CSE3207 Project #1 Report

Implementation of a simple database application

12191656 이채연

I. What you have implemented and what you have not

CSE3207 Database project #1 에서는 director, movie, actor에 대한 제공된Initial data를 기반으로 database를 구축하고 data를 테이블에 삽입하였다. 또한, 쿼리문들을 수행하면서 적절한 데이터를 집어넣었고, 특정 정보를 출력하였으며 특정 정보를 삭제하였다. 마지막에는 모든 table을 삭제하였다.
이번 프로젝트에서 요구한 9가지의 SQL문을 모두 수행하였다.

II. Brief explanation of your implementation (less than 1 page)

먼저 PostgreSQL jdbc driver를 등록한 뒤, Eclipse와 "project_movie" 라는 이름의 PostgreSQL server를 연동하였다.
데이터베이스에 table을 생성하기 전에 table이 있으면 해당 table을 drop하도록 하여 데이터베이스가 비어있도록 하였다.
그리고 Queries 1번에서 요구한대로 primary key와 foreign key를 고려하여 적절한 table들을 생성한 뒤, "Table created!" 문장을 출력하도록 하였다. 그리고 제공된 Initial Data를 기반으로 해당 table에 적절한 data를 insert해줬다. 이때 preparedstatement 객체를 이용하였다. 이 객체는 connection 객체의 prepareStatement(String query)를 통해 생성된다. 이 인자에 원하는 SQL문을 넣고 SQL문을 실행하기 위해 executeUpdate()함수를 호출하였다. Data를 insert할 때 preparedstatement 객체를 사용하여 SQL문에서 변수가 들어갈 자리는 '?'로 표시하였다.
그리고 Queries 2번과 3번에서 요구하는 알맞은 데이터를 삽입하기 위한 SQL문을 Statement 객체를 추가로 사용하여 실행하였고, 해당 table의 data 정보를 나타내기 위해 ResultSet 객체를 사용하였다.
Queries 4번, 5번, 6번에서 원하는 정보를 조합해서 출력하는 SQL문을 executeQuery() 함수를 호출하여 실행하고, 그 결과 테이블 정보를 출력하였다.
Queries 7번, 8번, 9번에서 요구하는 알맞은 데이터 혹은 테이블을 삭제하고 그 결과 테이블 정보를 출력하였다.

III. How to compile and run

Windows 환경에서 PostgreSQL과 JAVA eclipse를 연동하였고, 이때 JDBC driver를 Database connector로 이용하였다. PostgreSQL에 id는 postgres, password는 cse3207, port는 5432인 project_movie라는 database가 생성되어있다.

IV. Display result for all queries

Queries 1, Queries 2.1, Queries 2.2 Result

```
PostgreSQL JDBC Driver Registered!
org.postgresql.jdbc.PgConnection@470f1802
연결 성공
Table created!
Initial data inserted!

Statement : Winona Ryder won the “Best supporting actor” award in 1994
Translated SQL : select actorID from actor where actorName='Winona Ryder'
Translated SQL : insert into award values (1, 'Best supporting actor'
Translated SQL : select awardID from award where awardName='Best supporting actor
Translated SQL : insert into actorObtain values (2, 1, 1994)

-----<award>-----

                        awardID                    awardName
                              1         Best supporting actor

-----<actorObtain>-----

                        actorID                      awardID                         year
                              2                            1                         1994

Statement : Tom Hardy won the “Best supporting actor” award in 2018.
Translated SQL : select actorID from actor where actorName='Tom Hardy'
Translated SQL : select awardID from award where awardName='Best supporting actor
Translated SQL : insert into actorObtain values (9, 1, 2018)

-----<actorObtain>-----

                        actorID                      awardID                         year
                              2                            1                         1994
                              9                            1                         2018
```

Queries 2.3, Queries 2.4 Result

```
Statement : Heath Ledger won the “Best villain actor” award in 2009.
Translated SQL : select actorID from actor where actorName='Heath Ledger'
Translated SQL : insert into award values (2, 'Best villain actor')
Translated SQL : select awardID from award where awardName='Best villain actor'
Translated SQL : insert into actorObtain values (5, 2, 2009)

-----<award>-----

                      awardID                   awardName
                            1         Best supporting actor
                            2            Best villain actor

-----<actorObtain>-----

                      actorID                     awardID                        year
                            2                           1                        1994
                            9                           1                        2018
                            5                           2                        2009

Statement : Johnny Depp won the “Best main actor” award in 2011.
Translated SQL : select actorID from actor where actorName='Johnny Depp'
Translated SQL : insert into award values (3, 'Best main actor')
Translated SQL : select awardID from award where awardName='Best main actor'
Translated SQL : insert into actorObtain values (1, 3, 2011)

-----<award>-----

                      awardID                   awardName
                            1         Best supporting actor
                            2            Best villain actor
                            3               Best main actor

-----<actorObtain>-----

                      actorID                     awardID                        year
                            2                           1                        1994
                            9                           1                        2018
                            5                           2                        2009
                            1                           3                        2011
```

Queries 2.5, Queries 2.6 Result

```
Statement : Edward Scissorhands  won the “Best fantasy movie” award in 1991.
Translated SQL : select movieID from movie where movieName='Edward Scissorhands'
Translated SQL : insert into award values (4, 'Best fantasy movie')
Translated SQL : select awardID from award where awardName='Best fantasy movie'
Translated SQL : insert into movieObtain values (1, 4, 1991)

-----<award>-----

                      awardID                   awardName
                            1         Best supporting actor
                            2            Best villain actor
                            3               Best main actor
                            4            Best fantasy movie

-----<movieObtain>-----

                      movieID                     awardID                        year
                            1                           4                        1991

Statement : Alice In Wonderland won the “Best fantasy movie” award in 2011
Translated SQL : select movieID from movie where movieName='Alice In Wonderland'
Translated SQL : select awardID from award where awardName='Best fantasy movie'
Translated SQL : insert into movieObtain values (2, 4, 2011)

-----<movieObtain>-----

                      movieID                     awardID                        year
```

Queries 2.7 Result

```
Statement : The Dark Knight won the “Best picture” award in 2009
Translated SQL : select movieID from movie where movieName='The Dark Knight'
Translated SQL : insert into award values (5, 'Best picture')
Translated SQL : select awardID from award where awardName='Best picture'
Translated SQL : insert into movieObtain values (4, 5, 2009)

-----<award>-----

                      awardID                   awardName
                            1         Best supporting actor
                            2            Best villain actor
                            3               Best main actor
                            4            Best fantasy movie
                            5                  Best picture

-----<movieObtain>-----

                      movieID                     awardID                        year
                            1                           4                        1991
                            2                           4                        2011
                            4                           5                        2009
```

Queries 2.8, Queries 3.1 Result

```
Statement : Christopher Nolan won the “Best director” award in 2018
Translated SQL : select directorID from director where directorName='Christopher Nolan'
Translated SQL : insert into award values (6, 'Best director')
Translated SQL : select awardID from award where awardName='Best director'
Translated SQL : insert into directorObtain values (3, 6, 2018)

-----<award>-----

                      awardID                   awardName
                            1         Best supporting actor
                            2            Best villain actor
                            3               Best main actor
                            4            Best fantasy movie
                            5                  Best picture
                            6                Best director

-----<directorObtain>-----

                   directorID                     awardID                        year
                            3                           6                        2018

Statement : Ethan rates 5 to “Dunkirk”.
Translated SQL : select customerID from customer where customerName='Ethan'
Translated SQL : select movieID from movie where movieName='Dunkirk'
Translated SQL : insert into customerRate values (1, 5, 5)
Translated SQL : update movie set avgRate = 5 where movieID= 5

-----<customerRate>-----

                   customerID                     movieID                         rate
                            1                           5                     5.000000

-----<movie>-----

                      movieID                   movieName                      avgRate
                            1           Edward Scissorhands                     0.000000
                            2           Alice In Wonderland                     0.000000
                            3            The Social Network                     0.000000
                            4               The Dark Knight                     0.000000
                            5                       Dunkirk                     5.000000
```

Queries 3.2 Result

```
Statement : Bell rates 5 to the movies whose director is “Tim Burton”.
Translated SQL : select customerID from customer where customerName='Bell'
Translated SQL : select directorID from director where directorName='Tim Burton'
Translated SQL : select movieID from make where directorID = 1
Translated SQL : insert into customerRate values (5, 1, 5)
Translated SQL : insert into customerRate values (5, 2, 5)
Translated SQL : update movie set avgRate = 5 where movieID= 1
Translated SQL : update movie set avgRate = 5 where movieID= 2

-----<customerRate>-----
```

customerID	movieID	rate
1	5	5.00
5	1	5.00
5	2	5.00

```
-----<movie>-----
```

movieID	movieName	avgRate
1	Edward Scissorhands	5.00
2	Alice In Wonderland	5.00
3	The Social Network	0.00
4	The Dark Knight	0.00
5	Dunkirk	5.00

Queries 3.3 Result

```
Statement : Jill rates 4 to the movies whose main actor is female.
Translated SQL : select customerID from customer where customerName='Jill'
Translated SQL : select actorID from actor where gender='Female'
Translated SQL : select movieID from casting where (actorID = 2 or actorID = 3) and role = 'Main actor'
Translated SQL : insert into customerRate values (4, 1, 4)
Translated SQL : insert into customerRate values (4, 2, 4)
Translated SQL : update movie set avgRate = 4.5 where movieID= 1
Translated SQL : update movie set avgRate = 4.5 where movieID= 2

-----<customerRate>-----
```

customerID	movieID	rate
1	5	5.00
5	1	5.00
5	2	5.00
4	1	4.00
4	2	4.00

```
-----<movie>-----
```

movieID	movieName	avgRate
1	Edward Scissorhands	4.50
2	Alice In Wonderland	4.50
3	The Social Network	0.00
4	The Dark Knight	0.00
5	Dunkirk	5.00

Queries 3.4 Result

```
Statement : Hayden rates 4 to the fantasy movies.
Translated SQL : select customerID from customer where customerName='Hayden'
Translated SQL : select movieID from genreName where ='fantasy'
Translated SQL : select directorID from directorObtain where awardID = 6
Translated SQL : select movieID from make where directorID = 3
Translated SQL : insert into customerRate values (3, 1, 4)
Translated SQL : insert into customerRate values (3, 2, 4)
Translated SQL : update movie set avgRate = 4.25 where movieID= 1
Translated SQL : update movie set avgRate = 4.25 where movieID= 2

-----<customerRate>-----
```

customerID	movieID	rate
1	5	5.00
5	1	5.00
5	2	5.00
4	1	4.00
4	2	4.00
3	1	4.00
3	2	4.00

-----<movie>-----

movieID	movieName	avgRate
1	Edward Scissorhands	4.25
2	Alice In Wonderland	4.25
3	The Social Network	0.00
4	The Dark Knight	0.00
5	Dunkirk	5.00

Queries 3.5 Result

```
Statement : John rates 5 to the movies whose director won the “Best director” award.
Translated SQL : select customerID from customer where customerName='John'
Translated SQL : select awardID from award where awardName ='Best director'
Translated SQL : select directorID from directorObtain where awardID = 6
Translated SQL : select movieID from make where directorID = 3
Translated SQL : insert into customerRate values (2, 4, 5)
Translated SQL : insert into customerRate values (2, 5, 5)
Translated SQL : update movie set avgRate = 5 where movieID= 4

-----<customerRate>-----
```

customerID	movieID	rate
1	5	5.00
5	1	5.00
5	2	5.00
4	1	4.00
4	2	4.00
3	1	4.00
3	2	4.00
2	4	5.00
2	5	5.00

```
-----<movie>-----
```

movieID	movieName	avgRate
1	Edward Scissorhands	4.25
2	Alice In Wonderland	4.25
3	The Social Network	0.00
4	The Dark Knight	5.00
5	Dunkirk	5.00

Queries 4, Queries 5, Queries 6 Result

```
Statement : Select the names of the movies whose actor are dead.
Translated SQL : select movieName from movie where movieID in (select movieID from casting natural join actor where dateOfDeath is not null)

-----<Query 4 result table>-----

      movieName
The Dark Knight

Statement : Select the names of the directors who cast the same actor more than once.
Translated SQL : select directorName from director where directorID in (select directorID from casting natural join make as S
where (actorID, directorID) = some(select actorID, directorID from casting natural join make as T where T.movieID <> S.movieID))

-----<Query 5 result table>-----

  directorName
    Tim Burton

Statement : Select the names of the movies and the genres, where movies have the common genre.
Translated SQL : select movieName, genreName from movie natural join movieGenre as S
where movieName = some(select movieName from movie natural join movieGenre as T where S.genreName <> T.genreName)

-----<Query 6 result table>-----

      movieName      genreName
Edward Scissorhands    Fantasy
Edward Scissorhands    Romance
Alice In Wonderland    Fantasy
Alice In Wonderland  Adventure
Alice In Wonderland     Family
    The Dark Knight     Action
    The Dark Knight      Drama
    The Dark Knight    Mystery
    The Dark Knight   Thriller
            Dunkirk     Action
            Dunkirk      Drama
            Dunkirk   Thriller
            Dunkirk        War
```

Queries 7, Queries 8, Queries 9 Result

```
Statement : Delete the movies whose director or actor did not get any award and delete data from related tables.
Translated SQL : delete from movie where movieID in(select distinct movieID from (make natural join casting natural left outer join directorObtain)
left outer join actorObtain using(actorID) where directorObtain.awardID is null and actorObtain.awardID is null)

-----<movie>-----

movieID  movieName            releaseYear  releaseMonth  releaseDate  publisherName
1        Edward Scissorhands  1991         06            29           20th Century Fox Presents
4        The Dark Knight      2008         08            06           Warner Brothers Korea
5        Dunkirk              17           07            12           Warner Brothers Korea

-----<movieGenre>-----

movieID  genreName
1        Fantasy
1        Romance
4        Action
4        Drama
4        Mystery
4        Thriller
5        Action
5        Drama
5        Thriller
5        War

-----<movieObtain>-----

movieID  awardID  year
1        4        1991
4        5        2009

-----<casting>-----

movieID  actorID  role
1        1        Main actor
1        2        Main actor
4        4        Main actor
4        5        Main actor
5        8        Main actor
5        9        Supporting Actor

-----<make>-----

movieID  directorID
1        1
4        3
5        3

-----<customerRate>-----

customerID  movieID  rate
5           1        5.000
4           1        4.000
3           1        4.000
2           4        5.000
1           5        5.000
2           5        5.000

Statement : Delete all customers and delete data from related tables.
Translated SQL : delete from customer

-----<customer>-----

customerID  customerName  dateOfBirth  gender

-----<customerRate>-----

customerID  movieID  rate

Statement : Delete all tables and data (make the database empty).
Translated SQL : drop table customerRate, customer, movieGenre, movieObtain, actorObtain, directorObtain, casting, make, director, actor, movie, award, genre

-----<table 수>-----

Number of Table
0
```

V. Talk about your experience of doing this project.

이번 프로젝트를 진행하면서 직접 데이터베이스를 만들어 보고, 데이터를 삽입하고 출력하고 삭제하는 SQL문을 직접 작성하는 것이 뜻 깊은 경험이 되었다. 특히, SQL문을 어떻게 작성할지 생각하는 과정을 거치면서 Database 수업을 복습할 수 있었고 google로 다양한 SQL문을 찾아보면서 생각의 폭을 넓힐 수 있었다.

VI. Write your available contact information such as email address (just in case)

Email address : chae0382@gmail.com

Phone number : 01056640382