0.1 Question 1: Unboxing the Data

0.1.1 Question 1a

As mentioned above, we are working with just one month of data. In the full database (which we don't have access to), tables like the `data` table have billions of rows. What do you notice about the design of the database schema above that helps support the large amount of data and minimize redundancy? **Keep your response to at most two sentences.**

Hint: There is no need to examine any data here. What is a technique learned in lecture? Define that technique.

The schema is normalized to reduce redundancy by decomposing data into relations, as discussed in lecture. This design avoids update and deletion anomalies while supporting billions of rows through natural joins when needed.

0.1.2 Question 1d

Address the two questions below:

1. Can you uniquely determine the building given the sensor data? Why? (**Hint:** given a row in the `data` table, can you determine a **uniquely** associated row in `real_estate_metadata` table? Your answer should draw insights from 1b.)
2. Could `buildings_site_mapping.building` be a valid foreign key pointing to `real_estate_metadata.building_name`? (**Hint:** think about what kinds of columns can be a foreign key.)

Please keep your response to **exactly 1 sentence for each subpart and format your answer like so:**

1. YOUR ANSWER
2. YOUR ANSWER

1. No, because multiple real_estate_metadata rows can share the same building name, so sensor data cannot uniquely determine the building. In other words, the same building maps to multiple real-world buildings, and so we cant uniquely identify a building using just the name in the sensor data.
2. In 1c, we saw that one building name can map to multiple buildings (shortened versions), meaning the mapping isn't one-to-one. For a column to be a valid foreign key, it usually must reference a primary key or unique column- but building name isn't unique.

0.2 Question 3: Entity Resolution

0.2.1 Question 3a

There is a lot of mess in this dataset related to entity names. As a start, have a look at all of the distinct values in the `units` field of the `metadata` table, which contains the units of measurement for a particular piece of metadata (you can use the ungraded code cell below or the terminal).

If you are unfamiliar with a unit of measurement, try searching for it and its abbreviation online.

What do you notice about these values? Are there any duplicates? **Limit your response to one sentence.**

Several units are duplicated in different forms, such as "A" vs. "Amps" and "V" vs. "Volts", indicating inconsistent formatting in measurement units.

In [22]:
```
grading_util.run_sql("""
SELECT DISTINCT units
FROM metadata
ORDER BY units;
""")
```

Out[22]:

	units
0	A
1	Amps
2	Bottom
3	CF
4	CFm
..	…
29	uS
30	V
31	Volts
32	W
33	Wh

[34 rows x 1 columns]

0.2.2 Question 3d

Moving on, have a look at the `real_estate_metadata` table—starting with the distinct values in the `location` field! What do you notice about the spelling of some of these values? (If you're unfamiliar with these locations, search them up online.) **Keep your response to at most 1 sentence.**

Several locations have inconsistent spelling or typos, especially "San Francisco" and potentially "San Diego", which appears in multiple corrupted forms.

```
In [29]: grading_util.run_sql("""
         SELECT DISTINCT location
         FROM real_estate_metadata
         ORDER BY location;
         """)
```

Out[29]:

	location
0	AG FIELD STAT
1	BERKELEY
2	DAVIS
3	FRANCISC O
4	FRANCISC OSAN
5	FRANCISC SOAN
6	IRVINE
7	LOS ANGELES
8	MERCED
9	RIVERSIDE
10	SAN DIEGO
11	SAN DSAIENG O
12	SANTA BARBARA
13	SANTA CRUZ
14	SYSTEMWI DE