Project Part One

Design Decisions for "hospital" table

The table for "hospital" has the hospital id as its primary key, which is referenced by the "weekly logs" table and the "hospital_quality" table. All hospitals should have a name, so we restrict TEXT NOT NULL for "hospital_name". The "emergency_services" is a BOOLEAN because it indicates whether or not the hospital provided emergency services. Zipcode is a foreign key that references the "locations" table. The variable "geocoded_hospital_address" can be separated into the longitude and latitude coordinates for the hospital, which are likely not integers. All other attributes of the hospital -- "address", "type_of_hospital", and "type_of_ownership" -- are text-based and can have NULL values because it is information that may not be available but isn't crucial to have in order for the hospital to be included. Information such as the type of hospital, type of ownership, and emergency services are recorded here, despite being provided in the Hospital Quality data set. This is because we assume that these values are more or less fixed for a given hospital, and will not change very often (if at all).

```sql
%%sql
CREATE TABLE hospital (
    hospital_pk TEXT PRIMARY KEY,
    hospital_name TEXT NOT NULL,
    address TEXT,
    longitude FLOAT8,
    latitude FLOAT8,
    type_of_hospital TEXT,
    type_of_ownership TEXT,
    emergency_services BOOLEAN,
    zipcode INTEGER NOT NULL REFERENCES locations(zipcode)
);
```

Design Decisions for "weekly_logs" table

Each row in the "weekly_logs" table represents one hospital's activity for one week. It has a serial, autoincrementing column "id" as its primary key. The only other consideration we had for a primary key was to make the combination of the hospital ID and collection week the primary key, but decided against this because we believe we need the hospital ID as a foreign key referencing the "Hospital" table. We need a foreign key so that each record can be linked to an associated hospital. The hospital ID column is a TEXT column, as the primary key in the "Hospital" column, which this column references, is also a TEXT column. It also contains a DATE column which represents the week for which a given entry is from, which we restrict to be NOT NULL, as otherwise a weekly log would be meaningless. This table also contains FLOAT8 columns that represent the number of beds available/occupied for adult patients, pediatric care, and the ICU. All of these columns are floating point numeric columns as they are 7 day averages, so it is likely the values will not be integers. They are also constrained to be non-negative, as we can never have less than 0 beds available or less than 0 patients. We also constrain the number of beds in use to be less than or equal to the number of available beds for each category (adult, pediatric, and ICU), as the number of occupied beds should not exceed the total number of available beds. Lastly, we add a constraint to ensure that the total number of hospitalized patients with confirmed COVID is greater than or equal to the number of ICU patients with confirmed COVID.

```
In [ ]:  %%sql
         CREATE TABLE weekly_logs (
             id SERIAL PRIMARY KEY,
             collection_week DATE NOT NULL,
             adult_beds_available_avg FLOAT8 CHECK (adult_beds_available_avg >=0),
             pediatric_beds_available_avg FLOAT8 CHECK (pediatric_beds_available_avg
             adult_beds_occupied_avg FLOAT8 CHECK (adult_beds_occupied_avg >=0),
             pediatric_beds_occupied_avg FLOAT8 CHECK (pediatric_beds_occupied_avg >=
             icu_beds_available_avg FLOAT8 CHECK (icu_beds_available_avg >=0),
             ice_beds_occupied_avg FLOAT8 CHECK (ice_beds_occupied_avg >=0),
             confirmed_covid_hospitalized_avg FLOAT8 CHECK (confirmed_covid_hospitali
             confirmed_covid_icu_avg FLOAT8 CHECK (confirmed_covid_icu_avg >=0),
             hospital_pk TEXT NOT NULL REFERENCES hospital(hospital_pk),
             CHECK(adult_beds_occupied_avg <= adult_beds_available_avg),
             CHECK(pediatric_beds_occupied_avg <= pediatric_beds_available_avg),
             CHECK(ice_beds_occupied_avg <= icu_beds_available_avg),
             CHECK(confirmed_covid_icu_avg <= confirmed_covid_hospitalized_avg)
         );
```

Design Decisions for "hospital_quality" table

The table for "hospital_quality" stores quality ratings released by CMS for each hospital. We use a surrogate primary key (id) to uniquely identify each record, while hospital_pk serves as a foreign key referencing the "hospital" table to ensure that every rating corresponds to a valid facility. The quality_rating field is numeric and seems to have integer values for their ratings. The date_updated column records when the rating was published. We mark both quality_rating and date_updated as NOT NULL because a quality entry must always include a score and a date in order to be meaningful.

In [ ]:
```sql
%%sql
CREATE TABLE hospital_quality (
    id SERIAL PRIMARY KEY,
    quality_rating INTEGER NOT NULL,
    date_updated DATE NOT NULL,
    hospital_pk TEXT NOT NULL REFERENCES hospital(hospital_pk)
);
```

Design Decisions for "locations" table

The table for "locations" is designed to store geographic information associated with hospitals. We use zipcode as the primary key to represent each ZIP code once within our database and avoid repeating city, state, and county information across multiple hospital records. While multiple hospitals can share the same ZIP code in practice, this design choice keeps our schema simple and still allows us to associate hospitals with their regional attributes efficiently. Storing state, city, and fips_code with each ZIP supports geographic analysis without duplicating location attributes in the "hospital" table. ZIP codes are stored as text to preserve leading zeros and match real-world postal formatting, and the table structure remains easy to join from the "hospital" table through the foreign key reference.

In [ ]:
```sql
%%sql
CREATE TABLE locations (
    zipcode TEXT PRIMARY KEY,
    state TEXT NOT NULL,
    city TEXT NOT NULL,
    fips_code TEXT NOT NULL
);
```