과제명 : Project1

학번 : 12201752

이름 : 신유미

**What you have implemented and what you have not** : 1 ~ 9까지 전체 구현

**Brief explanation of your implementation** : 기본키가 하나만 존재하는 director, actor, movie, award, customer, genre를 기준이 되는 테이블로 지정하여 가장 먼저 생성해주었고, 이외의 테이블들은 director, actor, movie, award, customer, genre의 기본키인 directorID, actorID, movieID, awardID, customerID, genreName을 참조하는 외래키로 구성되도록 만들었습니다. 이후에 두 가지 컬럼으로 이루어진 기본키에 대해서 각 테이블마다 제약 조건을 추가해주었습니다.

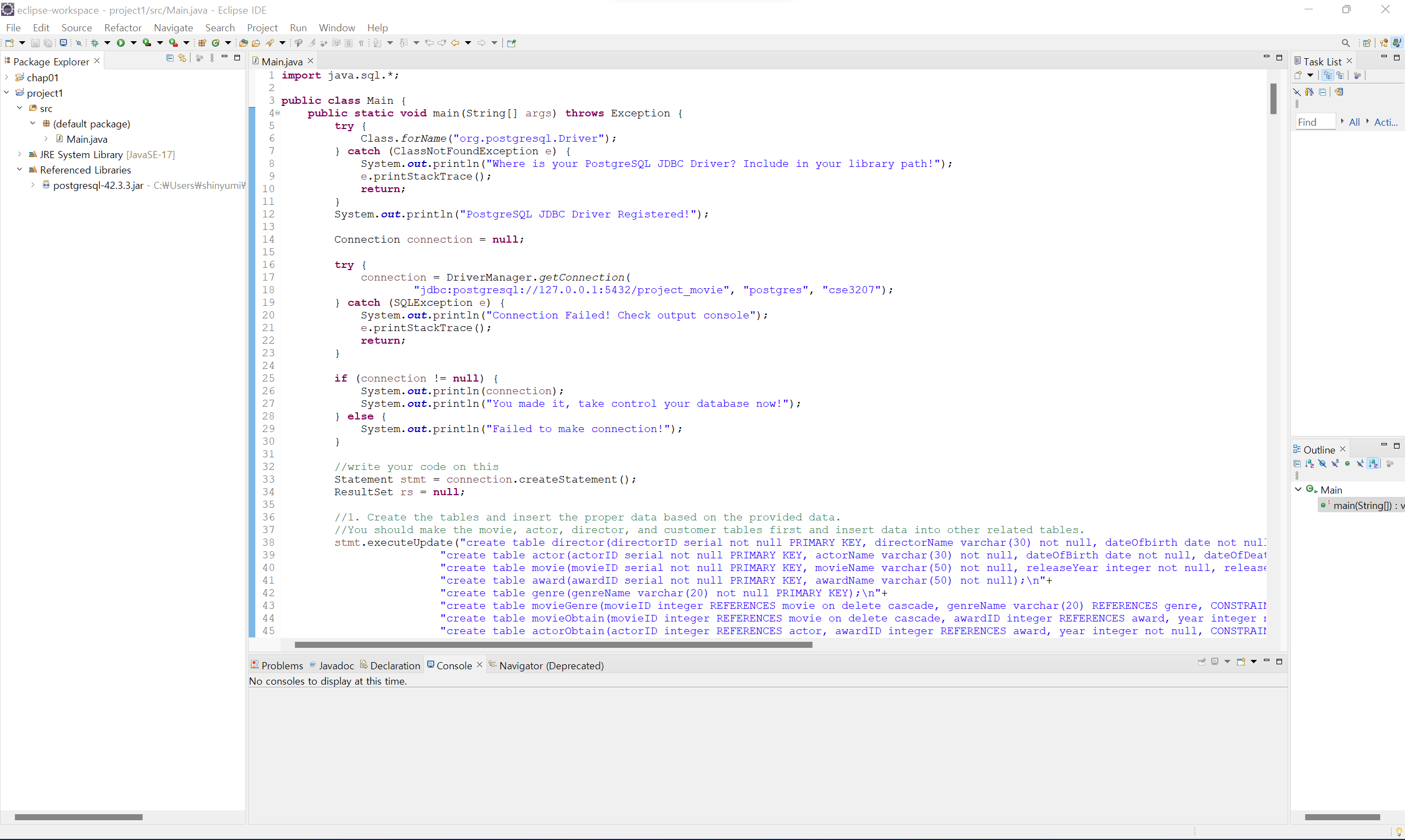
또한, 각각의 칼럼들의 데이터 타입을 정하는 과정에서 movieID, actorID, awardID에 대해서는 중복되는 값 없이 unique하게 지정되도록 serial이라는 타입을 활용하여 지정해주었습니다. 그 외의 칼럼들은 integer, char 등의 형식으로 넣어주었습니다. dateOfDeath, rate와 같은 특정 칼럼들은 아직 값이 존재하지 않을 수 있으므로 null을 허용하도록 하고, 기준이 되는 테이블과 연관된 테이블의 기본키에는 REFERENCES on delete cascade로 삭제 시에 연관된 테이블 또한 함께 삭제되도록 구현하였습니다.

초기 테이블 생성 이후 제공된 정보에 맞추어 데이터를 삽입할 시에는 기준이 되는 테이블의 정보는 그대로 삽입하되, 연관된 테이블에 넣어주는 정보는 기준 테이블에서 select문을 통해 가져와서 넣어주도록 했습니다. 또한, 여러 테이블에 정보를 분산해서 넣어야 하는 경우에도 본래 데이터가 위치하는 곳에서 select문을 통해 받아온 뒤에 insert문으로 연관된 테이블에 넣고, 새롭게 제공된 정보는 추가해서 삽입하도록 했습니다.

Rate가 업데이트될 시에 필수적으로 avgRate도 함께 수정되어야 하는 부분에 대해서는 제공된 정보가 여러 튜플을 반환하기도 하므로 for문을 사용하기 위해 PL/SQL function을 이용해서 record 형태로 튜플을 하나씩 받은 다음에 각각의 튜플을 통해 조건을 구성하고, 조건에 맞는 위치에 insert문으로 rate를 넣어주고, update문을 통해 movie table 내의 avgRate를 customerRate 테이블 내의 rate 값을 참조해서 업데이트하도록 구현했습니다.

세부적으로 rate를 넣어줄 때에 조건이 director에 관한 정보나 role과 gender가 혼합된 정보, genre에 대한 정보, 특정 상을 받은 것에 대한 정보 등이 주어지는 경우가 각각 구별이 되어있는데 이 경우에 참조하는 테이블이 다르고, 조건을 생성할 시에 필요한 칼럼이 다르기 때문에 각각 rate\_movie\_from\_director(), rate\_movie\_from\_gender\_role() … 등의 함수를 생성하여 반환되는 여러 튜플에 대해 동일한 동작을 반복하도록 설계했습니다.

**How to compile and run** : Java Application 프로젝트를 생성한 이후에 Eclipse IDE를 사용하여 postgreSQL JDBC 라이브러리를 추가하고, postgreSQL 내부에 만들어 놓은 project\_movie 데이터베이스와 연결되어 동작하도록 합니다.



**Talk about your experience of doing this project** : 이론적인 부분을 배웠을 당시에 데이터베이스, 테이블, 컬럼, 기본키, 외래키, 연결 관계 등에 대해 어느 정도 이해가 되었다고 생각했는데, 실제로 데이터베이스 내부의 테이블을 구성하고, 그 테이블의 칼럼, 칼럼 각각의 변수 타입, 테이블 간의 연결 관계 등을 직접 고려해보니 관계형 데이터베이스를 구축하는 것이 상당히 복잡하다는 것을 깨닫게 되었습니다. 데이터를 삽입할 때에도 기존의 변수 타입을 놓치면, 제약 조건을 위배하는 일이 발생하기도 하고, 중복되지 않아야 하는 데이터가 중복되어 삽입되기도 하는 경우를 직접 마주하면서 insert 시에 어떤 부분을 고려해야 하는지, 중복되지 않도록 어떻게 제약조건을 삽입해야 하는지를 여러 상황 속에서 경험해볼 수 있었습니다. 프로젝트를 진행하면서 막상 SQL문을 직접 입력해보고, 구성해보려고 하니 단순히 SQL문을 보고 이해하는 것만으로는 충분하지 못하다는 것을 깊이 깨닫기도 했습니다. 어렵고, 복잡한 과정이었지만 테이블 간의 관계를 설정하고, 테이블 간의 종속 관계를 통해 어떻게 연결이 되는지도 파악해볼 수 있었습니다. 가장 어려웠을 때는 처음에 구축하는 과정이었는데 직접 Aquery tool을 활용하여 ERD를 구축해보면서 확인을 했고, 그 이후부터는 ERD을 바탕으로 연결 관계를 파악한 뒤에 project\_movie 데이터베이스를 구축할 수 있었습니다. ERD 작성부터 구축까지 전부 실행해보니 오래 걸리고, 힘들었지만 그만큼 데이터베이스에 대한 이해에 도움이 많이 되었습니다.

**Write your available contact information such as a phone number** : 010-4932-1405

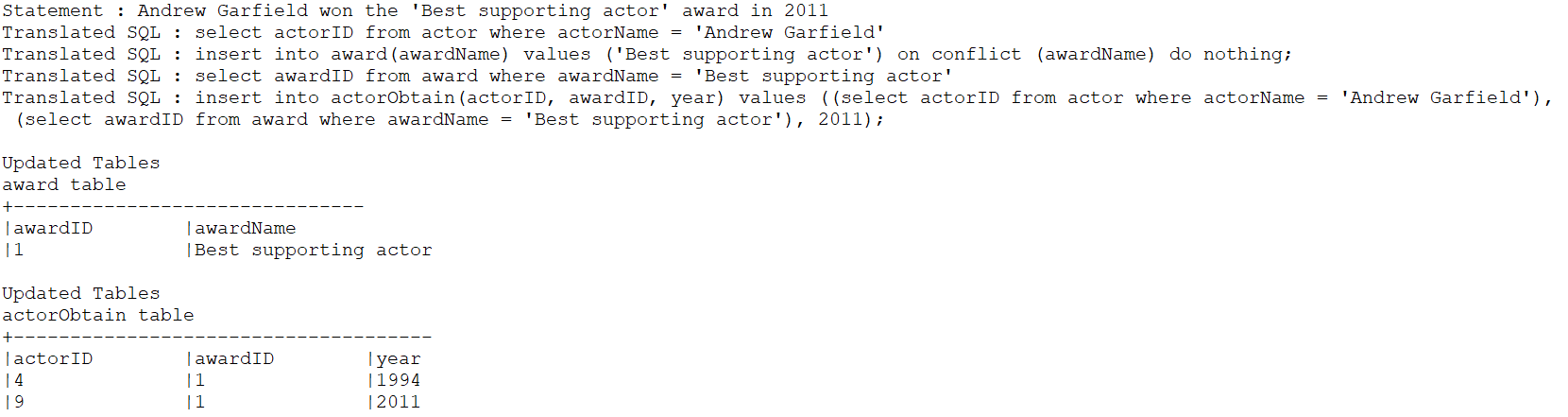
**The result of what I have implementation** :

2.1 Winona Ryder won the ‘Best supporting actor’ award in 1994

텍스트이(가) 표시된 사진

자동 생성된 설명

2.2 Andrew Garfield won the ‘Best supporting actor’ award in 2011



2.3 Jesse Eisenberg won the ‘Best main actor’ award in 2011

텍스트이(가) 표시된 사진

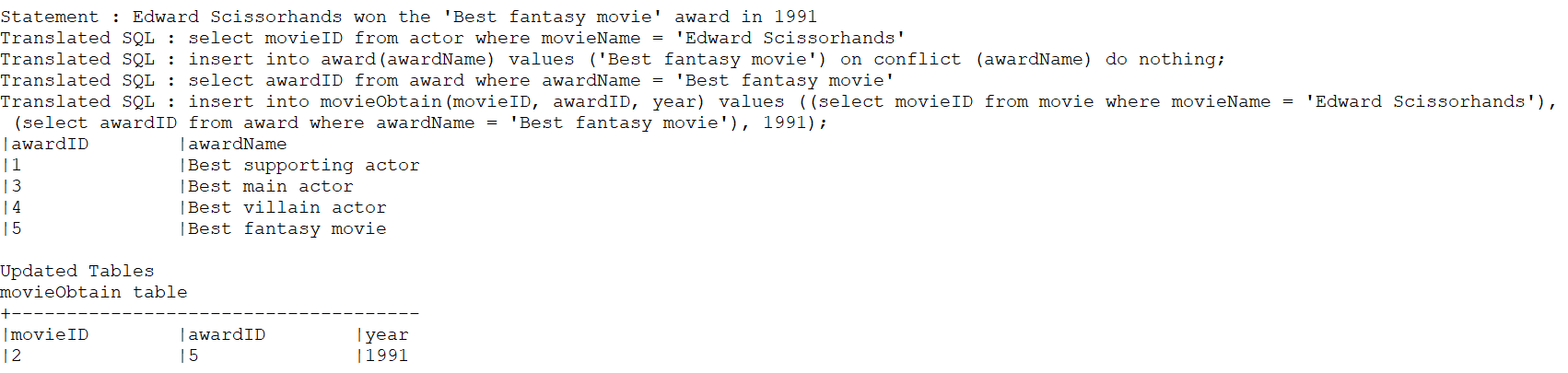
자동 생성된 설명

2.4 Johnny Depp won the ‘Best villain actor’ award in 2011

텍스트이(가) 표시된 사진

자동 생성된 설명

2.5 Edward Scissorhands won the ‘Best fantasy movie’ award in 1991



2.6 Alice In Wonderland won the ‘Best fantasy movie’ award in 2011

텍스트이(가) 표시된 사진

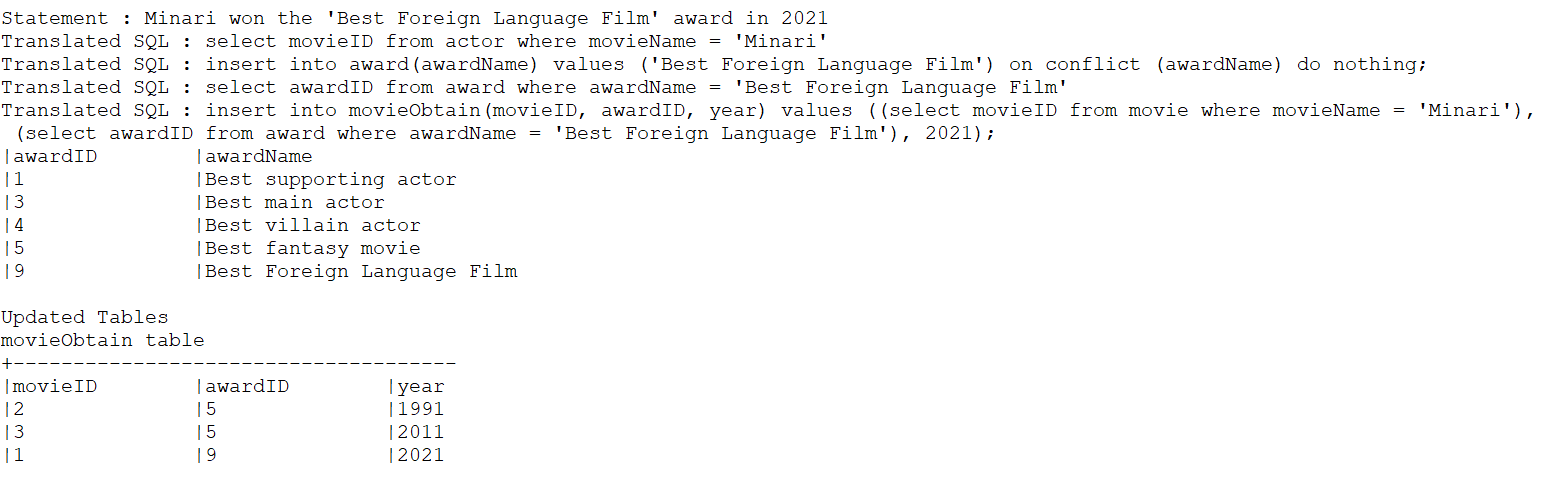
자동 생성된 설명

2.7 Youn Yuhjung won the ‘Best supporting actor’ award in 2021

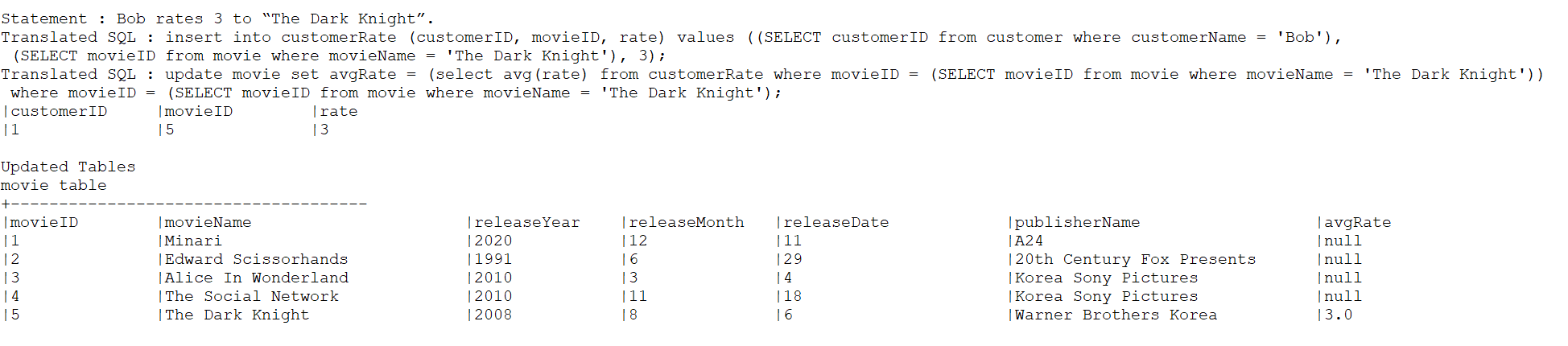
텍스트이(가) 표시된 사진

자동 생성된 설명

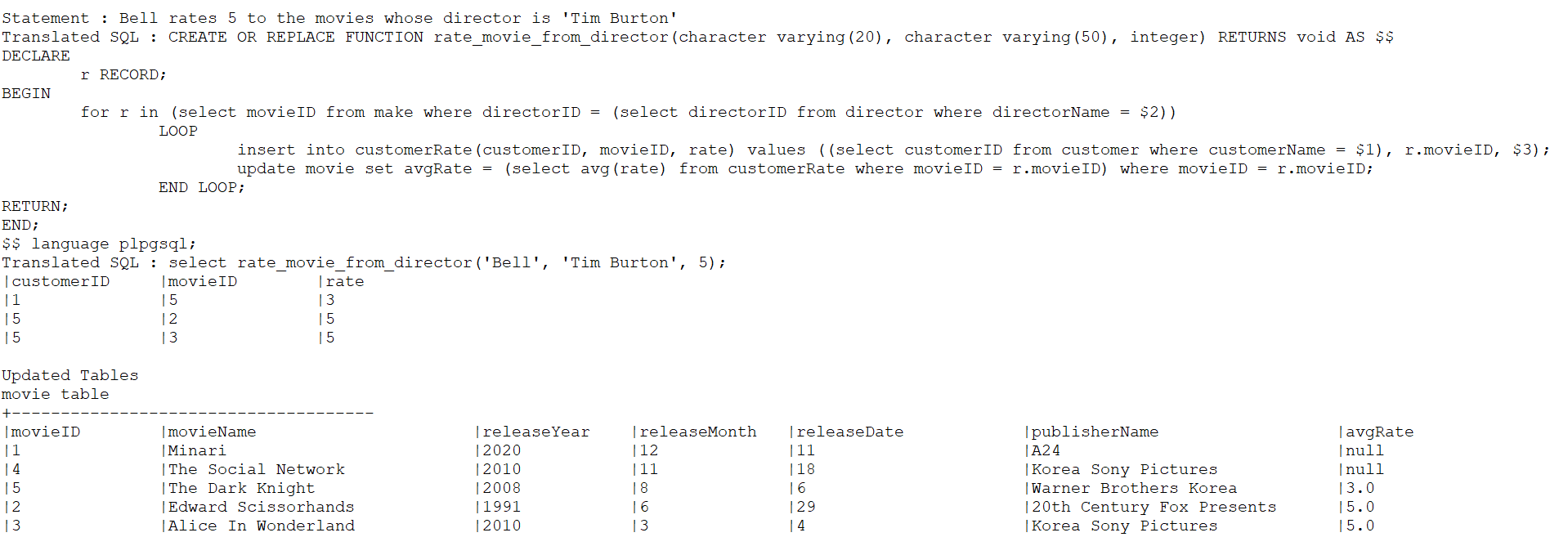
2.8 Minari won the ‘Best Foreign Language Film’ award in 2021



3.1 Bob rates 3 to “The Dark Knight”



3.2 Bell rates 5 to the movies whose director is Tim Burton

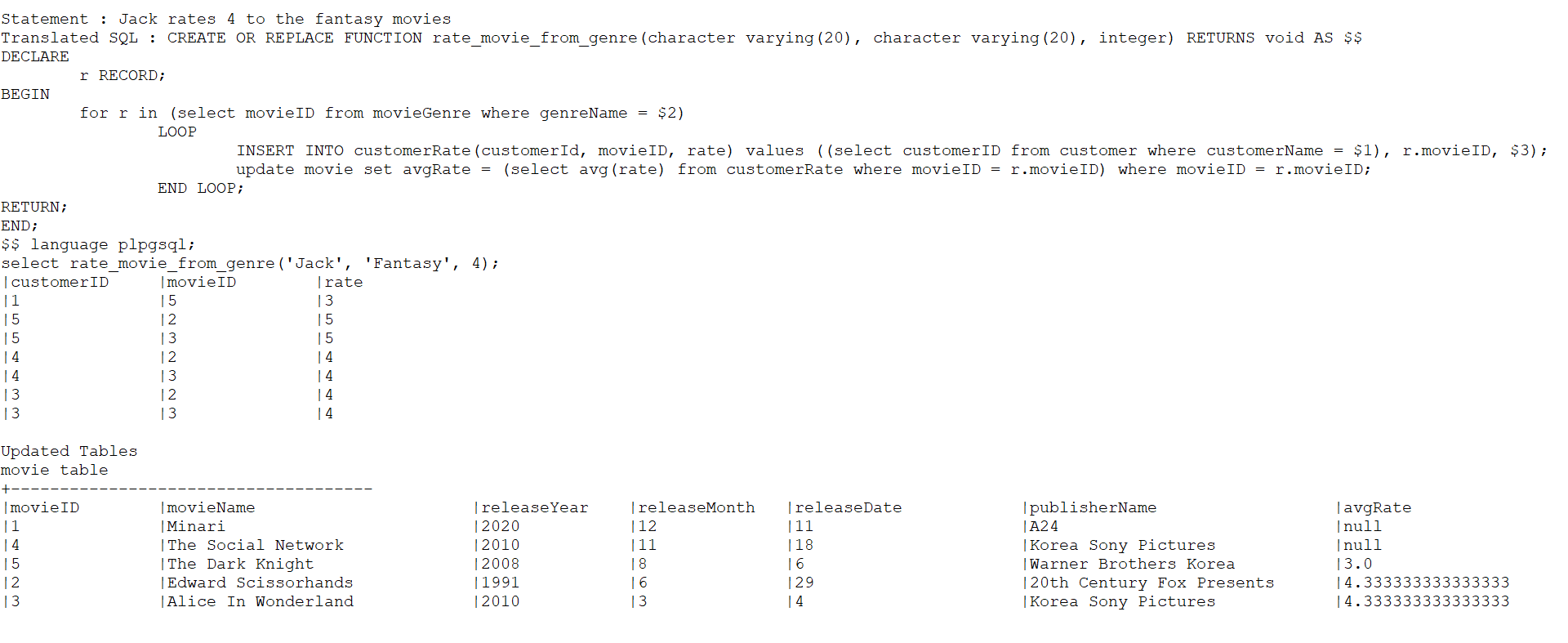


3.3 Jill rates 4 to the the movies whose main actor is female

텍스트이(가) 표시된 사진

자동 생성된 설명

3.4 Jack rates 4 to the fantasy movies

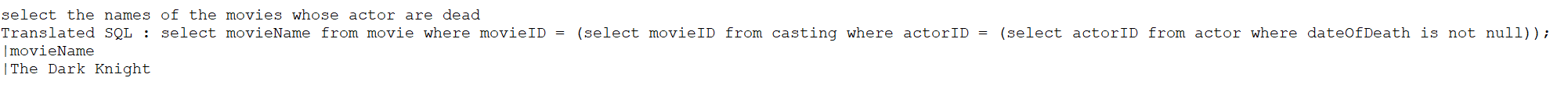


3.5 John rates 5 to the movies whose actor won the “Best supporting actor” award

텍스트이(가) 표시된 사진

자동 생성된 설명

4. Select the names of the movies whose actor are dead

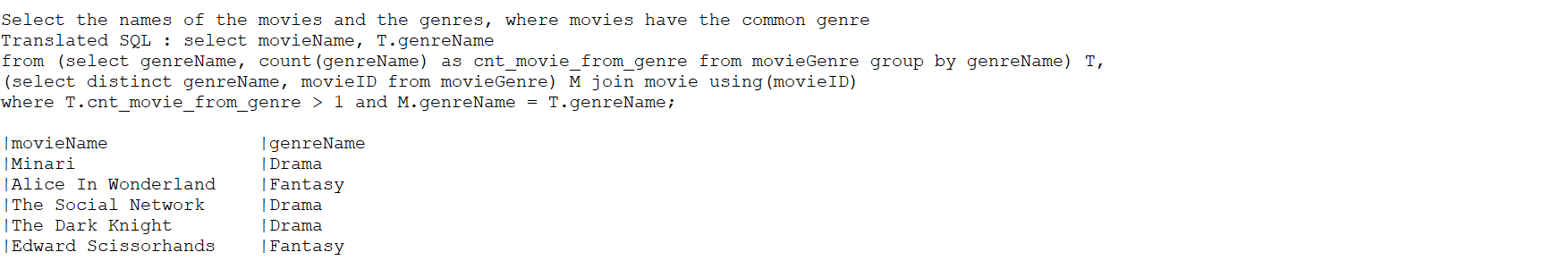


5. Select the names of the directors who cast the same actor more than once

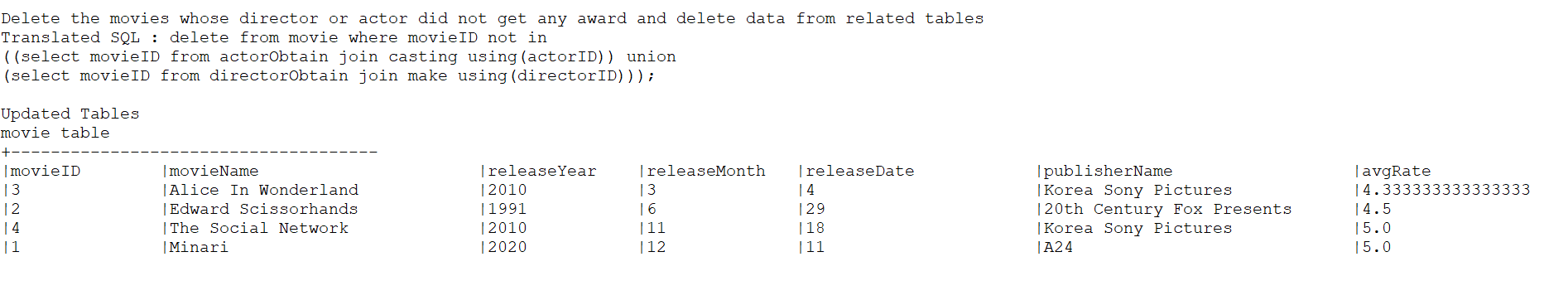
텍스트이(가) 표시된 사진

자동 생성된 설명

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



7. Delete the movies whose director or actor did not get any award and delete from related tables











8. delete all customers and delete data from related tables

