Energy Retailer Project

Technical Test

Business Context:

Our client is a retail electricity and power generation company based in Irving, Texas.
This company brings its products and services to market in 20 states and the District of Columbia, including six of the seven competitive wholesale markets in the U.S. and markets in Canada, as well.
The company is also the largest competitive power generator in the U.S. with a capacity of approximately 39,000 megawatts powered by a diverse portfolio, including natural gas, nuclear, solar, and battery energy storage facilities.

Technical Context:

Our client has provided you a sample copy of their database containing information of:

Markets: Governing entity that coordinates, controls, and monitors the operations of an electrical power system in which they and other competitors can retail electricity.

Units: In an industrial facility for the generation of electric power (Power plant) refers to a device that converts motive power (mechanical energy) into electric power. In the context of an energy storage facility refers to a battery.

Unit Types: The source of the energy generation for a unit. E.g.: Steam turbines, gas turbines, water turbines, internal combustion engines, wind turbines, etc.

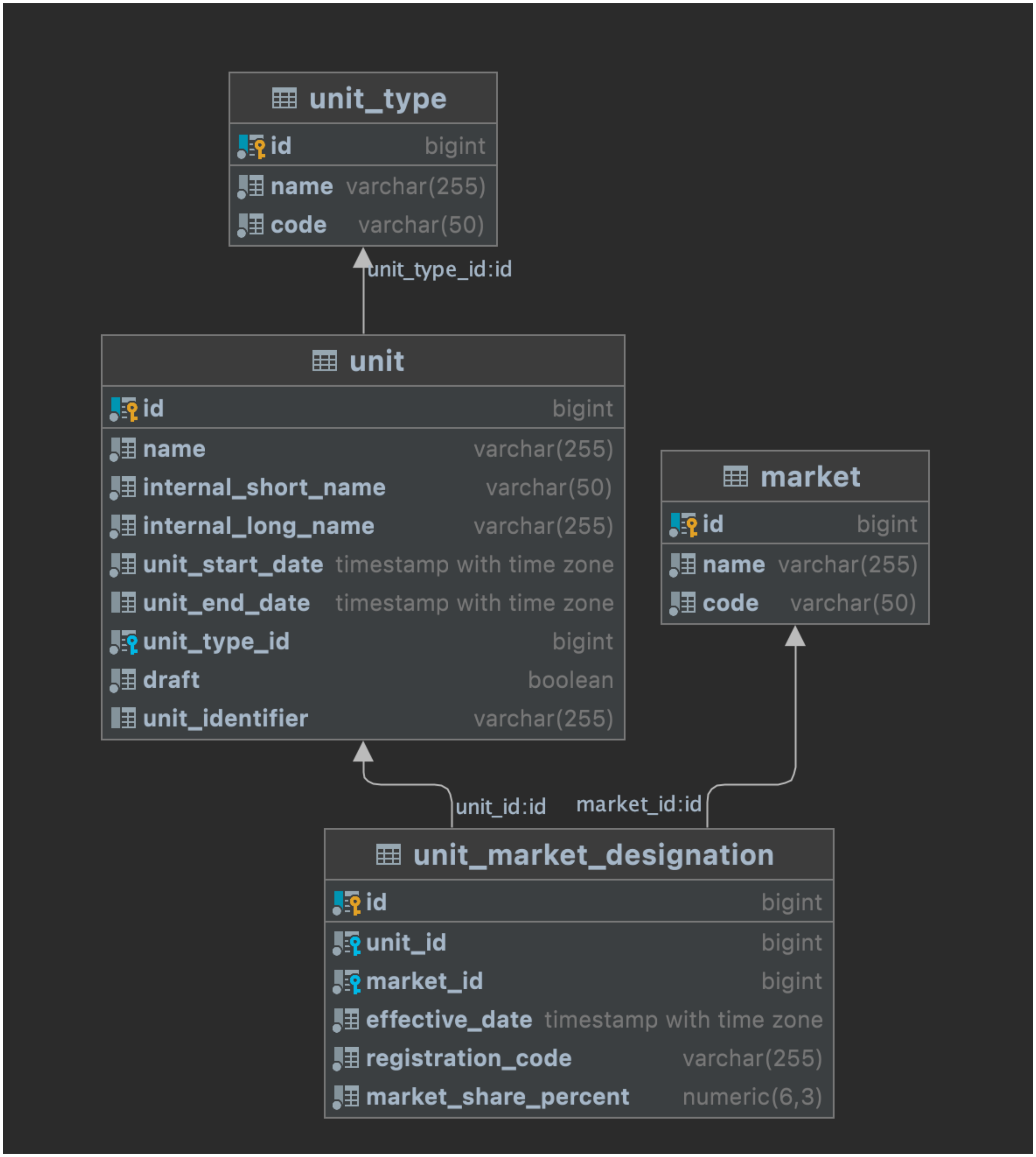
That sample copy can be found in the attached SQL file named "energy-retailer-technical-test.sql".

The company needs an application that, via a JSON API, allows them to:

- Manage unit market designations applying rigorous validations.
- List/Filter information for units according to different criteria.

Our team is in charge of the JSON API implementation.
And an external team in India will use such API to build a prototype UI using Angular JS.
Therefore, is critical that we meet all the requirements and match the specifications exactly to avoid future integration problems with that team.
Also is important to mention that the API is going to be deployed in servers protected by several layers of security only accessible to specific developers therefore **we don't need to implement any in app authentication / authorization.**

Database model:



unit_type

id	bigint
name	varchar(255)
code	varchar(50)

unit

id	bigint
name	varchar(255)
internal_short_name	varchar(50)
internal_long_name	varchar(255)
unit_start_date	timestamp with time zone
unit_end_date	timestamp with time zone
unit_type_id	bigint
draft	boolean
unit_identifier	varchar(255)

market

id	bigint
name	varchar(255)
code	varchar(50)

unit_market_designation

id	bigint
unit_id	bigint
market_id	bigint
effective_date	timestamp with time zone
registration_code	varchar(255)
market_share_percent	numeric(6,3)

Application Technical Requirements:

Create a Java 11 application named “Energy Retailer” implemented using:

- Maven (3.8.2+)
- Spring Boot (2.4.5+)
  - Spring Data JPA
  - Hibernate
  - JAX-RS
- PostgreSQL DB (11+)
- Liquibase (3.6.3+ This is desirable, but any other tool to apply database schema changes can be used)
- Git (latest)
  - Create a local repository and provide the project as zip for review

Unit Market Designation:

Context:

Unit market designation refers to a contract between a unit and 1 or more markets to retail a percent of the total power generated by the unit (market share percent) after certain date, also known as “effective date”. This contract doesn’t have an end date.
Instead, any contract created with a future effective date will be valid and will take effect after such date.
Per effective date, each market will keep a code referencing this contract known as “registration code”. This field is input by the user as it’s the number in an external system in the market.

E.g.: For the unit A, id = 1
On 2000-01-01 05:00:00 signed the contract to retail a 100% to the ERCOT market (id = 2) with the registration_code EC-A-12315

On 2010-09-04 00:00:00 signed the contract to retail an 80% to the ERCOT market (id = 2) with the registration_code EC-A-56789 and 20% to the CAISO market (id=1) with the registration_code CA-A-59383

On 2023-06-01 12:00:00 signed the contract to retail a 100% to the ERCOT market (id = 2) with the registration_code EC-A-87542

unit_id	market_id	effective_date	market_share_percent	registration_code
1	2	2000-01-01 05:00:00	100	EC-A-12315
1	2	2010-09-04 00:00:00	80	EC-A-56789
1	1	2010-09-04 00:00:00	20	CA-A-59383
1	2	2023-06-01 12:00:00	100	EC-A-87542

Technical Details:

POST /marketdesignations/{unitId}

This endpoint will set data records in the table master.unit_market_designation so all the data for the given unitId (path param) and effective date (body param) will math exactly the information in the marketDesignations field (body param). E.g.: The first request will insert records, every subsequent request will update or delete records.

Body

```
{
  "effectiveDate": {
    "date": "2000-01-01",
    "time": "05:00:00"
  },
  "marketDesignations": [
    {
      "registrationCode": "EC-A-12315",
      "marketShare": 50,
      "marketId": 2
    },
    {
      "registrationCode": "AC-A-14532",
      "marketShare": 50,
      "marketId": 1
    }
  ]
}
```

Response

204 on success

On error, describe the first exception or validation encounter:

```
{
  "errorMessage": "Unit not found"
}
```

Validations:

- UnitId must exist in the table master.unit
- The total marketShare for all the designations must be exactly 100
- All marketIds in the list must be different
- At least 1 market designation must be pass

- All fields are required
- Registration code is unique per unit and market

Unit List/Filter:

List units applying dynamic optional filters.

Technical Details:

This endpoint must be implemented using JPA Repository and JPA Specification / Criteria API. The results must be paginated in the database level (Not in memory)

GET /units

This endpoint returns records from the table master.unit applying filters on certain fields. Every null filter will be ignored. No duplicated records should be returned.

Query params

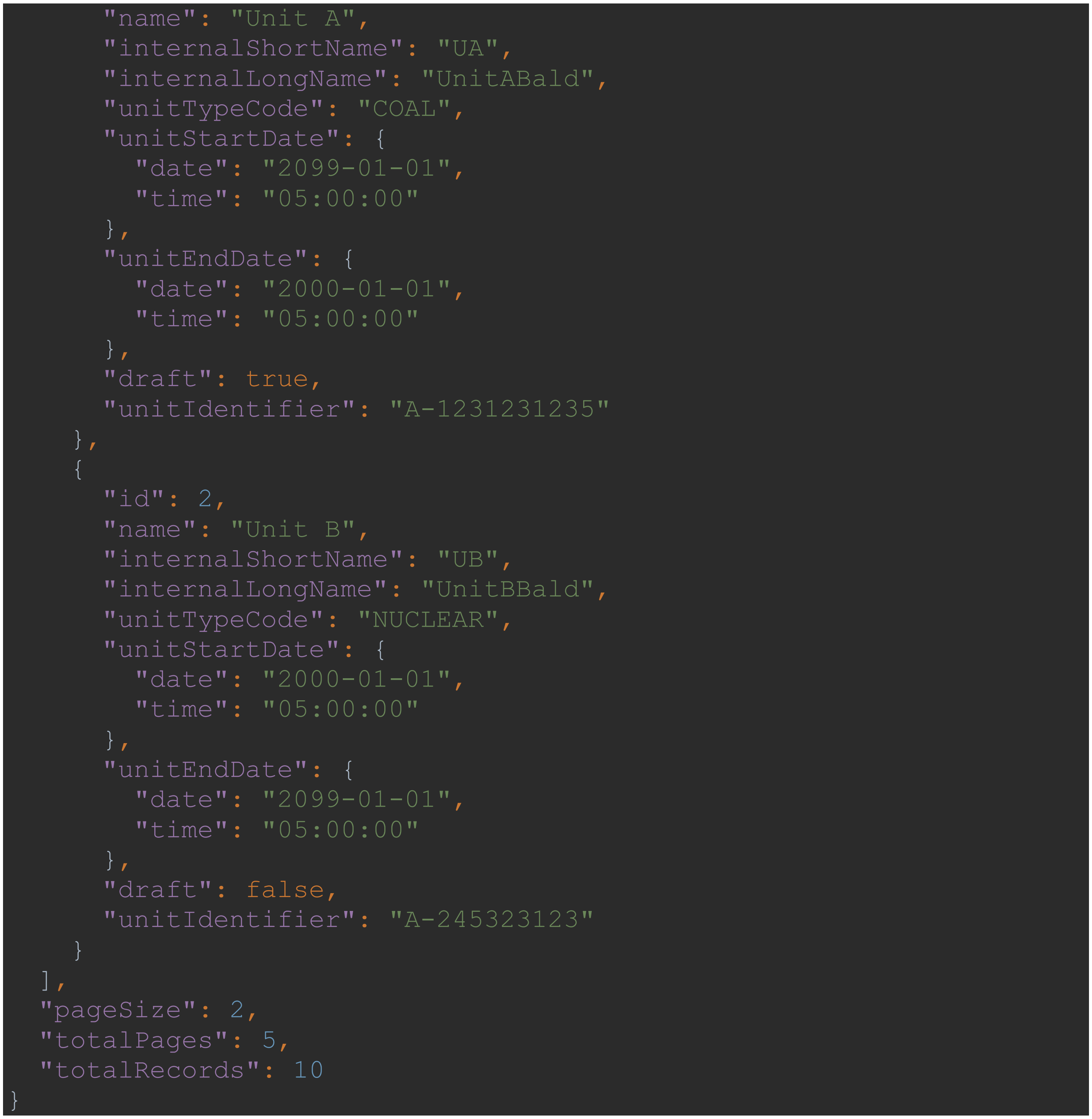
Name	type	Description
pageSize	Int	The number of records to return per page, default 10
name	String	Perform a LIKE check on the field unit.name
unitStartDate	String	String in the format yyyy-mm-dd to get unit whose unit_start_date is greater than or equal to this value
unitEndDate	String	String in the format yyyy-mm-dd to get unit whose unit_end_date is less than or equal to this value
unitTypeCode	String	Match exactly by master.unit_type.code using the fk field unit_type_id
draft	Boolean	Match exactly the field draft
unitIdentifier	String	Match exactly the field unit_identifier
marketCode	String	Match exactly by master.market.code using the middle table master.unit_market_designation but only consider records effective for the current date (now)

Body

N/A

Response

```
{
  "records": [
    {
      "id": 1,
```



```
      "name": "Unit A",
      "internalShortName": "UA",
      "internalLongName": "UnitABald",
      "unitTypeCode": "COAL",
      "unitStartDate": {
        "date": "2099-01-01",
        "time": "05:00:00"
      },
      "unitEndDate": {
        "date": "2000-01-01",
        "time": "05:00:00"
      },
      "draft": true,
      "unitIdentifier": "A-1231231235"
    },
    {
      "id": 2,
      "name": "Unit B",
      "internalShortName": "UB",
      "internalLongName": "UnitBBald",
      "unitTypeCode": "NUCLEAR",
      "unitStartDate": {
        "date": "2000-01-01",
        "time": "05:00:00"
      },
      "unitEndDate": {
        "date": "2099-01-01",
        "time": "05:00:00"
      },
      "draft": false,
      "unitIdentifier": "A-245323123"
    }
  ],
  "pageSize": 2,
  "totalPages": 5,
  "totalRecords": 10
}
```