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 three sentences.

Hint: There is no need to examine any data here. What is a technique learned in lecture 16? Define that technique.

We are using normalization for our database, that is to say we are splitting up large relations into smaller relations to reduce redundancy.

0.1.2 Question 1d

Do you see any issues with the schema given? In particular, please address the two questions below: - 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.) - Could `buildings_site_mapping.building` be a valid foreign key pointing to `real_estate_metadata.building_name`? (**Hint:** think about the definition / constraints of a foreign key.)

Please keep your response to **at most three sentences.**

Yes I think you can, I think this is the case since the sensor data was contained in the json entry in the table. Furthermore, I think buildings_site_mapping.building could be a foreign key pointing to real_estate_metadata.building_name since they would have values that match and thus it would be appropriate.

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

From looking at the distinct variables, I noticed that there is multiple instances of the same units, such as Amps being present multiple times, and also the same could be said for kVA and KVAR.

```
In [20]: %sql SELECT DISTINCT * FROM metadata LIMIT 5;
```

```
Running query in 'postgresql://jovyan@127.0.0.1:5432/ucb_buildings'
```

```
5 rows affected.
```

```
Out[20]: +--------------------------------------+------------------------+------------------------+-----
         |                  id                  |         class          |          site          | unit
         +--------------------------------------+------------------------+------------------------+-----
         | 02221b82-00fd-59a9-ba1c-c9d4dbbab069 |    3-Ph total (kVAR)   |  Valley Life Sciences  |  kVA
         | aef0c0b6-53fa-5ace-96d0-1eb1db42be59 |       kW b (kW)        |       Cory Hall        |   kW
         | 10c1efd1-34ff-5174-a546-792dbb183701 | I a peak demand (Amps) |     Chavez Center      |  Amp
         | 64c11ca8-b074-5ff6-b7bd-61e9b1b296a7 |      kVA b (kVA)       |    Hildebrand Hall     |  kVA
         | eadc9c75-7edb-5a83-a595-4b27a5ee361a |   I b demand (Amps)    | SRB1 and Oxford Tract  |  Amp
         +--------------------------------------+------------------------+------------------------+-----
```

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 these values? Keep your response to at most two sentences.

I noticed a lot common cities such as Berkeley being present multiple times, and I also noticed a lot of columns are misspelled and this may cause issues when it comes to joining the this table with others.

```
In [27]: %sql SELECT DISTINCT location FROM real_estate_metadata LIMIT 10;
```

```
Running query in 'postgresql://jovyan@127.0.0.1:5432/ucb_buildings'
```

```
10 rows affected.
```

```
Out[27]: +----------------+
         |    location    |
         +----------------+
         |  LOS ANGELES   |
         |   FRANCISC O   |
         |  SYSTEMWI DE   |
         | SAN DSAIENG O  |
         | FRANCISC SOAN  |
         |   SANTA CRUZ   |
         | AG FIELD STAT  |
         |     IRVINE     |
         |   RIVERSIDE    |
         |    BERKELEY    |
         +----------------+
         Truncated to displaylimit of 10.
```