DATA MANAGEMENT – ASSIGNMENT 1

MSC IN DATA SCIENCE – NATIONAL CENTRE FOR SCIENTIFIC RESEARCH “DEMOKRITOS”

1. DATABASE DESIGN CHOICES AND RATIONALE

The creation of the database should ensure the consistency of our data in a well structured database where information is not repeated so as to make easier the search of information in the database. For that reason I decided to split information that is repeated in our initial data to smaller tables in order to create a more flexible schema. Such information was:

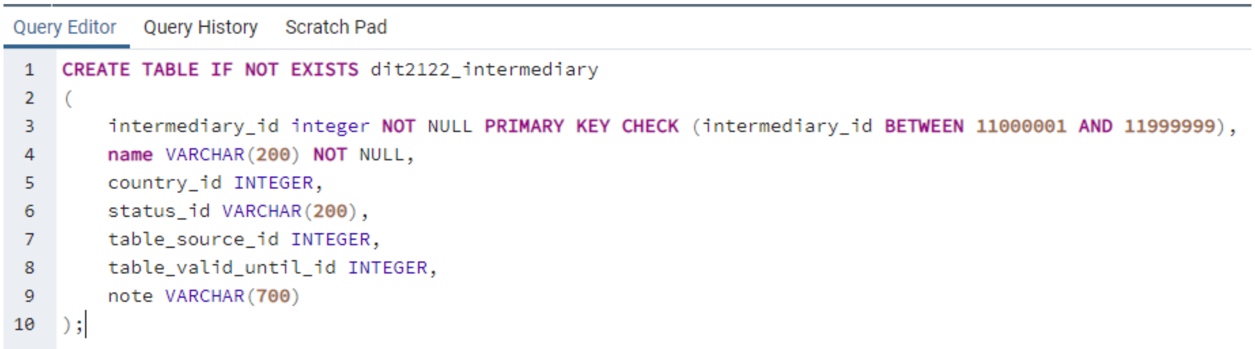
- countries
- country_codes
- jurisdiction
- jurisdiction_description
- sourceID
- valid_until
- service_provider
- status (in panama_papers.nodes.entity)
- status (in panama_papers.nodes.intermediary)

Such data are organized in distinct tables in which I have kept the unique id for every value of the above columns and we access to this information via the respective id column in the main tables*. To make it more clear the unique combination of the values of *countries* and *country_codes* have taken a unique id in the new table dit2122_countries and these two columns have been dropped and replaced with the new column country_id in the main tables. Moreover I created four more tables*[2] regarding the relations that were described in the panama_papers.edges file. In these tables I kept as foreign keys the primary keys of the tables with which are connected and as primary key the unique combination of the foreign keys.

* dit2122_address, dit2122_entity, dit2122_officer and dit2122_intermediary
*[2] dit2122_entity_address, dit2122_intermediary_entity, dit2122_officer_entity, dit2122_officer_address

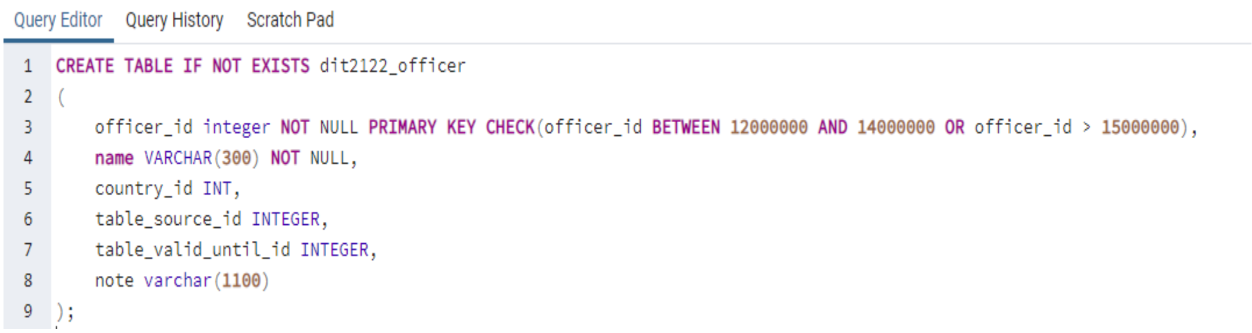
In the creation of the tables (dit2122_address, dit2122_entity, dit2122_officer and dit2122_intermediary) I used the CHECK command to reassure that the respective ids are unique and their values are between the numeric range that our initial data follow.

Example: dit2122_intermediary



```
CREATE TABLE IF NOT EXISTS dit2122_intermediary
(
    intermediary_id integer NOT NULL PRIMARY KEY CHECK (intermediary_id BETWEEN 11000001 AND 11999999),
    name VARCHAR(200) NOT NULL,
    country_id INTEGER,
    status_id VARCHAR(200),
    table_source_id INTEGER,
    table_valid_until_id INTEGER,
    note VARCHAR(700)
);
```

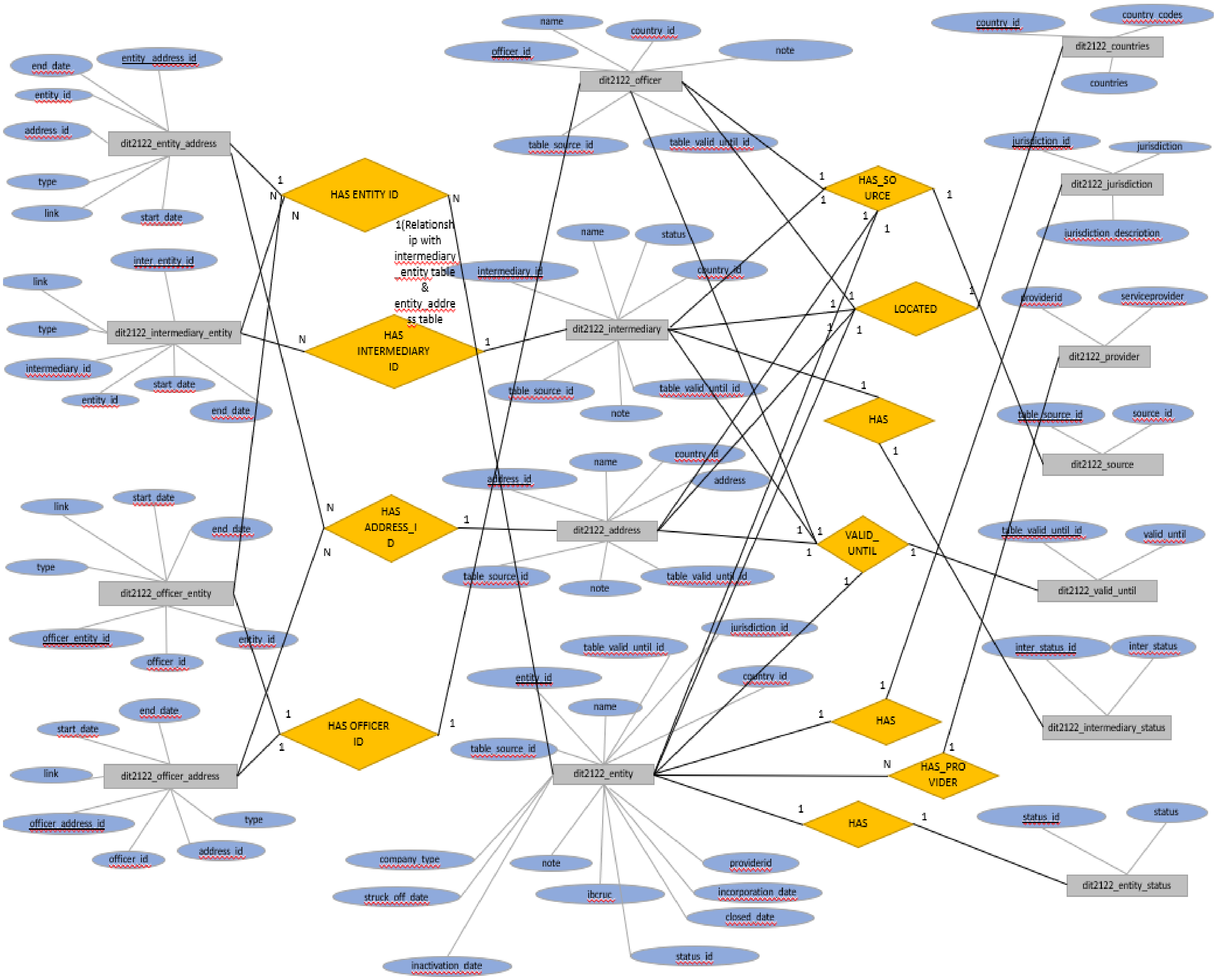
Example2: dit2122_officer



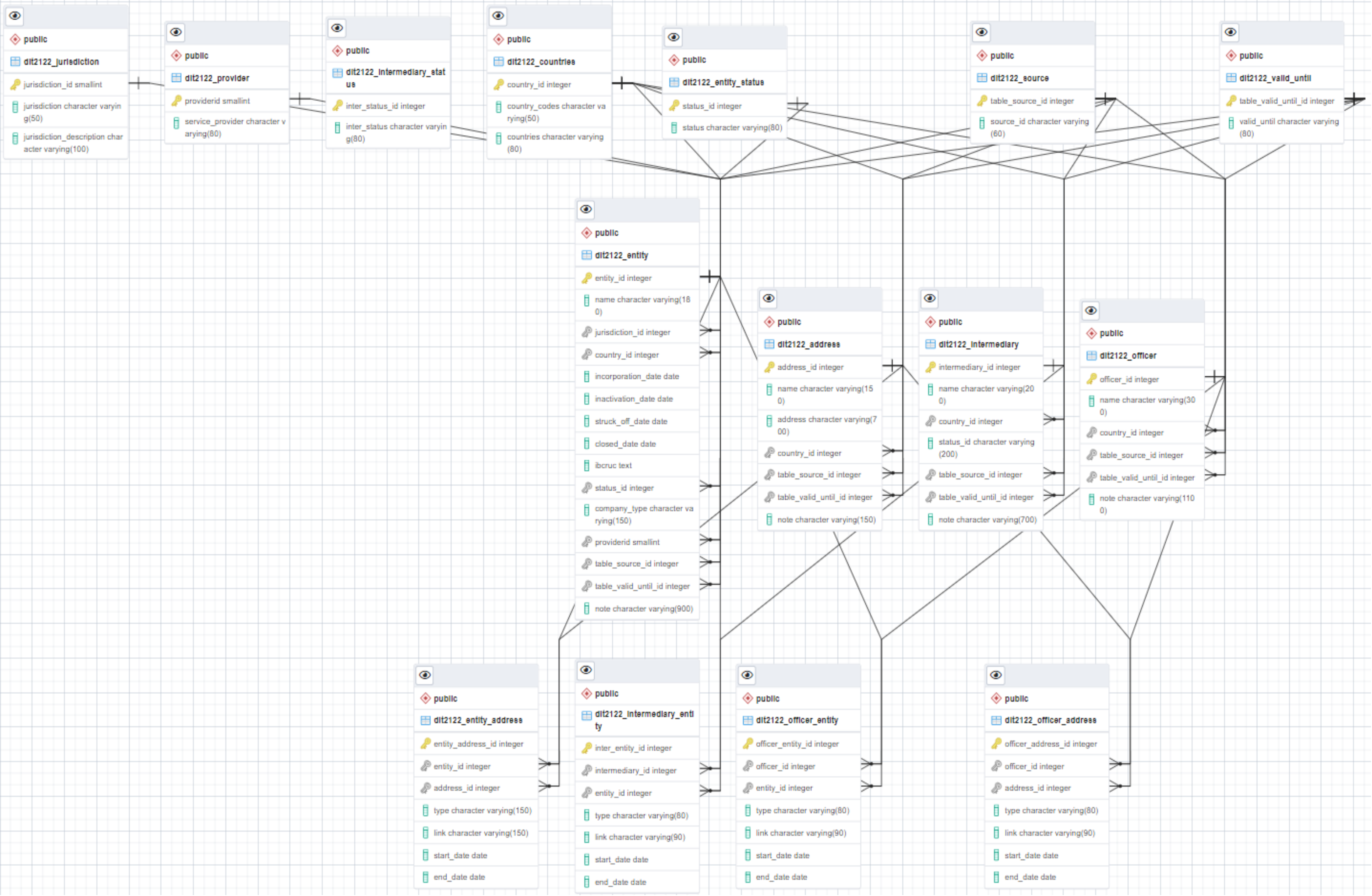
```
CREATE TABLE IF NOT EXISTS dit2122_officer
(
    officer_id integer NOT NULL PRIMARY KEY CHECK(officer_id BETWEEN 12000000 AND 14000000 OR officer_id > 15000000),
    name VARCHAR(300) NOT NULL,
    country_id INT,
    table_source_id INTEGER,
    table_valid_until_id INTEGER,
    note varchar(1100)
);
```

2. ER – MODEL

2.1 (Manually Created)



2.2 (Generated from PgAdmin)



3. RELATIONAL MODEL

dit2122_countries(PK country_id, country_codes, countries)

dit2122_jurisdiction(PK jurisdiction_id, jurisdiction, jurisdiction_description)

dit2122_intermediary_status(PK inter_status_id, inter_status)

dit2122_entity_status(PK status_id, status)

dit2122_source(PK table_source_id, source_id)

dit2122_valid_until(PK table_valid_until_id, valid_until)

dit2122_provider(PK providerid, service_provider)

dit2122_officer(PK officer_id, name, FK country_id, FK table_source_id, FK table_valid_until_id, note)

dit2122_entity(PK entity_id, name, FK jurisdiction_id, FK country_id, incorporation_date, inactivation_date)

dit2122_intermediary(PK intermediary_id, name, FK country_id, FK status_id, FK table_source_id, FK table_valid_until_id, note)

dit2122_address(PK address_id, name, address, FK country_id, FK table_source_id, FK table_valid_until_id, note)

dit2122_intermediary_entity(PK inter_entity_id, FK intermediary_id, FK entity_id, type, link, start_date, end_date)

dit2122_officer_entity(PK entity_id, name, FK jurisdiction_id, FK country_id, incorporation_date, inactivation_date, struck_off_date, closed_date, ibcruc, FK status_id, FK table_source_id, FK table_valid_until_id)

dit2122_officer_address(PK officer_address_id, FK officer_id, FK address_id, type, link, start_date, end_date)

dit2122_ entity _address(PK entity_address_id, FK entity_id, FK address_id, type, link, start_date, end_date)