Sol: Clean Energy Market Operation DBMS

IE6700: Project Showcase

Study Group 9

Siddharthan Singaravel and Akhila Tanubbodi

Introduction

Sol is a new-age clean energy procurement and market operator that strives to minimize the dependencies on petroleum-based energy sources by utilizing the residential solar-energy units hold.

Sol intends to build a centralized data infrastructure that keeps track of a variety of user and energy-related metrics.

Need? 

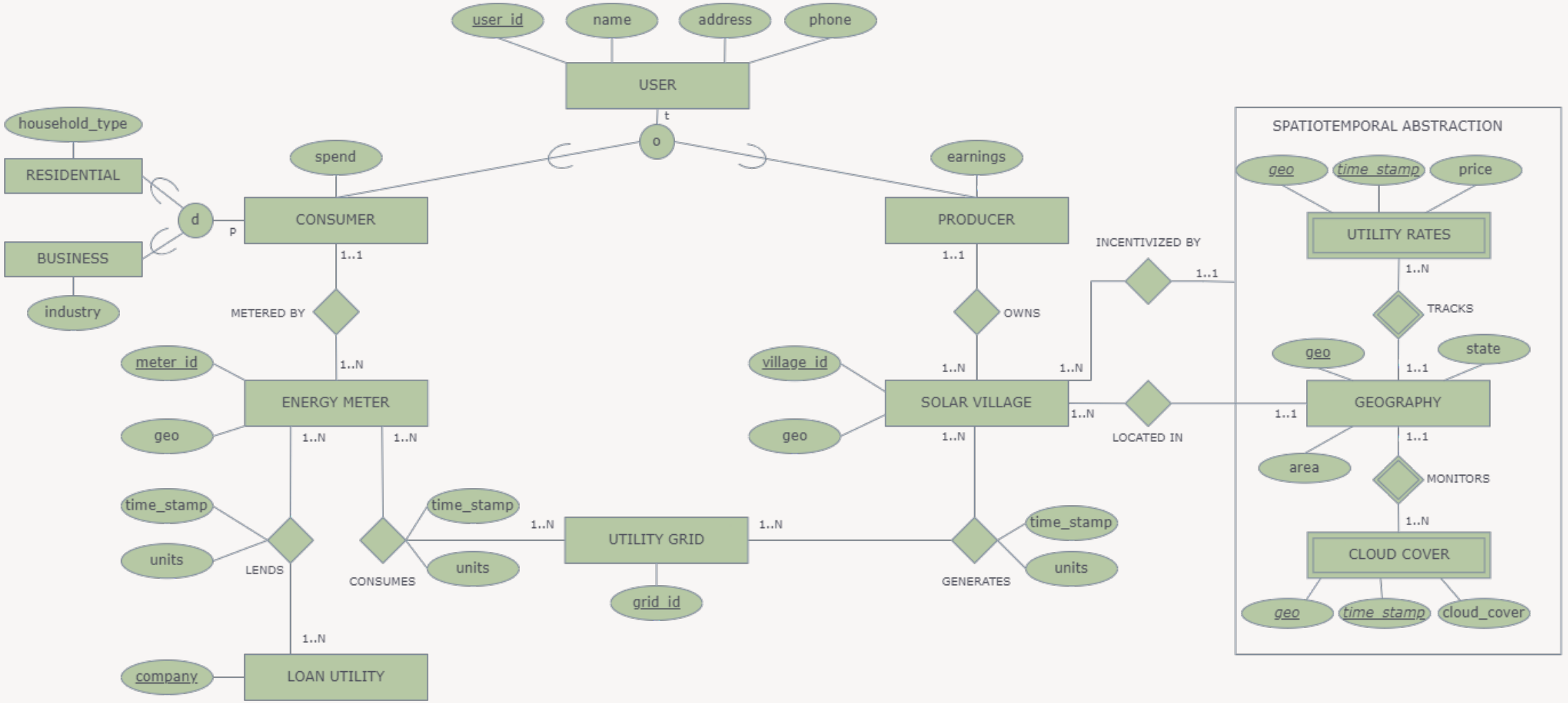
With sol_cleanergy SCHEMA, we have created a Database Management System in MySQL using Workbench. The schema houses all operational data objects required for Sol's operations.

Relational Model 

In addition to the relational model, we have incorporated a non-relational graph model using Neo4j for accessing village linkages.

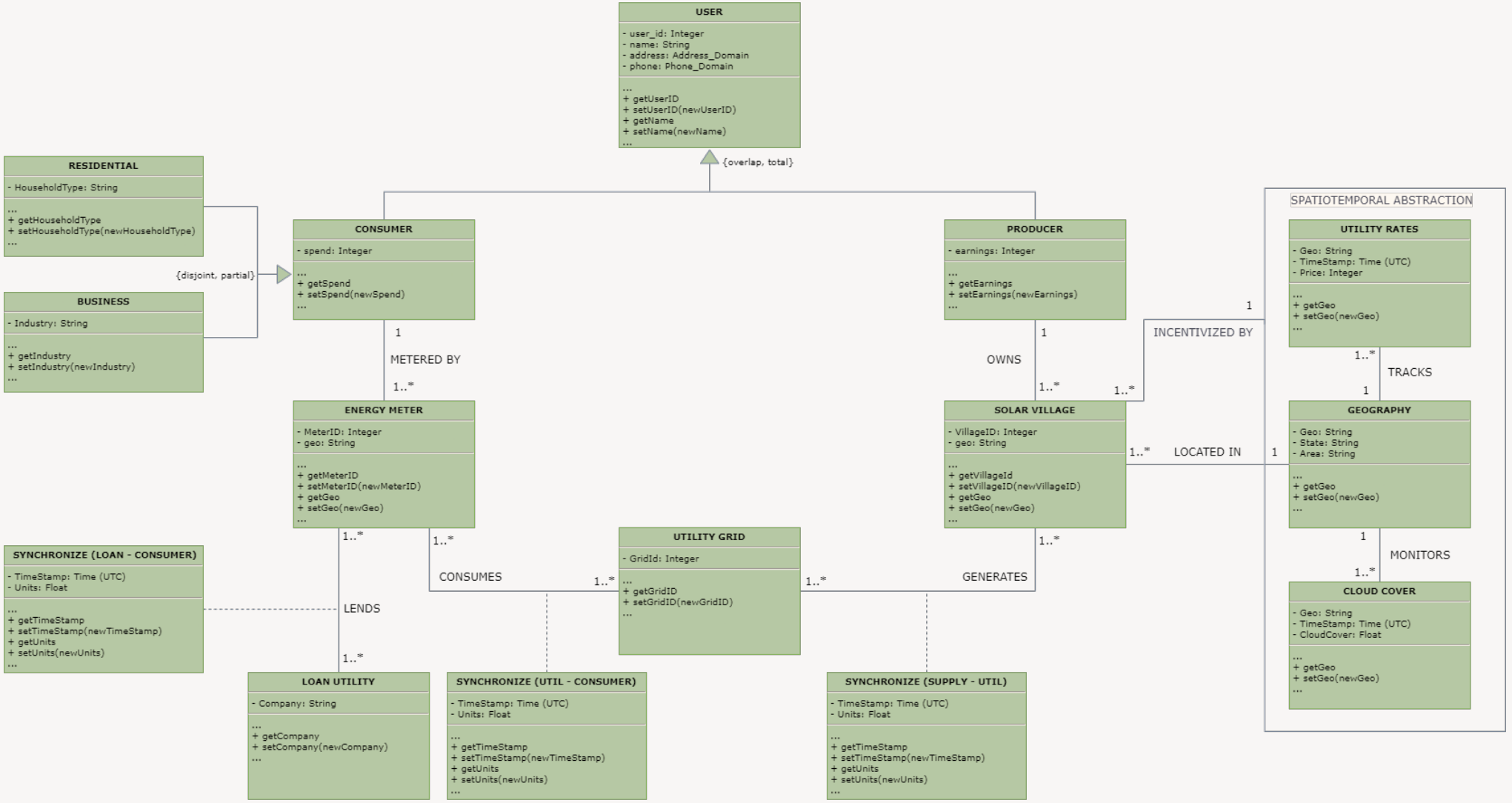
Non-relational Model 

Enhanced Entity Relationship Model



Sol: Clean Energy Market Operation DBMS

Unified Modeling Language Model



Sol: Clean Energy Market Operation DBMS

Relational Model

1

PRODUCER (user_id, name, add_streetnr, add_streetname, add_zipcode, add_city, phone, earnings)

CONSUMER (user_id, name, add_streetnr, add_streetname, add_zipcode, add_city, phone, spend)

> **RESIDENTIAL** (*user_id*, household_type, …)
> **BUSINESS** (*user_id*, industry, …)

SOLAR_VILLAGE (village_id, geo, *user_id*, …)

- user_id: NOT NULL

UTILITY_GRID (grid_id, …)

GENERATES (*grid_id*, *village_id*, time_stamp, units)

- grid_id: NOT NULL, village_id: NOT NULL

ENERGY_METER (meter_id, geo, user_id, …)

- user_id: NOT NULL

2

LOAN_UTILITY (company, …)

LENDS (*meter_id*, *company*, time_stamp, units)

- meter_id: NOT NULL, company: NOT NULL

CONSUMES (*grid_id*, *meter_id*, time_stamp, units)

- grid_id: NOT NULL, meter_id: NOT NULL

GEOGRAPHY (geo, state, area)

UTILITY_RATES (*geo*, time_stamp, price)

- geo: NOT NULL

CLOUD_COVER (*geo*, time_stamp, cloud_cover)

- geo: NOT NULL

The spatiotemporal aggregation is linked with the SOLAR VILLAGE at its most recent instance, signifying exactly one tuple being associated with a SOLAR VILLAGE entity from the aggregation. If there is a need to link the aggregation with SOLAR VILLAGE for all time stamped occasions, then we must introduce the attribute type time_stamp in the relationship itself. Then, a new relation, INCENTIVIZED_BY, needs to be constructed using the village_id from the SOLAR VILLAGE entity type along with the key attributes from the two temporal entity types inside the abstraction.

MySQL Queries

GROUP BY, ORDER BY

1

Loan Company: Units lent on 5th of January 2021

Nested Query, UNION

2

Consumers with the lowest and highest energy expenditure

EXISTS

3

PRODUCERS who are also CONSUMERS and earn less than what they spend

ALL (Multiset)

4

Village, total units generated, & owner for that village for villages that generated the highest number of units in that area

Subqueries inside SELECT

5

Village, and username from subquery inside SELECT

JOINS

6

Calculate the remuneration, and list the top 10 remuneration for Sol Users

VIEWS

7

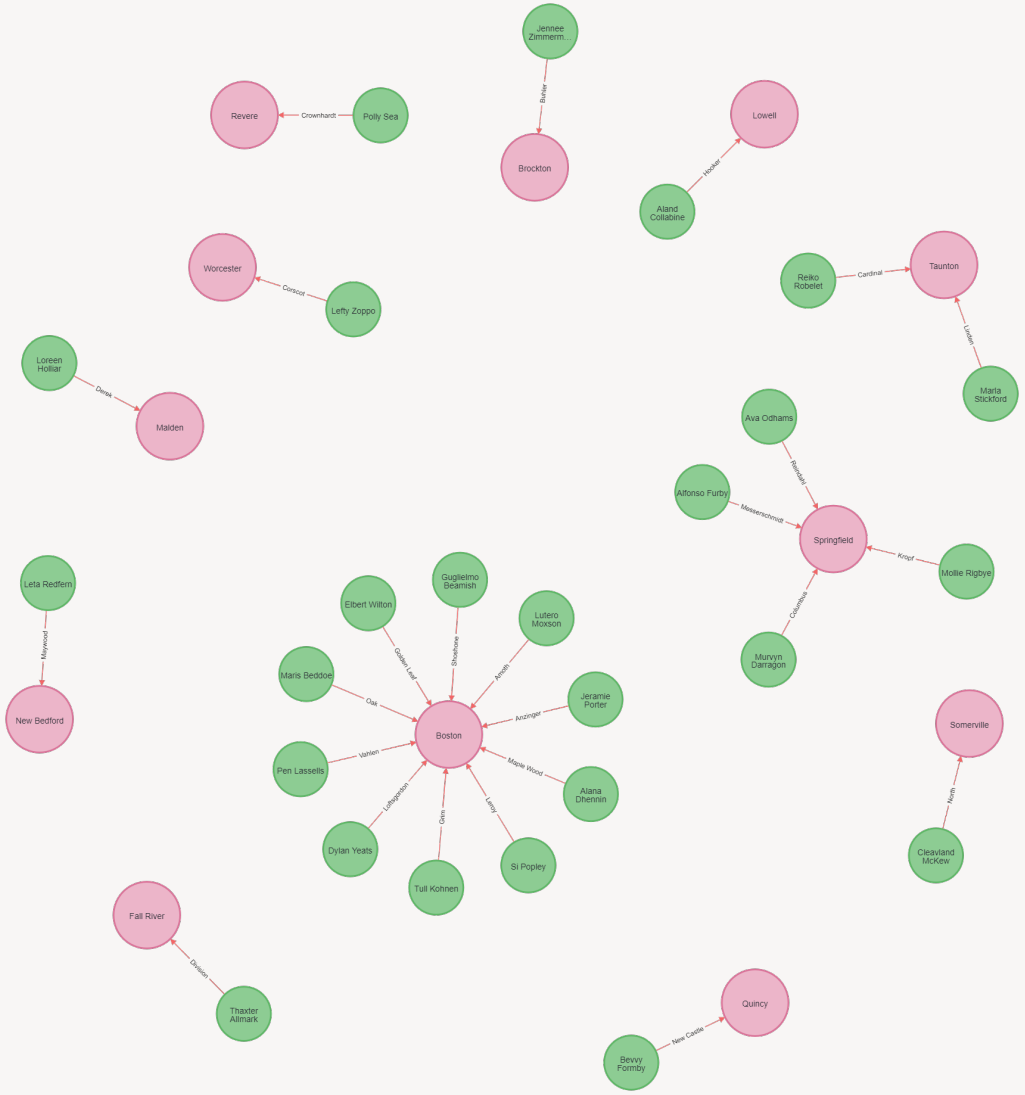
Join Producers and Consumer datasets into one

WITH

8

Lent units as a proportion of total generated units on that day

Graph Relationships [:RESIDES_IN]

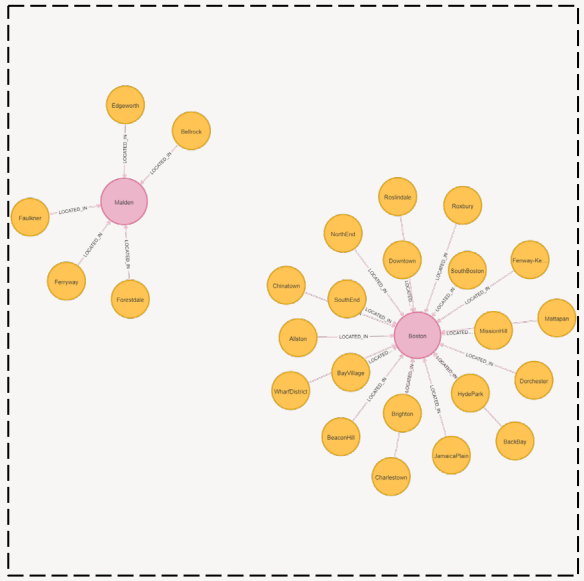


RESIDES_IN Relationship

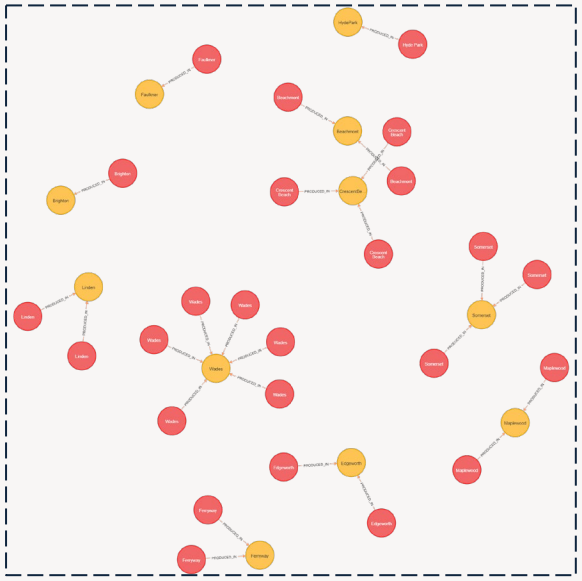
```
MATCH p=()-[r:RESIDED_IN]->() RETURN p LIMIT 25
```

Other Graph Relationships

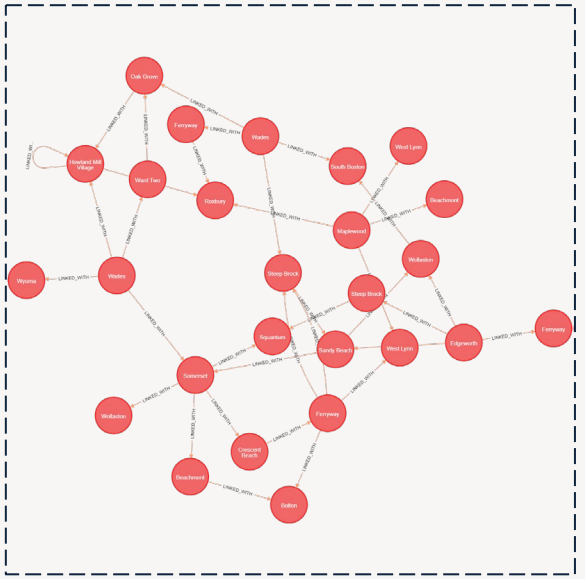
LOCATED_IN



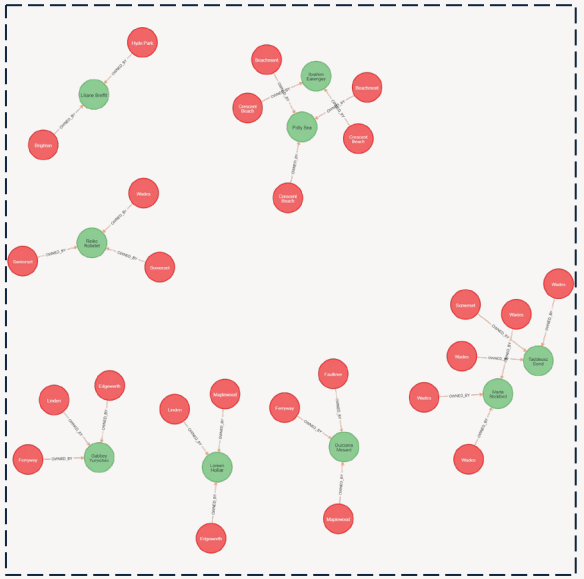
PRODUCED_IN



LINKED_WITH



OWNED_BY



Sol: Clean Energy Market Operation DBMS

Adding INCENTIVIZED_BY

1

PRODUCER (user_id, name, add_streetnr, add_streetname, add_zipcode, add_city, phone, earnings)

CONSUMER (user_id, name, add_streetnr, add_streetname, add_zipcode, add_city, phone, spend)

> **RESIDENTIAL** (*user_id*, household_type, …)
> **BUSINESS** (*user_id*, industry, …)

SOLAR_VILLAGE (village_id, geo, *user_id*, …)

- User_id: NOT NULL

UTILITY_GRID (grid_id, …)

GENERATES (*grid_id*, *village_id*, time_stamp, units)

- grid_id: NOT NULL, village_id: NOT NULL

ENERGY_METER (meter_id, geo, user_id, …)

- user_id: NOT NULL

2

LOAN_UTILITY (company, …)

LENDS (*meter_id*, *company*, time_stamp, units)

- meter_id: NOT NULL, company: NOT NULL

CONSUMES (*grid_id*, *meter_id*, time_stamp, units)

- grid_id: NOT NULL, meter_id: NOT NULL

GEOGRAPHY (geo, state, area)

UTILITY_RATES (*geo*, time_stamp, price)

- geo: NOT NULL

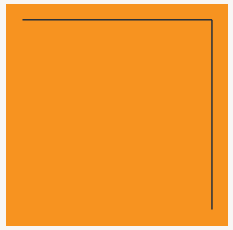
CLOUD_COVER (*geo*, time_stamp, cloud_cover)

- geo: NOT NULL

INCENTIVIZED_BY (*village_id*, geo, time_stamp, *price*, *cloud_cover*)

- village_id: NOT NULL, price: NOT NULL, cloud_cover: NOT NULL

The spatiotemporal aggregation is linked with the SOLAR VILLAGE at its most recent instance, signifying exactly one tuple being associated with a SOLAR VILLAGE entity from the aggregation. If there is a need to link the aggregation with SOLAR VILLAGE for all time stamped occasions, then we must introduce the attribute type time_stamp in the relationship itself. Then, a new relation, INCENTIVIZED_BY, needs to be constructed using the village_id from the SOLAR VILLAGE entity type along with the key attributes from the two temporal entity types inside the abstraction.



Questions?

singaravel.s@northeastern.edu

tanubbodi.a@northeastern.edu