ECE4721J - Lab 3 Report

Methods and Tools for Big Data

Kexuan Huang

June 6, 2022

3. Verifying the Data

The oldest movie

Query:

```
select primaryTitle,
    startYear
from title
where startYear <> "\N"
    and titleType = "movie"
order by startYear
limit 1;
```

Output:

```
primaryTitle  startYear
------------  ---------
Birmingham    1896
```

The longest movie in 2009

Query:

```
select primaryTitle,
    runtimeMinutes
from title
where startYear = "2009"
    and runtimeMinutes <> "\N"
    and titleType = "movie"
order by runtimeMinutes desc
limit 1;
```

Output:

```
primaryTitle       runtimeMinutes
-----------------  --------------
Native of Owhyhee  390
```

The year with the most movies

Query:

```
select startYear,
    count(*) as count
from title
where startYear <> "\N"
    and titleType = "movie"
group by startYear
order by count desc
limit 1;
```

Output:

```
startYear  count
---------  -----
2021       15898
```

The name of the person who contains in the most movies

Query:

```
select name.primaryName,
    count(*) as contained
from name,
    principal,
    title
where principal.tconst = title.tconst
    and principal.nconst = name.nconst
    and title.titleType = "movie"
group by principal.nconst
order by contained desc
limit 1;
```

Output:

```
primaryName  contained
-----------  ---------
Ilaiyaraaja  949
```

The principal crew of the movie with highest average ratings and more than 500 votes

Query:

```
select name.primaryName,
    principal.category
from name,
    principal
where name.nconst = principal.nconst
    and principal.tconst in (
        select rating.tconst
        from rating
        where rating.numVotes > 500
        order by rating.averageRating desc
        limit 1
    );
```

Output:

```
primaryName      category
---------------  --------
Melanie Zanetti  actress
David McCormack  actor
Joe Brumm        writer
David Barber     composer
```

The count of each `Pair<BirthYear, DeathYear>` of the people

Query:

```
select birthYear,
    deathYear,
    count(*) as count
from name
where birthYear <> "\N"
    and deathYear <> "\N"
group by birthYear,
    deathYear
order by count desc;
```

Output:

Too long, see `query.out`

4. Interaction with SQLite in Java / Python

Please refer to `insert.py`

5. Advanced Analysis with the new Tables

The top 3 most common professions among these people and also the average life span of these three professions

Query:

```
select profession,
    count(*) as count,
    avg(deathYear - birthYear) as avgLifeSpan
from name,
    name_profession
where name.nconst = name_profession.nconst
    and deathYear <> "\N"
    and birthYear <> "\N"
group by profession
order by count desc
limit 3;
```

Output:

```
profession  count   avgLifeSpan
----------  ------  ----------------
actor       126066  70.0966160582552
writer      64452   71.9769440824179
actress     55228   73.5529803722749
```

The top 3 most popular (received most votes) genres

Query:

```sql
select genre,
    sum(numVotes) as votes
from rating, title_genre
where rating.tconst = title_genre.tconst
group by genre
order by votes desc
limit 3;
```

Output:

```
genre   votes
------  ---------
Drama   532586552
Action  355071204
Comedy  326066948
```

The average time span (endYear - startYear) of the titles for each person

Query:

```sql
select name.primaryName,
    avg(title.endYear - title.startYear) as avgTimeSpan
from principal,
    name,
    title
where title.tconst = principal.tconst
    and name.nconst = principal.nconst
    and title.startYear <> "\N"
    and title.endYear <> "\N"
group by principal.nconst
order by avgTimeSpan desc;
```

Output:

Too long, see `query.out`