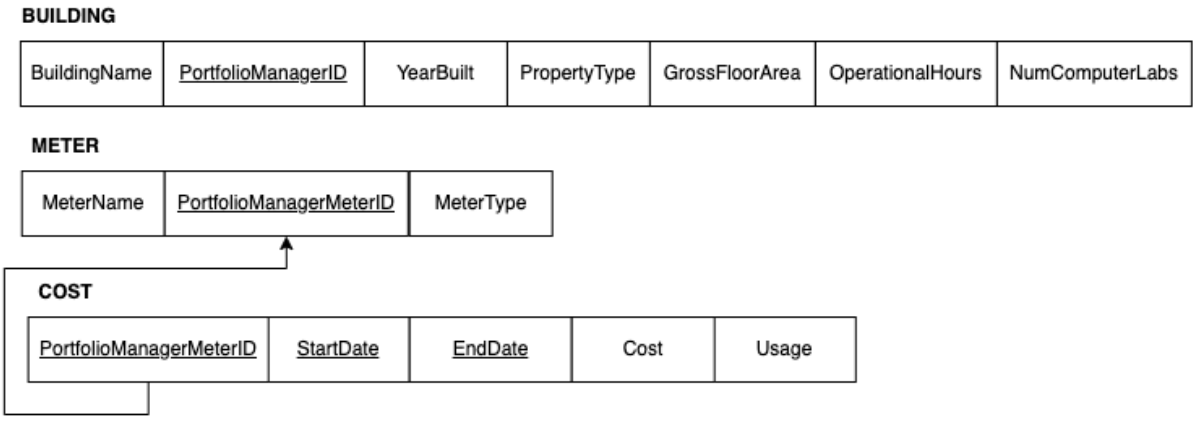
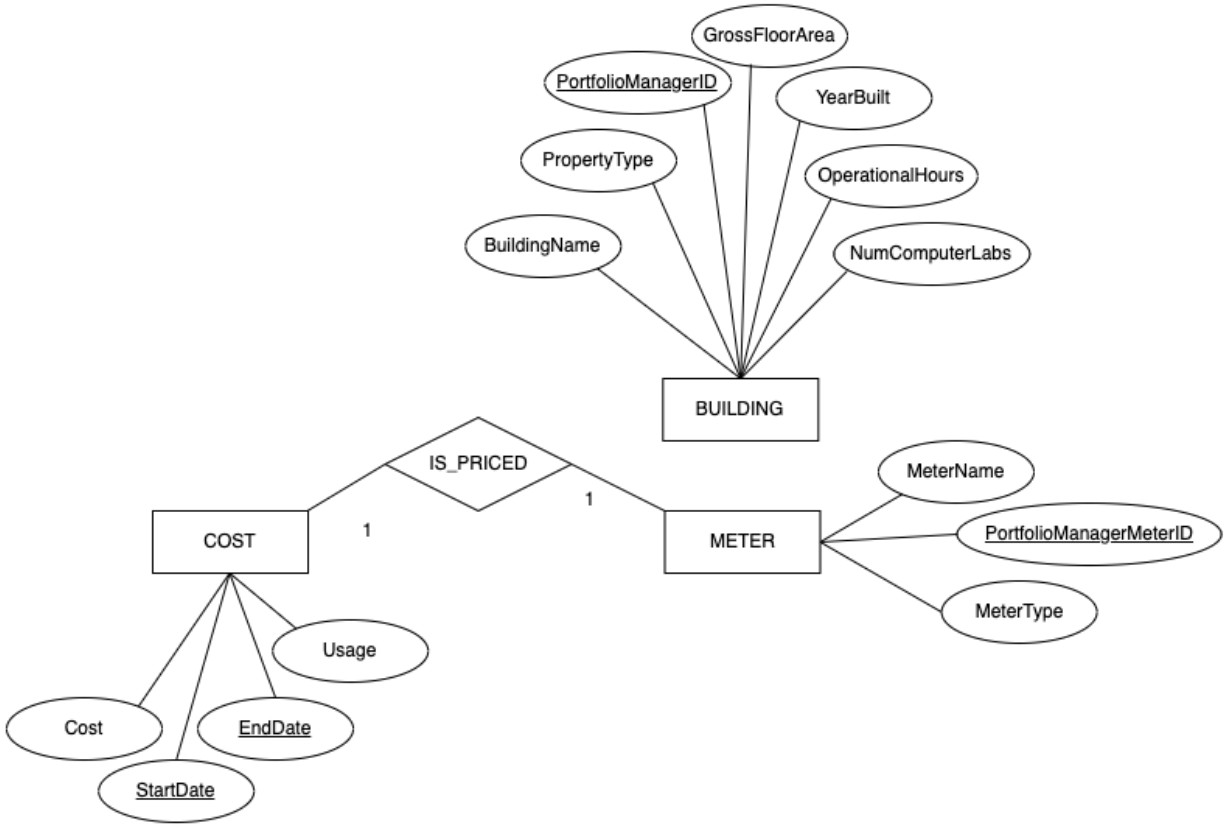
Stage Va – Construction: Tables in Postgre SQL, Queries in SQL

E.g., in an empty folder:
- git clone <path to your GitHub repository>
- cd <repository>
- ./build_db.sh # Stage V(a)
- ./run_app.sh   # Stage V(b)





1. Write and execute SQL data definition language (DDL) commands to create the tables and views in PostgreSQL. Ensure that all constraints are specified.

Create and execute .sql file(s) in an editor. DROPDB ...; CREATEDB ...; CREATE TABLE ...; CREATE VIEW …

```
dropdb energyDemand
createdb energyDemand
psql energyDemand

CREATE TABLE buildings(
        BuildingName text,
        PropertyType text,
        PortfolioManagerID char(7) PRIMARY KEY,
        GrossFloorArea int,
        YearBuilt smallint,
        OperationalHours text,
        NumComputerLabs int
);


CREATE TABLE meters(
        PortfolioManagerMeterID char(8) PRIMARY KEY,
        MeterName varchar(4) UNIQUE,
        MeterType text
);


CREATE TABLE costs(
        PortfolioManagerMeterID char(8),
        StartDate date,
        EndDate date,
        Cost float,
        Usage float,
        FOREIGN KEY (PortfolioManagerMeterID)
        REFERENCES meters (PortfolioManagerMeterID) MATCH FULL,
        PRIMARY KEY (PortfolioManagerMeterID, StartDate, EndDate)
);

CREATE VIEW meter_costs AS
SELECT PortfolioManagerMeterID,
```

```
CREATE VIEW holidays AS
SELECT event_id AS holiday_id, title AS name, starts AS date
FROM events
WHERE title LIKE '%Day%' AND venue_id IS NULL;
SELECT name, to_char(date, 'Month DD, YYYY') AS date
FROM holidays
WHERE date <= '2018-04-01';
ALTER TABLE events
ADD colors text ARRAY;
```

2. Write scripts / programs that may be required to obtain and format the data.

○ Write Python scripts, shell scripts, etc. Don't be lazy: it is important to automate this!!!
○ E.g., input = .csv file(s); output = a .sql file containing many, many INSERT ...

3. Populate the tables with valid data, with all constraints being enforced.

4. Write SQL data manipulation language (DML) commands that were designed in the previous stages. The queries must be elegant and make effective use of complex query constructs such as subqueries. Execute and test these queries to ensure that they work correctly. Examine the outputs carefully to verify that the queries do not return spurious tuples.

Execute a .sql file containing your queries from Stage IV

```
SELECT Cost, Usage
FROM meters NATURAL JOIN costs
WHERE MeterType = 'Electric - Grid' AND StartDate >= '2018-03-01' AND StartDate <=
'2018-03-31' ;

SELECT Cost, Usage
FROM meters NATURAL JOIN costs
WHERE MeterType = 'Natural Gas' AND StartDate >= '2018-03-01' AND StartDate <=
'2018-03-31' ;

SELECT BuildingName, YearBuilt, PropertyType, GrossFloorArea, OperationalHours,
NumComputerLabs
FROM buildings
WHERE PortfolioManagerID = '6610023';
```

5. Create issues to assign tasks to each group member and update existing ones. Create new issues for tasks that need to be completed or can be added in a future version of the application.