

PROJECT 2

Group 37

group members:
Min Jie
Erdong Luo
Numa Karolinski
Brian Cheng

Question1

Employee(eid , name, salary)

RestaurantManager(eid) (eid ref Employee)

Receptionist(eid) (eid ref Employee)

Housekeeper(eid) (eid ref Employee)

Restaurant(rname , type, eid) (eid ref RestaurantManager)

Customer(phone_number , cname, email_address)

Review(phone_number , date , time , rating, text) (phone_number ref Customer)

Room(room_number , type, room_status)

Reservation(rid , price_amount, start_date, end_date)

clean(room_number , eid) (room_number ref Room) (eid ref Housekeeper)

handle(rid , eid , phone_number) (rid ref Reservation) (eid ref Receptionist) (phone_number ref Customer)

belongs(rid , room_number) (rid ref Reservation) (room_number ref Room) dine(phone_number , rname , price) (phone_number ref Customer) (rname ref Restaurant)

check_in/out(room_number , eid , phone_number) (room_number ref Room) (eid ref Receptionist) (phone_number ref Customer)

Question 2

```
CREATE TABLE Employee(  
    eid INTEGER PRIMARY KEY,  
    name VARCHAR(30) NOT NULL,  
    salary INTEGER NOT NULL  
)
```

```
CREATE TABLE RestaurantManager(  
    eid INTEGER PRIMARY KEY,  
    FOREIGN KEY (eid)  
        REFERENCES Employee  
)
```

```
CREATE TABLE Receptionist(  
    eid INTEGER PRIMARY KEY,  
    FOREIGN KEY (eid)  
        REFERENCES Employee  
)
```

```
CREATE TABLE Housekeeper(  
    eid INTEGER PRIMARY KEY,  
    FOREIGN KEY (eid)  
        REFERENCES Employee  
)
```

```
CREATE TABLE Restaurant(  
    rname VARCHAR(30),  
    type VARCHAR(30) NOT NULL,  
    eid INTEGER NOT NULL,  
    PRIMARY KEY (rname),  
    FOREIGN KEY (eid)  
        REFERENCES RestaurantManager  
)
```

```
CREATE TABLE Meal(  
    mname VARCHAR(30),  
    rname VARCHAR(30) NOT NULL,  
    price DECIMAL(4, 2) NOT NULL,  
    PRIMARY KEY (mname),  
    FOREIGN KEY (rname)  
        REFERENCES Restaurant  
)
```

```
CREATE TABLE Customer(  
    phone_number BIGINT PRIMARY KEY ,  
    cname VARCHAR(30) NOT NULL,  
    email_address VARCHAR(50) NOT NULL  
)
```

```
CREATE TABLE Review(  
    phone_number BIGINT,  
    date DATE,  
    time TIME,
```

```
rating INTEGER,  
text VARCHAR(500),  
PRIMARY KEY(phone_number, date, time),  
FOREIGN KEY(phone_number)  
    REFERENCES Customer  
);
```

```
CREATE TABLE Room(  
room_number INTEGER PRIMARY KEY,  
type VARCHAR(20) NOT NULL,  
room_status VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE Reservation(  
rid INTEGER PRIMARY KEY,  
start_date DATE NOT NULL,  
end_date DATE NOT NULL,  
price_amount DECIMAL(7,2) NOT NULL  
);
```

```
CREATE TABLE Clean(  
room_number INTEGER,  
eid INTEGER,  
PRIMARY KEY (room_number, eid),  
FOREIGN KEY (room_number)  
    REFERENCES Room,  
FOREIGN KEY (eid)  
    REFERENCES Housekeeper  
);
```

```
CREATE TABLE Handle(  
rid INTEGER,  
eid INTEGER,  
phone_number BIGINT,  
PRIMARY KEY (rid, eid, phone_number),  
FOREIGN KEY (rid)  
    REFERENCES Reservation,  
FOREIGN KEY (eid)  
    REFERENCES Receptionist,  
FOREIGN KEY (phone_number)  
    REFERENCES Customer  
);
```

```
CREATE TABLE Belongs(  
rid INTEGER,  
room_number INTEGER,  
PRIMARY KEY(rid, room_number),  
FOREIGN KEY(rid)  
    REFERENCES Reservation,  
FOREIGN KEY(room_number)  
    REFERENCES Room  
);
```

```
CREATE TABLE Dine(
```

```
    phone_number BIGINT,  
    rname VARCHAR(30),  
    price DECIMAL(5, 2) NOT NULL,  
    PRIMARY KEY(phone_number, rname),  
    FOREIGN KEY(phone_number)  
        REFERENCES Customer,  
    FOREIGN KEY(rname)  
        REFERENCES Restaurant  
)
```

```
CREATE TABLE Check_in_out(  
    room_number INTEGER,  
    eid INTEGER,  
    phone_number BIGINT,  
    PRIMARY KEY(room_number, eid, phone_number),  
    FOREIGN KEY(room_number)  
        REFERENCES Room,  
    FOREIGN KEY(eid)  
        REFERENCES Receptionist,  
    FOREIGN KEY(phone_number)  
        REFERENCES Customer  
)
```

Question2 output

```
[cs421=> \d review
      Table "cs421g37.review"
Column | Type | Modifiers
-----+-----+-----
phone_number | integer | not null
date | date | not null
time | time without time zone | not null
rating | integer |
text | character varying(512) |
Indexes:
"review_pkey" PRIMARY KEY, btree (phone_number, date, "time")
Foreign-key constraints:
"review_phone_number_fkey" FOREIGN KEY (phone_number) REFERENCES customer(phone_number)
```

```
[cs421=> \d restaurantmanager
Table "cs421g37.restaurantmanager"
Column | Type | Modifiers
-----+-----+-----
eid | integer | not null
Indexes:
"restaurantmanager_pkey" PRIMARY KEY, btree (eid)
Foreign-key constraints:
"restaurantmanager_eid_fkey" FOREIGN KEY (eid) REFERENCES employee(eid)
Referenced by:
  TABLE "restaurant" CONSTRAINT "restaurant_eid_fkey" FOREIGN KEY (eid) REFERENCES restaurantmanager(eid)
```

```
[cs421=> \d restaurant
      Table "cs421g37.restaurant"
Column | Type | Modifiers
-----+-----+-----
rname | character varying(30) | not null
type | character varying(30) | not null
eid | integer | not null
Indexes:
"restaurant_pkey" PRIMARY KEY, btree (rname)
Foreign-key constraints:
"restaurant_eid_fkey" FOREIGN KEY (eid) REFERENCES restaurantmanager(eid)
Referenced by:
  TABLE "dine" CONSTRAINT "dine_rname_fkey" FOREIGN KEY (rname) REFERENCES restaurant(rname)
  TABLE "meal" CONSTRAINT "meal_rname_fkey" FOREIGN KEY (rname) REFERENCES restaurant(rname)
```

```
[cs421=> \d employee
      Table "cs421g37.employee"
Column | Type | Modifiers
-----+-----+
eid | integer | not null
name | character varying(30) | not null
salary | integer | not null
Indexes:
  "employee_pkey" PRIMARY KEY, btree (eid)
Referenced by:
  TABLE "housekeeper" CONSTRAINT "housekeeper_eid_fkey" FOREIGN KEY (eid) REFERENCES employee(eid)
  TABLE "receptionist" CONSTRAINT "receptionist_eid_fkey" FOREIGN KEY (eid) REFERENCES employee(eid)
  TABLE "restaurantmanager" CONSTRAINT "restaurantmanager_eid_fkey" FOREIGN KEY (eid) REFERENCES employee(eid)
```

```
[cs421=> \d receptionist
      Table "cs421g37.receptionist"
Column | Type | Modifiers
-----+-----+
eid | integer | not null
Indexes:
  "receptionist_pkey" PRIMARY KEY, btree (eid)
Foreign-key constraints:
  "receptionist_eid_fkey" FOREIGN KEY (eid) REFERENCES employee(eid)
Referenced by:
  TABLE "handle" CONSTRAINT "handle_eid_fkey" FOREIGN KEY (eid) REFERENCES receptionist(eid)
```

```
[cs421=> \d handle
      Table "cs421g37.handle"
Column | Type | Modifiers
-----+-----+
rid | integer | not null
eid | integer | not null
phone_number | integer | not null
Indexes:
  "handle_pkey" PRIMARY KEY, btree (rid, eid, phone_number)
Foreign-key constraints:
  "handle_eid_fkey" FOREIGN KEY (eid) REFERENCES receptionist(eid)
  "handle_phone_number_fkey" FOREIGN KEY (phone_number) REFERENCES customer(phone_number)
  "handle_rid_fkey" FOREIGN KEY (rid) REFERENCES reservation(rid)
```

```
[cs421=> \d housekeeper
Table "cs421g37.housekeeper"
Column | Type | Modifiers
-----+-----+
eid | integer | not null
Indexes:
"housekeeper_pkey" PRIMARY KEY, btree (eid)
Foreign-key constraints:
"housekeeper_eid_fkey" FOREIGN KEY (eid) REFERENCES employee(eid)
Referenced by:
TABLE "clean" CONSTRAINT "clean_eid_fkey" FOREIGN KEY (eid) REFERENCES house
keeper(eid)
```

```
[cs421=> \d clean
Table "cs421g37.clean"
Column | Type | Modifiers
-----+-----+
room_number | integer | not null
eid | integer | not null
Indexes:
"clean_pkey" PRIMARY KEY, btree (room_number, eid)
Foreign-key constraints:
"clean_eid_fkey" FOREIGN KEY (eid) REFERENCES housekeeper(eid)
"clean_room_number_fkey" FOREIGN KEY (room_number) REFERENCES room(room_numb
er)
```

```
[cs421=> \d room
Table "cs421g37.room"
Column | Type | Modifiers
-----+-----+
room_number | integer | not null
type | character varying(20) | not null
room_status | character varying(20) | not null
Indexes:
"room_pkey" PRIMARY KEY, btree (room_number)
Referenced by:
TABLE "belongs" CONSTRAINT "belongs_room_number_fkey" FOREIGN KEY (room_numb
er) REFERENCES room(room_number)
TABLE "clean" CONSTRAINT "clean_room_number_fkey" FOREIGN KEY (room_number)
REFERENCES room(room_number)
```

```
[cs421=> \d reservation
      Table "cs421g37.reservation"
      Column | Type | Modifiers
-----+-----+-----+
    rid | integer | not null
start_date | date | not null
end_date | date | not null
price_amount | numeric(7,2) | not null
Indexes:
    "reservation_pkey" PRIMARY KEY, btree (rid)
Referenced by:
    TABLE "belongs" CONSTRAINT "belongs_rid_fkey" FOREIGN KEY (rid) REFERENCES reservation(rid)
    TABLE "handle" CONSTRAINT "handle_rid_fkey" FOREIGN KEY (rid) REFERENCES reservation(rid)
```

```
[cs421=> \d meal
      Table "cs421g37.meal"
      Column | Type | Modifiers
-----+-----+-----+
  rname | character varying(30) | not null
  mname | character varying(30) | not null
  price | numeric(4,2) | not null
Indexes:
    "meal_pkey" PRIMARY KEY, btree (rname, mname)
Foreign-key constraints:
    "meal_rname_fkey" FOREIGN KEY (rname) REFERENCES restaurant(rname)
```

```
[cs421=> \d belongs
      Table "cs421g37.belongs"
      Column | Type | Modifiers
-----+-----+-----+
    rid | integer | not null
room_number | integer | not null
Indexes:
    "belongs_pkey" PRIMARY KEY, btree (rid, room_number)
Foreign-key constraints:
    "belongs_rid_fkey" FOREIGN KEY (rid) REFERENCES reservation(rid)
    "belongs_room_number_fkey" FOREIGN KEY (room_number) REFERENCES room(room_number)
```

```
[cs421=> \d dine
      Table "cs421g37.dine"
 Column |          Type          | Modifiers
-----+-----+-----+
 phone_number | integer | not null
 rname | character varying(30) | not null
 price | numeric(5,2) | not null
Indexes:
 "dine_pkey" PRIMARY KEY, btree (phone_number, rname)
Foreign-key constraints:
 "dine_phone_number_fkey" FOREIGN KEY (phone_number) REFERENCES customer(phone_number)
 "dine_rname_fkey" FOREIGN KEY (rname) REFERENCES restaurant(rname)]
```

```
[cs421=> \d customer
      Table "cs421g37.customer"
 Column |          Type          | Modifiers
-----+-----+-----+
 phone_number | integer | not null
 cname | character varying(30) | not null
 email_address | character varying(30) | not null
Indexes:
 "customer_pkey" PRIMARY KEY, btree (phone_number)
Referenced by:
 TABLE "dine" CONSTRAINT "dine_phone_number_fkey" FOREIGN KEY (phone_number)
 REFERENCES customer(phone_number)
 TABLE "handle" CONSTRAINT "handle_phone_number_fkey" FOREIGN KEY (phone_number)
 REFERENCES customer(phone_number)
 TABLE "review" CONSTRAINT "review_phone_number_fkey" FOREIGN KEY (phone_number)
 REFERENCES customer(phone_number)]
```

Question 3 & 4:

Employee:

| | eid | name | salary |
|----|------|-----------------------|--------|
| 1 | 1663 | Jonathon Lee | 74115 |
| 2 | 1357 | Giannis Antetokounmpo | 56700 |
| 3 | 1316 | Toby Lerone | 22222 |
| 4 | 1738 | Robert Rutledge | 45660 |
| 5 | 1358 | Jane Sturges | 36945 |
| 6 | 1317 | Nicholas Hesse | 26897 |
| 7 | 1739 | Dorcus Lewis | 77749 |
| 8 | 1665 | Sabine Heartilly | 22875 |
| 9 | 1985 | Sander Mercia | 39483 |
| 10 | 1359 | Jack Jaramillo | 44485 |
| 11 | 1318 | Mani Najera | 44689 |
| 12 | 1740 | Christian Stalin | 93326 |
| 13 | 1666 | Sarah Fanqula | 0 |
| 14 | 1986 | Summer Christensen | 89482 |
| 15 | 1360 | Amy Blake | 47645 |
| 16 | 1319 | Mary Dolabella | 73834 |
| 17 | 1741 | Laura Exley | 32465 |
| 18 | 1984 | Jenna Maul | 54956 |
| 19 | 1983 | Mos Def | 96066 |
| 20 | 1664 | Stevie Regis | 37967 |

```

SELECT * FROM restaurant;
SELECT * FROM belongs;
SELECT * FROM check_in_out;
SELECT * FROM clean;
SELECT * FROM customer;
SELECT * FROM dine;
SELECT * FROM employee;
SELECT * FROM housekeeper;
SELECT * FROM receptionist;
SELECT * FROM restaurantmanager;
SELECT * FROM handle;
SELECT * FROM review;
SELECT * FROM room;

```

RestaurantManager:

| | eid |
|---|------|
| 1 | 1663 |
| 2 | 1983 |
| 3 | 1357 |
| 4 | 1316 |
| 5 | 1738 |

Receptionist:

| | eid |
|---|------|
| 1 | 1664 |
| 2 | 1984 |
| 3 | 1358 |
| 4 | 1317 |
| 5 | 1739 |

Housekeeper:

| | eid |
|----|------|
| 1 | 1665 |
| 2 | 1985 |
| 3 | 1359 |
| 4 | 1318 |
| 5 | 1740 |
| 6 | 1666 |
| 7 | 1986 |
| 8 | 1360 |
| 9 | 1319 |
| 10 | 1741 |

Restaurant:

| | rname | type | eid |
|---|--------------------|----------------|------|
| 1 | Johns Bistro | Classic Bistro | 1663 |
| 2 | Spittin Fire | Spicy Food | 1983 |
| 3 | Santorini Souvlaki | Greek Food | 1357 |
| 4 | La Raclette | Swiss Food | 1316 |
| 5 | Baracuda Bobs | Mexican Brunch | 1738 |

Customer:

| | phone_number | cname | email_address |
|----|--------------|-------------------|-----------------------------------|
| 1 | 5204392847 | Santiago Rahl | s.rahl@gmail.com |
| 2 | 5208379384 | Hugo Mitchell | hugomitchell96024@gmail.com |
| 3 | 5204938758 | Samuel Campbell | bellcampFierce@live.com |
| 4 | 5143497231 | Juicer Shadow | shadowBoy01@aol.com |
| 5 | 3139812321 | Marshall Mathers | slimshady@aol.com |
| 6 | 5203428469 | Numa Karolinski | numa.karolinski@mail.mcgill.ca |
| 7 | 5149834734 | Jennifer Stukeley | jennyjenny33@hotmail.com |
| 8 | 5085488991 | Jan Stark | kingofthenorth@mail.mcgill.ca |
| 9 | 5085429999 | Earle Joker | notbatman@gmail.com |
| 10 | 5207778883 | Eugene Debbs | socialistpartyofamerica@gmail.com |

Review:

| | phone_number | date | time | rating | text |
|---|--------------|------------------------------|-----------------------------|--------|--|
| 1 | 5204392847 | Dec. 30, 2019, 12:00:00 a.m. | Jan. 1, 1970, 11:34:12 p.m. | 4 | |
| 2 | 5208379384 | Dec. 30, 2019, 12:00:00 a.m. | Jan. 1, 1970, 9:31:02 p.m. | | I enjoyed all the choice of Restaurants. I had a really good b |
| 3 | 5204938758 | Dec. 28, 2019, 12:00:00 a.m. | Jan. 1, 1970, 8:11:12 a.m. | 5 | Great Stay. Very comfortable bed. |
| 4 | 5085429999 | Jan. 2, 2010, 12:00:00 a.m. | Jan. 1, 1970, 5:51:55 a.m. | 5 | |

Room:

| | room_number | type | room_status |
|----|-------------|-------|-------------|
| 1 | 201 | King | OCC |
| 2 | 305 | King | V |
| 3 | 403 | Suite | V |
| 4 | 203 | King | OCC |
| 5 | 204 | Queen | OCC |
| 6 | 302 | Queen | OCC |
| 7 | 402 | Suite | OCC |
| 8 | 405 | King | OCC |
| 9 | 407 | Queen | OCC |
| 10 | 303 | Queen | OCC |

Reservation:

| | rid | start_date | end_date | price_amount |
|----|---------|------------------------------|------------------------------|--------------|
| 1 | 1001234 | Mar. 1, 2020, 12:00:00 a.m. | Mar. 5, 2020, 12:00:00 a.m. | 932.31 |
| 2 | 1000001 | Mar. 2, 2020, 12:00:00 a.m. | Mar. 4, 2020, 12:00:00 a.m. | 453.16 |
| 3 | 1000003 | Mar. 9, 2020, 12:00:00 a.m. | Mar. 22, 2020, 12:00:00 a.m. | 3029.52 |
| 4 | 1000005 | Mar. 9, 2020, 12:00:00 a.m. | Mar. 11, 2020, 12:00:00 a.m. | 532.98 |
| 5 | 1000008 | Mar. 11, 2020, 12:00:00 a.m. | Mar. 12, 2020, 12:00:00 a.m. | 287.21 |
| 6 | 1000010 | Mar. 11, 2020, 12:00:00 a.m. | Mar. 13, 2020, 12:00:00 a.m. | 572.48 |
| 7 | 1000011 | Mar. 12, 2020, 12:00:00 a.m. | Mar. 14, 2020, 12:00:00 a.m. | 498.64 |
| 8 | 1000013 | Mar. 12, 2020, 12:00:00 a.m. | Mar. 15, 2020, 12:00:00 a.m. | 694.31 |
| 9 | 1000015 | Mar. 13, 2020, 12:00:00 a.m. | Mar. 16, 2020, 12:00:00 a.m. | 643.98 |
| 10 | 1000018 | Mar. 15, 2020, 12:00:00 a.m. | Mar. 20, 2020, 12:00:00 a.m. | 1134.13 |

Clean:

| | room_number | eid |
|----|-------------|------|
| 1 | 201 | 1985 |
| 2 | 203 | 1665 |
| 3 | 204 | 1359 |
| 4 | 302 | 1986 |
| 5 | 303 | 1318 |
| 6 | 305 | 1740 |
| 7 | 402 | 1666 |
| 8 | 403 | 1319 |
| 9 | 405 | 1741 |
| 10 | 407 | 1985 |

Handle:

| | rid | eid | phone_number |
|----|---------|------|--------------|
| 1 | 1001234 | 1664 | 5208379384 |
| 2 | 1000001 | 1664 | 5203428469 |
| 3 | 1000003 | 1358 | 5207778883 |
| 4 | 1000005 | 1317 | 5203428469 |
| 5 | 1000015 | 1739 | 3139812321 |
| 6 | 1000018 | 1984 | 5085488991 |
| 7 | 1000013 | 1664 | 5204938758 |
| 8 | 1000010 | 1984 | 5085488991 |
| 9 | 1000008 | 1317 | 5149834734 |
| 10 | 1000011 | 1739 | 5143497231 |

Belongs:

| | rid | room_number |
|----|---------|-------------|
| 1 | 1001234 | 201 |
| 2 | 1001234 | 203 |
| 3 | 1001234 | 204 |
| 4 | 1000015 | 302 |
| 5 | 1000001 | 305 |
| 6 | 1000003 | 402 |
| 7 | 1000005 | 403 |
| 8 | 1000005 | 405 |
| 9 | 1000008 | 407 |
| 10 | 1000008 | 303 |

Dine:

| | phone_number | rname | price |
|---|--------------|--------------------|--------|
| 1 | 5204392847 | Santorini Souvlaki | 43.87 |
| 2 | 5208379384 | Santorini Souvlaki | 52.93 |
| 3 | 5208379384 | Spittin Fire | 44.23 |
| 4 | 5203428469 | Baracuda Bobs | 78.9 |
| 5 | 5085429999 | La Raclette | 153.98 |
| 6 | 5207778883 | Spittin Fire | 59.45 |

Check in/out:

| | room_number | eid | phone_number |
|---|-------------|------|--------------|
| 1 | 201 | 1664 | 5204392847 |
| 2 | 203 | 1664 | 5208379384 |
| 3 | 204 | 1984 | 5204392847 |
| 4 | 302 | 1358 | 5204938758 |
| 5 | 303 | 1317 | 5143497231 |
| 6 | 402 | 1317 | 5085488991 |
| 7 | 407 | 1739 | 5207778883 |
| 8 | 405 | 1358 | 5085429999 |

Question 3&4

```
INSERT INTO Restaurant VALUES(  
    'Johns Bistro', 'Classic Bistro', 1663  
)
```

```
INSERT INTO Restaurant VALUES(  
    'Spittin Fire', 'Spicy Food', 1983  
)
```

```
INSERT INTO Restaurant VALUES(  
    'Santorini Souvlaki', 'Greek Food', 1357  
)
```

```
INSERT INTO Restaurant VALUES(  
    'La Raclette', 'Swiss Food', 1316  
)
```

```
INSERT INTO Restaurant VALUES(  
    'Baracuda Bobs', 'Mexican Brunch', 1738  
)
```

```
INSERT INTO Employee VALUES (  
    1663, 'Jonathon Lee', 74115  
)
```

```
INSERT INTO Employee VALUES (  
    1983, 'Mos Def', 96024  
)
```

```
INSERT INTO Employee VALUES (  
    1357, 'Giannis Antetokounmpo', 56700  
)
```

```
INSERT INTO Employee VALUES (  
    1316, 'Toby Lerone', 22222  
)
```

```
INSERT INTO Employee VALUES(  
    1738, 'Robert Rutledge', 45660  
)
```

```
INSERT INTO Employee VALUES (  
    1664, 'Stevie Regis', 75935  
)
```

```
INSERT INTO Employee VALUES (  
    1984, 'Jenna Maul', 98583  
)
```

```
INSERT INTO Employee VALUES (  
    1358, 'Jane Sturges', 36945  
)
```

```
INSERT INTO Employee VALUES (  
    1317, 'Nicholas Hesse', 26897  
)
```

```
INSERT INTO Employee VALUES(  
1739, 'Dorcas Lewis', 77749  
);
```

```
INSERT INTO Employee VALUES (  
1665, 'Sabine Heartilly', 22875  
);
```

```
INSERT INTO Employee VALUES (  
1985, 'Sander Mercia', 39483  
);
```

```
INSERT INTO Employee VALUES (  
1359, 'Jack Jaramillo', 44485  
);
```

```
INSERT INTO Employee VALUES (  
1318, 'Mani Najera', 44689  
);
```

```
INSERT INTO Employee VALUES(  
1740, 'Christian Stalin', 93326  
);
```

```
INSERT INTO Employee VALUES (  
1666, 'Sarah Fangula', 0  
);
```

```
INSERT INTO Employee VALUES (  
1986, 'Summer Christensen', 89482  
);
```

```
INSERT INTO Employee VALUES (  
1360, 'Amy Blake', 47645  
);
```

```
INSERT INTO Employee VALUES (  
1319, 'Mary Dolabella', 73834  
);
```

```
INSERT INTO Employee VALUES(  
1741, 'Laura Exley', 32465  
);
```

```
INSERT INTO RestaurantManager VALUES(  
1663  
);
```

```
INSERT INTO RestaurantManager VALUES(  
1983  
);
```

```
INSERT INTO RestaurantManager VALUES(
```

1357
);

INSERT INTO RestaurantManager VALUES(
1316
);

INSERT INTO RestaurantManager VALUES(
1738
);

INSERT INTO Receptionist VALUES(
1664
);

INSERT INTO Receptionist VALUES(
1984
);

INSERT INTO Receptionist VALUES(
1358
);

INSERT INTO Receptionist VALUES(
1317
);

INSERT INTO Receptionist VALUES(
1739
);

INSERT INTO Housekeeper VALUES(
1665
);

INSERT INTO Housekeeper VALUES(
1985
);

INSERT INTO Housekeeper VALUES(
1359
);

INSERT INTO Housekeeper VALUES(
1318
);

INSERT INTO Housekeeper VALUES(
1740
);

INSERT INTO Housekeeper VALUES(
1666
);

```
INSERT INTO Housekeeper VALUES(
1986
);

INSERT INTO Housekeeper VALUES(
1360
);

INSERT INTO Housekeeper VALUES(
1319
);

INSERT INTO Housekeeper VALUES(
1741
);

INSERT INTO Customer VALUES(
5204392847, 'Santiago Rahl', 's.rahl@gmail.com'
);

INSERT INTO Customer VALUES(
5208379384, 'Hugo Mitchell', 'hugomitchell96024@gmail.com'
);

INSERT INTO Customer VALUES(
5204938758, 'Samuel Campbell', 'bellcampFierce@live.com'
);

INSERT INTO Customer VALUES(
5143497231, 'Juicer Shadow', 'shadowBoy01@aol.com'
);

INSERT INTO Customer VALUES(
3139812321, 'Marshall Mathers', 'slimshady@aol.com'
);

INSERT INTO Customer VALUES(
5203428469, 'Numa Karolinski', 'numa.karolinski@mail.mcgill.ca'
);

INSERT INTO Customer VALUES(
5149834734, 'Jennifer Stukeley', 'jennyjenny33@hotmail.com'
);

INSERT INTO Customer VALUES(
5085488991, 'Jan Stark', 'kingofthenorth@mail.mcgill.ca'
);

INSERT INTO Customer VALUES(
5085429999, 'Earle Joker', 'notbatman@gmail.com'
);

INSERT INTO Customer VALUES(
```

5207778883, 'Eugene Debbs', 'socialistpartyofamerica@gmail.com'
);

INSERT INTO Review (phone_number, date, time, rating) VALUES(
5204392847, '2019-12-30', '23:34:12', 4
);

INSERT INTO Review (phone_number, date, time, text) VALUES(
5208379384, '2019-12-30', '21:31:02', 'I enjoyed all the choice of Restaurants. I had a really good burrito at Baracuda Bobs, and the manager was quite lovely. He taught me a lot about Neutron Stars.'
);

INSERT INTO Review (phone_number, date, time, rating, text) VALUES(
5204938758, '2019-12-28', '08:11:12', 5, 'Great Stay. Very comfortable bed.'
);

INSERT INTO Review (phone_number, date, time, rating) VALUES(
5143497231, '2019-12-31', '11:43:34', 2
);

INSERT INTO Review (phone_number, date, time, rating) VALUES(
5085429999, '2010-01-02', '05:51:55', 5
);

INSERT INTO Clean VALUES(
201, 1985
);

INSERT INTO Clean VALUES(
203, 1665
);

INSERT INTO Clean VALUES(
204, 1359
);

INSERT INTO Clean VALUES(
302, 1986
);

INSERT INTO Clean VALUES(
303, 1318
);

INSERT INTO Clean VALUES(
305, 1740
);

INSERT INTO Clean VALUES(
402, 1666
);

INSERT INTO Clean VALUES(
);

403, 1319
);

INSERT INTO Clean VALUES(
405, 1741
);

INSERT INTO Clean VALUES(
407, 1985
);

INSERT INTO Handle VALUES(
1001234, 1664, 5208379384
);

INSERT INTO Handle VALUES(
1000001, 1664, 5203428469
);

INSERT INTO Handle VALUES(
1000003, 1358, 5207778883
);

INSERT INTO Handle VALUES(
1000005, 1317, 5203428469
);

INSERT INTO Handle VALUES(
1000015, 1739, 3139812321
);

INSERT INTO Handle VALUES(
1000018, 1984, 5085488991
);

INSERT INTO Handle VALUES(
1000013, 1664, 5204938758
);

INSERT INTO Handle VALUES(
1000010, 1984, 5085488991
);

INSERT INTO Handle VALUES(
1000008, 1317, 5149834734
);

INSERT INTO Handle VALUES(
1000011, 1739, 5143497231
);

INSERT INTO Belongs VALUES(
1001234, 201
);

INSERT INTO Belongs VALUES(
1001234, 203
);

```
INSERT INTO Belongs VALUES(  
1001234, 204  
);
```

```
INSERT INTO Belongs VALUES(  
1000015, 302  
);
```

```
INSERT INTO Belongs VALUES(  
1000001, 305  
);
```

```
INSERT INTO Belongs VALUES(  
1000003, 402  
);
```

```
INSERT INTO Belongs VALUES(  
1000005, 403  
);
```

```
INSERT INTO Belongs VALUES(  
1000005, 405  
);
```

```
INSERT INTO Belongs VALUES(  
1000008, 407  
);
```

```
INSERT INTO Belongs VALUES(  
1000008, 303  
);
```

```
INSERT INTO Dine VALUES(  
5204392847, 'Santorini Souvlaki', 43.87  
);
```

```
INSERT INTO Dine Values(  
5208379384, 'Santorini Souvlaki', 52.93  
);
```

```
INSERT INTO Dine VALUES(  
5208379384, 'Spittin Fire', 44.23  
);
```

```
INSERT INTO Dine VALUES(  
5203428469, 'Baracuda Bobs', 78.90  
);
```

```
INSERT INTO Dine VALUES(  
5085429999, 'La Raclette', 153.98  
);
```

```
INSERT INTO Dine VALUES(
```

```
5207778883, 'Spittin Fire', 59.45
);

INSERT INTO Check_in_out VALUES(
201, 1664, 5204392847
);

INSERT INTO Check_in_out VALUES(
203, 1664, 5208379384
);

INSERT INTO Check_in_out VALUES(
204, 1984, 5204392847
);

INSERT INTO Check_in_out VALUES(
302, 1358, 5204938758
);

INSERT INTO Check_in_out VALUES(
303, 1317, 5143497231
);

INSERT INTO Check_in_out VALUES(
402, 1317, 5085488991
);

INSERT INTO Check_in_out VALUES(
407, 1739, 5207778883
);
INSERT INTO Check_in_out VALUES(
405, 1358, 5085429999
);

INSERT INTO room VALUES(
201, 'King', 'OCC')
;
INSERT INTO room VALUES(
203,'King','OCC')
;
INSERT INTO room VALUES(
204,'Queen', 'OCC')
;
INSERT INTO room VALUES(
302,'Queen', 'OCC')
;
INSERT INTO room VALUES(
303,'Queen','OCC')
;
INSERT INTO room VALUES(
305,'King', 'V')
;
INSERT INTO room VALUES(
402,'Suite','OCC')
```

```
;  
INSERT INTO room VALUES(  
403,'Suite','V')  
;  
INSERT INTO room VALUES(  
405,'King','OCC')  
;  
INSERT INTO room VALUES(  
407,'Queen','OCC')  
;  
  
INSERT INTO reservation VALUES(  
1001234, '2020-03-01','2020-03-05','932.31'  
);  
INSERT INTO reservation VALUES(  
1000001,'2020-03-02','2020-03-04','453.16'  
);  
INSERT INTO reservation VALUES(  
1000003,'2020-03-09','2020-03-22','3029.52'  
);  
INSERT INTO reservation VALUES(  
1000005,'2020-03-09','2020-03-11','532.98'  
);  
INSERT INTO reservation VALUES(  
1000008,'2020-03-11','2020-03-12','287.21'  
);  
INSERT INTO reservation VALUES(  
1000010,'2020-03-11','2020-03-13','572.48'  
);  
INSERT INTO reservation VALUES(  
1000011,'2020-03-12','2020-03-14','498.64'  
);  
INSERT INTO reservation VALUES(  
1000013,'2020-03-12','2020-03-15','694.31'  
);  
INSERT INTO reservation VALUES(  
1000015,'2020-03-13','2020-03-16','643.98'  
);  
INSERT INTO reservation VALUES(  
1000018,'2020-03-15','2020-03-20','1134.13'  
);
```

```
1 -- Here contains all the INSERT statements ran in a single run,
2 -- statements were reordered to avoid foreign key constraint errors
3
4 INSERT INTO Employee VALUES (
5 1663, 'Jonathon Lee', 74115
6 );
7
8 INSERT INTO Employee VALUES (
9 1983, 'Mos Def', 96024
10 );
11
12 INSERT INTO Employee VALUES (
13 1357, 'Giannis Antetokounmpo', 56700
14 );
15
16 INSERT INTO Employee VALUES (
17 1316, 'Toby Lerone', 22222
18 );
19
20 INSERT INTO Employee VALUES (
```

Data Output Explain **Messages** Notifications

INSERT 0 1

Query returned successfully in 92 msec.

Question 5

1.

```
1 --select name of the manager of the restaurant that Earle Joker has dined in
2 SELECT name FROM employee WHERE eid IN
3 (SELECT eid FROM restaurant WHERE rname IN
4 (SELECT rname FROM customer c JOIN dine d ON
5 c.phone_number=d.phone_number AND c.cname='Earle Joker'));
```

Data Output Explain Messages Notifications

| | name character varying (30) | 🔒 |
|---|--------------------------------|---|
| 1 | Toby Lerone | |

2.

```
1 --select the customer name who has checked in the room and dine in the restaurant
2 SELECT cname FROM customer WHERE phone_number IN
3 ((SELECT customer.phone_number from dine
4   JOIN customer ON customer.phone_number=dine.phone_number)
5 INTERSECT
6 (SELECT customer.phone_number from check_in_out
7   JOIN customer ON customer.phone_number=check_in_out.phone_number))
8 ;
9
10 |
```

Data Output Explain Messages Notifications

| | cname character varying (30) | 🔒 |
|---|---------------------------------|---|
| 1 | Santiago Rahl | |
| 2 | Eugene Debbs | |
| 3 | Hugo Mitchell | |
| 4 | Earle Joker | |

3.

```
1 --select the #rooms that the housekeepers, whose salary>40000, cleaned
2 SELECT eid, count(room_number) AS num_of_cleaned_room FROM clean WHERE eid IN
3 (SELECT e.eid FROM housekeeper h JOIN employee e ON h.eid=e.eid WHERE salary>40000)
4 GROUP BY eid
5 ORDER BY eid;
```

Data Output Explain Messages Notifications

| | eid integer | num_of_cleaned_room bigint |
|---|----------------|-------------------------------|
| 1 | 1318 | 1 |
| 2 | 1319 | 1 |
| 3 | 1359 | 1 |
| 4 | 1740 | 1 |
| 5 | 1986 | 1 |

4.

```
1 --select the name of the customers who has a reservation's price more than 1000
2 SELECT cname FROM customer WHERE phone_number IN
3 (SELECT phone_number FROM handle WHERE rid IN
4 (SELECT rid FROM reservation WHERE price_amount>1000));
5
6
```

Data Output Explain Messages Notifications

| | cname character varying (30) |
|---|---------------------------------|
| 1 | Jan Stark |
| 2 | Eugene Debbs |

5. & Question9 Complex AnalyticalQueries

```
7 -- QUESTION 5 :  
8 -- get the percentage of customers in the hotel who dines in  
9 -- 'Baracuda Bobs' and the sum of money people spend there  
10 -- multiply the two values to get a 'score' for this restaurant  
11 -- this query could be useful when it comes to evaluating a restaurant.  
12 -- The score is balanced by the profit of the restaurant as well as  
13 -- the popularity. If the hotel decides to close down a restaurant or replace  
14 -- one with another, they can make choices based on this score.  
15 SELECT SUM(price)*(SELECT CAST(COUNT(DISTINCT phone_number) as float)/  
16           (SELECT CAST((SELECT COUNT(phone_number) FROM Customer as cus)as float))  
17           FROM Dine WHERE (rname='Baracuda Bobs')) FROM Dine  
18           WHERE rname='Baracuda Bobs';  
^~
```

Data Output Explain Messages Notifications

| | ?column? double precision | ? |
|---|------------------------------|---|
| 1 | 7.89 | |

Question 6

1).

The function:

```
20 -- QUESTION 6:  
21 -- half the salary of the employee who handled the reservation 10001234  
22 UPDATE Employee  
23 SET salary=(SELECT salary/2 FROM Employee  
24     WHERE eid=(SELECT eid FROM Handle WHERE rid=1001234))  
25 WHERE eid=(SELECT eid FROM Handle WHERE rid=1001234);  
26
```

Data Output Explain Messages Notifications

UPDATE 1

Query returned successfully in 63 msec.

Before modification:

```
16  
17 -- Before modification:  
18 SELECT * FROM Employee WHERE eid=1664;  
19
```

Data Output Explain Messages Notifications

| | eid [PK] integer | name character varying (30) | salary integer | |
|---|---------------------|--------------------------------|-------------------|--|
| 1 | 1664 | Stevie Regis | 75935 | |

After modification:

```
27 -- after modification  
28 SELECT * FROM Employee WHERE eid=1664;  
29
```

Data Output Explain Messages Notifications

| | eid [PK] integer | name character varying (30) | salary integer | |
|---|---------------------|--------------------------------|-------------------|--|
| 1 | 1664 | Stevie Regis | 37967 | |

2).

The function:

```
21 -- raise the salary of the restaurant manager of Spittin Fire
22 -- by 20% of the sum of the restaurant's income
23 UPDATE Employee
24 SET salary=(SELECT salary+(SELECT sum(price)*0.2 FROM Dine WHERE rname='Spittin Fire')
25           FROM (SELECT * FROM Employee WHERE
26                   eid=(SELECT eid FROM Restaurant
27                         WHERE rname='Spittin Fire'))as SFmngr)
28 WHERE eid=(SELECT eid FROM Restaurant WHERE rname='Spittin Fire');
```

Data Output Explain Messages Notifications

UPDATE 1

Query returned successfully in 230 msec.

Before modification:

```
17 -- before modification
18 SELECT * FROM Employee where eid=1983;
```

Data Output Explain Messages Notifications

| | eid [PK] integer | name character varying (30) | salary integer |
|---|---------------------|--------------------------------|-------------------|
| 1 | 1983 | Mos Def | 96024 |

```
19 SELECT SUM(price) FROM Dine WHERE rname='Spittin Fire';
```

Data Output Explain Messages Notifications

| | sum numeric |
|---|----------------|
| 1 | 103.68 |

After modification:

```
30 -- after modification:
31 SELECT * FROM Employee where eid=1983;
```

Data Output Explain Messages Notifications

| | eid [PK] integer | name character varying (30) | salary integer |
|---|---------------------|--------------------------------|-------------------|
| 1 | 1983 | Mos Def | 96045 |

3).

The function:

```
21 -- delete all the reviews that are under 2 in 2019
22 DELETE FROM review WHERE (rating BETWEEN 0 AND 2)
23 AND (date BETWEEN '2019-01-01' AND '2019-12-31');
```

Data Output Explain Messages Notifications

DELETE 1

Query returned successfully in 83 msec.

Before modification:

```
17 -- before modification
18 SELECT * FROM review;
```

Data Output Explain Messages Notifications

| | phone_number [PK] bigint | date [PK] date | time [PK] time without time zone | rating integer | text character varying (500) |
|---|-----------------------------|-------------------|-------------------------------------|-------------------|------------------------------------|
| 1 | 5204392847 | 2019-12-30 | 23:34:12 | 4 | [null] |
| 2 | 5208379384 | 2019-12-30 | 21:31:02 | [null] | I enjoyed all the choice of Res... |
| 3 | 5204938758 | 2019-12-28 | 08:11:12 | 5 | Great Stay. Very comfortable ... |
| 4 | 5085429999 | 2010-01-02 | 05:51:55 | 5 | [null] |
| 5 | 5143497231 | 2019-12-31 | 11:43:34 | 2 | [null] |

Note that there is a null value in the table, but this was before the CHECK constraint was implemented. This value will cause an error later

After modification:

```
25 -- after modification
26 SELECT * FROM review;
```

Data Output Explain Messages Notifications

| | phone_number [PK] bigint | date [PK] date | time [PK] time without time zone | rating integer | text character varying (500) |
|---|-----------------------------|-------------------|-------------------------------------|-------------------|------------------------------------|
| 1 | 5204392847 | 2019-12-30 | 23:34:12 | 4 | [null] |
| 2 | 5208379384 | 2019-12-30 | 21:31:02 | [null] | I enjoyed all the choice of Res... |
| 3 | 5204938758 | 2019-12-28 | 08:11:12 | 5 | Great Stay. Very comfortable ... |
| 4 | 5085429999 | 2010-01-02 | 05:51:55 | 5 | [null] |

4).

The function:

```
13 --change the lowest salary of the employees to the average salary
14 UPDATE Employee
15 SET salary = (SELECT AVG(salary) FROM employee)
16 WHERE eid = (SELECT eid FROM employee WHERE salary = (SELECT MAX(salary) FROM employee));
```

Data Output Explain Messages Notifications

UPDATE 1

Query returned successfully in 63 msec.

Before modification:

```
18 -- before modification
19 SELECT * FROM employee WHERE salary = (SELECT MAX(salary) FROM employee);
```

Data Output Explain Messages Notifications

| | eid [PK] integer | name character varying (30) | salary integer |
|---|---------------------|--------------------------------|-------------------|
| 1 | 1984 | Jenna Maul | 98583 |

After modification:

```
18 -- after modification
19 SELECT * FROM employee WHERE eid=1984;
```

Data Output Explain Messages Notifications

| | eid [PK] integer | name character varying (30) | salary integer |
|---|---------------------|--------------------------------|-------------------|
| 1 | 1984 | Jenna Maul | 53058 |

Question 7

```
1) 2 -- this view returns a table with customer informations and their future check in dates\3 -- this can be useful when the hotel wants to notify or confirm a reservation with a customer\4 CREATE VIEW Customer_Reservation5 AS SELECT cname,phone_number,x.rid,start_date FROM6 (SELECT cname,rid,customer.phone_number FROM customer INNER JOIN handle7 ON customer.phone_number=handle.phone_number) AS x8 inner join reservation ON x.rid=reservation.rid;
```

Data Output Explain **Messages** Notifications

CREATE VIEW

Query returned successfully in 149 msec.

```
10 11 SELECT * FROM Customer_Reservation ;
```

Data Output Explain Messages Notifications

| | cname character varying (30) | phone_number bigint | rid integer | start_date date |
|----|---------------------------------|------------------------|----------------|--------------------|
| 1 | Hugo Mitchell | 5208379384 | 1001234 | 2020-03-01 |
| 2 | Numa Karolinski | 5203428469 | 1000001 | 2020-03-02 |
| 3 | Eugene Debbs | 5207778883 | 1000003 | 2020-03-09 |
| 4 | Numa Karolinski | 5203428469 | 1000005 | 2020-03-09 |
| 5 | Marshall Mathers | 3139812321 | 1000015 | 2020-03-13 |
| 6 | Jan Stark | 5085488991 | 1000018 | 2020-03-15 |
| 7 | Samuel Campbell | 5204938758 | 1000013 | 2020-03-12 |
| 8 | Jan Stark | 5085488991 | 1000010 | 2020-03-11 |
| 9 | Jennifer Stukeley | 5149834734 | 1000008 | 2020-03-11 |
| 10 | Juicer Shadow | 5143497231 | 1000011 | 2020-03-12 |

2)

```
3 --select the room that is vacant now, and has reservation in the future
4 CREATE VIEW VACANT_ROOM(room_number,type,room_status)
5 AS SELECT DISTINCT r.room_number , r.type , r.room_status
6 FROM room r JOIN belongs b ON r.room_number= b.room_number
7 WHERE (r.room_status ='VD' OR r.room_status='V')
8 ;
9
```

Data Output Explain **Messages** Notifications

CREATE VIEW

Query returned successfully in 2 secs 26 msec.

```
9  SELECT * FROM VACANT_ROOM;
10
```

Data Output Explain **Messages** Notifications

| | room_number integer | type character varying (20) | room_status character varying (20) |
|---|------------------------|--------------------------------|---------------------------------------|
| 1 | 403 | Suite | V |
| 2 | 305 | King | V |

Question 8:

1).

Adding the constraint

```
12 -- add a CHECK constraint on the Dine table
13 -- to make the price > 0
14 ALTER TABLE Dine add constraint PriceCheck
15 CHECK(price>0);
```

Data Output Explain Messages Notifications

ALTER TABLE

Query returned successfully in 76 msec.

INSERT statement that violates the constraint

```
17 -- modification that violates the constraints
18 INSERT INTO Dine VALUES(
19 5204392847, 'Santorini Souvlaki', 0
20 );
21
```

Data Output Explain Messages Notifications

ERROR: new row for relation "dine" violates check constraint "pricecheck"
DETAIL: Failing row contains (5204392847, Santorini Souvlaki, 0.00).
SQL state: 23514

2).

Adding the constraint

```
12 -- add a CHECK constraint on the Review table
13 -- to make the rating > 0 and < 6
14 ALTER TABLE Review add constraint RatingCheck
15 CHECK(rating>0 AND rating<6);
```

Data Output Explain **Messages** Notifications

ALTER TABLE

Query returned successfully in 55 msec.

INSERT statement that violates the constraint

```
12 -- modification that violates the constraint
13 INSERT INTO Review VALUES(
14 5143487231,'2019-3-20', '9:13:20', 10,
15      'The stay was SO good that a 5 doesnt suffice'
16 );
```

Data Output Explain **Messages** Notifications

```
ERROR: new row for relation "review" violates check constraint "ratingcheck"
DETAIL: Failing row contains (5143487231, 2019-03-20, 09:13:20, 10, The stay was SO good that a 5 doesnt suffice).
SQL state: 23514
```

revised review schema

```
cs421=> \d review
      Table "cs421g37.review"
 Column |          Type          | Modifiers
-----+-----+
phone_number | bigint           | not null
date        | date             | not null
time        | time without time zone | not null
rating      | integer           |
text        | character varying(500) |

Indexes:
 "review_pkey" PRIMARY KEY, btree (phone_number, date, "time")
Check constraints:
 "ratingcheck" CHECK (rating > 0 AND rating < 6)
Foreign-key constraints:
 "review_phone_number_fkey" FOREIGN KEY (phone_number) REFERENCES customer(phone_number)
```

revised dine schema

```
[cs421=> \d dine
      Table "cs421g37.dine"
Column   |      Type       | Modifiers
-----+-----+-----+
phone_number | bigint      | not null
rname        | character varying(30) | not null
price        | numeric(5,2)    | not null
Indexes:
  "dine_pkey" PRIMARY KEY, btree (phone_number, rname)
Check constraints:
  "pricecheck" CHECK (price > 0::numeric)
Foreign-key constraints:
  "dine_phone_number_fkey" FOREIGN KEY (phone_number) REFERENCES customer(phone_number)
  "dine_rname_fkey" FOREIGN KEY (rname) REFERENCES restaurant(rname)
```