**CSE 3207 Project # 1**

**Implementation of a simple database application**

**12161567 박기수**

**E-mail : rltn2121@naver.com**

1. **What I have implemented**

* JDBC와 PostgreSQL의 연동
* 13개의 relational table 생성 및 삭제
* 각 테이블의 primary key, foreign key 관계 설정
* customerRate 테이블이 갱신됐을 때 movie 테이블의 avgRate 자동 갱신
* 1~9번 쿼리의 statement를 SQL로 번역 및 처리

1. **Brief explanation of implementation**

* JDBC, PostgreSQL 연동 및 1~9번 쿼리 실행을 각각의 method로 구현
* JDBC, PostgreSQL 연동: **void connect\_to\_PostgreSQL()**
* 1~9번 쿼리: **void do\_1st\_query() ~ void do\_9th\_query()**
* Main 함수에서 connect\_to\_PostgreSQL()를 호출한 후 statement 생성. 이후 각 쿼리를 실행하는 method 호출
* 모든 ID는 varchar(5) 형식이며, 첫 자리 숫자로 유형 구분

**directorID:** 10001부터 시작

**actorID:** 20001부터 시작

**movieID**: 30001부터 시작

**customerID:** 40001부터 시작

**awardID**: 50001부터 시작

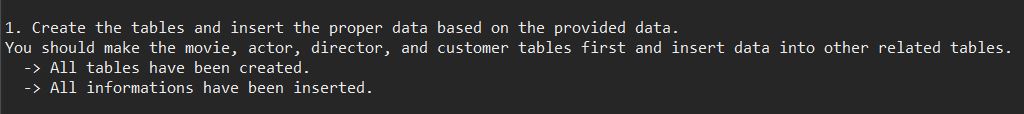
* Method의 종류와 기능

|  |  |
| --- | --- |
| **Method** | **기능** |
| public static void connect\_to\_PostgreSQL () | JDBC와 PostgreSQL을 연결하는 method |
| public static void insertTable\_(table\_name) (…) | (table\_name) 테이블에 tuple을 삽입하는 method |
| public static void printTable\_(table\_name) () | (table\_name) 테이블의 정보를 출력하는 method |
| public static void updateTable\_movieRate () | movie 테이블의 avgRate를 갱신하는 method |
| Public static void do\_nth\_query () | n번째 쿼리를 실행하는 method |

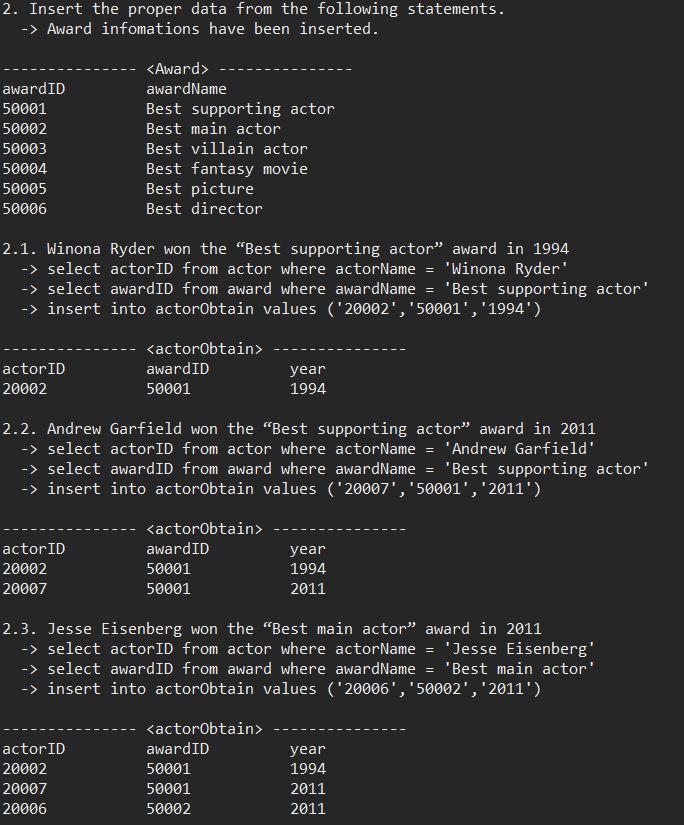
1. **How to Compile and run**

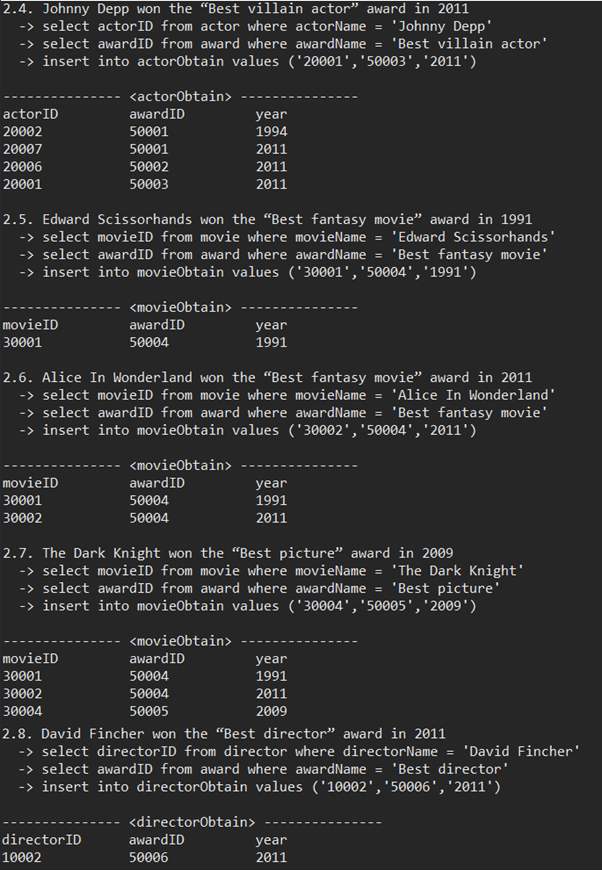
사용한 JAVA idle : Eclipse neon.2

1. **Display result for all queries**
2. **1번 쿼리 실행 결과**

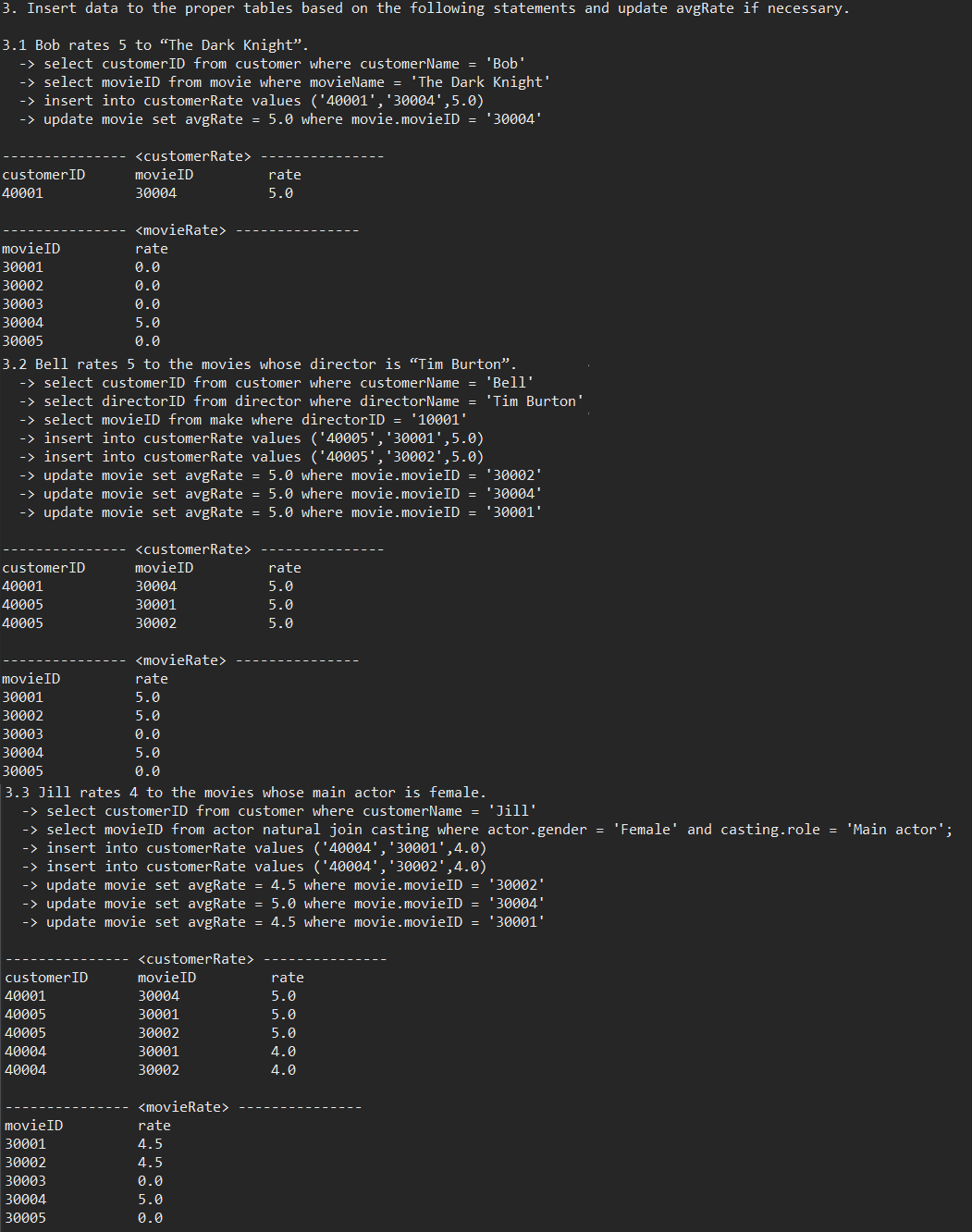


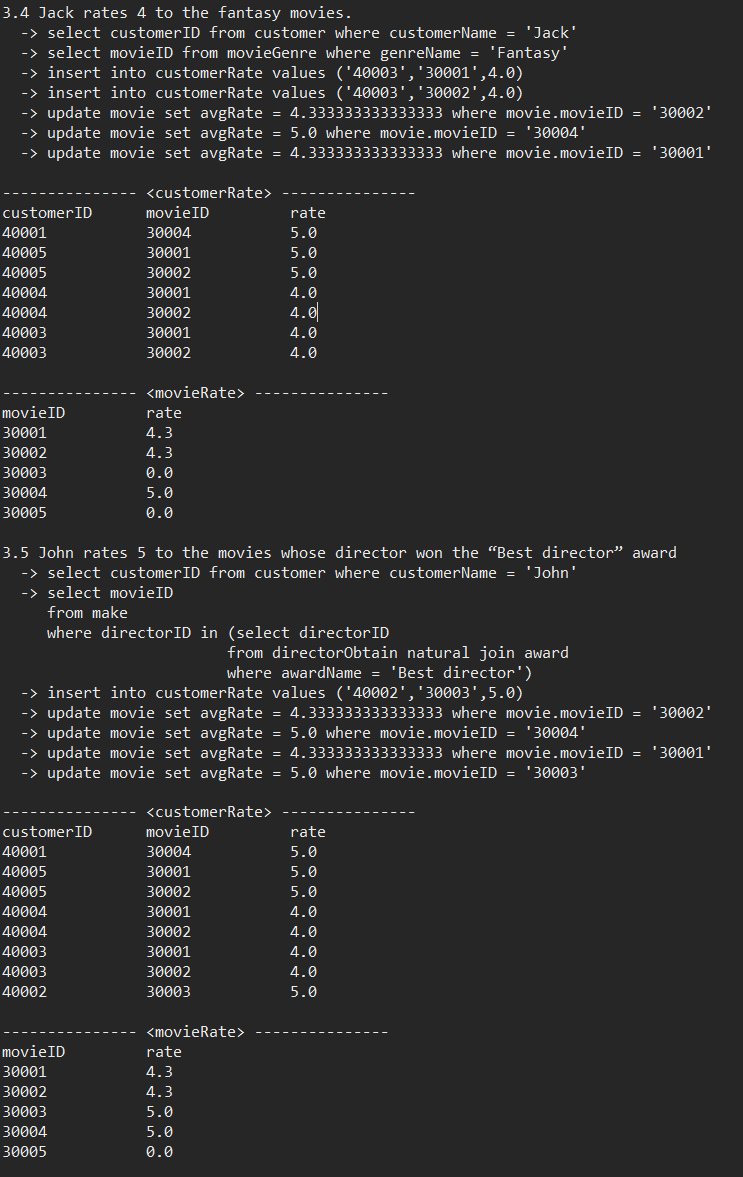
1. **2번 쿼리 실행 결과**



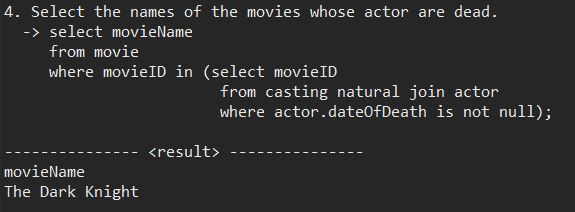


1. **3번 쿼리 실행 결과**

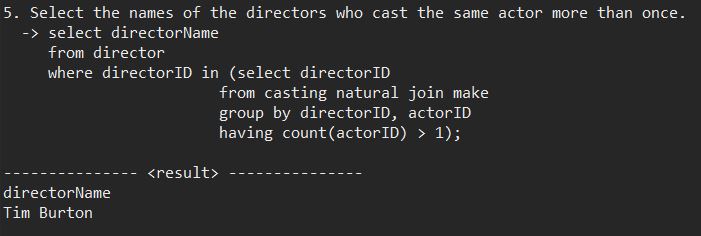




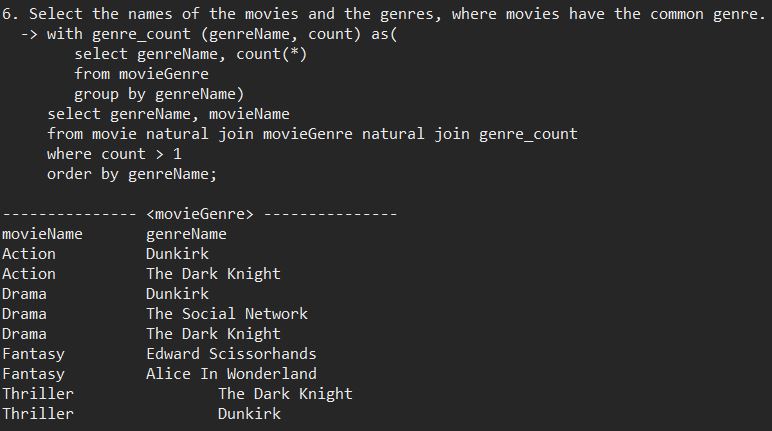
1. **4번 쿼리 실행 결과**

****

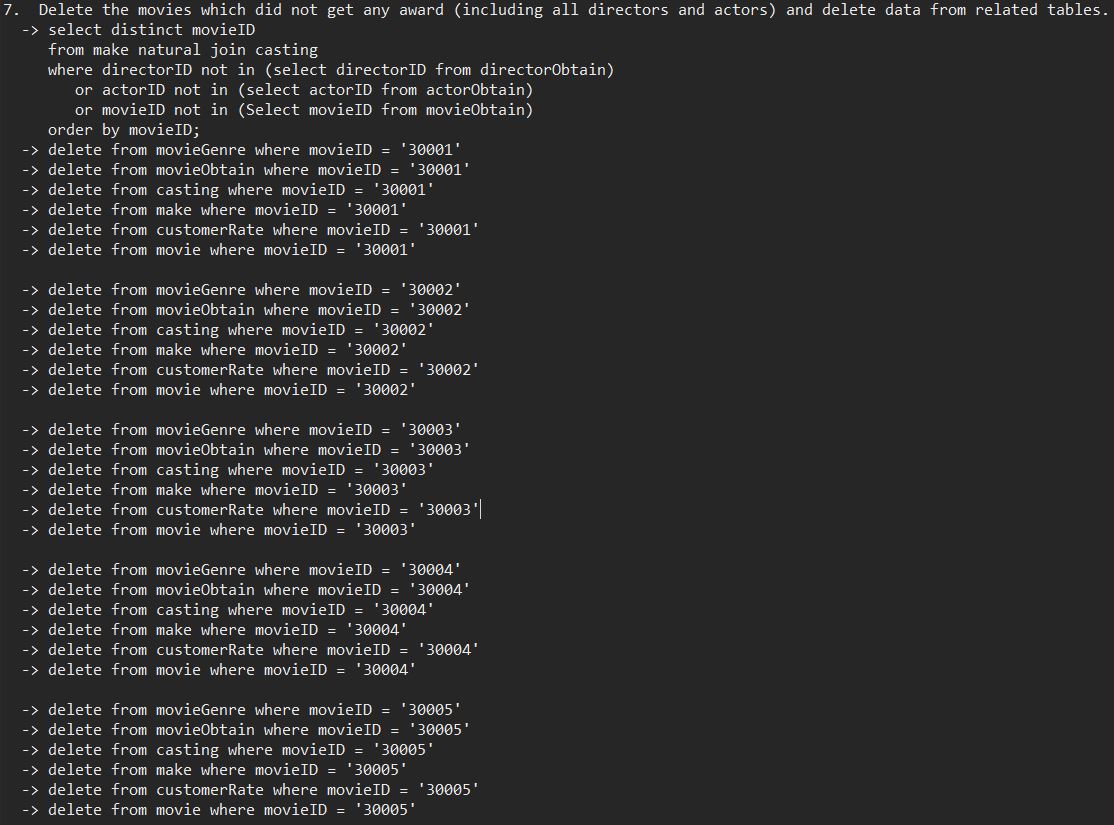
1. **5번 쿼리 실행 결과**

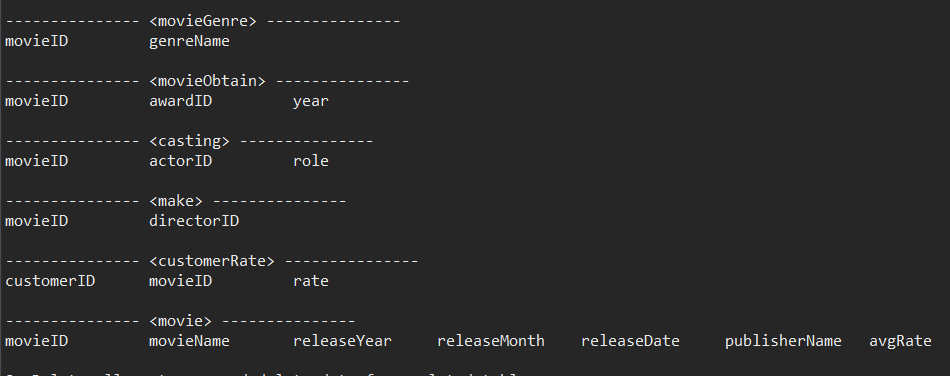


1. **6번 쿼리 실행 결과**

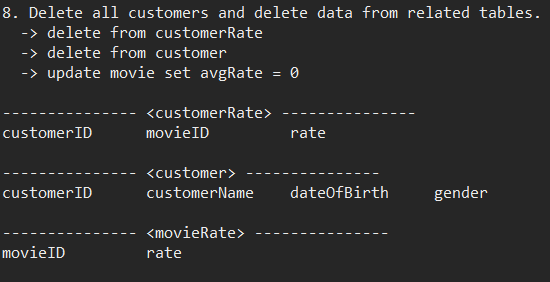


1. **7번 쿼리 실행 결과**

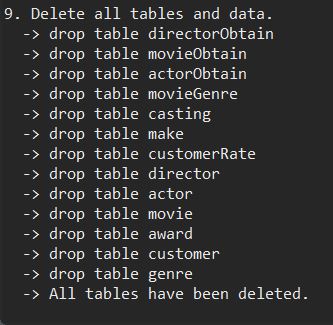




1. **8번 쿼리 실행 결과**



1. **9번 쿼리 실행 결과**



1. **Experience of doing this project**

statement를 쿼리로 번역하는 작업을 우선적으로 처리했습니다. With절, where절 하위 등 Nested 쿼리의 사용법이 익숙하지 않아서 교재를 찾아보면서 했습니다. JDBC와 PostgreSQL 연동을 완료하고 번역한 쿼리를 실행했는데, 하나의 쿼리로 statement를 처리하려고 하니 처리는 되는데 가독성이 떨어졌습니다. 그래서 하나의 쿼리를 여러 개의 쿼리로 나눠서 수행했습니다. 또한, JDBC tutorial에서 일반 statement보단 prepare statement가 더 효율적이라고 하셔서 prepared statement를 최대한 사용했습니다. 초기 데이터 삽입, 테이블 삭제 등과 같은 반복적인 작업을 처리할 때 먼저 배열에 관련된 정보들을 삽입하고, for문과 prepare statement을 통해서 쿼리를 반복적으로 수행하여 코드를 간결하게 작성했습니다. 모든 삽입, 삭제 과정을 for문과 prepare statement를 이용해서 처리하고 싶었지만, statement마다 조건이 조금씩 달랐기 때문에 처리하지 못했습니다. 하나의 statement를 처리하는데 일반 statement를 사용하느냐, prepare statement를 사용하느냐, 하나의 쿼리로 처리하느냐, 여러 개의 쿼리로 나눠서 처리하느냐 등 쿼리를 실행하는 방법이 아주 많았는데, 어떠한 방법이 최적의 방법인지 잘 모르겠습니다.