

## **PROJECT 2 (Group 57)**

**Note: PostgreSQL was used for this project.**

### **Part 1**

The updated relational model is shown below. For full details, refer to the Project 1 writeup.

#### **Entities:**

**Lease(lease\_id, start\_date, end\_date, room\_no, monthly\_rent, address)**

room\_no references Apartment.room\_no

address references Apartment.address

**apt\_building(address)**

**Employee(eid, first\_name, last\_name)**

Note: Employee relation is combined with Manager and Superintendent, where only managers have email

**Superintendent(eid)**

Eid references employee

**Manager (eid, email)**

Eid references employee

**Tenant (tid, first\_name, last\_name, email, phone\_number)**

**legal\_action (legal\_id , name, date, lease\_id)**

lease\_id references Lease.lease\_id

**Complain(cid,description)**

**Guarantor (SIN, first\_name, last\_name, email, phone\_number,lease\_id)**

Leaseid references lease.lease\_id

#### **Weak Entities:**

**Apartment(room\_no, address, size)**

address references Apt\_Building.address

**Payment(month ,lease\_id,status)**

Leaseid references lease.leaseid

#### **Relationship:**

**Manages (eid,address)**

Eid references superintendent.eid

Address references aptBuilding.addr

**Issues (cid,tid,date\_issued)**

Id references Tenant.tid

Cid references Complaint.cid

**Resolves (cid,eid,date\_resolved)**

Cid references Complaint.cid

Eid references manager.eid

**Sign (lease\_id, tid,date)**

Leaseid references lease.leaseld

Tid references Tenant.tid

## **Part 2**

**(i) CREATE TABLE commands:**

```
CREATE TABLE employee (eid INTEGER, first_name VARCHAR(50), last_name  
VARCHAR(50) , PRIMARY KEY(eid));
```

```
CREATE TABLE apt_building  
(address VARCHAR(250) NOT NULL, PRIMARY KEY(address));
```

```
CREATE TABLE Superintendent  
(eid INT, PRIMARY KEY(eid),FOREIGN KEY(eid) REFERENCES employee);
```

```
CREATE TABLE manager (eid INT NOT NULL, email VARCHAR(320),PRIMARY  
KEY(eid),FOREIGN KEY(eid) REFERENCES employee) ;
```

```
CREATE TABLE apartment (room_no INTEGER, address VARCHAR(250),size  
FLOAT, PRIMARY KEY(address, room_no),FOREIGN KEY(address) REFERENCES  
apt_building) ;
```

```
CREATE TABLE lease  
(  
    lease_id INTEGER,  
    start_date DATE,  
    end_date DATE,  
    room_no INTEGER NOT NULL,  
    monthly_rent FLOAT,  
    address VARCHAR(250) NOT NULL,  
    PRIMARY KEY(lease_id) ,  
    FOREIGN KEY(address,room_no) REFERENCES apartment  
);
```

```
CREATE TABLE complaint
```

```
(  
    cid INTEGER,  
    description TEXT,  
    PRIMARY KEY(cid),  
);
```

```
CREATE TABLE legal_action
```

```
(  
    legal_id INTEGER,  
    name VARCHAR(100),  
    date DATE,  
    lease_id INTEGER NOT NULL,  
    PRIMARY KEY(legal_id),  
    FOREIGN KEY(lease_id) REFERENCES lease  
);
```

```
CREATE TABLE guarantor
```

```
(  
    SIN VARCHAR(12),  
    first_name VARCHAR(100),  
    last_name VARCHAR(100),  
    email VARCHAR(320) ,  
    phone_number VARCHAR(20),  
    lease_id INTEGER NOT NULL,  
    PRIMARY KEY(SIN),  
    FOREIGN KEY(lease_id) REFERENCES lease  
);
```

```
CREATE TABLE tenant
```

```
(  
    tid INTEGER,  
    first_name VARCHAR(100),  
    last_name VARCHAR (100),  
    email VARCHAR(320),  
    phone_number VARCHAR(20) ,  
    PRIMARY KEY(tid)  
);
```

```
CREATE TABLE payment(
```

```
    month DATE,  
    lease_id INTEGER,
```

```

        status BOOLEAN DEFAULT FALSE,
        PRIMARY KEY(month,lease_id) ,
        FOREIGN KEY (lease_id ) REFERENCES lease

);

CREATE TABLE sign
(
    lease_id INTEGER,
    tid INTEGER NOT NULL,
    date DATE,
    PRIMARY KEY(lease_id) ,
    FOREIGN KEY(lease_id) REFERENCES lease,
    FOREIGN KEY(tid) REFERENCES tenant
);

CREATE TABLE resolves
(
    cid INTEGER,
    eid INTEGER,
    date_resolved DATE,
    PRIMARY KEY(cid) ,
    FOREIGN KEY(cid) REFERENCES complaint,
    FOREIGN KEY(eid) REFERENCES manager
);

CREATE TABLE issues
(
    cid INTEGER,
    tid INTEGER,
    date_issued DATE,
    PRIMARY KEY(cid) ,
    FOREIGN KEY(cid) REFERENCES complaint,
    FOREIGN KEY(tid) REFERENCES tenant
);

CREATE TABLE manages
(
    eid INTEGER,
    address VARCHAR(250) ,
    PRIMARY KEY(eid,address) ,
    FOREIGN KEY(eid) REFERENCES superintendent,
    FOREIGN KEY(address) REFERENCES apt_building

```

);

**(ii) The \d command for each table:**

```
\d apartment
\d apt_building
\d complaint
\d employee
\d guarantor
\d issues
\d lease
\d legal_action
\d manager
\d manages
\d resolves
\d sign
\d superintendent
\d tenant
\d payment
```

**(iii) Output for each \d command:**

```
Table "cs421g57.apartment"
Column |          Type          | Modifiers
-----+-----+-----
room_no | integer                | not null
address | character varying(250) | not null
size    | double precision       |
Indexes:
    "apartment_pkey" PRIMARY KEY, btree (address, room_no)
Foreign-key constraints:
    "apartment_address_fkey" FOREIGN KEY (address) REFERENCES
apt_building(address, room_no)
Referenced by:
    TABLE "lease" CONSTRAINT "lease_address_fkey" FOREIGN KEY
(address, room_no)
REFERENCES apartment(address, room_no)
```

```
Table "cs421g57.apt_building"
Column |          Type          | Modifiers
-----+-----+-----
address | character varying(250) | not null
Indexes:
```

"apt\_building\_pkey" PRIMARY KEY, btree (address)

Referenced by:

TABLE "apartment" CONSTRAINT "apartment\_address\_fkey" FOREIGN KEY (address)

REFERENCES apt\_building(address)

TABLE "manages" CONSTRAINT "manages\_address\_fkey" FOREIGN KEY (address) REFE

RENCES apt\_building(address)

Table "cs421g57.complaint"

Column	Type	Modifiers
--------	------	-----------

-----+-----+-----

cid	integer	not null
-----	---------	----------

description	text	
-------------	------	--

Indexes:

"complaint\_pkey" PRIMARY KEY, btree (cid)

Referenced by:

TABLE "issues" CONSTRAINT "issues\_cid\_fkey" FOREIGN KEY (cid)

REFERENCES com

plaint(cid)

TABLE "resolves" CONSTRAINT "resolves\_cid\_fkey" FOREIGN KEY (cid)

REFERENCES

complaint(cid)

Table "cs421g57.employee"

Column	Type	Modifiers
--------	------	-----------

-----+-----+-----

eid	integer	not null
-----	---------	----------

first_name	character varying(50)	
------------	-----------------------	--

last_name	character varying(50)	
-----------	-----------------------	--

Indexes:

"employees\_pkey" PRIMARY KEY, btree (eid)

Referenced by:

TABLE "manager" CONSTRAINT "manager\_eid\_fkey" FOREIGN KEY (eid)

REFERENCES e

mployee(eid)

TABLE "superintendent" CONSTRAINT "superintendent\_eid\_fkey"

FOREIGN KEY (eid

) REFERENCES employee(eid)

Table "cs421g57.guarantor"

Column	Type	Modifiers
--------	------	-----------

-----+-----+-----

```

sin          | character varying(12) | not null
first_name   | character varying(100) |
last_name    | character varying(100) |
email        | character varying(320) |
phone_number | character varying(20)  |
lease_id     | integer                 | not null

```

Indexes:

```
"guarantor_pkey" PRIMARY KEY, btree (sin)
```

Foreign-key constraints:

```
"guarantor_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES
lease(lease_id)
```

Table "cs421g57.issues"

Column	Type	Modifiers
cid	integer	not null
tid	integer	
date_issued	date	

Indexes:

```
"issues_pkey" PRIMARY KEY, btree (cid)
```

Foreign-key constraints:

```
"issues_cid_fkey" FOREIGN KEY (cid) REFERENCES complaint(cid)
"issues_tid_fkey" FOREIGN KEY (tid) REFERENCES tenant(tid)
```

Table "cs421g57.lease"

Column	Type	Modifiers
lease_id	integer	not null
start_date	date	
end_date	date	
room_no	integer	not null
monthly_rent	double precision	
address	character varying(250)	not null

Indexes:

```
"lease_pkey" PRIMARY KEY, btree (lease_id)
```

Foreign-key constraints:

```
"lease_address_fkey" FOREIGN KEY (address, room_no) REFERENCES
apartment(address, room_no)
```

Referenced by:

```
TABLE "guarantor" CONSTRAINT "guarantor_lease_id_fkey" FOREIGN
KEY (lease_id
) REFERENCES lease(lease_id)
```

```

TABLE "legal_action" CONSTRAINT "legal_action_lease_id_fkey"
FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
TABLE "sign" CONSTRAINT "sign_lease_id_fkey" FOREIGN KEY
(lease_id) REFERENCES lease(lease_id)

```

```

Table "cs421g57.legal_action"
Column |          Type          | Modifiers
-----+-----+-----
legal_id | integer                | not null
name      | character varying(100) |
date      | date                   |
lease_id | integer                | not null

```

Indexes:

```
"legal_action_pkey" PRIMARY KEY, btree (legal_id)
```

Foreign-key constraints:

```

"legal_action_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES
lease(lease_id)

```

```

Table "cs421g57.manager"
Column |          Type          | Modifiers
-----+-----+-----
eid | integer                | not null
email | character varying(320) |

```

Indexes:

```
"manager_pkey" PRIMARY KEY, btree (eid)
```

Foreign-key constraints:

```
"manager_eid_fkey" FOREIGN KEY (eid) REFERENCES employee(eid)
```

Referenced by:

```

TABLE "resolves" CONSTRAINT "resolves_eid_fkey" FOREIGN KEY (eid)
REFERENCES
manager(eid)

```

```

Table "cs421g57.manages"
Column |          Type          | Modifiers
-----+-----+-----
eid | integer                | not null
address | character varying(250) | not null

```

Indexes:

```
"manages_pkey" PRIMARY KEY, btree (eid, address)
```

Foreign-key constraints:



```

    "manages_address_fkey" FOREIGN KEY (address) REFERENCES
apt_building(address
)
    "manages_eid_fkey" FOREIGN KEY (eid) REFERENCES
superintendent(eid)

```

```

Table "cs421g57.resolves"
Column      |  Type   | Modifiers
-----+-----+-----
cid          | integer | not null
eid          | integer |
date_resolved | date   |

```

Indexes:

```

"resolves_pkey" PRIMARY KEY, btree (cid)

```

Foreign-key constraints:

```

"resolves_cid_fkey" FOREIGN KEY (cid) REFERENCES complaint(cid)
"resolves_eid_fkey" FOREIGN KEY (eid) REFERENCES manager(eid)

```

```

Table "cs421g57.sign"
Column |  Type   | Modifiers
-----+-----+-----
lease_id | integer | not null
tid      | integer | not null
date     | date    |

```

Indexes:

```

"sign_pkey" PRIMARY KEY, btree (lease_id)

```

Foreign-key constraints:

```

"sign_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES
lease(lease_id)
"sign_tid_fkey" FOREIGN KEY (tid) REFERENCES tenant(tid)

```

```

Table "cs421g57.superintendent"
Column |  Type   | Modifiers
-----+-----+-----
eid    | integer | not null

```

Indexes:

```

"superintendent_pkey" PRIMARY KEY, btree (eid)

```

Foreign-key constraints:

```

"superintendent_eid_fkey" FOREIGN KEY (eid) REFERENCES
employee(eid)

```

Referenced by:

```
TABLE "manages" CONSTRAINT "manages_eid_fkey" FOREIGN KEY (eid)
REFERENCES s
    uperintendent(eid)
```

```
Table "cs421g57.tenant"
Column      |      Type      | Modifiers
-----+-----+-----
tid         | integer        | not null
first_name  | character varying(100) |
last_name   | character varying(100) |
email       | character varying(320) |
phone_number | character varying(20)  |
```

Indexes:

```
"tenant_pkey" PRIMARY KEY, btree (tid)
```

Referenced by:

```
TABLE "issues" CONSTRAINT "issues_tid_fkey" FOREIGN KEY (tid)
REFERENCES ten
    ant(tid)
TABLE "sign" CONSTRAINT "sign_tid_fkey" FOREIGN KEY (tid)
REFERENCES tenant(
    tid)
```

```
Table "cs421g57.payment"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
month  | date |           | not null |
lease_id | integer |           | not null |
status | boolean |           |         | false
```

Indexes:

```
"payment_pkey" PRIMARY KEY, btree (month, lease_id)
```

Foreign-key constraints:

```
"payment_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES
lease(lease_id)
```

## Part 3

### (i) Five INSERT commands:

```
INSERT INTO tenant (tid, first_name, last_name, email, phone_number)
VALUES (1, 'Stephen', 'Harper', 'stephenharper@mcgill.ca', 438827000);
```

```
INSERT INTO tenant (tid, first_name, last_name, email, phone_number)
VALUES (2, 'Justin', 'Trudeau', 'justintrudeau@mcgill.ca', 498827001);
```

```
INSERT INTO tenant (tid, first_name, last_name, email, phone_number)
```

```
VALUES (3, 'Paul', 'Martin', 'paulmartin@mcgill.ca', 438827023);

INSERT INTO tenant (tid, first_name, last_name, email, phone_number)
VALUES (4, 'kim', 'Campbell', 'kimcampbell@mcgill.ca', 438824002);

INSERT INTO tenant (tid, first_name, last_name, email, phone_number)
VALUES (5, 'John', 'Turner', 'johnturner@mcgill.ca', 438827002);

SELECT * FROM tenant;
```

## (ii) Output of INSERT commands:

```
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

tid	first_name	last_name	email	phone_number
1	Stephen	Harper	stephenharper@mcgill.ca	438827000
2	Justin	Trudeau	justintrudeau@mcgill.ca	498827001
3	Paul	Martin	paulmartin@mcgill.ca	438827023
4	kim	Campbell	kimcampbell@mcgill.ca	438824002
5	John	Turner	johnturner@mcgill.ca	438827002

## Part 4

### (i) Commands after insertion of data:

```
SELECT * FROM apartment LIMIT 5;
SELECT * FROM apt_building LIMIT 5;
SELECT * FROM complaint LIMIT 5;
SELECT * FROM employee LIMIT 5;
SELECT * FROM guarantor LIMIT 5;
SELECT * FROM issues LIMIT 5;
SELECT * FROM lease LIMIT 5;
SELECT * FROM legal_action LIMIT 5;
SELECT * FROM manager LIMIT 5;
SELECT * FROM manages LIMIT 5;
SELECT * FROM payment LIMIT 5;
SELECT * FROM resolves LIMIT 5;
SELECT * FROM sign LIMIT 5;
SELECT * FROM superintendent LIMIT 5;
SELECT * FROM tenant LIMIT 5;
```

**(ii) Output of commands:**

```
room_no | address      | size
-----+-----+-----
      57 | 1717 Shopko Way | 674.42
      19 | 66 Esker Center | 990
      11 | 3 Eastwood Center | 435.45
      49 | 1938 Novick Road | 690.72
      57 | 5 McBride Way    | 989.83
(5 rows)
```

```
address
-----
1717 Shopko Way
66 Esker Center
3 Eastwood Center
1938 Novick Road
5 McBride Way
(5 rows)
```

```
cid |
description

-----+-----
--

-----+-----
--

-----+-----
1 | just wanted to say hi
2 | drinking the tap water makes me hallucinate
3 | 20lbs of roof collapsed onto my head yesterday and I had to go to the
emergency room. Not wanting to make a big deal about it though. I know you
have a lot on your plate
4 | I rate the shower 10/10 but the toilet -5/10 pls fix
5 | My roommate goes on daily 20-minute rants about the toilet. Would it be
possible to fix it please?
(5 rows)
```

```
eid | first_name | last_name
-----+-----+-----
    1 | Shaurya    | Mehta
    2 | Jonathan   | Zhang
    3 | Cassandra  | Chan
    4 | Tigran     | Io
    5 | Joseph     | Dsilva
(5 rows)
```

lease_id	sin	first_name	last_name	email	phone_number
456126359	Ashley	Brooke	abrooke@customdomain.net	562-456-5555	
1	516265892	Cameron	Danson	myheadisround@yahoo.com	862-999-1234
2	156248795	Eve	Finnet	efinnet@gmail.com	556-153-4985
3	156245789	Gordon	Ham	saltypork@hotmail.com	516-489-4458
4	954652448	Imley	Jones	imjoe@global.net	878-448-4489
5					

(5 rows)

cid	tid	date_issued
1	1	2019-01-25
2	4	2017-01-02
3	4	2019-07-10
4	5	2007-07-10
5	3	2019-05-01

(5 rows)

lease_id	start_date	end_date	room_no	monthly_rent	address
1	2016-09-26	2021-09-14	57	596.42	1717 Shopko Way
2	2016-02-26	2023-03-17	19	827.17	66 Esker Center
3	2017-08-15	2023-09-06	11	909.63	3 Eastwood
4	2016-10-12	2020-12-20	49	709.94	1938 Novick
5	2017-07-30	2022-10-30	57	742.75	5 McBride Way

(5 rows)

legal_id	name	date	lease_id
1	Eviction	2020-02-28	8
2	Noise Fine	2019-01-01	50
3	Repair Fine	2019-12-25	99
4	Repair Fine	2016-05-23	23

5 | Eviction | 2019-01-25 | 10  
(5 rows)

eid	email
5	jdsilva@hotmail.com
1	smehta@gmail.com
2	jonathan@mcgill.ca
6	potus@whitehouse.gov
3	awesomesocks@gmail.com

(5 rows)

eid	address
1	1717 Shopko Way
3	66 Esker Center
4	3 Eastwood Center
5	5 McBride Way
6	8 Gerald Park

(5 rows)

month	lease_id	status
June	30	f
July	31	f
August	32	t
May	33	t
March	33	t

(5 rows)

cid	eid	date_resolved
1	3	2019-02-27
2	2	2017-03-20
3	1	2020-01-05
4	1	2008-08-20
5	6	2019-05-05

(5 rows)

lease_id	tid	date
89	1	2019-04-25
30	2	2018-02-04
1	3	2020-01-03
97	4	2018-08-08
55	5	2017-03-03

(5 rows)

```
eid
-----
1
3
4
5
6
(5 rows)
```

```
tid | first_name | last_name | email | phone_number
-----+-----+-----+-----+-----
1 | Stephen | Harper | stephenharper@mcgill.ca | 438827000
2 | Justin | Trudeau | justintrudeau@mcgill.ca | 498827001
3 | Paul | Martin | paulmartin@mcgill.ca | 438827023
4 | kim | Campbell | kimcampbell@mcgill.ca | 438824002
5 | John | Turner | johnturner@mcgill.ca | 438827002
(5 rows)
```

## Part 5

### 1. Find the superintendent who manages the tenant's building (Assume tid=3 for the example).

```
Select * from employee where eid=(select eid from manages where  
address=(select address from lease where lease_id=(select  
lease_id from sign where tid=3)))
```

```
cs421=> \d employee
Table "cs421g57.employee"
Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
eid     | integer                |           | not null |
first_name | character varying(50) |           |          |
last_name | character varying(50) |           |          |
Indexes:
    "employees_pkey" PRIMARY KEY, btree (eid)
Referenced by:
    TABLE "manager" CONSTRAINT "manager_eid_fkey" FOREIGN KEY (eid) REFERENCES employee(eid)
    TABLE "superintendent" CONSTRAINT "superintendent_eid_fkey" FOREIGN KEY (eid) REFERENCES employee(eid)

cs421=> \d manages
Table "cs421g57.manages"
Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
eid     | integer                |           | not null |
address | character varying(250) |           | not null |
Indexes:
    "manages_pkey" PRIMARY KEY, btree (eid, address)
Foreign-key constraints:
    "manages_address_fkey" FOREIGN KEY (address) REFERENCES apt_building(address)
    "manages_eid_fkey" FOREIGN KEY (eid) REFERENCES superintendent(eid)
```



```

cs421=> \d address
Did not find any relation named "address".
cs421=> \d lease
               Table "cs421g57.lease"
   Column   |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
lease_id    | integer         |           | not null |
start_date  | date            |           |          |
end_date    | date            |           |          |
room_no     | integer         |           | not null |
monthly_rent | double precision |           |          |
address     | character varying(250) |         | not null |
Indexes:
    "lease_pkey" PRIMARY KEY, btree (lease_id)
Check constraints:
    "minimum_rent" CHECK (monthly_rent > 0::double precision)
    "valid_lease_date" CHECK (start_date < end_date AND (start_date - end_date) < 366)
Foreign-key constraints:
    "lease_address_fkey" FOREIGN KEY (address, room_no) REFERENCES apartment(address, room_no)
Referenced by:
    TABLE "guarantor" CONSTRAINT "guarantor_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    TABLE "legal_action" CONSTRAINT "legal_action_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    TABLE "payment" CONSTRAINT "payment_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    TABLE "sign" CONSTRAINT "sign_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)

cs421=> \d sign
               Table "cs421g57.sign"
   Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
lease_id | integer |         | not null |
tid      | integer |         | not null |
date     | date   |         |          |
Indexes:
    "sign_pkey" PRIMARY KEY, btree (lease_id)
Foreign-key constraints:
    "sign_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    "sign_tid_fkey" FOREIGN KEY (tid) REFERENCES tenant(tid)

cs421=> Select * from employee where eid=(select eid from manages where address=(select address from lease where lease_id=(select lease_id from sign where tid=3)));
eid | first_name | last_name
-----+-----+-----
  1 | Shaurya    | Mehta
(1 row)

```

## 2. Find the number of complaints.

```
SELECT COUNT(*) from complaint;
```

```

cs421=> \d complaint
               Table "cs421g57.complaint"
   Column   |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
cid         | integer         |           | not null |
description | text            |           |          |
Indexes:
    "complaint_pkey" PRIMARY KEY, btree (cid)
Referenced by:
    TABLE "issues" CONSTRAINT "issues_cid_fkey" FOREIGN KEY (cid) REFERENCES complaint(cid)
    TABLE "resolves" CONSTRAINT "resolves_cid_fkey" FOREIGN KEY (cid) REFERENCES complaint(cid)

cs421=> SELECT COUNT(*) from complaint;
count
-----
     6
(1 row)

```

### 3 Find the cid of the unresolved complaints .

```
SELECT cid FROM complaint
except
SELECT cid FROM resolves;
```

```
cs421=> \d complaint
          Table "cs421g57.complaint"
  Column      | Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 cid          | integer   |           | not null |
 description   | text      |           |          |
Indexes:
    "complaint_pkey" PRIMARY KEY, btree (cid)
Referenced by:
    TABLE "issues" CONSTRAINT "issues_cid_fkey" FOREIGN KEY (cid) REFERENCES complaint(cid)
    TABLE "resolves" CONSTRAINT "resolves_cid_fkey" FOREIGN KEY (cid) REFERENCES complaint(cid)

cs421=> \d resolves
          Table "cs421g57.resolves"
  Column      | Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 cid          | integer   |           | not null |
 eid          | integer   |           |          |
 date_resolved | date      |           |          |
Indexes:
    "resolves_pkey" PRIMARY KEY, btree (cid)
Foreign-key constraints:
    "resolves_cid_fkey" FOREIGN KEY (cid) REFERENCES complaint(cid)
    "resolves_eid_fkey" FOREIGN KEY (eid) REFERENCES manager(eid)

cs421=> SELECT cid FROM complaint
cs421-> except
cs421-> SELECT cid FROM resolves;
 cid
----
    6
(1 row)
```

### 4. Find the number of vacant apartments.

```
select COUNT(*) from (select address, room_no from lease
                        except
                        select address, room_no from lease l where
exists (
                                select * from sign where sign.lease_id =
l.lease_id)) as foo;
```

```

cs421=> \d lease;

               Table "cs421g57.lease"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 lease_id     | integer        |           | not null |
 start_date   | date           |           |          |
 end_date     | date           |           |          |
 room_no      | integer        |           | not null |
 monthly_rent | double precision |          |          |
 address      | character varying(250) |          | not null |
Indexes:
    "lease_pkey" PRIMARY KEY, btree (lease_id)
Check constraints:
    "minimum_rent" CHECK (monthly_rent > 0::double precision)
    "valid_lease_date" CHECK (start_date < end_date AND (start_date - end_date) < 366)
Foreign-key constraints:
    "lease_address_fkey" FOREIGN KEY (address, room_no) REFERENCES apartment(address, room_no)
Referenced by:
    TABLE "guarantor" CONSTRAINT "guarantor_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    TABLE "legal_action" CONSTRAINT "legal_action_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    TABLE "payment" CONSTRAINT "payment_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    TABLE "sign" CONSTRAINT "sign_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)

cs421=> \d sign

               Table "cs421g57.sign"
  Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
 lease_id | integer |          | not null |
 tid      | integer |          | not null |
 date     | date   |          |          |
Indexes:
    "sign_pkey" PRIMARY KEY, btree (lease_id)
Foreign-key constraints:
    "sign_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    "sign_tid_fkey" FOREIGN KEY (tid) REFERENCES tenant(tid)

cs421=> select COUNT(*) from (select address, room_no from lease
cs421(>      except
cs421(>      select address, room_no from lease l where exists (
cs421(>      select * from sign where sign.lease_id = l.lease_id))  as foo;
count
-----
    93
(1 row)

```

**5. Find the average monthly rent of vacant apartments compared to the average monthly rent of all leased apartments.**

```

select avg(monthly_rent) from lease l
group by exists
(select * from sign where l.lease_id = sign.lease_id);

```

cs421=> \d lease

Table "cs421g57.lease"				
Column	Type	Collation	Nullable	Default
lease_id	integer		not null	
start_date	date			
end_date	date			
room_no	integer		not null	
monthly_rent	double precision			
address	character varying(250)		not null	

Indexes:

"lease\_pkey" PRIMARY KEY, btree (lease\_id)

Check constraints:

"minimum\_rent" CHECK (monthly\_rent > 0::double precision)

"valid\_lease\_date" CHECK (start\_date < end\_date AND (start\_date - end\_date) < 366)

Foreign-key constraints:

"lease\_address\_fkey" FOREIGN KEY (address, room\_no) REFERENCES apartment(address, room\_no)

Referenced by:

TABLE "guarantor" CONSTRAINT "guarantor\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

TABLE "legal\_action" CONSTRAINT "legal\_action\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

TABLE "payment" CONSTRAINT "payment\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

TABLE "sign" CONSTRAINT "sign\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

cs421=> \d sign

Table "cs421g57.sign"				
Column	Type	Collation	Nullable	Default
lease_id	integer		not null	
tid	integer		not null	
date	date			

Indexes:

"sign\_pkey" PRIMARY KEY, btree (lease\_id)

Foreign-key constraints:

"sign\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

"sign\_tid\_fkey" FOREIGN KEY (tid) REFERENCES tenant(tid)

cs421=> select avg(monthly\_rent) from lease l

cs421-> group by exists

cs421-> (select \* from sign where l.lease\_id = sign.lease\_id)

cs421-> ;

```
      avg
-----
772.261580645161
716.656
(2 rows)
```

## Part 6

1. If one tenant, with say, for example, tid=1, paid the rent, the status of the payment associated with the tenant will be changed to true.

```
UPDATE payment
SET status='TRUE'
FROM sign, tenant
WHERE payment.lease_id =
      (SELECT lease_id
       FROM sign
       WHERE sign.tid = 1
      );
```

Before:

```
cs421=> SELECT * FROM PAYMENT;
month | lease_id | status
-----+-----+-----
July  |      31 | f
August |      32 | t
May    |      33 | t
March  |      33 | t
March  |      34 | t
June   |      30 | f
June   |      89 | f
(7 rows)
```

Run:

```
cs421=> UPDATE payment
cs421-> SET status='TRUE'
cs421-> FROM sign, tenant
cs421-> WHERE payment.lease_id =
cs421->      (SELECT lease_id
cs421(>      FROM sign
cs421(>      WHERE sign.tid = 1
cs421(> );
UPDATE 1
```



After:

```
cs421=> SELECT * FROM PAYMENT;
month | lease_id | status
-----+-----+-----
July  |         31 | f
August |         32 | t
May    |         33 | t
March  |         33 | t
March  |         34 | t
June   |         30 | f
June   |         89 | t
(7 rows)
```

```
cs421=> SELECT * FROM PAYMENT;
month | lease_id | status
-----+-----+-----
July  |         31 | f
August |         32 | t
May    |         33 | t
March  |         33 | t
March  |         34 | t
June   |         30 | f
(6 rows)
```

```
cs421=> \d lease
```

Column	Type	Collation	Nullable	Default
lease_id	integer		not null	
start_date	date			
end_date	date			
room_no	integer		not null	
monthly_rent	double precision			
address	character varying(250)		not null	

Indexes:

"lease\_pkey" PRIMARY KEY, btree (lease\_id)

Check constraints:

"minimum\_rent" CHECK (monthly\_rent > 0::double precision)

"valid\_lease\_date" CHECK (start\_date < end\_date AND (start\_date - end\_date) < 366)

Foreign-key constraints:

"lease\_address\_fkey" FOREIGN KEY (address, room\_no) REFERENCES apartment(address, room\_no)

Referenced by:

TABLE "guarantor" CONSTRAINT "guarantor\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

TABLE "legal\_action" CONSTRAINT "legal\_action\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

TABLE "payment" CONSTRAINT "payment\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

TABLE "sign" CONSTRAINT "sign\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

```
cs421=> \d sign
```

Column	Type	Collation	Nullable	Default
lease_id	integer		not null	
tid	integer		not null	
date	date			

Indexes:

"sign\_pkey" PRIMARY KEY, btree (lease\_id)

Foreign-key constraints:

"sign\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

"sign\_tid\_fkey" FOREIGN KEY (tid) REFERENCES tenant(tid)

```

cs421=> \d tenant

          Table "cs421g57.tenant"
   Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
   tid    | integer                |           | not null |
 first_name | character varying(100) |           |          |
 last_name | character varying(100) |           |          |
  email   | character varying(320) |           |          |
phone_number | character varying(20) |           |          |
Indexes:
    "tenant_pkey" PRIMARY KEY, btree (tid)
Referenced by:
    TABLE "issues" CONSTRAINT "issues_tid_fkey" FOREIGN KEY (tid) REFERENCES tenant(tid)
    TABLE "sign" CONSTRAINT "sign_tid_fkey" FOREIGN KEY (tid) REFERENCES tenant(tid)

cs421=> \d lease

          Table "cs421g57.lease"
   Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 lease_id | integer                |           | not null |
start_date | date                  |           |          |
 end_date | date                  |           |          |
  room_no | integer                |           | not null |
monthly_rent | double precision      |           |          |
  address | character varying(250) |           | not null |
Indexes:
    "lease_pkey" PRIMARY KEY, btree (lease_id)
Check constraints:
    "minimum_rent" CHECK (monthly_rent > 0::double precision)
    "valid_lease_date" CHECK (start_date < end_date AND (start_date - end_date) < 366)
Foreign-key constraints:
    "lease_address_fkey" FOREIGN KEY (address, room_no) REFERENCES apartment(address, room_no)
Referenced by:
    TABLE "guarantor" CONSTRAINT "guarantor_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    TABLE "legal_action" CONSTRAINT "legal_action_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    TABLE "payment" CONSTRAINT "payment_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
    TABLE "sign" CONSTRAINT "sign_lease_id_fkey" FOREIGN KEY (lease_id) REFERENCES lease(lease_id)

```

## 2. Increase the monthly rent of the leases by 10 percent which start after 2018.

UPDATE lease

SET monthly\_rent =monthly\_rent\*1.1

WHERE (date\_part('year',start\_date)>2016 );

Before:

```

cs421=> SELECT * FROM lease ORDER BY monthly_rent DESC LIMIT 5;
 lease_id | start_date | end_date | room_no | monthly_rent | address
-----+-----+-----+-----+-----+-----
       39 | 2019-09-27 | 2021-02-01 |        6 |    1195.8067 | 335 Spaight Court
       66 | 2019-03-08 | 2023-09-08 |       90 |    1160.9829 | 1781 Trailsway Place
       53 | 2019-04-19 | 2020-04-20 |       49 |    1068.3695 | 811 Forest Road
       91 | 2017-11-25 | 2020-07-30 |       25 |     1065.603 | 40 Evergreen Pass
       99 | 2018-09-29 | 2023-05-30 |       86 |     1063.425 | 0967 Oneill Avenue
(5 rows)

```

After:

```
cs421=> SELECT * FROM lease WHERE (lease_id=39 OR lease_id=66 or lease_id=53 or lease_id=91 or lease_id=99);
```

lease_id	start_date	end_date	room_no	monthly_rent	address
91	2017-11-25	2020-07-30	25	1172.1633	40 Evergreen Pass
99	2018-09-29	2023-05-30	86	1169.7675	0967 Oneill Avenue
39	2019-09-27	2021-02-01	6	1315.38737	335 Spaight Court
53	2019-04-19	2020-04-20	49	1175.20645	811 Forest Road
66	2019-03-08	2023-09-08	90	1277.08119	1781 Trailsway Place

(5 rows)

```
cs421=> \d lease;
```

Column	Type	Collation	Nullable	Default
lease_id	integer		not null	
start_date	date			
end_date	date			
room_no	integer		not null	
monthly_rent	double precision			
address	character varying(250)		not null	

Indexes:

- "lease\_pkey" PRIMARY KEY, btree (lease\_id)

Check constraints:

- "minimum\_rent" CHECK (monthly\_rent > 0::double precision)
- "valid\_lease\_date" CHECK (start\_date < end\_date AND (start\_date - end\_date) < 366)

Foreign-key constraints:

- "lease\_address\_fkey" FOREIGN KEY (address, room\_no) REFERENCES apartment(address, room\_no)

Referenced by:

- TABLE "guarantor" CONSTRAINT "guarantor\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)
- TABLE "legal\_action" CONSTRAINT "legal\_action\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)
- TABLE "payment" CONSTRAINT "payment\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)
- TABLE "sign" CONSTRAINT "sign\_lease\_id\_fkey" FOREIGN KEY (lease\_id) REFERENCES lease(lease\_id)

**3 . If one complaint, with say, for example, cid=1, is resolved, one record will be inserted into the resolved table.**

```
INSERT INTO resolves (cid, eid, date_resolved)
VALUES (6, 1, '2020-10-30');
```

**Before:**



```

cs421=> \d resolves
          Table "cs421g57.resolves"
   Column      | Type   | Collation | Nullable | Default
-----+-----+-----+-----+-----
 cid           | integer |           | not null |
 eid           | integer |           |          |
 date_resolved | date    |           |          |
Indexes:
    "resolves_pkey" PRIMARY KEY, btree (cid)
Foreign-key constraints:
    "resolves_cid_fkey" FOREIGN KEY (cid) REFERENCES complaint(cid)
    "resolves_eid_fkey" FOREIGN KEY (eid) REFERENCES manager(eid)

cs421=> SELECT * FROM resolves;
 cid | eid | date_resolved
-----+-----+-----
   1 |   3 | 2019-02-27
   2 |   2 | 2017-03-20
   4 |   1 | 2008-08-20
   5 |   6 | 2019-05-05
(4 rows)

```

After:

```

cs421=> INSERT INTO resolves (cid, eid, date_resolved)
cs421-> VALUES (6, 1, '2020-10-30');
INSERT 0 1
cs421=> SELECT * FROM resolves;
 cid | eid | date_resolved
-----+-----+-----
   1 |   3 | 2019-02-27
   2 |   2 | 2017-03-20
   4 |   1 | 2008-08-20
   5 |   6 | 2019-05-05
   6 |   1 | 2020-10-30
(5 rows)

```

4. Update the email for a certain tenant's guarantor (assume tid=3).

```

UPDATE guarantor
SET email='abc@gmail.com'
FROM sign, tenant
WHERE guarantor.lease_id =
      (SELECT lease_id
       FROM sign
       WHERE sign.tid = 3)

```

);

Before:

```
cs421=> SELECT * FROM guarantor;
      sin      | first_name | last_name | email                  | phone_number | lease_id
-----+-----+-----+-----+-----+-----
456126359 | Ashley    | Brooke    | abrooke@customdomain.net | 562-456-5555 | 1
516265892 | Cameron   | Danson    | myheadisround@yahoo.com  | 862-999-1234 | 2
156248795 | Eve       | Finnet    | efinnet@gmail.com        | 556-153-4985 | 3
156245789 | Gordon    | Ham       | saltypork@hotmail.com    | 516-489-4458 | 4
954652448 | Imley     | Jones     | imjoe@global.net         | 878-448-4489 | 5
(5 rows)
```

After:

```
cs421=> UPDATE guarantor
cs421-> SET email='abc@gmail.com'
cs421-> FROM sign, tenant
cs421-> WHERE guarantor.lease_id =
cs421->      (SELECT lease_id
cs421->      FROM sign
cs421->      WHERE sign.tid = 3
cs421-> );
UPDATE 1
cs421=> SELECT * FROM guarantor;
      sin      | first_name | last_name | email                  | phone_number | lease_id
-----+-----+-----+-----+-----+-----
516265892 | Cameron   | Danson    | myheadisround@yahoo.com | 862-999-1234 | 2
156248795 | Eve       | Finnet    | efinnet@gmail.com        | 556-153-4985 | 3
156245789 | Gordon    | Ham       | saltypork@hotmail.com    | 516-489-4458 | 4
954652448 | Imley     | Jones     | imjoe@global.net         | 878-448-4489 | 5
456126359 | Ashley    | Brooke    | abc@gmail.com            | 562-456-5555 | 1
(5 rows)
```

## **Part 7**

(1):

```
CREATE VIEW unresolved_complaints AS SELECT complaint.cid FROM
complaint EXCEPT (select complaint.cid FROM complaint,resolves WHERE
resolves.cid=complaint.cid)
```

```
CREATE VIEW
```

```
cs421=> select * from unresolved_complaints;
```

```
cid
```

```
-----
```

```
6
```

```
(1 row)
```

```
cs421=> UPDATE unresolved_complaints SET cid=6 WHERE cid=6;
```

```
ERROR: cannot update view "unresolved_complaints"
```

```
DETAIL: Views containing UNION, INTERSECT, or EXCEPT are not
automatically updatable.
```

```
HINT: To enable updating the view, provide an INSTEAD OF UPDATE
trigger or an unconditional ON UPDATE DO INSTEAD rule.
```

(2):

```
CREATE VIEW expensive_places AS SELECT lease_id, address FROM lease
WHERE monthly_rent>950;
```

```
CREATE VIEW
```

```
cs421=> SELECT * FROM expensive_places;
```

```
lease_id | address
```

```
-----+-----
```

```
8 | 9496 Reindahl Junction
24 | 779 Washington Crossing
44 | 78133 Washington Trail
54 | 75 Rusk Avenue
57 | 10 Springview Drive
91 | 40 Evergreen Pass
99 | 0967 Oneill Avenue
39 | 335 Spaight Court
53 | 811 Forest Road
61 | 4 Melrose Crossing
66 | 1781 Trailsway Place
```

```
(11 rows)
```

```
cs421=> UPDATE expensive_places SET address='555 Test Lane' WHERE
lease_id>60;
```

```
ERROR: insert or update on table "lease" violates foreign key
constraint "lease_address_fkey"
```

**DETAIL:** Key (address, room\_no)=(555 Test Lane, 25) is not present in table "apartment".

The first view represents the cid of complaints that have not yet been resolved.

The second view represents the lease\_id and address of places that have a rent of greater than 950.

Neither of these views are updatable. The first cannot be updated as it contains the except command. The second cannot be updated as it would violate foreign key constraints.

To be updateable:

A view must originate from only one table.

A view must not contain set operations

## **Part 8**

**Constraint 1:** Make sure rent is greater than 0.

**(i) Command:**

```
ALTER TABLE lease ADD CONSTRAINT minimum_rent CHECK
(
    monthly_rent > 0
);
```

**(ii) Revised Schema:**

Table "cs421g57.lease"				
Column	Type	Collation	Nullable	Default
-----+-----+-----+-----+-----				
--				
lease_id	integer		not null	
start_date	date			
end_date	date			
room_no	integer		not null	
monthly_rent	double precision			
address	character varying(250)		not null	

Indexes:

```

        "lease_pkey" PRIMARY KEY, btree (lease_id)
Check constraints:
        "minimum_rent" CHECK (monthly_rent > 0::double precision)
Foreign-key constraints:
        "lease_address_fkey" FOREIGN KEY (address, room_no) REFERENCES
apartment(address, room_no)
Referenced by:
        TABLE "guarantor" CONSTRAINT "guarantor_lease_id_fkey" FOREIGN
KEY (lease_id) REFERENCES lease(lease_id)
        TABLE "legal_action" CONSTRAINT "legal_action_lease_id_fkey"
FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
        TABLE "payment" CONSTRAINT "payment_lease_id_fkey" FOREIGN KEY
(lease_id) REFERENCES lease(lease_id)
        TABLE "sign" CONSTRAINT "sign_lease_id_fkey" FOREIGN KEY
(lease_id) REFERENCES lease(lease_id)

```

### (iii) Invalid Insert/Updates:

Command (Update):

```

UPDATE lease
SET
    monthly_rent=-20
WHERE
    lease_id=5;

```

Output:

```

ERROR:  new row for relation "lease" violates check constraint
"minimum_rent"
DETAIL:  Failing row contains (5, 2017-07-30, 2020-10-30, 57, -20, 5
Mcbride Way).

```

Command (Insert):

```

INSERT INTO lease (lease_id, start_date, end_date, room_no,
monthly_rent, address)
VALUES(98, '2019-01-01', '2020-01-01', 40, -50, '3445 Stanley
Street');

```

Output:

```

ERROR:  new row for relation "lease" violates check constraint
"minimum_rent"
DETAIL:  Failing row contains (98, 2019-01-01, 2020-01-01, 40, -50,
3445 Stanley Street).

```

**Constraint 1:** Make sure start\_date is earlier than end\_date in lease and that the lease is at most a year long.

**(i) Command:**

```
ALTER TABLE lease ADD CONSTRAINT valid_lease_date CHECK
(
    start_date < end_date
    AND start_date - end_date < 366
);
```

**(ii) Revised Schema:**

```
Table "cs421g57.lease"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
lease_id     | integer        |           | not null |
start_date   | date           |           |          |
end_date     | date           |           |          |
room_no      | integer        |           | not null |
monthly_rent | double precision |           |          |
address      | character varying(250) |           | not null |
```

Indexes:

```
"lease_pkey" PRIMARY KEY, btree (lease_id)
```

Check constraints:

```
"minimum_rent" CHECK (monthly_rent > 0::double precision)
```

```
"valid_lease_date" CHECK (start_date < end_date AND (start_date -
end_date) < 366)
```

Foreign-key constraints:

```
"lease_address_fkey" FOREIGN KEY (address, room_no) REFERENCES
apartment(address, room_no)
```

Referenced by:

```
TABLE "guarantor" CONSTRAINT "guarantor_lease_id_fkey" FOREIGN
KEY (lease_id) REFERENCES lease(lease_id)
```

```
TABLE "legal_action" CONSTRAINT "legal_action_lease_id_fkey"
FOREIGN KEY (lease_id) REFERENCES lease(lease_id)
```

```
TABLE "payment" CONSTRAINT "payment_lease_id_fkey" FOREIGN KEY
(lease_id) REFERENCES lease(lease_id)
```

```
TABLE "sign" CONSTRAINT "sign_lease_id_fkey" FOREIGN KEY
(lease_id) REFERENCES lease(lease_id)
```

**(iii) Invalid Insert/Updates:**

Commands + Output:

For reference, the start\_date for lease with lease\_id=5 is '2017-07-30'.

#1. Attempting to set end\_date earlier than start\_date.

```
UPDATE lease
SET
    end_date='2013-10-30'
WHERE
    lease_id=5;
```

```
ERROR:  new row for relation "lease" violates check constraint
"valid_lease_date"
DETAIL:  Failing row contains (5, 2017-07-30, 2013-10-30, 57, 742.75,
5 McBride Way).
```

#2. Attempting to add a lease with lease length of 2 years (greater than 1).

```
INSERT INTO lease (lease_id, start_date, end_date, room_no,
monthly_rent, address)
VALUES(101, '2019-01-01', '2021-01-01', 40, 50, '3445 Stanley
Street');
ERROR:  insert or update on table "lease" violates foreign key
constraint "lease_address_fkey"
DETAIL:  Failing row contains (101, 2019-01-01, 2021-01-01, 40, 50,
3445 Stanley Street).
```





## Part 9: Creativity Points

### Automated data generation & real world data:

We inserted 100 real-world-like tuples for the tables apartment, apt\_building, and lease. The data was taken from <https://mockaroo.com/>, and the Python code for formatting the data into insert statements is below.

Evidence:

	lease_id [PK] integer	start_date date	end_date date	room_no integer	monthly_rent double precision	address character varying (250)
69	69	2018-06-22	2020-08-23	37	898.01	86 Scofield Circle
70	70	2017-09-24	2023-12-29	65	852.43	145 Esker Road
71	71	2019-12-17	2022-11-01	19	713.75	10900 Bashford Trail
72	72	2019-02-21	2021-02-24	7	504	0 Corben Pass
73	73	2018-04-19	2020-06-06	65	944.96	2 8th Drive
74	74	2018-04-23	2020-06-16	29	716.17	29 Kenwood Pass
75	75	2017-08-16	2021-11-07	34	656.42	30784 Gina Lane
76	76	2017-11-22	2020-07-25	15	870.1	3 Ridgeway Plaza
77	77	2019-04-18	2022-03-22	23	730.75	9190 Forest Dale Center
78	78	2016-03-29	2023-02-05	82	948.68	389 Ludington Circle
79	79	2016-04-26	2022-06-20	57	612.25	2 Artisan Hill
80	80	2018-03-14	2020-01-04	25	887.12	1631 Southoff Hill
81	81	2018-05-12	2020-06-28	24	604.96	65 Vernon Plaza
82	82	2018-04-20	2021-02-26	71	588.35	68 Rutledge Center
83	83	2017-05-16	2021-03-21	69	818.4	64870 Fairview Avenue
84	84	2019-09-03	2023-10-03	90	786.99	3 Birchwood Alley
85	85	2017-07-25	2020-09-18	60	534.87	443 Toban Way
86	86	2018-06-19	2022-02-03	66	855.56	0166 Oak Valley Park
87	87	2018-04-26	2022-06-12	76	670.18	8661 Hintze Junction
88	88	2019-10-22	2020-12-01	54	788.34	02536 Nobel Circle
89	89	2019-02-26	2023-02-25	31	751.6	9920 Gale Pass
90	90	2017-02-25	2020-12-24	98	680.14	2 Golf Course Plaza
91	91	2017-11-25	2020-07-30	25	968.73	40 Evergreen Pass
92	92	2017-01-13	2023-01-04	74	890.7	93937 Packers Center
93	93	2017-03-26	2020-04-08	69	839.19	20912 Dawn Way

insertStatements.py

```
import csv
import random

csv1 = []
output = []
rooms = [57, 19, 11, 49, 57, 11, 86, 77, 17, 87, 48, 18, 14, 11, 40, 23, 52,
90, 95, 63, 93, 4, 100, 66, 5, 43, 68, 32, 63, 24, 28, 89, 76, 26, 19, 79, 63,
84, 6, 7, 36, 61, 60, 50, 53, 80, 69, 40, 37, 31, 54, 4, 49, 66, 10, 93, 34,
93, 21, 25, 12, 3, 16, 47, 38, 90, 34, 67, 37, 65, 19, 7, 65, 29, 34, 15, 23,
82, 57, 25, 24, 71, 69, 90, 60, 66, 76, 54, 31, 98, 25, 74, 69, 63, 56, 46,
32, 11, 86, 88]
startDates = []
endDates = []

for i in range(0,100):
    startDates.append('\'' + str(random.randint(2016, 2019)) + '-' +
str(random.randint(1, 12)) + '-' + str(random.randint(1, 30)) + '\'')

for i in range(0,100):
    endDates.append('\'' + str(random.randint(2020, 2023)) + '-' +
str(random.randint(1, 12)) + '-' + str(random.randint(1, 30)) + '\'')

with open('D:\Downloads\MOCK_DATA(1).csv') as file:
    reader = csv.reader(file, delimiter=',')
    for row in reader:
        csv1.append('\'' + row[0] + '\'')

for num in range (0, 100):
    print('(' + str(num+1) + ', ' + startDates[num] + ', ' + endDates[num] + ', '
+ str(rooms[num]) + ', ' + str(round(random.uniform(500.00, 1000.00), 2)) + ', '
+ csv1[num] + '), ' )
```