**데이터베이스 – MySQL**

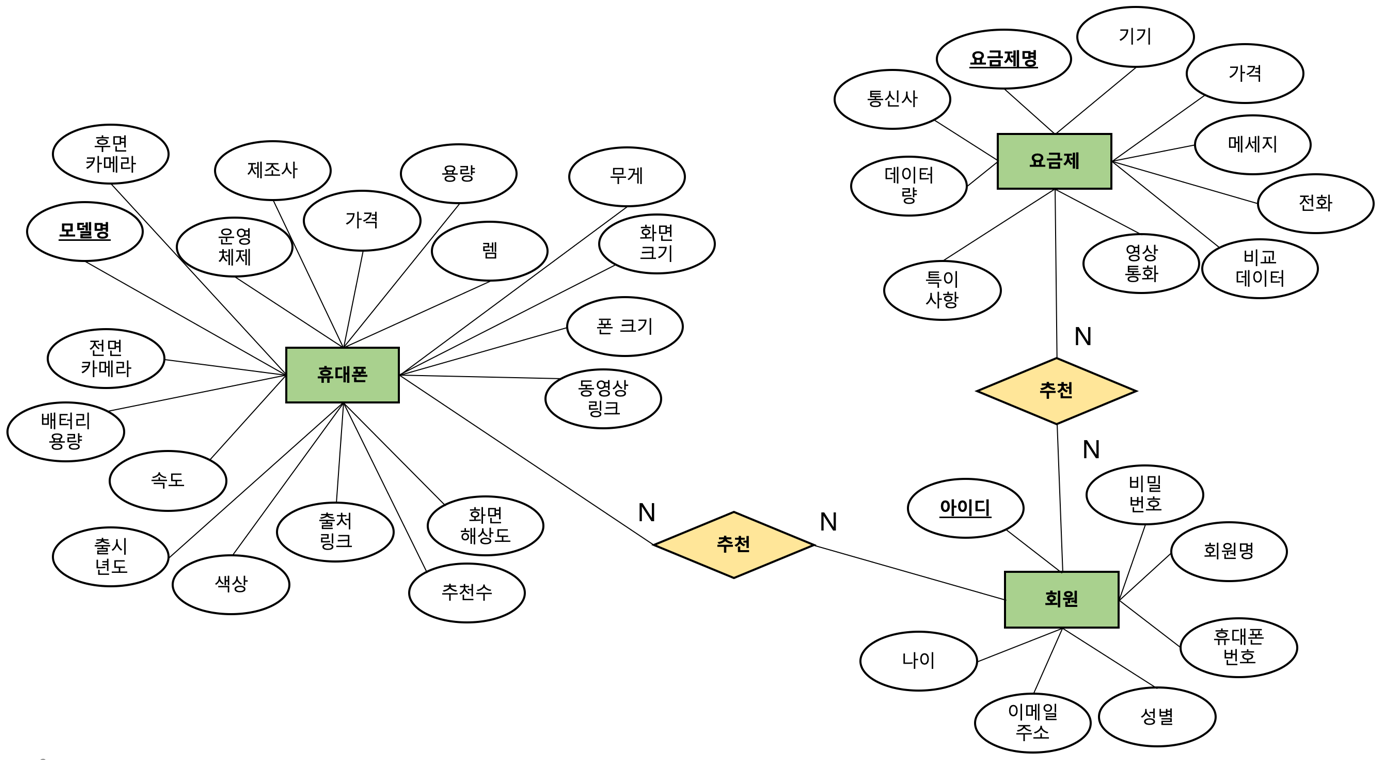


그림 1 추천 시스템 데이터베이스(recDB) ER Diagram

1. 추천 시스템 데이터베이스는 휴대폰, 요금제, 회원의 엔티티가 있다.
2. 휴대폰은 모델명, 운영체제, 제조사, 가격, 용량, 렘, 무게, 화면 크기, 폰 크기, 전면 카메라, 후면 카메라, 배터리 용량, 속도, 출시 년도, 색상, 출처 링크, 화면 해상도, 동영상 링크, 추천 수의 속성이 있으며 모델명이 키 속성이다.
3. 요금제는 요금제 명, 기기, 통신사, 가격, 데이터 량, 메세지, 전화, 영상통화, 비교 데이터, 특이 사항의 속성이 있으며 요금제 명이 키 속성이다.
4. 회원은 아이디, 비밀번호, 회원 명, 휴대폰 번호, 성별, 나이, 이메일 주소의 속성이 있으며 아이디가 키 속성이다.
5. 회원과 요금제는 추천 관계가 있다.
6. 회원과 휴대폰은 추천 관계가 있다.
7. 회원은 여러 휴대폰을 추천 받으며 휴대폰은 여러 회원들에게 추천될 수 있다.
8. 회원은 여러 요금제를 추천 받으며 요금제는 여러 회원들에게 추천될 수 있다.

Eclipse에 MySQL을 연동시켜 데이터베이스를 구현했다. DBTest 클래스를 생성하여 메소드로 다양한 기능을 넣었다.



그림 2 DB 서버 연결 메소드

프로젝트 라이브러리에 mysql connector를 추가하면 위의 connectDB() 메소드에서 jdbc 드라이버를 등록한다. 그 후Connection 객체를 생성하여 경로를 지정해주고DB 서버와 연결시킨다. 여기서 mysql 서버의 경로 설정 시 타임 존을 인식하지 못하는 에러가 발생하여 mysql서버의 타임존을 Asia/Seoul로 지정해주었다. “com.mysql.jdbc.Driver” 클래스를 로드하지 못할 경우의 예외처리(ClassNotFoundException e)와 데이터 소스에 접근하는 동안 발생하는 오류에 대한 예외처리(SQLException e)도 해주었다.

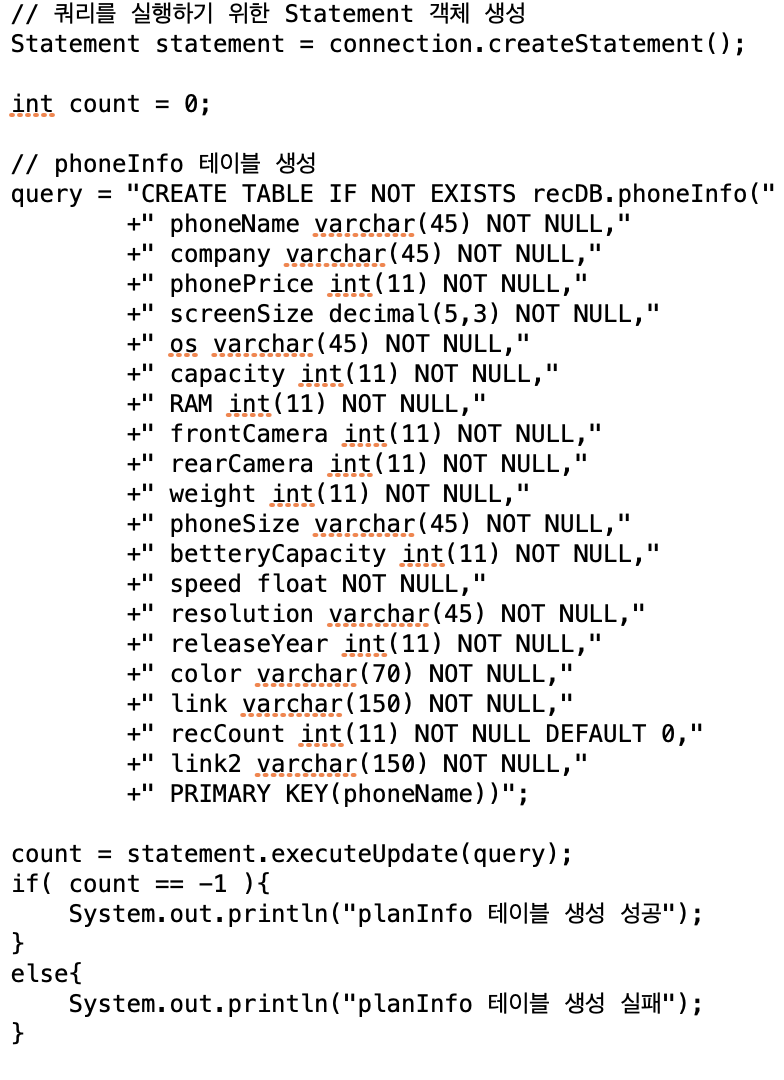
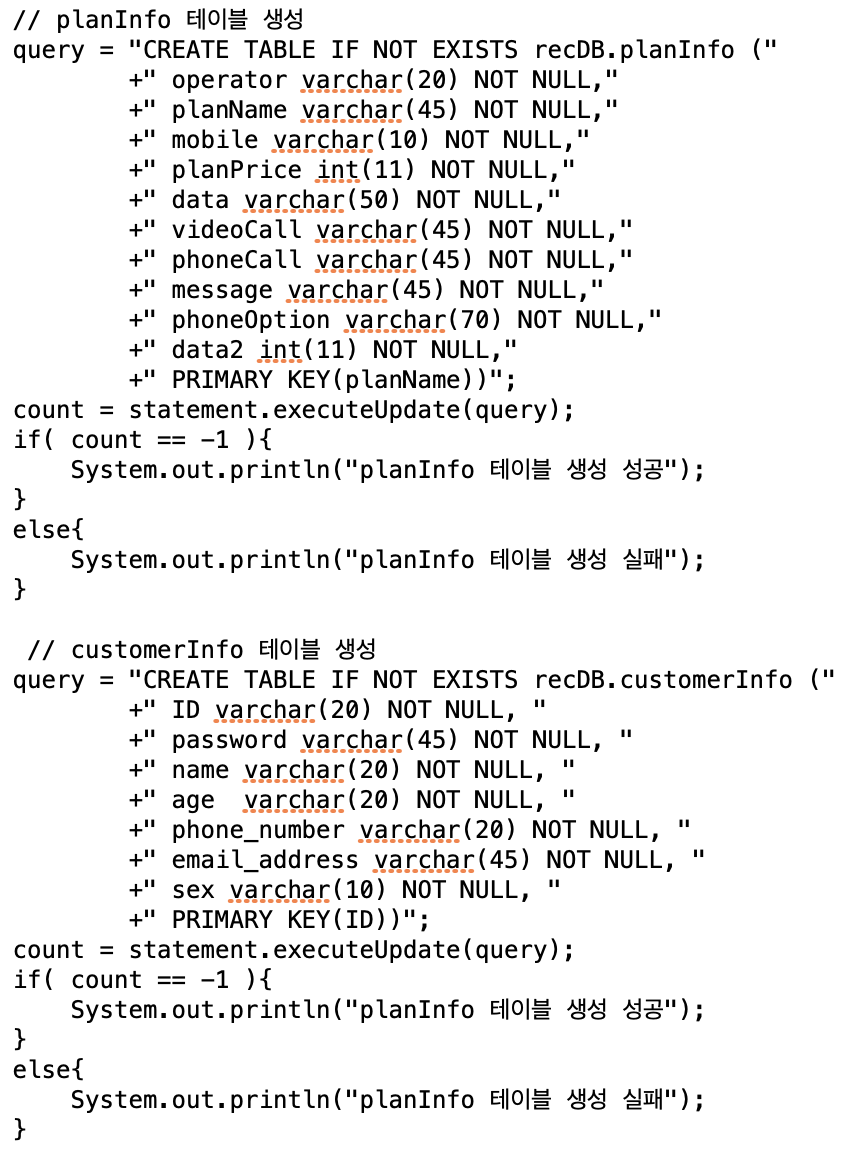
 

그림 3 데이터 로드 메소드

loadData() 메소드이다. 우선 해당 테이블이 없으면 생성시키도록 하는 쿼리를 작성하여 Statement 객체에 담아 실행시킨다. executeUpdate() 메소드는 Create/Drop 명령 성공 시 -1을 반환하고 그 외에는 반영된 레코드 건수를 반환하므로 int 형 변수 count에 반환 값을 저장하여 테이블 생성 성공 여부를 확인할 수 있도록 구현하였다. 테이블 생성 쿼리는 column 정보 가독성이 좋도록 한 눈에 보이게 나눠 작성하였다. 테이블을 생성했다면 미리 만들어놓은 엑셀 파일의 데이터를 데이터베이스 서버로 로드한다. 엑셀 파일은 CSV 파일로 변환 후 프로젝트 파일 밑으로 위치시켰다. 이 때 CSV 파일의 구분자는 ‘,’이기 때문에 데이터의 ‘,’는 모두 없앴다. 마찬가지로 executeUpdate() 메소드로 데이터 로드 성공 여부를 확인할 수 있다. 그리고 데이터 소스에 접근하는 동안 발생하는 오류에 대한 예외처리(SQLException e)를 해주었다.

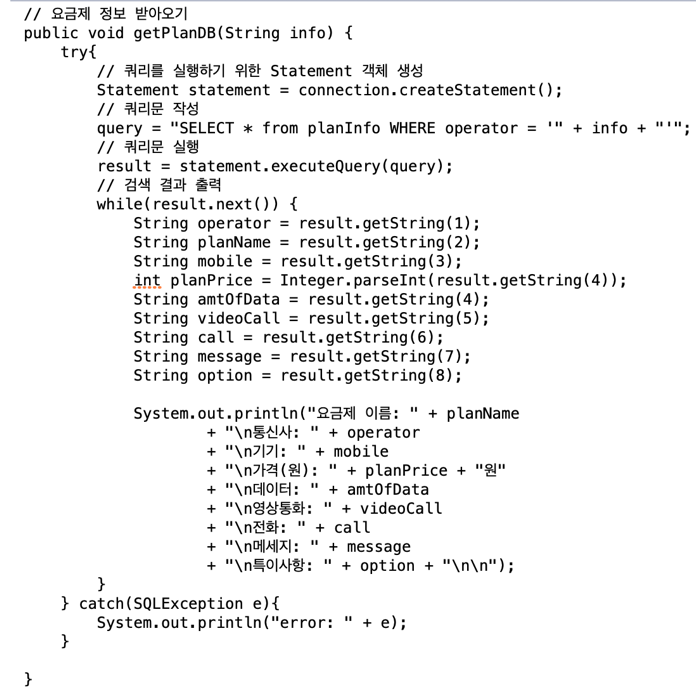
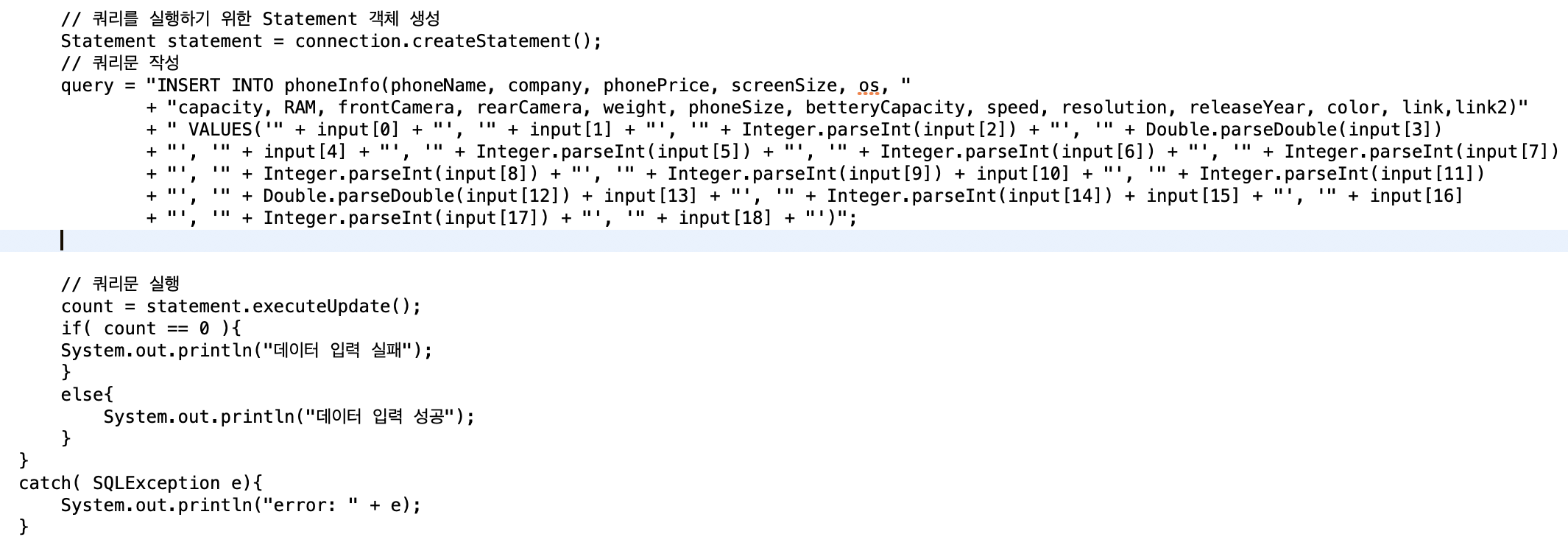


그림 4 데이터 선택 메소드

getPhoneDB() 메소드에서는 휴대폰 모델명을 검색하면 해당 모델 정보를 가져오는 쿼리를 작성하여 Statement 객체에 담아 실행시킨다. 모델명은 메소드를 호출할 때 파라미터로 받아온다. 이 때는 결과값이 존재하므로 executeQuery() 함수를 사용하여 Result 객체인 result에 결과값을 담는다. 그리고 다음 내용이 없을 때 까지 순서대로 값을 뽑아내 각 데이터 형식에 맞는 변수에 저장시킨 후 출력한다.

getPlanDB() 메소드에서는 통신사를 검색하면 해당 통신사의 요금제 정보를 가져오는 쿼리를 작성해 실행시킨다. getPhoneDB()와 마찬가지 과정으로 정보를 출력시킨다.

각 메소드에는 데이터 소스에 접근하는 동안 발생하는 오류에 대한 예외처리(SQLException e)가 구현되어 있고, 검색 정보 범위는 기능에 맞춰 수정이 가능하다.



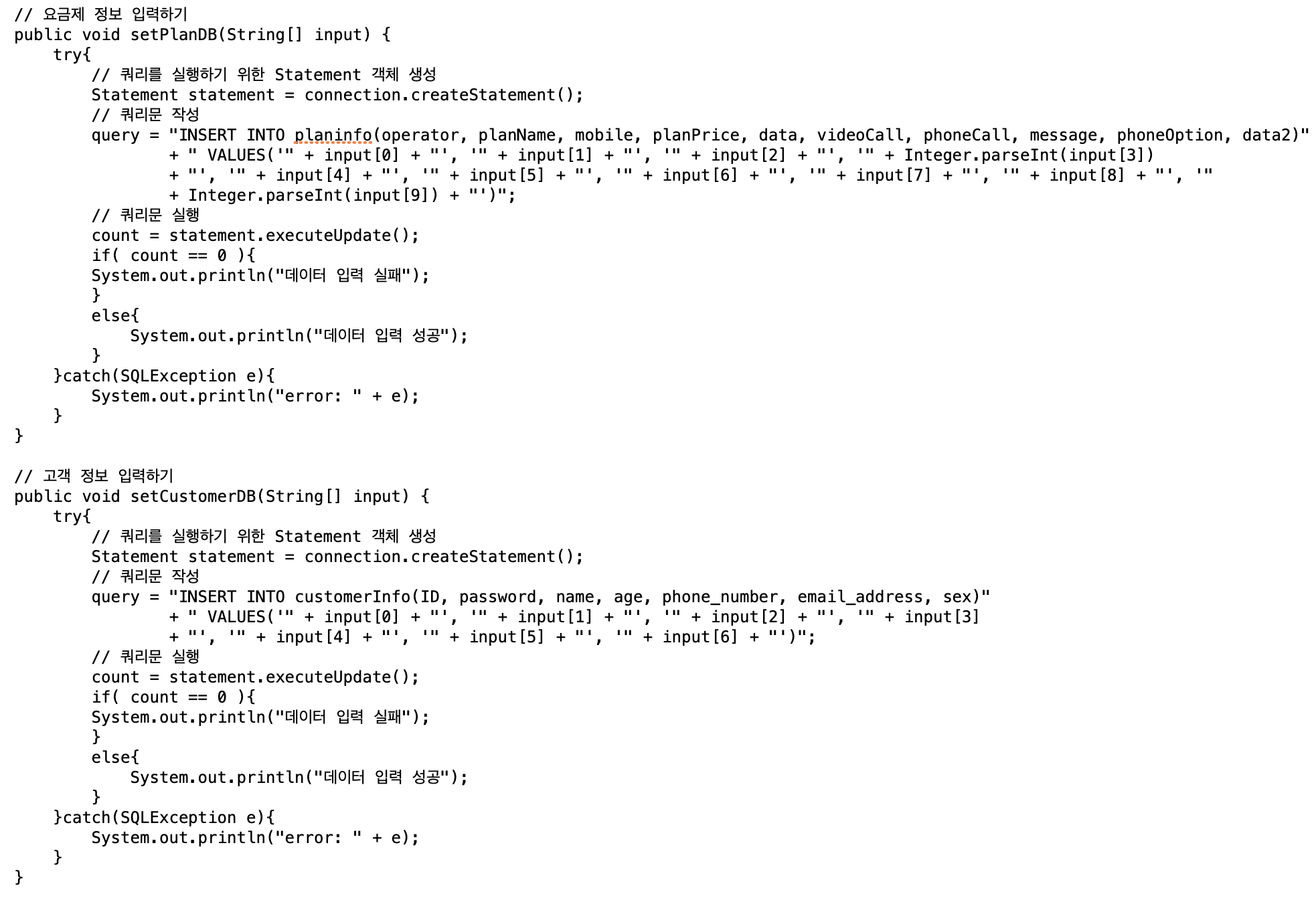


그림 5 데이터 입력 메소드

위에서부터 차례로 휴대폰, 요금제, 회원 정보 입력 메소드이다. 각 메소드는 입력할 데이터를 배열 파라미터로 받아온다. 배열 원소를 형식에 맞게 바꿔 데이터 입력 쿼리를 작성해준 후 Statement 객체에 담아 실행한다. executeUpdate() 메소드로 반영된 레코드수를 count에 받아와 데이터 입력의 성공 여부를 확인할 수 있다. 다른 메소드들과 마찬가지로 데이터 소스에 접근하는 동안 발생하는 오류에 대한 예외처리(SQLException e)가 구현되어 있다.

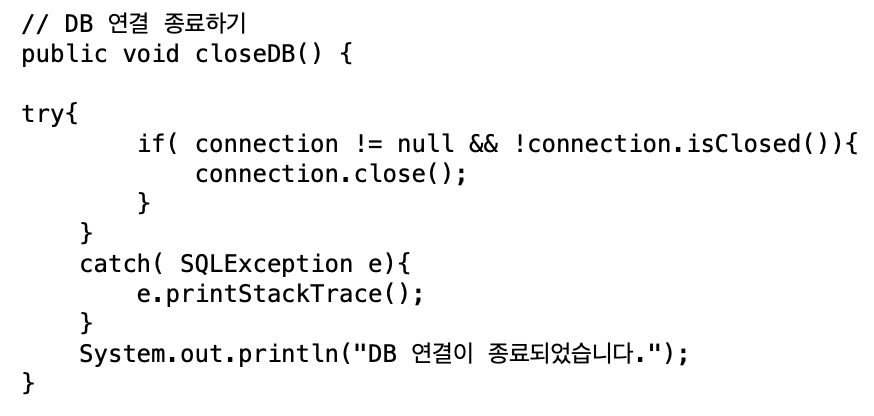


그림 6 DB 서버 연결 종료 메소드

DB 서버로부터의 연결을 종료하는 메소드로 메인 함수에서 DBTest에 있는 필요한 모든 메소드를 모두 호출한 후 마지막에 꼭 이 메소드를 호출하여 DB 서버 연결을 종료해 줘야 한다. 데이터 소스에 접근하는 동안 발생하는 오류에 대한 예외처리(SQLException e) 후 모든 과정이 끝나면 종료되었다는 메세지가 출력된다.