-What you have implemented and what you have not

주어진 Statement를 다 해결했습니다.

-Brief explanation of your implementation

**Q1.Create the tables and insert the proper data based on the provided data. You should make movie, actor, director, and customer tables first and insert data into other related tables.**

create table문을 사용하여서 table을 만들었습니다. Initial data를 삽입할 때, reference되는 key를 미리 primary key로 지정을 해놓고, reference할 때 모든 foreign key는 “on delete cascade” 속성으로 지정하여, delete를 할 때 reference하고있는 키 값이 삭제되면 함께 삭제되도록 했습니다.

**Q2.Insert the proper data from the following statements.**

actor가 award를 받았을땐 actorObtain과 award, movie가 award를 받았을땐 award와 movieObtain, director가 award를 받았을땐 directorObtain과 award의 value값을 새로 insert해주었다.

Ex2.5)stmt.executeUpdate("insert into award values(4,'Best fantasy movie')");

stmt.executeUpdate("insert into movieObtain values(1,4,1991)");

**Q3. Insert data to the proper tables based on the following statements and update avgRate if necessary.**

각 statement가 원하는 조건의 data가 있는 table을 찾아서, movieID를 알아낼 수 없는 table은 movieID를 알아 낼 수 있는 table과 natural join하여 movieID를 알아 내 준 후, customerRate table에 data들을 insert해주었다.그 후, customerRate에 있는 data값들 중 movie Id가 같은 값들끼리 묶어 주어서, 평균값을 구하였다.

EX3-5))stmt.executeUpdate("insert into customerRate select 2 as customerID, movieID, 5.0 as rate from make natural join directorObtain natural join award where awardName ='Best Director'");

stmt.executeUpdate("update movie M set avgRate =(select avg(rate) from customerRate where M.movieID = customerRate.movieID)");

**Q4. Select the names of the movies whose actor are dead**

actor table에서 dateofDeath가 NULL이 아닌 data를 select하여서,casting table과 actorID를 natural join하여 movieID를 구해낸 후, movie ID에 맞는 movieName을 출력하였다.

**Q5. Select the names of the directors who cast the same actor more than once.**

stmt.executeQuery("select directorName from director natural join casting natural join make group by actorID,directorID,directorName having count(\*) > 1 ");

director가 actorID를 2번이상 사용하였을 때 directName을 select하도록 하였다.

**Q6. Select the names of the movies and the genres, where movies have the common genre**

Genrename을 groupby문을 이용하여, count가 1을 넘을 때 movieName을 출력하도록 하였다.

select genreName,count(\*) from movieGenre group by genreName

**Q7. Delete the movies which did not get any award(including all directors and actors) and delete data from related tables**

stmt.executeUpdate("delete from movie where movieID not in (select movieID from actorObtain natural join casting) and movieID not in(select movieID from directorObtain natural join make)");

**Q8. Delete all customers and delete data from related table.**

stmt.executeUpdate("delete from customer"); 를 해주면 reference된 key들도 같이 삭제됨

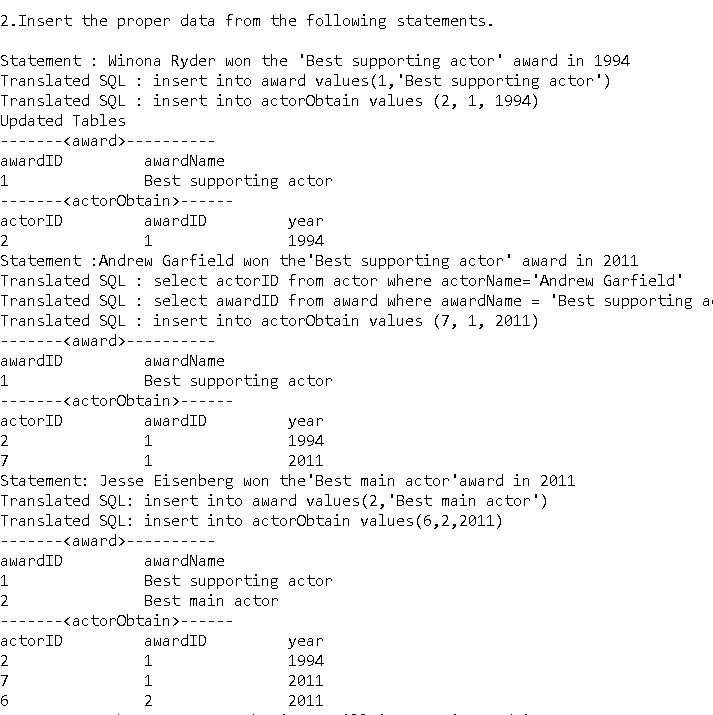
**Q9. Delete all tables and data. -**reference된 key들을 고려하여 drop문을 써주었음.

-How to compile and run

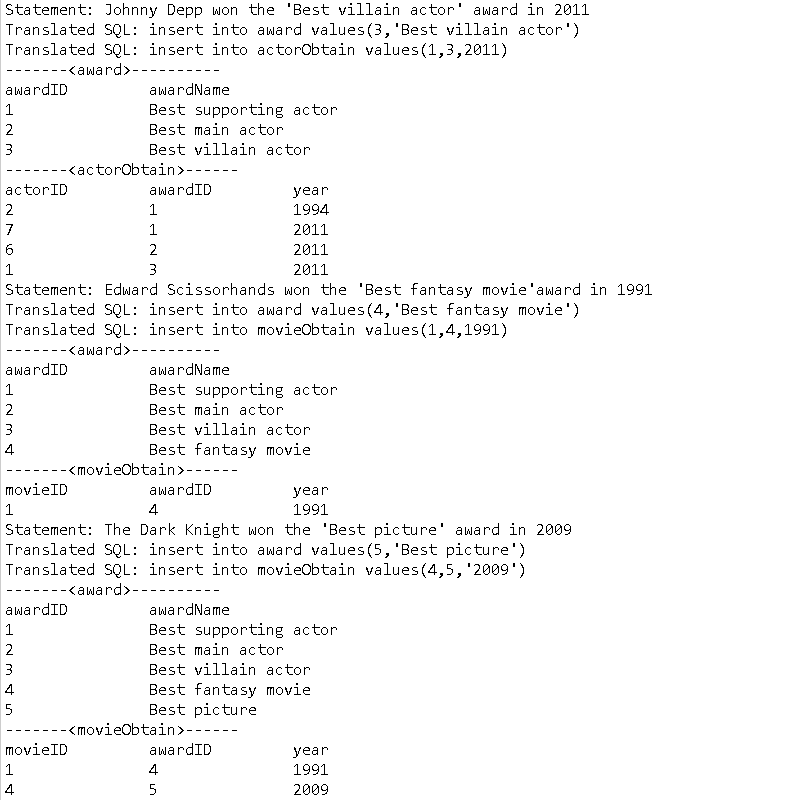
eclipse에서 Jdbc를 이용하여 쿼리문을 코딩하고, compile시켜줬습니다.

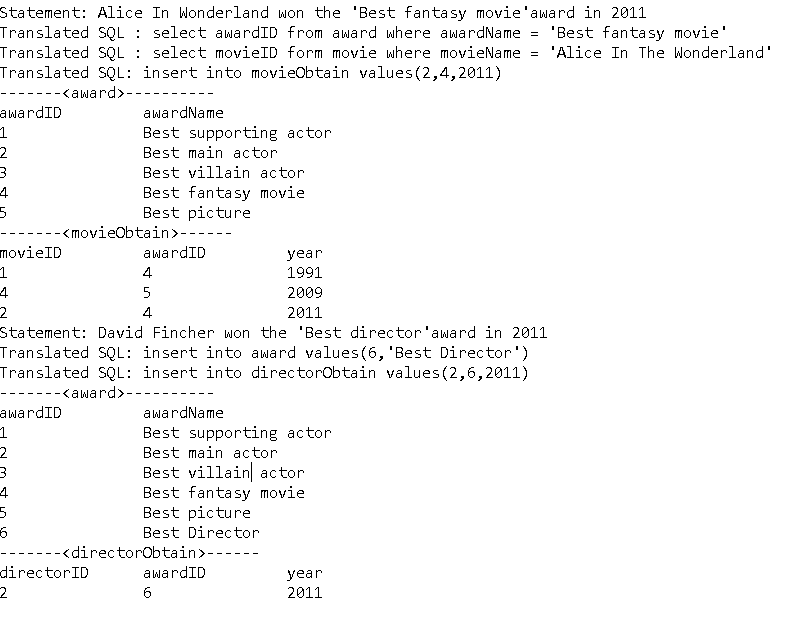
-Display result for all queries

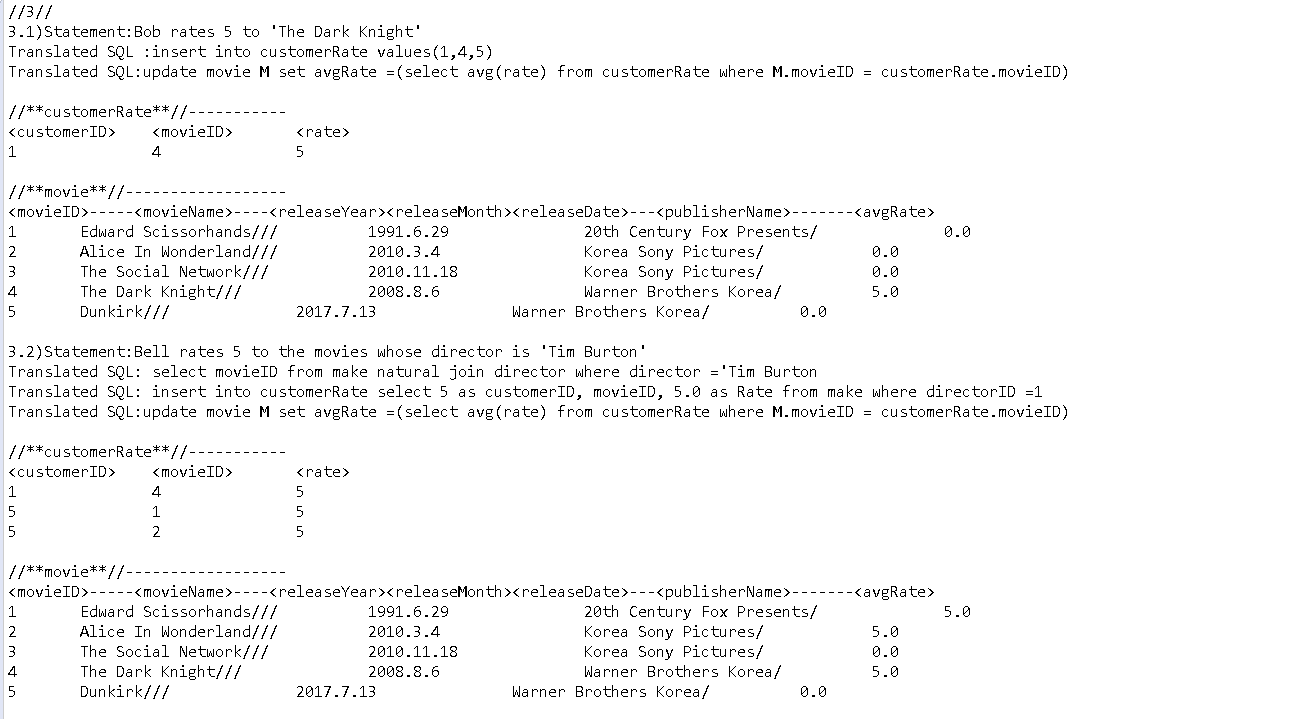
2.->2.1~2.3

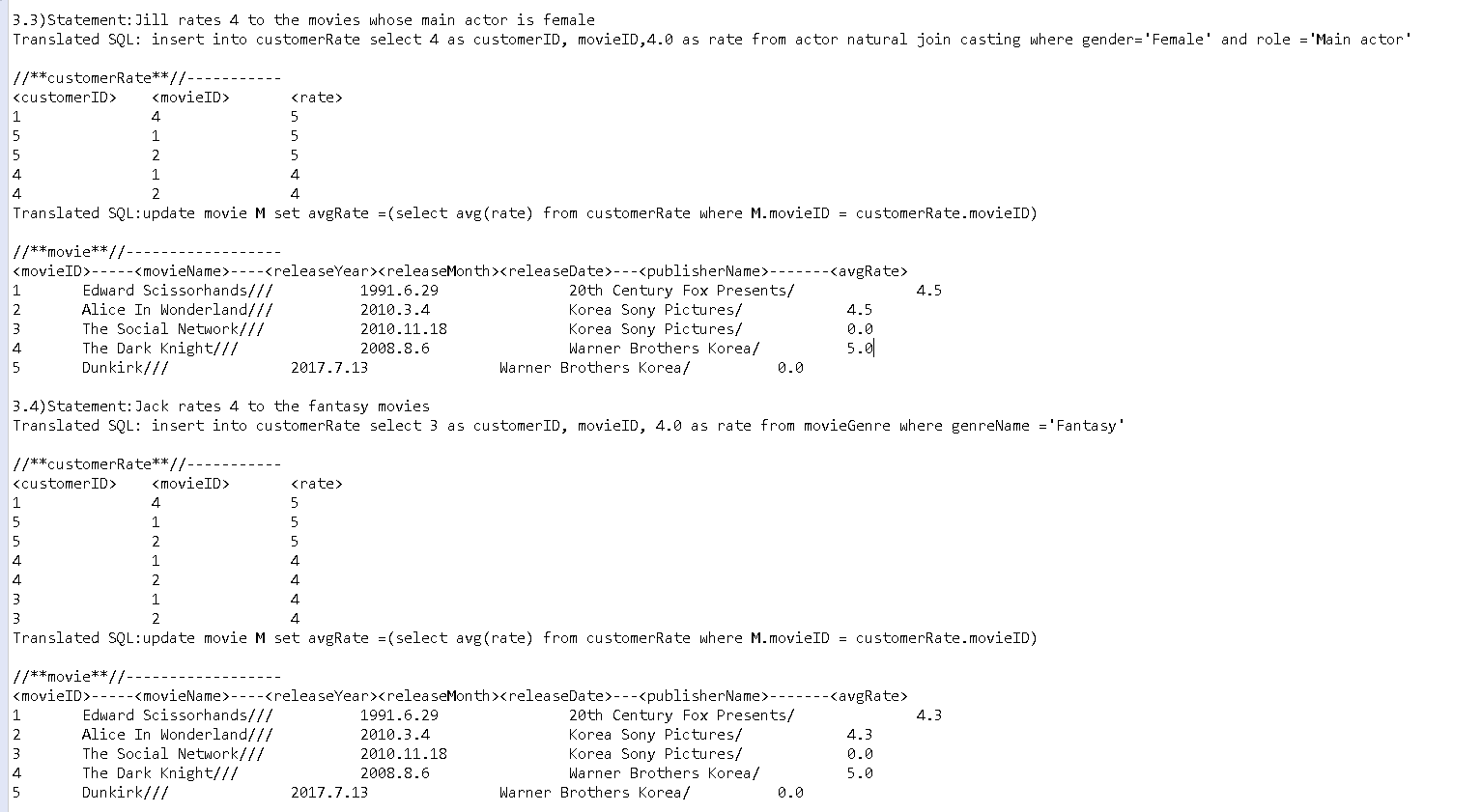


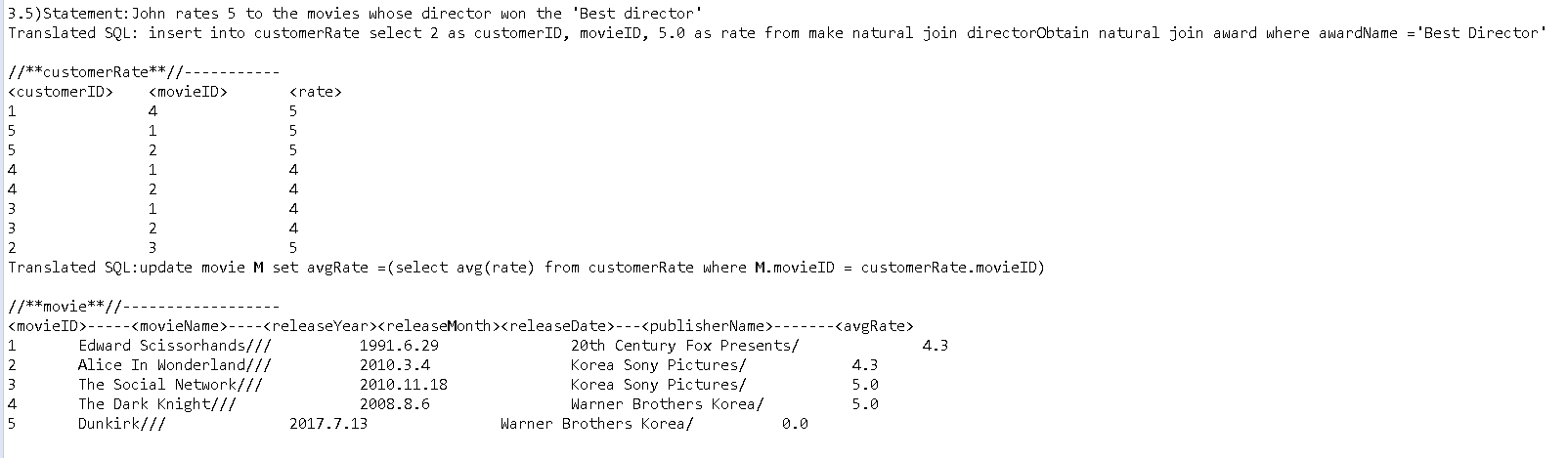
2.4-2.8

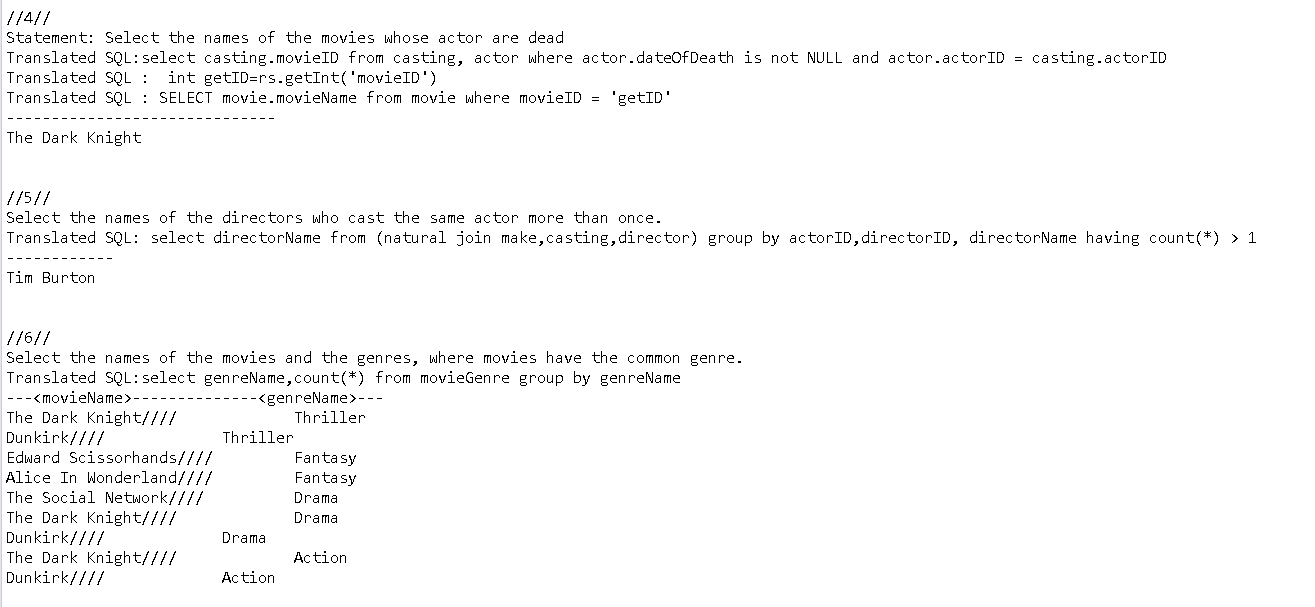


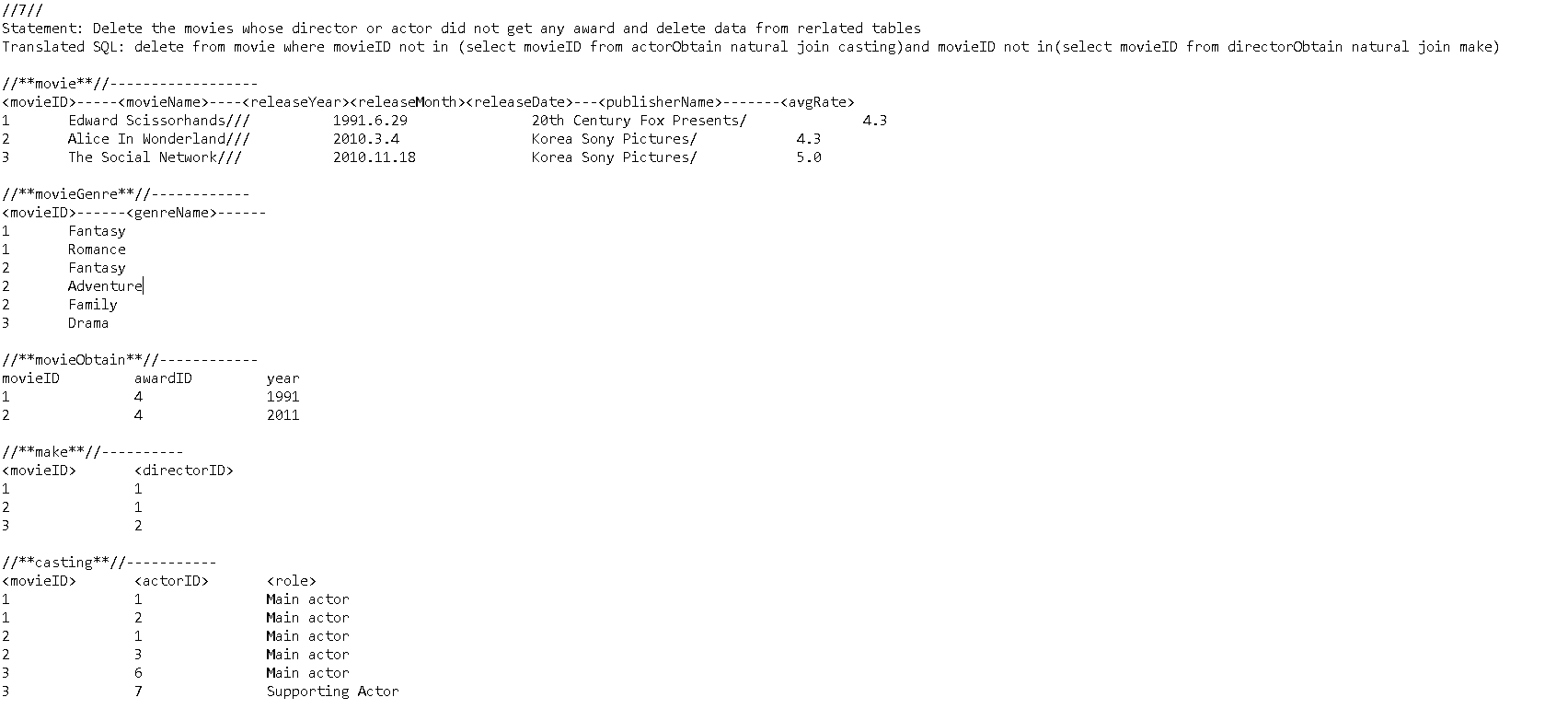


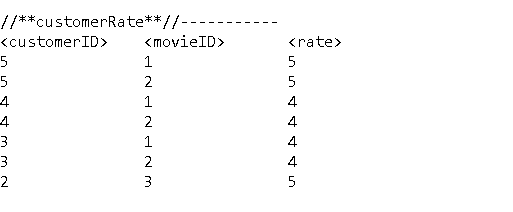












-Talk about your experience of doing this project

연결은 쉽겠지? 하고 pgadmin에서 미리 쿼리문을 짜뒀었는데, 사실 연결을 할때부터 난관이 시작되었습니다. 비밀번호 설정을 cse3207로 해두지않아서, 인터넷에 postgresql비밀번호 바꾸는법을 검색해서 다시 설정하였습니다.

처음 쿼리문들을 풀때는 여러가지 함수들, 또 natural join같은것들을 생각도못하고, 오로지 select과 insert , from과 where정도만 생각하여서 풀었습니다.(4번,6번 풀이가 아마 그럴것입니다.) 하지만 이것은 단지 자바의 도움을 받은것이지 제대로 된 쿼리문이 아닌것같아서, 다른 문제들을 풀면서 계속 여러가지 쿼리문을 써보려고 노력하고,(avg, natural join, count등등) 최대한 효율적인 쿼리문이 어떻게나올지 오래 고민하였습니다. 시간은 오래 걸렸지만, 몇번 연습하고 나니 술술 풀리게 되어서 기분이 좋았습니다. 조금 더 다른 문제들로 여러가지 쿼리문을 써보고싶다는 생각이들었습니다. 사실 보고서를 작성하면서, “아,,이것보다는 이렇게 쓰면 훨씬 편하고 좋은데, 내가 바보같았네” 라는 생각을 할 정도로, 지금은 쿼리문,함수들에 대한 이해도가 높아진것같아서 기분이 좋아졌습니다. (그래도 초반에는 내가 저렇게 작성했구나~ 라는 생각을 하게 되어 고치지는 않았습니다 ㅎㅎ)

-Write your available contact information -

12184102양유진

Email:cary1234@naver.com

HP:010-3411-5166