Stage IV – Elaboration: Database Design

2. Demonstrate that all the relations in the relational schema are normalized to Boyce–Codd normal form (BCNF).

- For each table, specify whether it is in BCNF or not, and explain why.

Meters: This table is not in BCNF normalization, because there exists a transitive dependency between start_date and end_date; knowing start_date allows one to infer end_date, despite start_date not being a primary key.

Energy Metrics: This table is not in BCNF normalization, since there is a transitive dependency between type and usage_units, as type implies usage_units, despite type not being a unique identifier.

Green Energy: This table is in BCNF normalization. There are no transitive or partial dependencies in the table, and the table has a primary key.

Emissions: This table is in BCNF normalization. There are no transitive or partial dependencies in the table, and the table has a primary key

FDs: *consumption_id -> {type, name, start_date, end_date, cost, usage_quantity, usage_units, etype}*
type -> avg_emissions

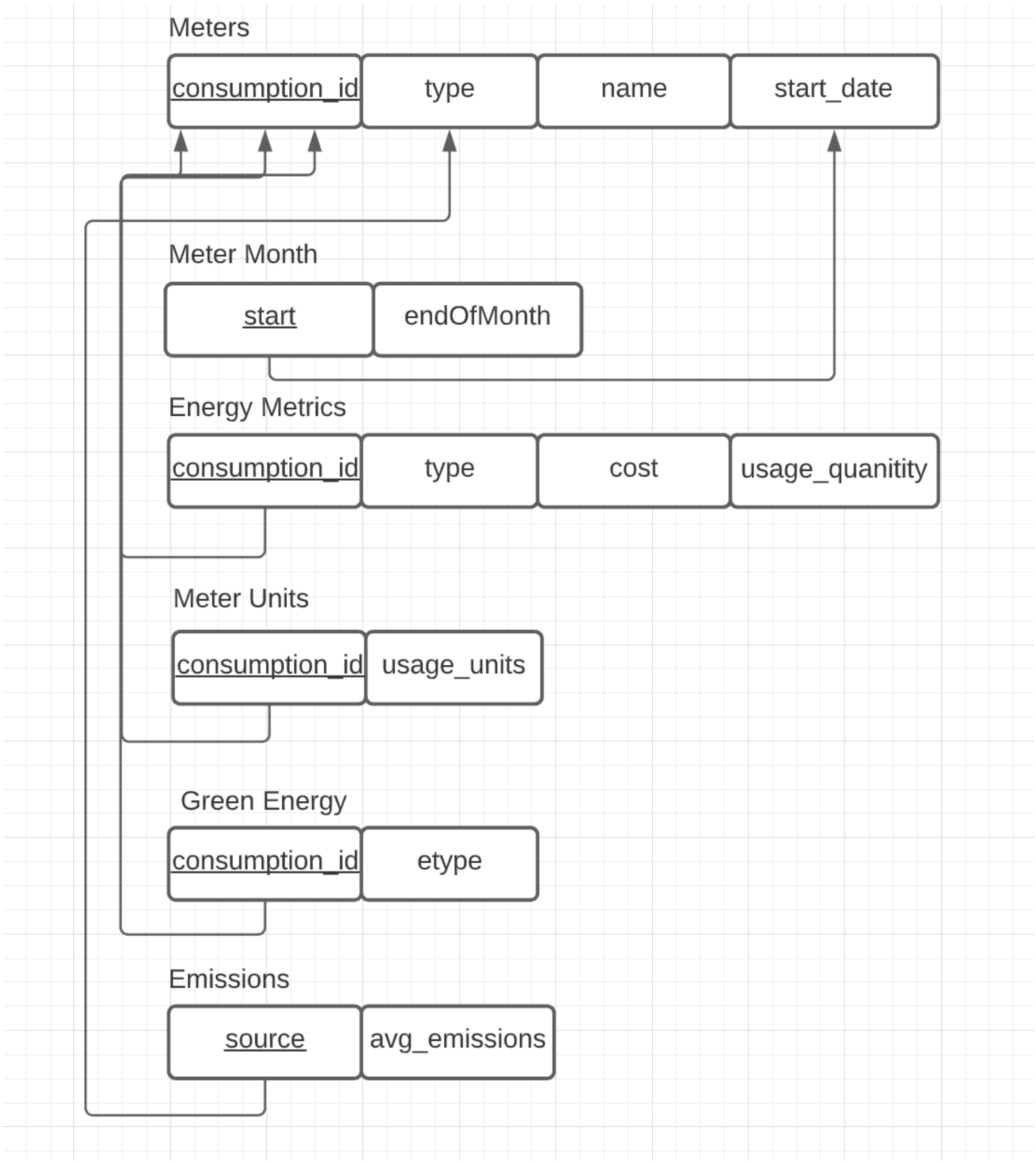
TDs: *{start_date->end_date} in Meters*
{type->usage_units} in Energy Metrics

- For each table that is not in BCNF, show the process that normalizes it to BCNF.

Meters: Break the table up into 2 smaller tables to get rid of the transitive dependency. We will have our meters table include {consumption_id, type, name, and start_date}. The second table would be named Meter_Month and include {start_date and end_date}. These two tables would not be in BCNF since there will be no transitive dependencies inside of them as well as no partial dependencies.

Energy Metrics: This table can be split into Energy Metrics {consumption_id, type, cost, usage_quantity} and Meter Units, which is {consumption_id, usage_units}.

Updated Relational Schema:



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.

Views:

- Total energy data
- Average emissions per energy source
- Minimum cost per meter
- Maximum cost per meter
- Average cost per type
- Average cost per meter
- Total emissions per type

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.

Total energy data:

```
SELECT * FROM meters
JOIN emissions ON meters.type = emissions.source
JOIN energy_metrics ON meters.consumption_id = energy_metrics.consumption_id
JOIN meter_units ON meters.consumption_id = meter_units.consumption_id;
```

Average emissions per energy source:

```
SELECT * FROM meters
JOIN emissions ON type = source;
GROUP BY type;
```

Minimum cost per meter:

```
SELECT consumption_id, type, MIN(cost), usage_quantity
FROM energy_metrics
GROUP BY name;
```

Maximum cost per meter:

```
SELECT consumption_id, type, MAX(cost), usage_quantity
FROM energy_metrics
GROUP BY name;
```

Average cost per type:

```
SELECT type, AVG(cost), usage_quantity
FROM energy_metrics
GROUP BY type;
```

Average cost per meter:

```
SELECT type, AVG(cost), usage_quantity
FROM energy_metrics
GROUP BY type
NATURAL JOIN meters;
```

Total emissions per type

```
SELECT source, SUM(emissions.avg_emissions)
FROM emissions
GROUP BY source;
```