost, type)
VALUES (‘100’, ‘100’, ‘knives’)

INSERT
INTO Inventory1(amount, cost, type)
VALUES (‘50’, ‘700’, ‘blankets’)
```

Inventory2:

```
INSERT
INTO Inventory2(amount, id, type)
VALUES (‘100’, ‘12345678’, ‘chairs’)

INSERT
INTO Inventory2(amount, id, type)
VALUES (‘50’, ‘58349544’, ‘spoons’)

INSERT
INTO Inventory2(amount, id, type)
VALUES (‘100’, ‘54982365’, ‘forks’)

INSERT
INTO Inventory2(amount, id, type)
VALUES (‘100’, ‘45217893’, ‘knives’)

INSERT
INTO Inventory2(amount, id, type)
VALUES (‘50’, ‘99875757’, ‘blankets’)
```

BelongsTo:

```
INSERT
INTO BelongsTo(id, reservation_id)
VALUES (‘123455’, ‘101234’)
```

```
INSERT
INTO BelongsTo(id, reservation_id)
VALUES (‘123456’, ‘101235’)

INSERT
INTO BelongsTo(id, reservation_id)
VALUES (‘123457’, ‘101236’)

INSERT
INTO BelongsTo(id, reservation_id)
VALUES (‘123458’, ‘101237’)

INSERT
INTO BelongsTo(id, reservation_id)
VALUES (‘123459’, ‘101238’)
```

Keeps:

```
INSERT
INTO Keeps(id, employee_id)
VALUES (‘12345678’, ‘123455’)

INSERT
INTO Keeps(id, employee_id)
VALUES (‘58349544’, ‘123456’)

INSERT
INTO Keeps(id, employee_id)
VALUES (‘54982365’, ‘123457’)

INSERT
INTO Keeps(id, employee_id)
VALUES (‘45217893’, ‘123458’)



INSERT
INTO Keeps(id, employee_id)
VALUES (‘99875757’, ‘123459’)
```

Checks_in1:

```
INSERT
INTO Checks_in1(date, employee_id)
VALUES (‘October 5, 2021’, ‘123455’)

INSERT
INTO Checks_in1(date, employee_id)
VALUES (‘October 6, 2021’, ‘123456’)

INSERT
INTO Checks_in1(date, employee_id)
VALUES (‘October 7, 2021’, ‘123456’)

INSERT
INTO Checks_in1(date, employee_id)
VALUES (‘October 8, 2021’, ‘123457’)

INSERT
INTO Checks_in1(date, employee_id)
VALUES (‘October 9, 2021’, ‘123457’)
```

Checks_in2:

```
INSERT
INTO Keeps(date, guest_id)
VALUES (‘October 5, 2021’, ‘123456’)

INSERT
INTO Keeps(date, guest_id)
VALUES (‘October 6, 2021’, ‘222222’)

INSERT
INTO Keeps(date, guest_id)
VALUES (‘October 7, 2021’, ‘333333’)
```

```
INSERT
INTO Keeps(date, guest_id)
VALUES (‘October 8, 2021’, ‘444444’)

INSERT
```

```
INTO Keeps(date, guest_id)
VALUES (‘October 9, 2021’, ‘666666’)
```

Receives:

```
INSERT
INTO Receives(id, employee_id)
VALUES(123455, 123455)

INSERT
INTO Receives(id, employee_id)
VALUES(123456, 123456)

INSERT
INTO Receives(id, employee_id)
VALUES(123457, 123457)

INSERT
INTO Receives(id, employee_id)
VALUES(123458, 123458)

INSERT
INTO Receives(id, employee_id)
VALUES(123459, 123459)
```

EmployeeManagesWorks:

```
INSERT
INTO EmployeeManagesWorks(employee_id, name, email, worker_type, manager_id, id)
VALUES (‘123455’,‘Alan’, ‘alan@email.com’, ‘Kitchen’, ‘123458’, ‘1234’)

INSERT
INTO EmployeeManagesWorks(employee_id, name, email, worker_type, manager_id, id)
VALUES (‘123456’,‘Bob’, ‘bob@email.com’, Front_Desk, ‘123458’, ‘3243’)



INSERT
INTO EmployeeManagesWorks(employee_id, name, email, worker_type, manager_id, id)
VALUES (‘123457’,‘Cole’, ‘cole@email.com’, ‘House_Keeping’, ‘123458’, ‘2342’)

INSERT
```

```
INTO EmployeeManagesWorks(employee_id, name, email, worker_type, manager_id, id)
VALUES (‘123458’,‘David’, ‘david@email.com’, Manager, ‘123458’, ‘4532’)

INSERT
INTO EmployeeManagesWorks(employee_id, name, email, worker_type, manager_id, id)
VALUES (‘123459’,‘Elliot’, ‘elliot@email.com’, ‘Kitchen’, ‘123458’, ‘8978’)
```

TransactionMadeTo1:

```
INSERT
INTO TransactionMadeTo1(hours_worked, amount, wage)
VALUES (‘1’, ‘10’, 15)

INSERT
INTO TransactionMadeTo1(hours_worked, amount, wage)
VALUES (‘2’, ‘11’, ‘16’)

INSERT
INTO TransactionMadeTo1(hours_worked, amount, wage)
VALUES (‘3’, ‘12’, ‘17)

INSERT
INTO TransactionMadeTo1(hours_worked, amount, wage)
VALUES (‘4’, ‘13’, ‘18’)

INSERT
INTO TransactionMadeTo1(hours_worked, amount, wage)
VALUES (‘5’, ’14’, ‘19’)
```

TransactionMadeTo2:

```
INSERT
INTO TransactionMadeTo2(id, date, type, period, hotel_id, amount, hours_worked)
VALUES (‘123455’, ’February 20, 2002’, ‘Room Payment’, ‘NULL’, ‘1234’, ‘9999’, NULL)



INSERT
INTO TransactionMadeTo2(id, date, type, period, hotel_id, amount, hours_worked)
VALUES (‘123456’, ’February 21, 2002’, Room Payment,   ‘NULL’, ‘3243’, ‘1000’, NULL,)

INSERT
```

```
INTO TransactionMadeTo2(id, date, type, period, hotel_id, amount, hours_worked)
VALUES (‘123457’,  ’February 22, 2002’, ‘Cancellation Fee’,   ‘NULL’, ’3243’, ‘3214’, ‘NULL’,)

INSERT
INTO TransactionMadeTo2(id, date, type, period, hotel_id, amount, hours_worked)
VALUES (‘123458’,  ’February 23, 2002’, ‘Cancellation Fee’,  ‘October, 10, 2002 - February, 10,
2002’, ‘1234’, ’13’, ‘4’)

INSERT
INTO TransactionMadeTo2(id, date, type, period, hotel_id, amount, hours_worked)
VALUES (‘123459’,  ’February 24, 2002’, ‘Cancellation Fee’,  ‘October, 10, 2002 - February, 10,
2002’, ‘3243’, ‘14’, ‘5’)
```