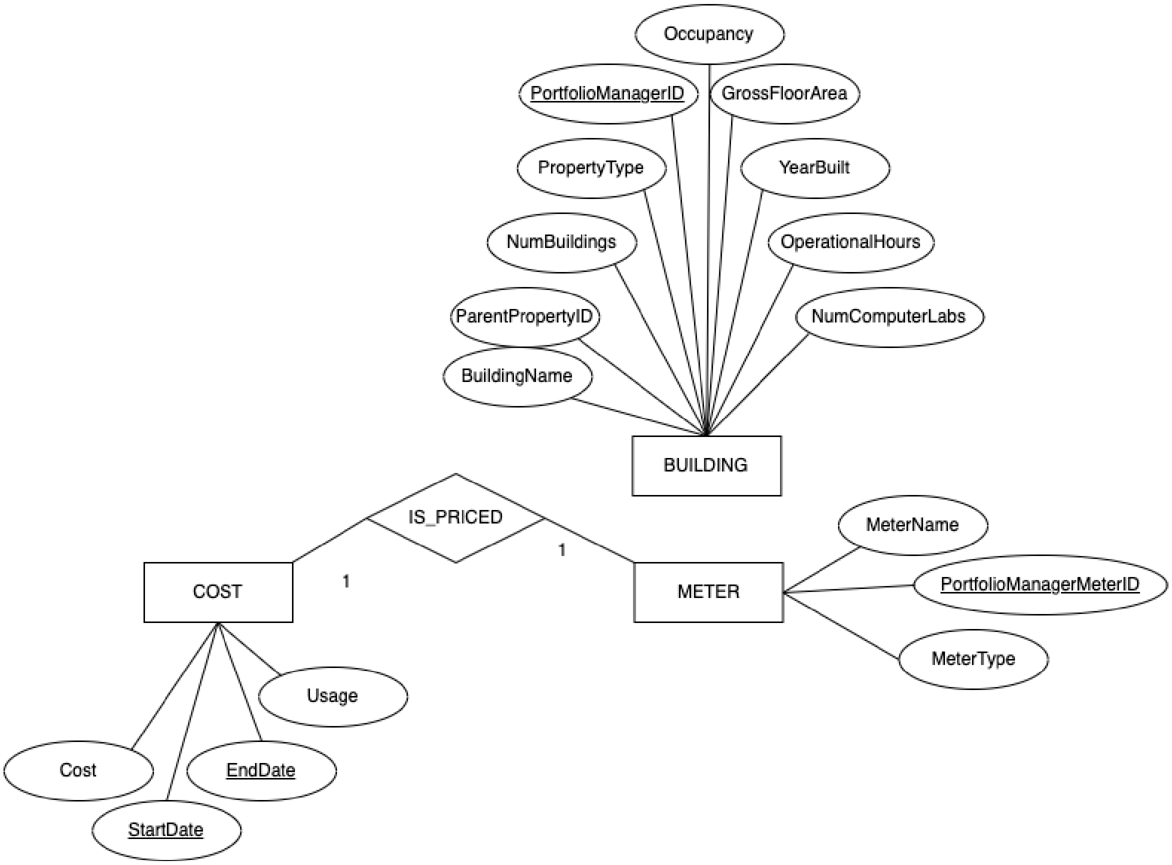
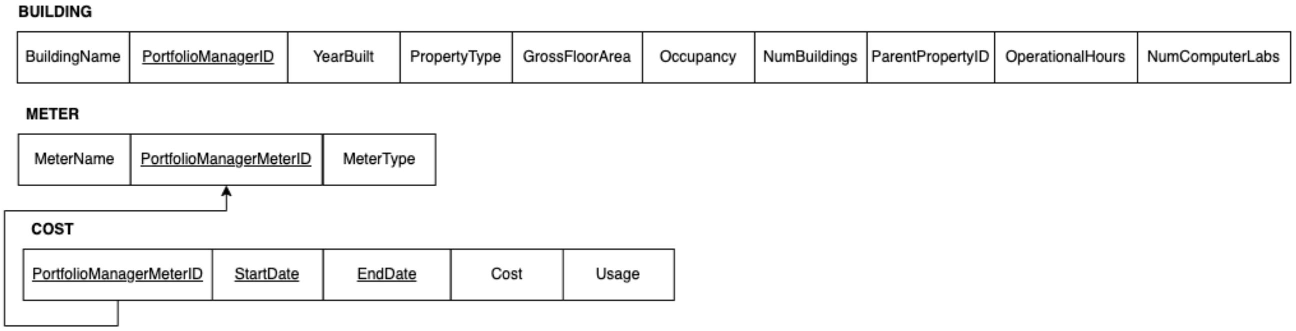
Thomas Camilli, Zane Mrezek, Julianna Iannini
Stage IV – Elaboration: Database Design

1. Review the Database Model document with the stakeholder, and update the model as needed.





**BUILDING**

| BuildingName | PortfolioManagerID | YearBuilt | PropertyType | GrossFloorArea | Occupancy | NumBuildings | ParentPropertyID | OperationalHours | NumComputerLabs |
|---|---|---|---|---|---|---|---|---|---|

**METER**

| MeterName | PortfolioManagerMeterID | MeterType |
|---|---|---|

**COST**

| PortfolioManagerMeterID | StartDate | EndDate | Cost | Usage |
|---|---|---|---|---|

We removed the relationship relation between "Building" and "Meter" because there is none in the data. Our approach for querying the database will be gathering necessary cost information from each meter and then dividing that cost data by what building we are approximating energy usage for based on gross floor area, hours of operation, the number of computer labs, age of the building, and any other factors that could be significant.

We also modified the cost relation to include the portfolio manager meter id, because the previous model's "is_priced" relation redundantly stored start and end dates.

We also removed the redundant attribute "MeterType" from cost because it is already in the "Meter" relation.

While converting to 2NF we noticed that usage units partially depend on the primary key of the "cost" relation, it does not depend on the date, only the "usage" values will fully depend on the primary key so we decided to remove "UsageUnits". In fact, the units for usage were actually already included in the "Meter" relation so it made sense to eliminate"UsageUnits" from the "cost" relation.

When converting to 3NF we noticed that there was a transitive dependency where units depended on "MeterType" which is not a prime attribute. Thus we removed units from the "Meter" relation. We could create another table with the primary key of "MeterType" and store the units of the meter that way, however, that would add an extra step in querying the data by joining an extra table. Instead, we plan to include the units (kilowatts or therms) in the UI when the user selects a view.

2. Demonstrate that all the relations in the relational schema are normalized to Boyce-Codd normal form (BCNF).
    a. For each table, specify whether it is in BCNF or not, and explain why.
        i. Each relation has only a single valued, atomic attribute. Thus it is in 1NF.
        ii. Every non prime attribute is fully (not partially) functionally dependent on the primary key. The schema is also in 1NF thus it is 2NF.
            1. Because our primary keys in "Building" and "Meter" are single attributes, it is impossible to have any partial dependencies.
        iii. No attribute depends on a non prime attribute, thus there are no transitive dependencies. The relation is also in 2NF thus the relation is in 3NF.
        iv. No prime attribute is dependent on a non prime attribute, and the relation is in 3NF. Thus this relation is in Boyce-Codd normal form.

    b. For each table that is not in BCNF, show the complete process that normalizes it to BCNF.
        i. All tables are in BCNF

3. Define the different views (virtual tables) required. For each view list the data and transaction requirements. Give a few examples of queries, in English, to illustrate.

Given Electricity & Month

- Data:
    - YearBuilt
    - PropertyType
    - GrossFloorArea
    - Occupancy
    - NumBuildings
    - OperationalHours
    - NumComputerLabs
    - Cost
    - Usage
- Transaction Requirements: Transactions are primarily used when updating data rather than retrieving. The data for the most part remains constant and will hardly ever change so there is no need for extensive design of transactions. For example, a building's gross floor area will only change if that building happens to get an addition built on it which is very rare.

Given Natural Gas & Month

- Data:
    - YearBuilt
    - PropertyType
    - GrossFloorArea
    - Occupancy
    - NumBuilding
    - OperationalHours
    - NumComputerLabs
    - Cost
    - Usage
- Transaction Requirements: Transactions are primarily used when updating data rather than retrieving. The data for the most part remains constant and will hardly ever change so there is no need for extensive design of transactions. For example, a building's gross floor area will only change if that building happens to get an addition built on it which is very rare.

Query Examples:

- Select cost and usage for electric type and a certain date.
- Select cost and usage for natural gas and a certain date.
- Select building data for a specific building.

4. Design a complete set of SQL queries to satisfy the transaction requirements identified in the previous stages, using the relational schema and views defined in tasks 2 and 3 above.

```
SELECT Cost,Usage
FROM Meter NATURAL JOIN Cost
WHERE MeterType = ‘Electric’ AND Date = ‘01/01/2018’;
```

```
SELECT Cost,Usage
FROM Meter NATURAL JOIN Cost
WHERE MeterType = ‘Natural Gas’ AND Date = ‘01/01/2018’;
```

```
SELECT
YearBuilt,PropertyType,GrossFloorArea,Occupancy,NumBuilding,OperationalHours,NumComputerLabs
FROM Building
WHERE PortfolioManagerID = ‘123456789’;
```