ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Project Report

Panagiotis Katsos
Epaminondas Ioannou
Petros Tsotsi

2019-2020

1 Data Pre-processing

1.1 Creating the tables

Initially, we created the tables with the use of the gen_ ddl python script and we also defined the appropriate data types for each field. The sql scripts that were used can be found in partA.sql. For example, the following sql script was used to create the Links table:

```
CREATE TABLE 'Links' (
    movieId int,
    imdbId int,
    tmdbId int
);
```

1.2 Duplicates Deletion

After creating all tables, primary key constraints were added in each table. For these constraints to be added, we located the duplicates in every table (except Ratings_Small)and then proceeded to delete them. The process we followed for deleting duplicates was to find the tuples that exist more than once and delete the additional ones, so that they become unique. Duplicates were found in all tables. Specifically:

- 44 duplicates were detected and deleted in Credits table (45475 records before deletion, 45431 records after deletion)
- 987 duplicates were detected and deleted in Keywords table (46419 records before deletion, 45432 records after deletion)
- 30 duplicates were detected and deleted in Links table (45843 records before deletion, 45813 records after deletion)

- 30 duplicates were detected and deleted in Movies_Metadata table (45463 records before deletion, 45433 records after deletion)

For example, the following sql script was used to delete duplicates in Credits table:

```
DELETE FROM 'Credits' AS a USING (
    SELECT MIN(ctid) AS ctid, id
        FROM 'Credits'
        GROUP BY id HAVING COUNT(*) > 1
    ) AS b
    WHERE a.id = b.id
    AND a.ctid <> b.ctid;
```

All records that have *count* > 1 are deleted, until only 1 unique record remains, which is ensured by the second condition in the WHERE clause.

1.3 Movies Deletion

After deleting the duplicates we proceeded to delete movies which did not exist in the Movies_Metadata table but were present in one of the other tables. This was the case for Ratings_Small and Links table. Specifically:

- 55015 records were detected and deleted in Ratings_Small table (100004 records before deletion, 449898 after deletion)
- 350 records were detected and deleted in Links table (45813 records before deletion, 45433 records after deletion)

For example, the following sql script was used to delete movies in Links table:

```
DELETE FROM 'Links' WHERE movieid IN
    (SELECT movieid FROM 'Links'
    LEFT JOIN 'Movies“Metadata'
    ON 'Links'.tmdbid = 'Movies_Metadata'.id
    WHERE 'Movies_Metadata'.id is NULL );
```

The use of left join and this specific where clause delete the Links records with a movieid that does not exist in the Movies_Metadata table. In this case a left join was required because by using it we can retrieve the values that interest us and have NULL value in Movies_Metadata.

1.4 Additional Changes (json datatype conversion)

An additional change that was made to the data was a small edit in the genres field of the Movies_Metadata table. The single quote was replaced with a double quote. This field is required for partB queries and we had to convert it to json type in order to have easier access to the data.