
Goldi-Locks Self-Storage Relational Database

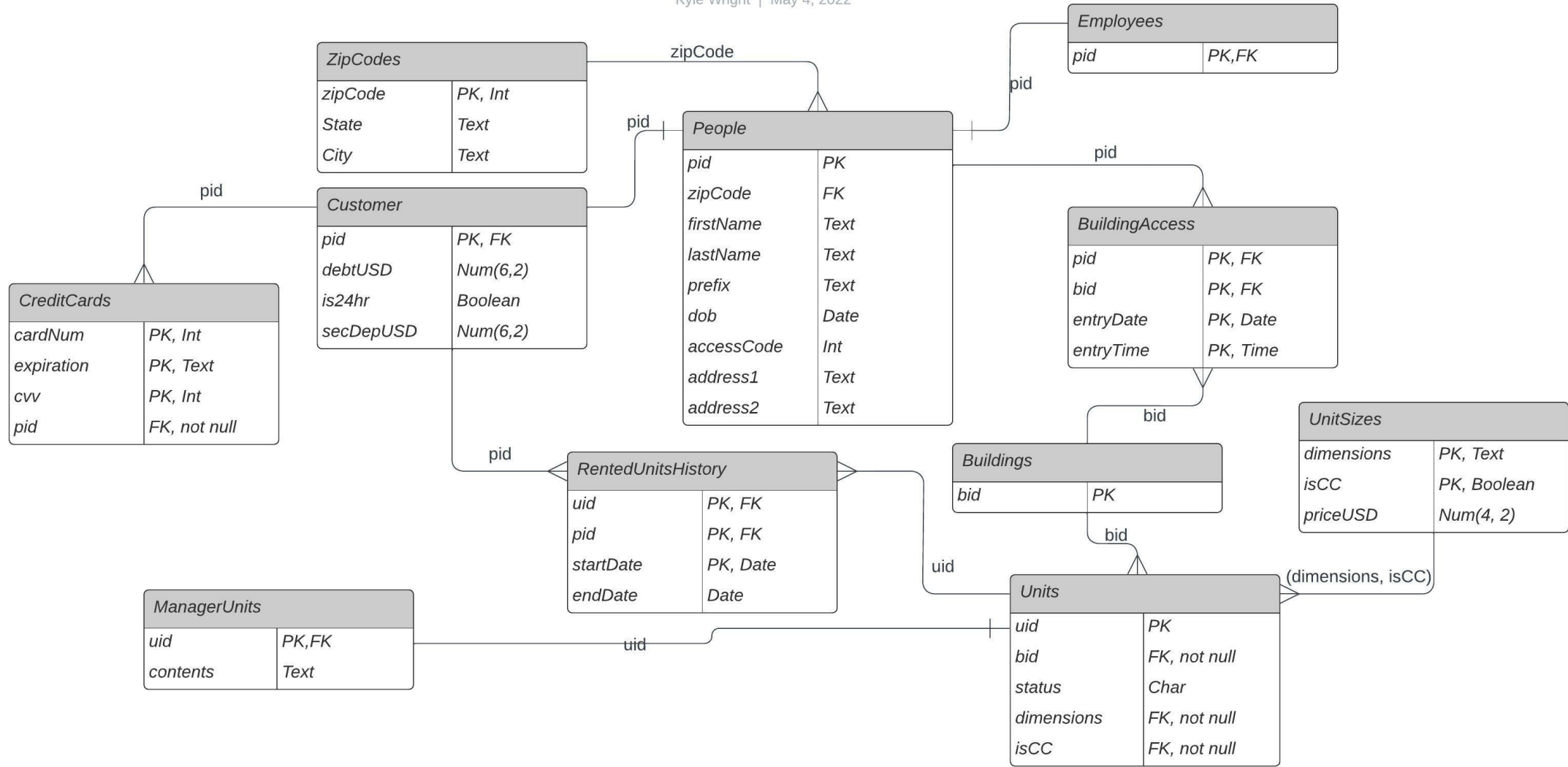
Built by Kyle Wright
Powered by PostgreSQL

Overview


Objectives

Goldi-Locks Self-Storage

Kyle Wright | May 4, 2022






```
--Buildings--  
CREATE TABLE Buildings (  
    bid          int not null,  
    primary key(bid)  
);
```

	bid [PK] integer 
1	1
2	2




--UnitSizes--

```
CREATE TABLE UnitSizes (  
    dimensions  text not null, --not strictly atomic  
    isCC        boolean not null, --is climate controlled  
    priceUSD    numeric(6,2), --monthly rate  
    primary key(dimensions, isCC)  
);
```

	dimensions [PK] text 	iscc [PK] boolean 	priceusd numeric (6,2) 
1	5x5	true	65.00
2	10x10	true	110.00
3	10x10	false	80.00
4	10x20	false	130.00

--ZipCodes--

```
CREATE TABLE ZipCodes (  
    zipCode      text not null,  
    state        text,  
    city         text,  
    primary key(zipCode)  
);
```

	zipcode 	state 	city 
▲	[PK] text	text	text
1	06415	Connecticut	Colchester
2	06249	Connecticut	Lebanon

--Units--

```
CREATE TABLE Units (  
  uid          int not null,  
  bid          int not null references Buildings(bid),  
  status       char, --V=vacant,0=occupied,R=restricted  
  dimensions   text not null,  
  isCC         boolean not null,  
  foreign key(dimensions, isCC) references UnitSizes(dimensions, isCC),  
  primary key(uid)  
);
```

	uid [PK] integer	bid integer	status character (1)	dimensions text	iscc boolean
1	1	1	O	10x10	false
2	2	1	O	10x20	false
3	3	1	O	10x10	true
4	4	2	V	5x5	true
5	5	2	R	5x5	true
6	6	2	O	10x10	false
7	7	2	O	10x20	false

--ManagerUnits--

```
CREATE TABLE ManagerUnits (  
    uid          int not null references Units(uid),  
    contents     text --not strictly atomic  
);
```

	uid integer	contents text
1	1	tools, light bulbs, spare boxes
2	7	Oldsmobile Cutlass 442


--People--

```
CREATE TABLE People (  
  pid          int not null,  
  zipCode      text references ZipCodes(zipCode),  
  address1     text, --line 1  
  address2     text, --line 2  
  prefix       text,  
  firstName    text not null,  
  lastName     text,  
  dob          date,  
  accessCode   int UNIQUE,  
  primary key(pid)  
);
```

	pid [PK] integer	zipcode text	address1 text	address2 text	prefix text	firstname text	lastname text	dob date	accesscode integer
1	10	06415	30 Main St.	[null]	Dr.	Alan	Labouseur	1972-02-13	2633
2	20	06415	786 Park Blvd.	[null]	Dr.	Donald	Schwartz	2061-07-05	5714
3	30	06249	13 Smith St.	Apt. A	[null]	Rick	Treu	1984-05-16	[null]
4	40	06415	89 Lebanon Ave.	[null]	Mr.	Greg	Vilardo	2001-04-26	1198
5	50	06249	2 Cragin Dr.	[null]	Ms.	Linda	Baker	1972-02-13	[null]

--Employees--

```
CREATE TABLE Employees (  
    pid          int not null references People(pid),  
    primary key(pid)  
);
```

	pid [PK] integer 
1	40

--Customers--

```
CREATE TABLE Customers (  
    pid          int not null references People(pid),  
    debtUSD      numeric(8,2) not null,  
    is24hr       boolean not null,  
    secDepUSD    numeric(8,2),  
    primary key(pid)  
);
```

	pid [PK] integer	debtusd numeric (8,2)	is24hr boolean	secdepusd numeric (8,2)
1	10	0.00	true	245.00
2	20	100.00	false	65.00
3	40	20.00	false	110.00
4	30	0.00	false	0.00

--BuildingAccess--

```
CREATE TABLE BuildingAccess (  
    pid          int not null references People(pid),  
    bid          int not null references Buildings(bid),  
    entryDate    date not null,  
    entryTime    time not null,  
    primary key(pid, bid, entryDate, entryTime)  
);
```

	pid [PK] integer	bid [PK] integer	entrydate [PK] date	entrytime [PK] time without time zone
1	40	1	2022-03-02	13:17:09
2	10	1	2022-03-02	22:31:43
3	40	2	2022-03-03	09:03:54
4	20	1	2022-03-04	11:42:43
5	20	2	2022-03-04	10:14:27

--CreditCards--

```
CREATE TABLE CreditCards (  
    cardNum    text not null,  
    expiration text not null,  
    cvv       text not null,  
    pid        int not null references Customers(pid),  
    primary key(cardNum, expiration, cvv)  
);
```

	cardnum [PK] text	expiration [PK] text	cvv [PK] text	pid integer
1	5927384994129411	10/25	905	10

--RentedUnitsHistory--

```
CREATE TABLE RentedUnitsHistory (  
  uid          int not null references Units(uid),  
  pid          int not null references Customers(pid),  
  startDate    date not null,  
  endDate      date CHECK (endDate is null or endDate > startDate),  
  primary key(uid, pid, startDate)  
);
```

	uid [PK] integer	pid [PK] integer	startdate [PK] date	enddate date
1	2	10	2022-03-01	[null]
2	6	10	2022-03-01	[null]
3	5	20	2021-11-15	[null]
4	3	40	2021-05-12	[null]
5	4	30	2021-12-03	2022-02-28
6	2	20	2021-11-15	2022-02-17

--VacantUnits--

CREATE VIEW VacantUnits (uid, bid, status, dimensions, isCC)

AS

select u.uid, u.bid, u.status, u.dimensions, u.isCC

from Units u

where status = 'V'

;

	uid integer	bid integer	status character (1)	dimensions text	iscc boolean
1	4	2	V	5x5	true

```
--RestrictedUnits--
```

```
CREATE VIEW RestrictedUnits (uid, bid, status, dimensions, isCC)
```

```
AS
```

```
    select u.uid, u.bid, u.status, u.dimensions, u.isCC
    from Units u
    where status = 'R' and u.uid not in(select uid
                                         from ManagerUnits)
;
```

	uid integer	bid integer	status character (1)	dimensions text	iscc boolean
1	5	2	R	5x5	true

--ManagerUnits--

CREATE VIEW ManagerUnitsView (uid, bid, status, dimensions, isCC, contents)

AS

```
select u.uid, u.bid, u.status, u.dimensions, u.isCC, mu.contents
from Units u inner join ManagerUnits mu on(u.uid = mu.uid)
;
```

	uid integer	bid integer	status character (1)	dimensions text	iscc boolean	contents text
1	1	1	0	10x10	false	tools, light bulbs, spare boxes
2	7	2	0	10x20	false	Oldsmobile Cutlass 442

--CustomersInDebt--

CREATE VIEW CustomersInDebt (pid, firstName, lastName, debtUSD, address1, address2, city, state, zipCode)

AS

select c.pid, p.firstName, p.lastName, c.debtUSD, p.address1, p.address2, zip.city, zip.state, zip.zipCode

from Customers c inner join People p on(c.pid = p.pid)

inner join ZipCodes zip on(p.zipCode = zip.ZipCode)

where c.debtUSD > 0

;

	pid integer	firstname text	lastname text	debtusd numeric (8,2)	address1 text	address2 text	city text	state text	zipcode text
1	20	Donald	Schwartz	100.00	786 Park Blvd.	[null]	Colchester	Connecticut	06415
2	40	Greg	Vilardo	20.00	89 Lebanon Ave.	[null]	Colchester	Connecticut	06415

```
--TotalDebtAndSecDep--
```

```
CREATE VIEW TotalDebtAndSecDep (totalDebtUSD, totalSecDepUSD)
```

```
AS
```

```
    select sum(c.debtUSD), sum(c.secDepUSD)
```

```
    from Customers c
```

```
    ;
```

	totaldebtusd. numeric	totalsecdepusd. numeric
1	120.00	420.00

--EmployeeAccess--

CREATE VIEW EmployeeAccess (pid, building, entryDate, entryTime)

AS

select ba.pid, ba.bid, ba.entryDate, ba.entryTime

from BuildingAccess ba

where pid **in**(**select** pid
 from Employees)

order by entryDate **desc**

;

	pid integer	building integer	entrydate date	entrytime time without time zone
1	40	2	2022-03-03	09:03:54
2	40	1	2022-03-02	13:17:09

--CustomerAccess--

CREATE VIEW CustomerAccess (pid, building, entryDate, entryTime)

AS

select ba.pid, ba.bid, ba.entryDate, ba.entryTime

from BuildingAccess ba

where pid **in**(**select** pid

from Customers

where pid **not in**(**select** pid

from Employees))

order by entryDate **desc**

;

	pid integer	building integer	entrydate date	entrytime time without time zone
1	20	1	2022-03-04	11:42:43
2	20	2	2022-03-04	10:14:27
3	10	1	2022-03-02	22:31:43

```
--UnitsQuickReport--
```

```
CREATE VIEW UnitsQuickReport (total, manager, vacant, sold, restricted)
```

```
AS
```

```
    select count(u.uid),  
           count(m.uid),  
           count(v.uid),  
           count(s.uid),  
           count(r.uid)
```

```
    from Units u left outer join ManagerUnits m on(u.uid = m.uid)  
              left outer join VacantUnits v on(u.uid = v.uid)  
              left outer join SoldUnits s on(u.uid = s.uid)  
              left outer join RestrictedUnits r on(u.uid = r.uid)
```

```
;
```

	total bigint	manager bigint	vacant bigint	sold bigint	restricted bigint
1	7	2	1	3	1