CSCI-620 Data Management with the IMDb Dataset

[Developing tools to interact with the IMDb dataset]

Aishwarya Rao
ar2711@rit.edu

Apurav Khare
ak2816@rit.edu

Martin Qian
jq3513@rit.edu

Prateek Kalasannavar
pk6685@rit.edu

ABSTRACT

This project aims to explore a dataset by understanding it, modeling it to a normalized relational schema so that it can be stored and retrieved from a relational database management system. The project also focuses on developing an interface that allows fast and easy access to the dataset by abstracting complex query scenarios, like search by specific parameters within and across tables, and aggregate queries.

1. PROJECT STATUS

As established in Phase 0 and Phase 1, the deliverable of Phase 2 in Data Management includes three things. One, a document specifying different query scenarios on the IMDb dataset, their equivalent queries and sample outputs. Two, a script that handles requests from the user interface and maps these requests to the relevant query and responds with appropriate results. Finally, a user interface that allows the user to pick scenarios, specify various filters and view the results in a readable format. The following sections cover the technology and methodology used to implement these tasks, query scenarios, and screenshots of the output.

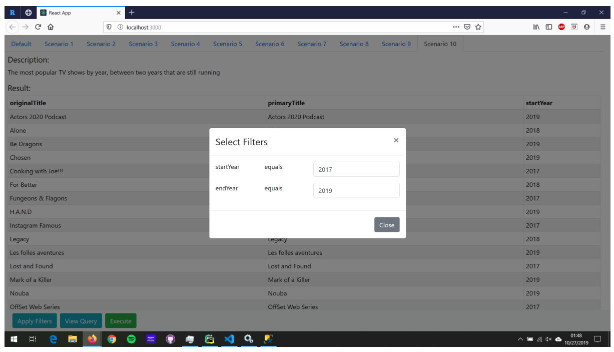
1.1 Technology Used

We used a web interface that allows users to interact with our database through a browser. The front end is built with ReactJS and BootStrap. It consists of tabs for every query scenario and allows entry of appropriate filters as seen in 11. On submitting a customized query, the request is handled by a Python script which uses the the user entered values to build a query and sends it to the database. It collects the response and sends it back to ReactJS for the user to view. The database is stored using Microsoft SQL Server 2017 and the queries are built accordingly.

1.2 Requirements and running the code

Our development phases uses the following technologies,

- Python 3.6 with Flask



Description:
The most popular TV shows by year, between two years that are still running

Result:

originalTitle	primaryTitle	startYear
Actors 2020 Podcast	Actors 2020 Podcast	2019
Alone		2018
Be Dragons		2019
Chosen		2019
Cooking with Joe!!!		2017
For Better		2018
Fungeons & Flagons		2017
H.A.N.D		2019
Instagram Famous		2017
Legacy	Legacy	2018
Les folles aventures	Les folles aventures	2019
Lost and Found	Lost and Found	2017
Mark of a Killer	Mark of a Killer	2019
Nouba	Nouba	2019
OffSet Web Series	OffSet Web Series	2017

startYear	equals	2017
endYear	equals	2019

Figure 1: Front-end - Applying applicable filters

- ReactJS
- Microsoft SQL Server 2017

Due to memory and storage constraints, the database size was cut down to only those entries after the year 2000. The related tables are adjusted accordingly.

1.3 Performance considerations

SQL Server creates clustered indexes for all columns in the primary key. In addition to that, we added non-clustered indexes on columns that were frequently used for look-up in the tables. These include the year and name columns in the title and name tables, and genre column in the titleGenres table. Indexes are added using the following scripts:

```sql
CREATE INDEX IX_title_name on title(originalTitle);
CREATE INDEX IX_title_year on title(startYear);
CREATE INDEX IX_name_name on name(primaryName);
CREATE INDEX IX_name_year on name(deathYear);
CREATE INDEX IX_genre on titleGenres(genre);
```

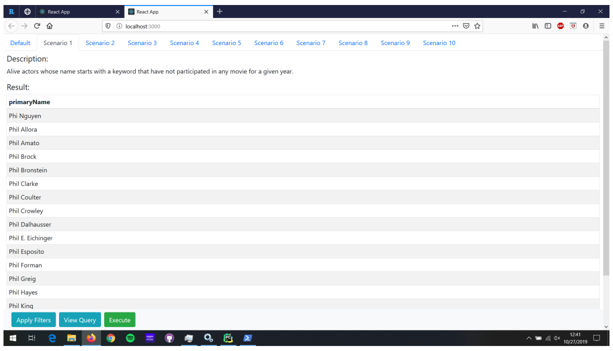
Overall, a 10-15% performance improvement was observed in the queries that filter on the non-text (deathYear and startYear) columns after applying the indexes.

1.4 Query Scenarios

1. List the names of alive actors whose name starts with a given keyword (such as “Phi”)and did not participate in any movie in a given year (such as 2014)

```sql
SELECT name.primaryName FROM name, movie,
    principalCast
```

```
WHERE name.primaryName = principalCast.primaryName
AND name.birthYear = principalCast.birthYear AND
     principalCast.category = 'actor'
AND movie.startYear NOT IN (SELECT startYear FROM
     movie WHERE startYear <> 2017)
AND name.deathYear IS NULL AND
     principalCast.primaryTitle =
     movie.primaryTitle
AND principalCast.originalTitle =
     movie.originalTitle AND
     principalCast.startYear = movie.startYear
AND name.primaryName LIKE 'phi';
```


Default | Scenario 1 | Scenario 2 | Scenario 3 | Scenario 4 | Scenario 5 | Scenario 6 | Scenario 7 | Scenario 8 | Scenario 9 | Scenario 10

Description:

Alive actors whose name starts with a keyword that have not participated in any movie for a given year.

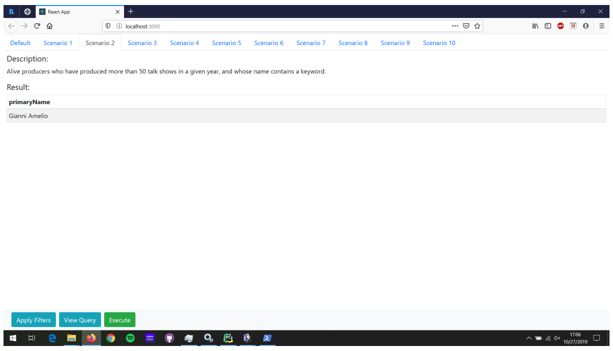
Result:

primaryName
Phi Nguyen
Phil Allora
Phil Amato
Phil Brock
Phil Bronstein
Phil Clarke
Phil Coulter
Phil Crowley
Phil Dalhausser
Phil E. Eichinger
Phil Esposito
Phil Forman
Phil Greig
Phil Hayes
Phil King

Figure 2: Scenario 1 Result

2. List the names of alive producers who have produced more than a given number (such as 50) of talk shows in a given year (such as 2017) and whose name contains a given keyword (such as "Gill")

```
SELECT n.primaryName
FROM name n
inner join principalCast pc on pc.primaryName =
     n.primaryName and pc.birthYear = n.birthYear
inner join titleGenres tg on tg.originalName =
     pc.originalName and tg.primaryTitle =
     pc.primaryTitle and tg.startYear =
     pc.startYear
where n.deathYear is null and tg.genre =
     'talk-show' and tg.startYear = 2017 and
     n.primaryName like 'gill%'
group by n.primaryName
having count(tg.originalTitle) >= 50;
```


Default | Scenario 1 | Scenario 2 | Scenario 3 | Scenario 4 | Scenario 5 | Scenario 6 | Scenario 7 | Scenario 8 | Scenario 9 | Scenario 10

Description:

Alive producers who have produced more than 50 talk shows in a given year, and whose name contains a keyword.

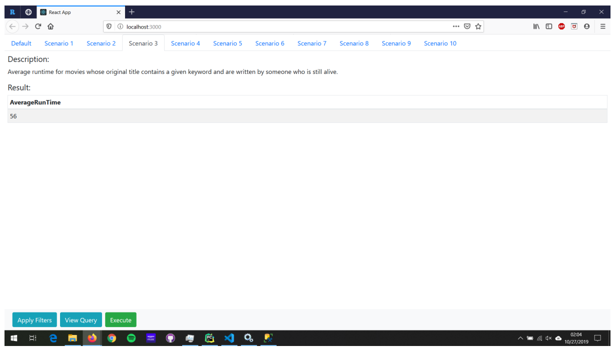
Result:

primaryName
Gianni Amelio

Figure 3: Scenario 2 Result

3. List the average runtime for movies whose original title contain a given keyword such as ("star")and were written by somebody who is still alive

```
SELECT top 100 avg(runtimeInMinutes) as
     AverageRunTime FROM movie, title,writers, name
WHERE movie.primaryTitle = title.primaryTitle
AND movie.originalTitle = title.originalTitle AND
     movie.startYear = title.startYear
AND writers.primaryTitle = title.primaryTitle AND
     writers.originalTitle = title.originalTitle
AND writers.startYear = title.startYear AND
     writers.primaryName = name.primaryName
AND writers.birthYear = name.birthYear AND
     name.deathYear IS NULL
AND movie.originalTitle LIKE '%gill%';
```


Default | Scenario 1 | Scenario 2 | Scenario 3 | Scenario 4 | Scenario 5 | Scenario 6 | Scenario 7 | Scenario 8 | Scenario 9 | Scenario 10

Description:

Average runtime for movies whose original title contains a given keyword and are written by someone who is still alive.

Result:

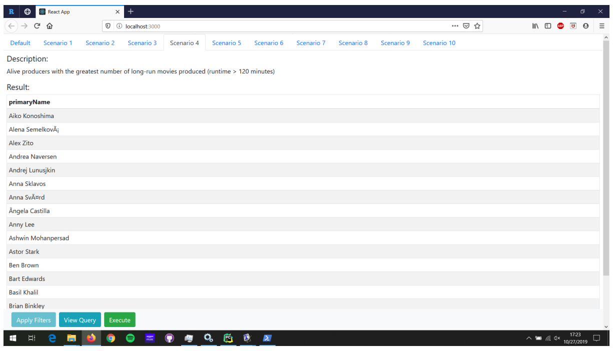
AverageRunTime
56

Figure 4: Scenario 3 Result

4. List the names of alive producers with the greatest number of long-run movies produced (runtime greater than 120min)

```
SELECT name.primaryName from name, movie, title,
     principalCast
WHERE principalCast.primaryName = name.primaryName
AND principalCast.birthYear = name.birthYear
AND principalCast.primaryTitle = title.primaryTitle
AND principalCast.originalTitle =
     title.originalTitle
AND principalCast.startYear = title.startYear
AND movie.primaryTitle = title.primaryTitle
AND movie.originalTitle = title.originalTitle
AND movie.startYear = title.startYear
AND title.runtimeInMinutes > 120
AND name.deathYear IS NULL
GROUP BY name.primaryName
HAVING count(movie.originalTitle) >=
(Select Max(totalCounts) from
(SELECT COUNT(movie.originalTitle) AS totalcounts
     FROM name n, movie m, title t, principalCast p
WHERE p.primaryName = n.primaryName
AND p.birthYear = n.birthYear
AND p.primaryTitle = t.primaryTitle
AND p.originalTitle = t.originalTitle
AND p.startYear = t.startYear
AND m.primaryTitle = t.primaryTitle
AND m.originalTitle = t.originalTitle
AND m.startYear = t.startYear
AND t.runtimeInMinutes > 120
AND n.deathYear IS NULL) tc)
```

5. List the unique name pairs of actors who have acted together in more than a given number (such as 2) movies and sort them by average movie rating (of those they acted together).



Description:

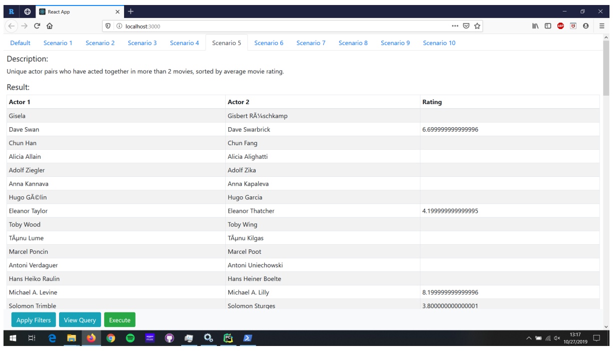
Alive producers with the greatest number of long-run movies produced (runtime > 120 minutes)

Result:

primaryName
Aiko Konoshima
Alena SemelkovÃ¡
Alex Zito
Andrea Naversen
Andrej Lunusjkin
Anna Sklavos
Anna SvÃ¤rd
Ãngela Castilla
Anny Lee
Ashwin Mohanpersad
Astor Stark
Ben Brown
Bart Edwards
Basil Khalil
Brian Binkley

Figure 5: Scenario 4 Result

```
SELECT Aname.primaryName, Bname.primaryName,
     AVG(averageRating)
FROM name Aname,name Bname, movie, title,
     principalCast Apc, principalCast Bpc
WHERE Aname.primaryName = Apc.primaryName
AND Aname.birthYear = Apc.birthYear
AND Bname.primaryName = Bpc.primaryName
AND Bname.birthYear = Bpc.birthYear
AND Aname.birthYear <> Bname.birthYear
AND Aname.primaryName <> Bname.primaryName
AND movie.originalTitle = title.originalTitle
AND movie.primaryTitle = title.primaryTitle
AND movie.startYear = title.startYear
AND Apc.originalTitle = title.originalTitle
AND Apc.primaryTitle = title.primaryTitle
AND Apc.startYear = title.startYear
AND Bpc.originalTitle = title.originalTitle
AND Bpc.primaryTitle = title.primaryTitle
AND Bpc.startYear = title.startYear
GROUP BY Aname.primaryName, Bname.primaryName
HAVING COUNT(movie.originalTitle)>=2;
```



Description:

Unique actor pairs who have acted together in more than 2 movies, sorted by average movie rating.

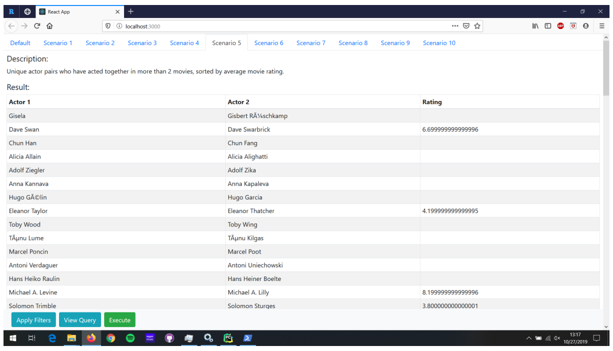
Result:

Actor 1	Actor 2	Rating
Gisela	Gisbert RÃ¼schkamp	
Dave Swan	Dave Swarbrick	6.6999999999999996
Chun Han	Chun Fang	
Alicia Allain	Alicia Alighatti	
Adolf Ziegler	Adolf Zika	
Anna Kannava	Anna Kapaleva	
Hugo GÃ©lin	Hugo Garcia	
Eleanor Taylor	Eleanor Thatcher	4.1999999999999995
Toby Wood	Toby Wing	
TÃµnu Lume	TÃµnu Kilgas	
Marcel Poncin	Marcel Poot	
Antoni Verdaguer	Antoni Uniechowski	
Hans Heiko Raulin	Hans Heiner Boelte	
Michael A. Levine	Michael A. Lilly	8.1999999999999996
Solomon Trimble	Solomon Sturges	3.800000000000001

Figure 6: Scenario 5 Result

6. List the actors that have worked in x movies (say 10) from one genre (say horror)

```
SELECT n.primaryName FROM name n
inner join principalCast pc on pc.primaryName =
     n.primaryName and pc.birthYear = n.birthYear
inner join movie m on m.originalTitle =
     pc.originalTitle and m.primaryTitle =
     pc.primaryTitle and m.startYear = pc.startYear
inner join titleGenres tg on tg.originalTitle =
     m.originalTitle and tg.primaryTitle =
     m.primaryTitle and tg.startYear = m.startYear
where tg.genre = 'horror' and m.movieType =
     'movie' and pc.category = 'actor'
group by n.primaryName, n.birthYear
having COUNT(CONCAT(pc.originalTitle,
     pc.primaryTitle, pc.startYear)) > 10;
```



Description:

Unique actor pairs who have acted together in more than 2 movies, sorted by average movie rating.

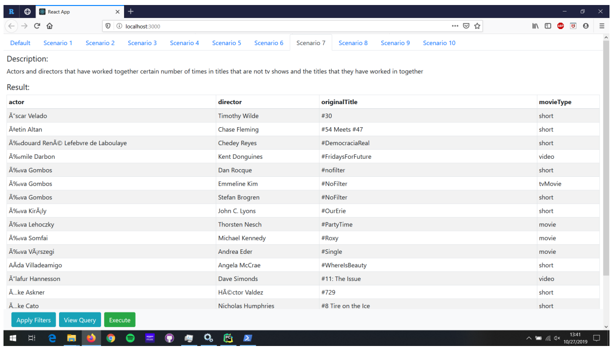
Result:

Actor 1	Actor 2	Rating
Gisela	Gisbert RÃ¼schkamp	
Dave Swan	Dave Swarbrick	6.6999999999999996
Chun Han	Chun Fang	
Alicia Allain	Alicia Alighatti	
Adolf Ziegler	Adolf Zika	
Anna Kannava	Anna Kapaleva	
Hugo GÃ©lin	Hugo Garcia	
Eleanor Taylor	Eleanor Thatcher	4.1999999999999995
Toby Wood	Toby Wing	
TÃµnu Lume	TÃµnu Kilgas	
Marcel Poncin	Marcel Poot	
Antoni Verdaguer	Antoni Uniechowski	
Hans Heiko Raulin	Hans Heiner Boelte	
Michael A. Levine	Michael A. Lilly	8.1999999999999996
Solomon Trimble	Solomon Sturges	3.800000000000001

Figure 7: Scenario 6 Result

7. Actors and directors that have worked together certain number of times in titles that are not tv shows and the titles that they have worked in together.

```
select n.primaryName as actor, d.primaryName as
     director, d.originalTitle, m.movieType
from name n
inner join principalCast pc on pc.primaryName =
     n.primaryName and pc.birthYear = n.birthYear
inner join directors d on d.originalTitle =
     pc.originalTitle and d.startYear =
     pc.startYear
inner join movie m on m.originalTitle =
     d.originalTitle and m.primaryTitle =
     d.primaryTitle and m.startYear = d.startYear
where pc.category = 'actor'
group by n.primaryName, n.birthYear,
     d.primaryName, d.birthYear
having COUNT(CONCAT(d.originalTitle,
     d.primaryTitle, d.startYear)) > 5;
```



Description:

Actors and directors that have worked together certain number of times in titles that are not tv shows and the titles that they have worked in together

Result:

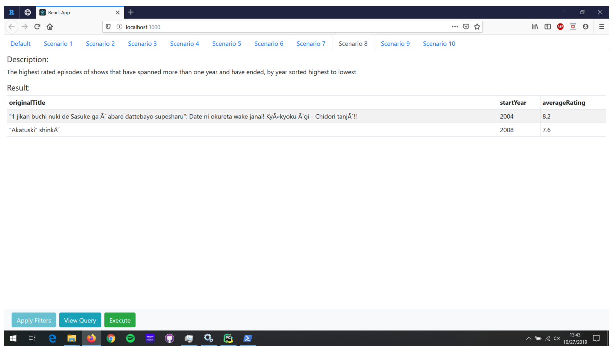
actor	director	originalTitle	movieType
Ã"scar Velado	Timothy Wilde	#30	short
Ã‡etin Altan	Chase Fleming	#54 Meets #47	short
Ã‰douard RenÃ© Lefebvre de Laboulaye	Chedey Reyes	#DemocraciaReal	short
Ã‰mile Darbon	Kent Donguines	#FridaysForFuture	video
Ã‰va Gombos	Dan Rocque	#nofilter	short
Ã‰va Gombos	Emmeline Kim	#NoFilter	tvMovie
Ã‰va Gombos	Stefan Brogren	#NoFilter	short
Ã‰va KirÃ¡ly	John C. Lyons	#OurErie	short
Ã‰va Lehoczky	Thorsten Nesch	#PartyTime	movie
Ã‰va Somfai	Michael Kennedy	#Roxy	movie
Ã‰va VÃ¡rszegi	Andrea Eder	#Single	movie
AÃda Villadeamigo	Angela McCrae	#WhereIsBeauty	short
Ã"lafur Hannesson	Dave Simonds	#11: The Issue	video
Ã…ke Askner	HÃ©ctor Valdez	#729	short
Ã…ke Cato	Nicholas Humphries	#8 Tire on the Ice	short

Figure 8: Scenario 7 Result

8. The highest rated episodes of shows that have spanned more than one year and have ended, by year sorted highest to lowest.

```
select top 100 e.originalTitle, e.startYear,
     t.averageRating
```

```
from tvEpisode e
inner join title t on e.originalTitle =
     t.originalTitle and e.primaryTitle =
     t.primaryTitle and e.startYear = t.startYear
inner join ( select s.originalTitle,
     s.primaryTitle, s.startYear,
     MAX(t.averageRating) as maxRating
from tvEpisode e
inner join title t on e.originalTitle =
     t.originalTitle and e.primaryTitle =
     t.primaryTitle and e.startYear = t.startYear
inner join tvSeries s on s.originalTitle =
     e.seriesOriginalTitle and s.primaryTitle =
     e.seriesPrimaryTitle and s.startYear =
     e.seriesStartYear
where s.endYear is not null and s.startYear <>
     s.endYear
group by s.originalTitle, s.primaryTitle,
     s.startYear ) r on r.originalTitle =
     e.seriesOriginalTitle and r.primaryTitle =
     e.seriesPrimaryTitle and r.startYear =
     e.seriesStartYear
where t.averageRating = r.maxRating;
```


Default Scenario 1 Scenario 2 Scenario 3 Scenario 4 Scenario 5 Scenario 6 Scenario 7 Scenario 8 Scenario 9 Scenario 10

Description:

The highest rated episodes of shows that have spanned more than one year and have ended, by year sorted highest to lowest

Result:

originalTitle	startYear	averageRating
"1 jikan buchi nuki de Sasuke ga Ã´ abare dattebayo supesharu": Date ni okureta wake janai! KyÅ»kyoku Ã´gi - Chidori tanjÃ´!!	2004	8.2
"Akatuski" shinkÃ´	2008	7.6

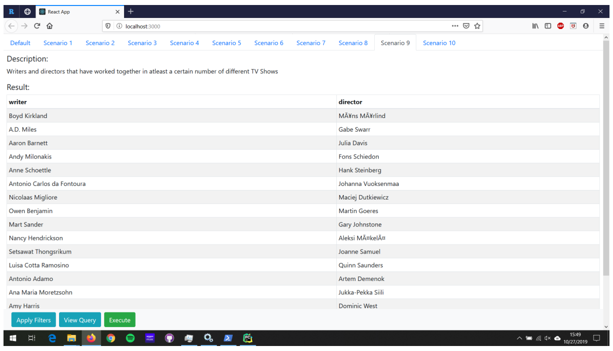
Figure 9: Scenario 8 Result

9. Writer, director that have worked together in atleast x different TV Shows

```
select top 100 w.primaryName as writer,
     d.primaryName as director
from writers w
inner join directors d on d.originalTitle =
     d.originalTitle and d.startYear = w.startYear
inner join tvSeries s on s.originalTitle =
     d.originalTitle and s.startYear = d.startYear
group by w.primaryName, w.birthYear,
     d.primaryName, d.birthYear
having COUNT(CONCAT(d.originalTitle,
     d.primaryTitle, d.startYear)) > 5;
```

10. List the most popular TV shows between the years x and y that are still running.

```
select top 100 s.originalTitle, s.primaryTitle,
     s.startYear
from tvSeries s
inner join title t on s.originalTitle =
     t.originalTitle and s.primaryTitle =
     t.primaryTitle and s.startYear = t.startYear
inner join ( select t.startYear,
     MAX(t.averageRating) as maxRating
from title t
```


Default Scenario 1 Scenario 2 Scenario 3 Scenario 4 Scenario 5 Scenario 6 Scenario 7 Scenario 8 Scenario 9 Scenario 10

Description:

Writers and directors that have worked together in atleast a certain number of different TV Shows

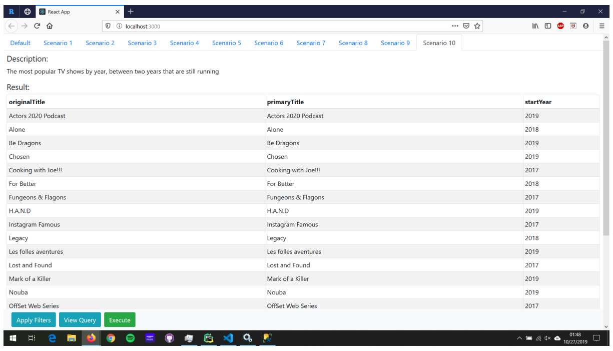
Result:

writer	director
Boyd Kirkland	MÃ¥ns MÃ¥rlind
A.D. Miles	Gabe Swarr
Aaron Barnett	Julia Davis
Andy Milonakis	Fons Schiedon
Anne Schoettle	Hank Steinberg
Antonio Carlos da Fontoura	Johanna Vuoksenmaa
Nicolaas Migliore	Maciej Dutkiewicz
Owen Benjamin	Martin Goeres
Mart Sander	Gary Johnstone
Nancy Hendrickson	Aleksi MÃ¤kelÃ¤
Setsawat Thongsrikum	Joanne Samuel
Luisa Cotta Ramosino	Quinn Saunders
Antonio Adamo	Artem Demenok
Ana Maria Moretzsohn	Jukka-Pekka Siili
Amy Harris	Dominic West

Figure 10: Scenario 9 Result

```
inner join tvSeries s on s.originalTitle =
     t.originalTitle and s.primaryTitle =
     t.primaryTitle and s.startYear = t.startYear
where s.endYear is not null and s.startYear
     between 2017 and 2019
group by t.startYear ) r on r.startYear =
     s.startYear
where t.averageRating =
     r.maxRating;'.format(startYear, endYear);
```

Result:


Default Scenario 1 Scenario 2 Scenario 3 Scenario 4 Scenario 5 Scenario 6 Scenario 7 Scenario 8 Scenario 9 Scenario 10

Description:

The most popular TV shows by year, between two years that are still running

Result:

originalTitle	primaryTitle	startYear
Actors 2020 Podcast	Actors 2020 Podcast	2019
Alone	Alone	2018
Be Dragons	Be Dragons	2019
Chosen	Chosen	2019
Cooking with Joe!!!	Cooking with Joe!!!	2017
For Better	For Better	2018
Fungeons & Flagons	Fungeons & Flagons	2017
H.A.N.D	H.A.N.D	2019
Instagram Famous	Instagram Famous	2017
Legacy	Legacy	2018
Les folles aventures	Les folles aventures	2019
Lost and Found	Lost and Found	2017
Mark of a Killer	Mark of a Killer	2019
Nouba	Nouba	2019
OffSet Web Series	OffSet Web Series	2017

Figure 11: Scenario 10 Result