THE SHAPE OF THINGS

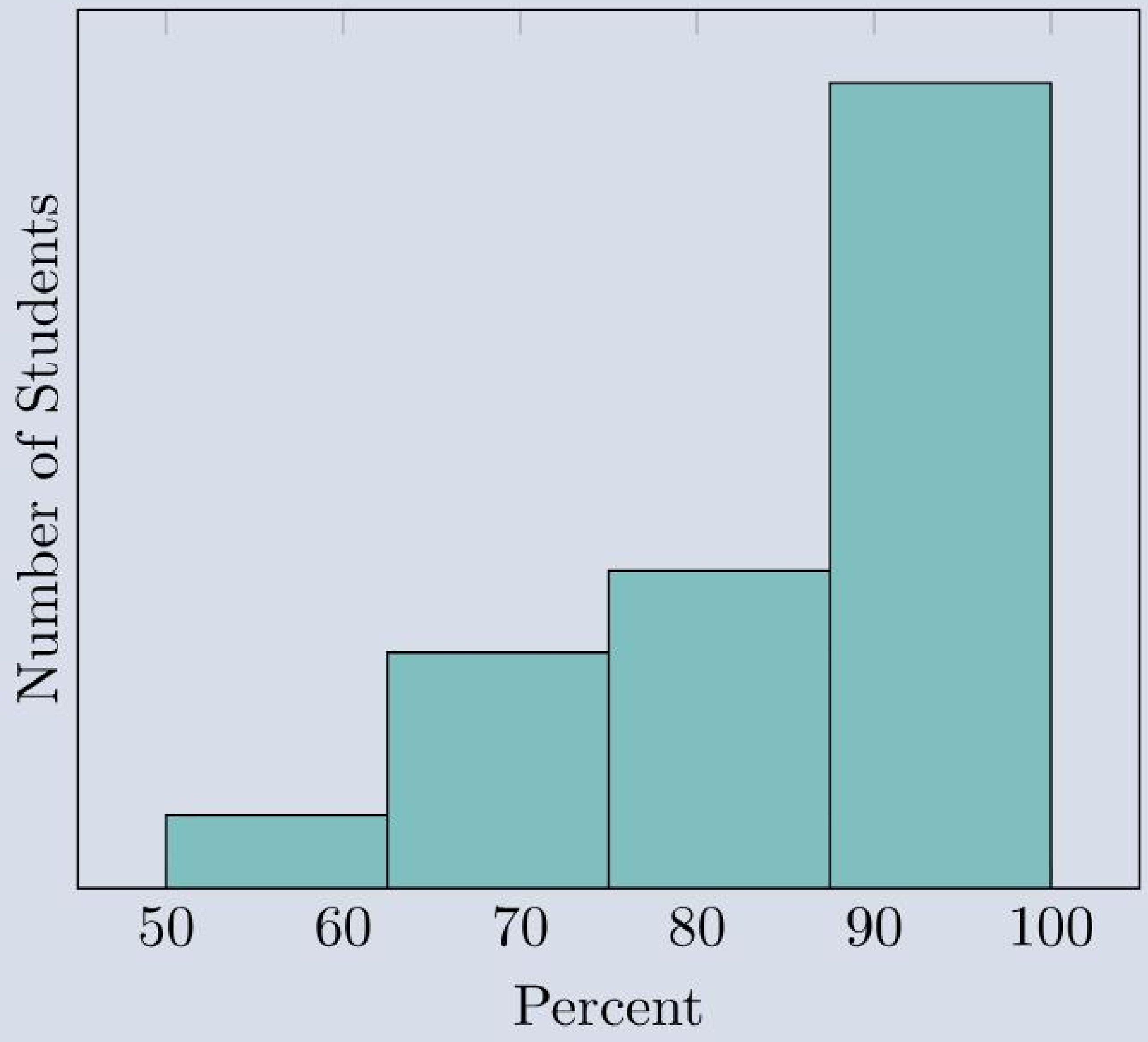
Jed Rembold

Wednesday, November 16, 2022

ANNOUNCEMENTS

- Homework 9 due on Friday
- Project partner assignments and guidelines going out by Friday night!
  - A poll about who you might want to work with (or not work with) will be going out tonight
- We’ll go over the test in a moment
- Polling today: rembold-class.ddns.net

TEST RESULTS



- Average: 82.6%
- Median: 86.8%
- St Dev: 15.9%

PROBLEM 1

```sql
CREATE INDEX bdindex ON to_index_or_not (birthday);
```

```sql
SELECT SUM(favorite_num)
FROM to_index_or_not
WHERE name ILIKE '%Ben%'
```

```sql
SELECT name, birthday
FROM to_index_or_not
WHERE favorite_num = 8
```

```sql
SELECT name
FROM to_index_or_not
WHERE birthday = 'Jan 5, 2014'
```

```sql
SELECT MIN(name)
FROM to_index_or_not
WHERE birthday < 'Nov 2, 2022'
```

PROBLEM 2

```sql
INSERT INTO tab3 VALUES
(7, -4, 'Katy', '2022-02-14');
```

```sql
DELETE FROM tab1
WHERE A = 'Katy';
```

```sql
ALTER TABLE tab3
  ADD FOREIGN KEY (H)
  REFERENCES tab1 (A);
```

```sql
ALTER TABLE tab3
  ALTER COLUMN G
  SET DATA TYPE INT;
```

```sql
UPDATE tab2
SET E = -4.83;
```

```sql
ALTER TABLE tab1
  ADD PRIMARY KEY (C);
```

PROBLEM 3

- Part A

```
SELECT tab2.D, AVG(tab3.F)
FROM tab3
RIGHT JOIN tab2
   ON tab2.D = tab3.I
GROUP BY tab2.D;
```

- Part B

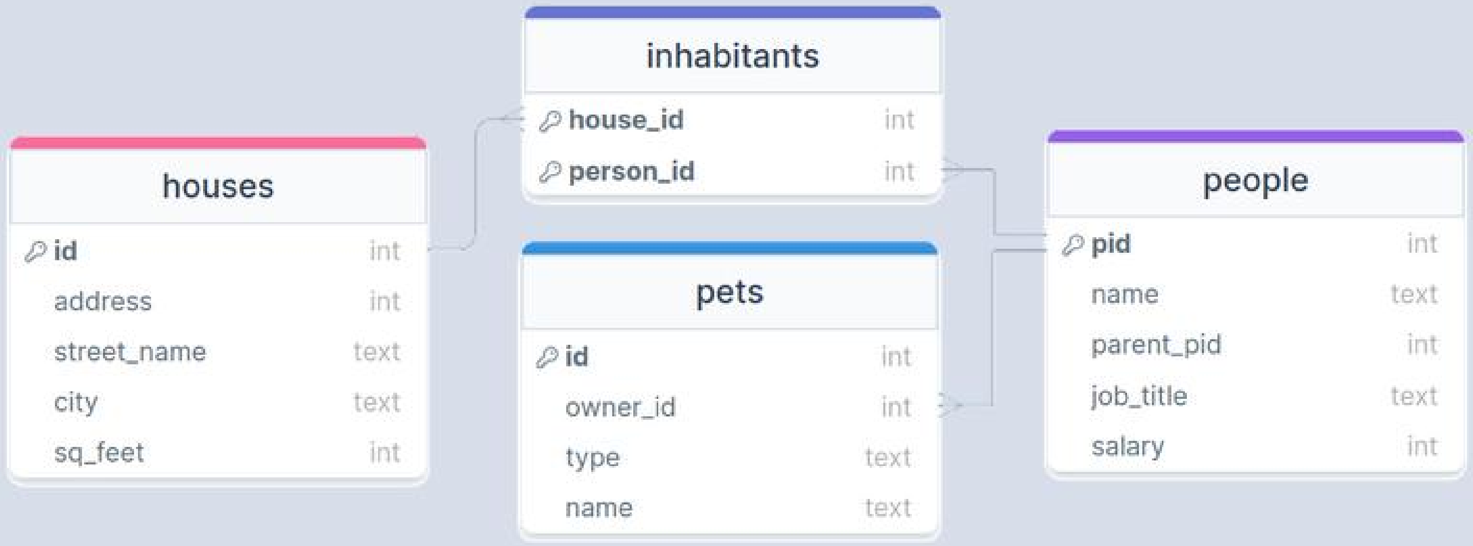
```
SELECT COUNT(*) - MAX(tab1.B + tab3.F)
FROM tab1
FULL OUTER JOIN tab3
  ON tab1.B = tab3.F;
```

PROBLEM 3

- Part C

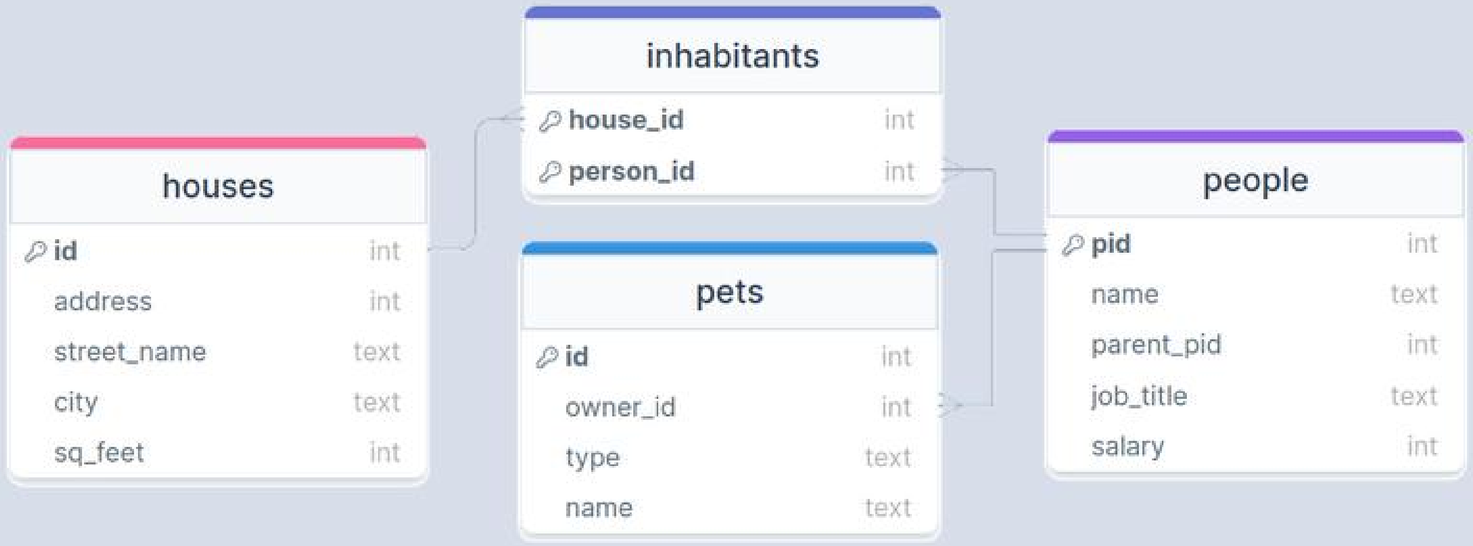
```
SELECT AVG(tab1.B) + COUNT(tab2.D)
FROM tab1
LEFT JOIN tab2
  ON tab1.C = tab2.E
JOIN tab1 AS ttab1
  ON ttab1.B > tab1.C;
```

PROBLEM 4A



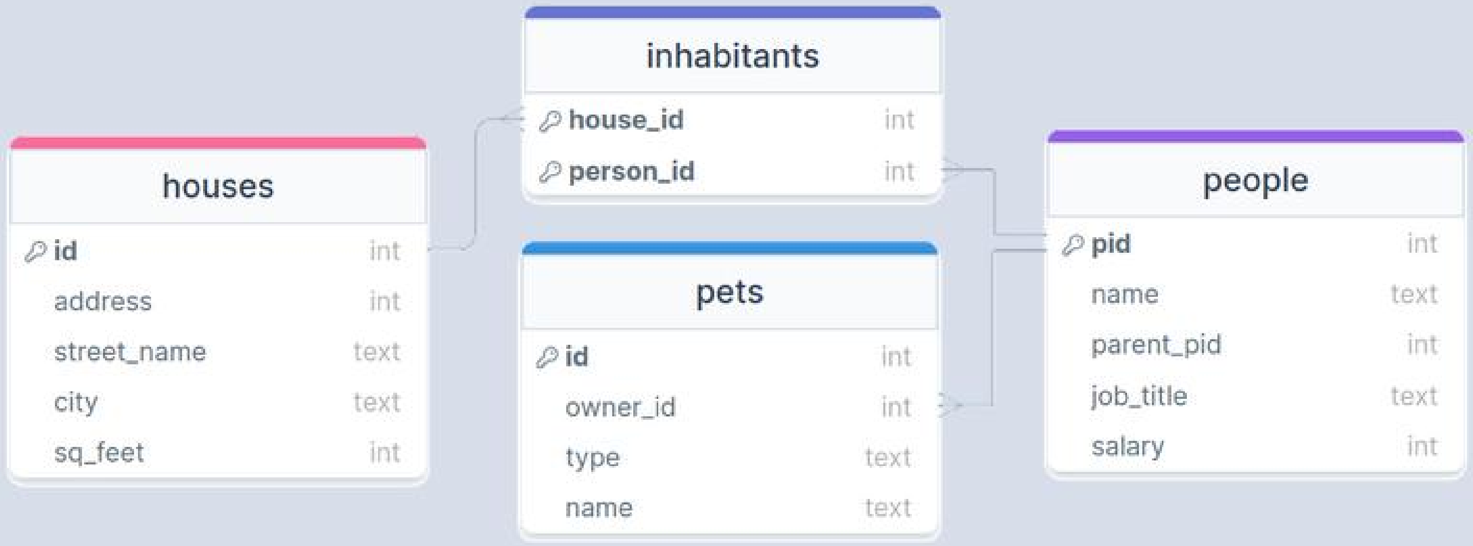
How many houses have been foreclosed, having nobody living within them?

PROBLEM 4B



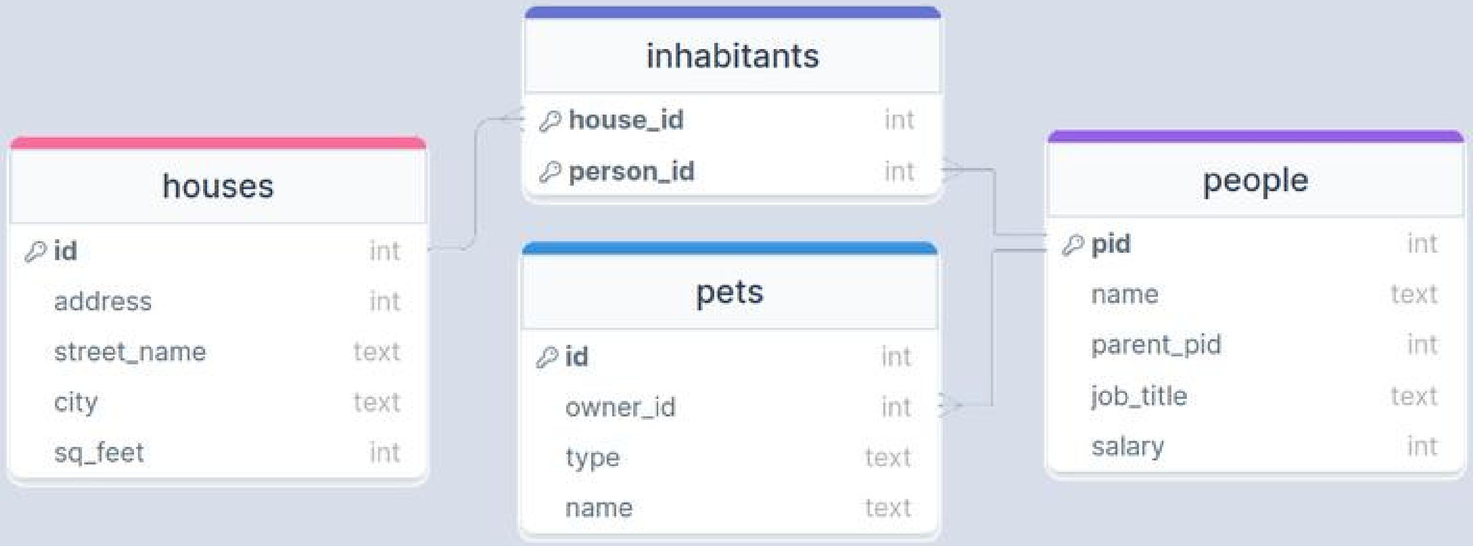
What are the names of the `'CEO'`'s who own more than 3 pets?

PROBLEM 4C



What is the full address (number, street, and city) of the household with the greatest number of pet dogs?

PROBLEM 4D



How many people are living with a parent?

SHAPEFILES

- Life would be very painstaking if you had to recreate complex polygons point by point
- Instead, most sources of spatial information that is more than a single point distribute that information in what is commonly called a *shapefile*
  - Shapefiles are technically the data format developed for the ArcGIS platform
  - Basically a zip which includes several files that contain the necessary information (.shp, .shx, and .dbf, at least)
  - The shapefile contains all the same information as we want, concerning lines, polygons, points, etc, as well as extra explanatory information or annotations
- The general plan is to import the shapefile information into its own table within our database

SHP2PGSQL

- PostGIS comes with a command line utility called `shp2pgsql` on all operating systems
  - Windows users can run a graphical version of the same program, but it is not available on Mac and Linux
    - If you want to run the graphical version, I'll direct you to the book's explanation, which is covered in depth
- Like many command line utilities, `shp2pgsql` utilizes several flags to control its behavior
  - `-I` → sets up a GIST index on the geometry column
  - `-s` → specifies a specific SRID
  - `-W` → specifies a particular encoding if needed (sometimes necessary for location names)

```
shp2pgsql -I -s SRID -W ENCODING SHAPEFILE.SHP TABLE_NAME
```

BRINGING INTO PGSQL

- By itself, `shp2pgsql` will just generate SQL
- You could save or copy that output and then run it in your database, but it can be more useful to pass that SQL directly into your database as it is created
- This can be done with the `|` (pipe) operator
- All together then, the command would look like below (all on one line)

```
shp2pgsql -I -s SRID -W ENCODING SHAPEFILE.SHP TABLE_NAME
| psql -d DATABASE -U postgres
```

- Shapefiles will usually create geometry objects, which you could then cast to geography as needed

BACK TO TEXT

- Since the shapefile spatial information will be encoded directly to a geometry type, it can be tricky to know what exactly you are working with at times
- You can call the `ST_AsText()` function on any geometry (or geography) object to output its WKT representation
  - This can also be useful if you need to get it into a text form to copy into another location

```
SELECT ST_AsText(geom)
FROM table_name
LIMIT 1;
```

VISUALIZATION

- In general, you would need to take your information to another program for visualization purposes
- For a quick view though, this site will let you enter in a WKT which it will then display
- Can be used in conjunction with `ST_AsText` to grab results for quick visualization
- You can use `ST_Collect` to aggregate an entire column of singular geometries into one Multi-geometry object for each of representation

POSTGIS POLYGON FUNCTIONS

- Working with shapefiles gives an easy way to gain access to complex polygonal spatial information
- PostGIS has several useful functions to interact with polygons:
  - `ST_Area(poly)` will return the area of the provided polygon. This will be in SRID specified units if geometry or square meters if geography
  - `ST_Within(point, poly)` will return a True/False as to whether the given point lies within the provided polygon
    - Make sure your SRID values match for point and poly! Or you could get bizarre results!

CROSSINGS

- PostGIS can also determine information about intersections between various geometries
- `ST_Intersects(geom1, geom2)` will return a True/False if there exists an intersection between the two geometries
- `ST_Intersection(geom1, geom2)` will return a new geometry representing the intersection between the two geometries
  - This might be a point for the intersection between two lines or a line for the intersection between a line and a polygon, or a polygon for the intersection between two polygons