**음원 스트리밍 서비스를 위한 음원 데이터베이스**

**Project 4: DBMS를 활용한 프로그램 개발**

2019055078 컴퓨터소프트웨어학부 신채영

**Project 3 수정사항:**

1. 음원 테이블:

이름, 번호, 제목, 유료여부, 길이, 관리자번호, 등록날짜

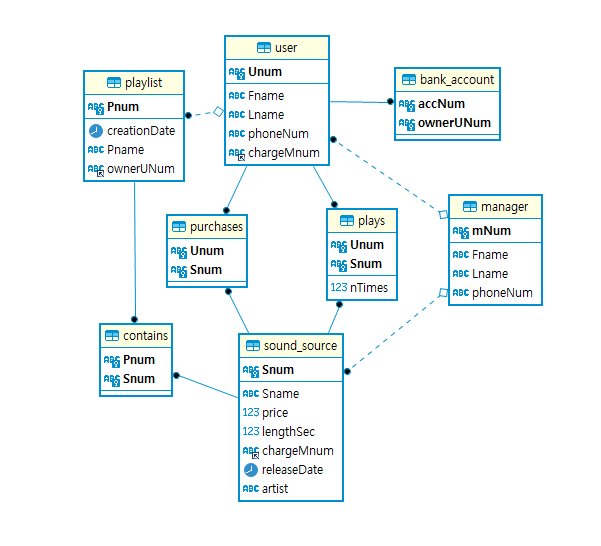
* 이름, 번호, **가수**, 유료여부, 길이, 관리자번호, 등록날짜 (오타)

1. 플레이리스트 테이블:

이름, 구독자번호, 생성날짜

* 번호, 이름, 구독자 번호, 생성날짜

플레이리스트 PK를 새로 만들어 줌. (관리하기 더 편리함)

**변경 후 DB entity 관계도 (Dbeaver):**

**프로그램 기능 구조:**

1. **Exit**

* 프로그램 종료

1. **유저 메뉴**

* 유저 아이디 입력
  + 이전 화면
  + 유저 정보
  + 내 플레이리스트(유저 당 한 개씩 소유)

-플레이리스트 정보

1. 이전 화면
2. 플레이리스트 수록곡 보기

-플레이리스트 번호 입력

-수록곡 목록

1) 노래 틀기

-듣고 싶은 노래 입력

-재생

2) 플레이리스트 수정하기(추가, 삭제, 모든 곡 삭제)

3) 이전 화면

* + 노래 듣기 (온라인 play)

-모든 노래 목록

1. 이전 화면
2. 듣고 싶은 노래 입력 (없는 노래일 경우 경고)

-재생

* + 유료 음원 구매 (구매하면 오프라인으로 유료 음악 감상 가능)

-모든 유료 음원 목록

* + - * 1. 이전 화면
        2. 사고 싶은 음원 입력

-구매

* + Top 10 차트

-총 재생 수 순서로 1~10위 음원 목록

1. 이전 화면
2. 듣고 싶은 노래 입력
   * 유저 정보 수정
     + 전화번호 수정

-기존 전화번호 출력

* + - 1. 변경 안함
      2. 새 번호
    - 계좌번호 수정

-기존 계좌번호 출력

* + - 1. 변경 안함
      2. 새 번호

1. **관리자 메뉴**

* 관리자 아이디 입력
  + 이전 화면
  + 관리자 정보
  + 관리하는 음원

-관리 음원 목록

* + 관리하는 유저

-관리 유저 목록

1. 이전 화면
2. 듣고 싶은 노래 입력

-재생

* + 음원 등록

-음원을 등록하시겠습니까? (Y/N)

-새 음원 정보 입력

* + 음원 삭제

-음원을 삭제하시겠습니까? (Y/N)

-삭제할 음원 번호 입력 (없는 노래일 경우 경고)

* + Top 10 차트

-총 재생 수 순서로 1~10위 음원 목록

1. 이전 화면
2. 듣고 싶은 노래 입력
   * 관리자 정보 수정

-기존 전화번호 출력

* + - 1. 변경 안함
      2. 새 번호

**JAVA 코드 및 해설:**

2: Connection, Driver 등을 사용하기 위해 java.sql을 import해준다.

6: MariaDB 연결 드라이버

12: 드라이버를 로드한다. 접속 성공시 자동으로 객체가 생성되고 DriverManager에 등록된다.

13~16: 호스트주소, 사용할 계정, 비밀번호로 getConnection

접속실패시 에러 경고

Main 문: 구조체 생성 (Database 생성자 호출)

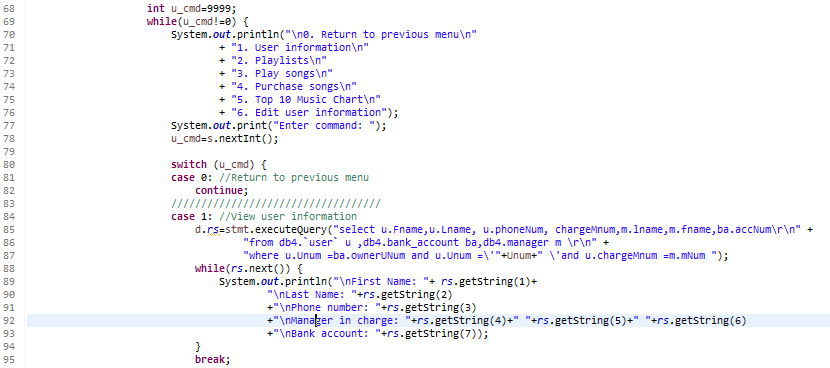
메뉴를 출력하고 cmd에 숫자를 받아 명령을 받는다Switch를 이용해 입력 받은 cmd에 따라 명령을 수행한다.

**0일 때:** exit – 프로그램 종료

Connection과 statement를 close해준 뒤 프로그램을 종료시킨다.

**1일 때:** 유저 메뉴

유저 아이디를 입력 받고 SELECT EXISTS와 as success를 이용해 아이디 존재 여부를 0과 1로 알아 낸다.

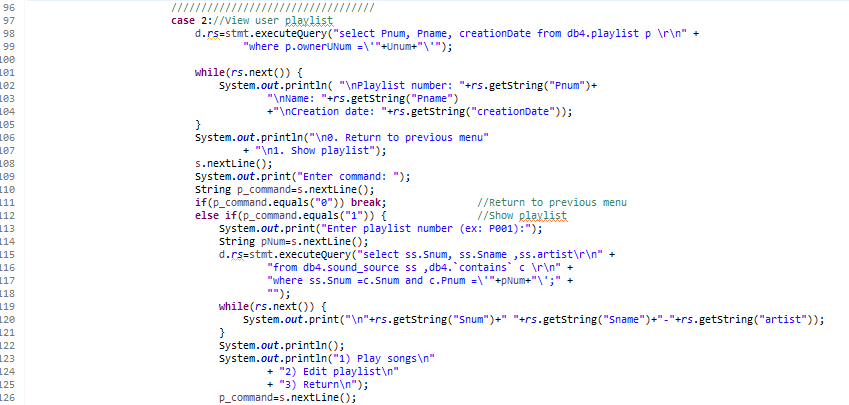
존재하지 않는 아이디라면 에러 메세지를 출력한다.

유저 메뉴에서 명령할 때 사용할 변수 u\_cmd 선언 (68)

유저 메뉴들 출력 (70)

**0일 때:** 이전 화면 (switch가 while문에 있기 때문에 continue하면 0. Exit… 부분으로 돌아감) (81)

**1일 때:** 유저 정보 (SELECT를 이용해 Unum이 입력한 아이디인 user column 데이터를 전부 출력함) (84)

****

**2일 때:** 플레이리스트

SELECT을 이용해 Playlist 테이블에서 소유자 번호(Unum)이 일치하는 모든 플레이리스트의 정보들을 출력함 (98)

p\_command 입력: (109)

* 0일 때 이전 화면 (111)
* 1일 때 플레이리스트 수록곡 보기 (112)

(사용자가 자기가 가지고 있는 플레이리스트들 중 하나를 선택해서 볼 수 있는 기능)

pNum 입력: 원하는 플레이리스트의 번호를 pNum에 입력 받고 (114)

SELECT를 이용해 입력 받은 pNum과 contains테이블에 수록된 음원 번호들을 찾고

sound\_source 테이블에서 해당 음원번호들의 정보를 출력한다. (115)

p\_command 입력: (126)

1일 때: 특정 음원 한 곡 재생

2일 때: 플레이리스트 편집

3일 때: 이전 화면

1일 때: 원하는 음원 번호 입력(129)

UPDATE를 이용해 plays 테이블에서 해당 유저가 그 노래를 들은 횟수 1 더해주기(131)

2일 때: 리스트 편집

-A 입력 시: 음원 추가

-D 입력 시: 음원 삭제

-DP 입력 시: 플레이리스트 속 모든 음원 삭제

3일 때: 이전 화면

다 아닐 때: “에러 메시지: 틀린 명령어 입니다. 이전 화면으로 돌아갑니다…”

A: 추가할 노래 번호를 입력하고 INSERT를 이용해 contains에 플레이리스트번호와 노래 번호를 추가함 (137)

D: 삭제할 노래 번호를 입력하고 DELETE을 이용해 contains 테이블에서 해당 라인을 삭제함 (146)

DP: 정말 삭제를 원하는지 다시 한번 확인 (Y/N) (154)

Y인 경우 DELETE을 이용해 해당 플레이 리스트에 해당되는 contains 데이터와 플레이리스트 데이터를 삭제해준다. (153)

N인 경우 따로 조건문을 넣지 않았다 (while문 시작으로 돌아감)

**3일 때:** 노래 듣기

SELECT를 이용해 sound\_source 테이블에서 모든 음원의 번호, 이름, 가수를 출력한다(178)

X입력시: 이전 화면

음원 번호 입력 시: 해당 음원 재생

존재하는 음원인지 SELECT EXISTS확인하고 존재하지 않을 경우 경고문

존재하는 음원인 경우 UPDATE를 이용해 현재 유저의 해당 음원 재생 횟수 +1

**4일 때:** 음원 구매

SELECT을 이용해 모든 유료 음원의, 번호, 이름, 가수, 가격을 출력한다 (206)

X입력시: 이전 화면

음원 번호 입력 시: 해당 음원 구매

INSERT를 이용해 purchases 클래스에 음원과 유저 이름 추가

**5일 때:** top10 음원 리스트

Snum으로 plays와 soundSource를 JOIN해준 뒤 SELECT로 번호, 이름, 가수, SUM(재생횟수)를 찾아낸다. 그리고 음원 별로 총 재생횟수를 order by desc로 내림차순으로 출력한다.(226)’

While문이 10번만 돌아가서 재생횟수 상위 10등까지만 출력한다.(234)

-X입력 시: 이전 화면(244)

-음원 번호 입력 시: 해당 음원 재생 (UPDATE를 이용해 재생 횟수+1)(246)

**6일 때:** 유저 정보 수정

1입력 시: 전화번호 수정

SELECT를 이용해 해당 유저 기존 전화번호 출력

-X입력시: 이전 화면 (269)

-새로운 번호 입력 시: UPDATE를 이용해 전화번호 수정 (271)

2입력 시: 계좌번호 수정

SELECT를 이용해 해당 유저 기존 계좌번호 출력

-X입력시: 이전 화면 (286)

-새로운 번호 입력 시: UPDATE를 이용해 계좌번호 수정 (288)

**2일 때:** 관리자 메뉴

매니저 아이디를 입력 받고 SELECT EXISTS와 as success를 이용해 아이디 존재 여부를 0과 1로 알아 낸다.

존재하지 않는 아이디라면 에러 메세지를 출력한다. (301)

메뉴 출력 후 명령 입력: (313)

**0일 때:** 이전 화면 (324)

**1일 때:** 매니저 정보 (SELECT를 이용해 Mnum이 입력한 아이디와 같은 매니저 데이터를 전부 출력함) (328)

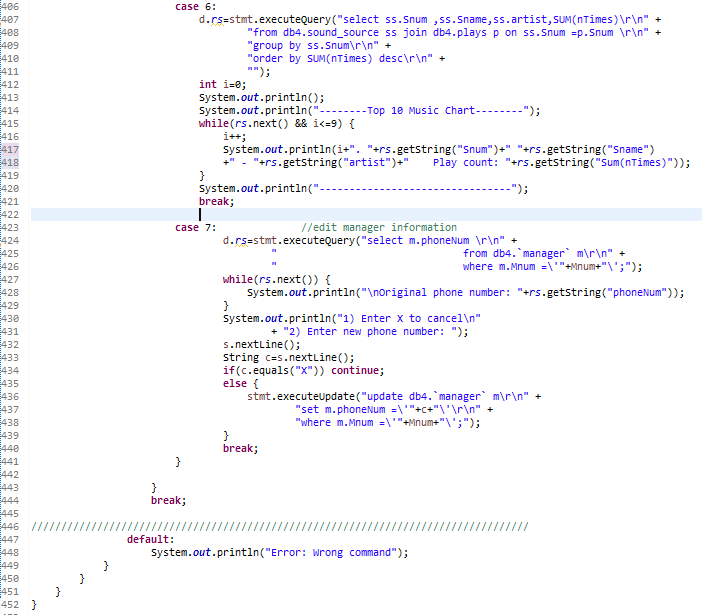
**2일 때:** 관리 중인 음원 정보 출력. SELECT를 이용해 sound\_source에서 담당관리자 번호가 로그인한 관리자 번호와 같은 음원 정보를 출력한다. (339)

**3일 때:** 관리 중인 유저 정보 출력. SELECT를 이용해 user에서 담당관리자 번호가 로그인한 관리자 번호와 같은 유저 정보를 출력한다.

**4일 때:** 새로운 음원 등록. 음원을 등록할 것인지 다시 확인하고 (Y/N) 음원 번호, 이름, 가수, 가격, 길이, 등록날짜를 순서대로 입력한다. 입력 받은 음원 정보를 INSERT를 이용해 sound\_source에 추가한다.

**5일 때:** 음원 삭제. 음원을 삭제할 것인지 다시 확인하고 (Y/N) 음원 번호를 입력 받는다. (384)

* SELECT EXIST를 이용해 존재하는 음원인지 확인한다. 존재하지 않는다면 에러 메시지를 출력한다. (391)
* DELETE를 이용해 음원번호를 FK, PK로 사용하는 purchase에서 해당 음원을 삭제한다. (399)
* DELETE를 이용해 음원번호를 FK, PK로 사용하는 plays에서 해당 음원을 삭제한다.(400)
* DELETE를 이용해 음원번호를 FK, PK로 사용하는 contains에서 해당 음원을 삭제한다. (401)
* DELETE를 이용해 soundsource에서 해당 음원을 삭제한다. (402)

**6일 때:** 유저 메뉴에서 실행된 top 10 차트 코드와 같은 코드이다.(407)

(SELECT, order by ~ DESC, group by 그리고 while문을 이용한 재생 수 1~10위 음원 출력, 음악 재생 기능은 없음)

**7일 때:** 매니저 정보 수정.

매니저는 수정할 수 있는 정보가 전화번호 밖에 없다.

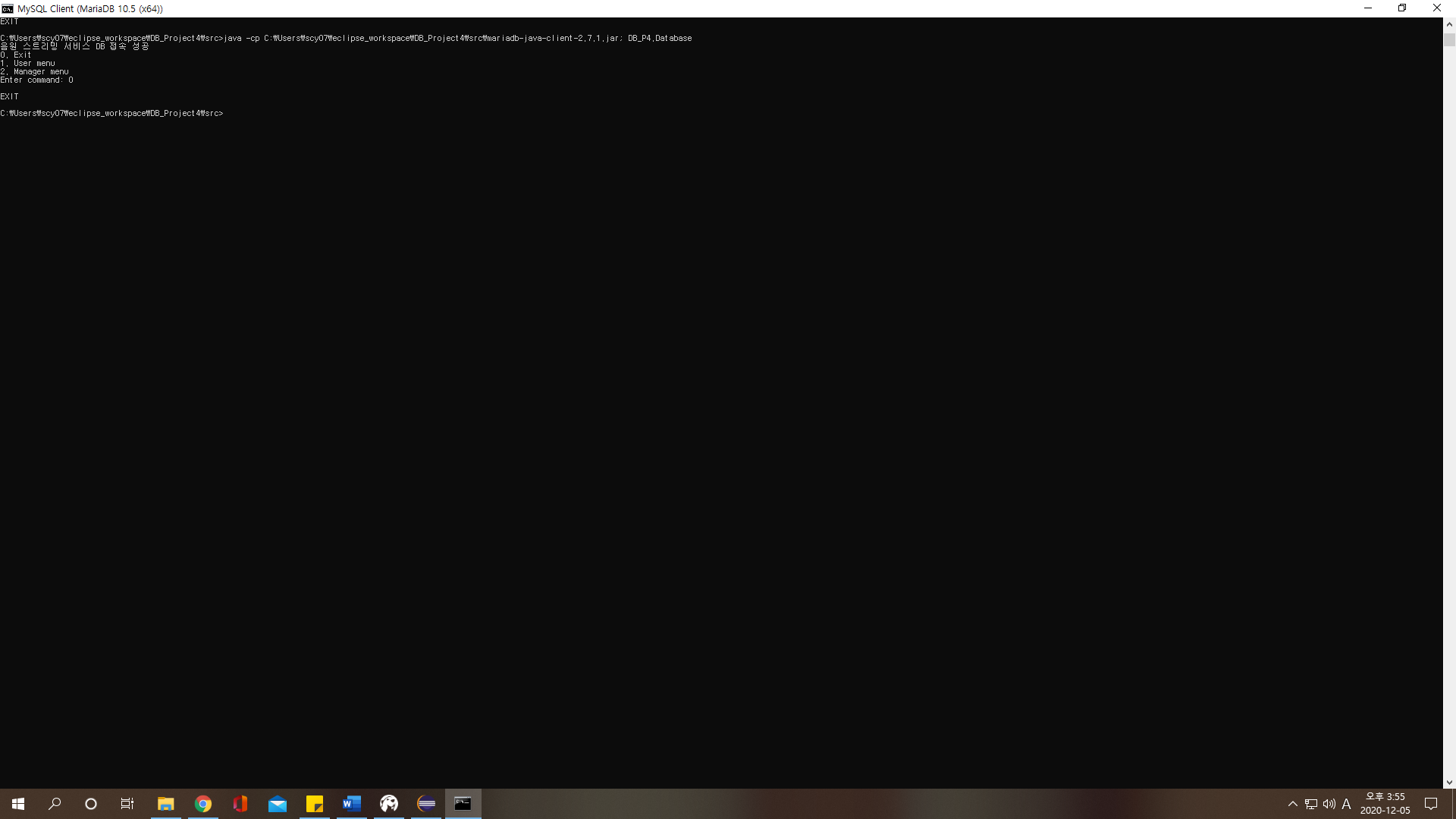
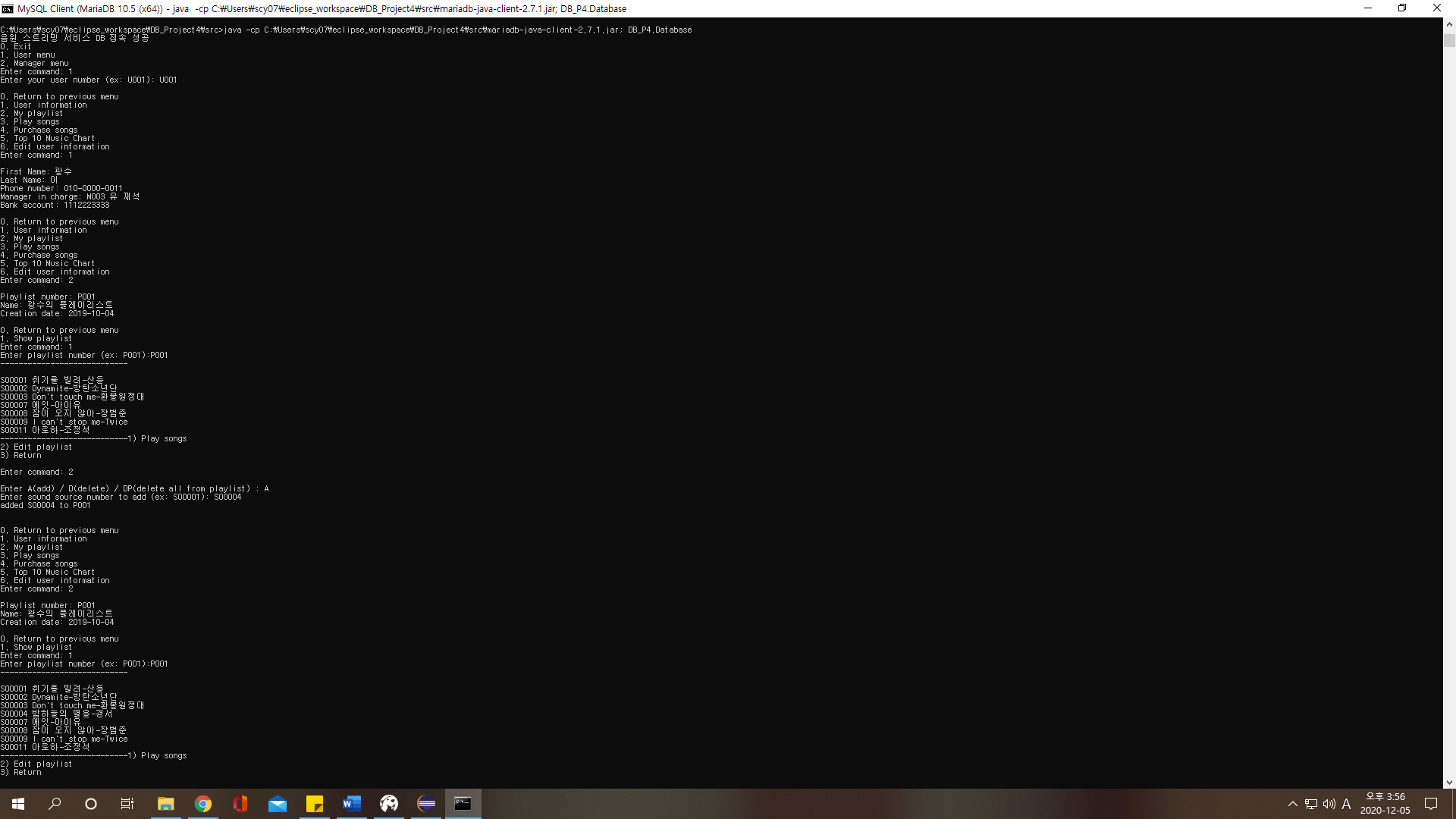
* SELECT를 이용해 매니저의 기존 전화번호를 출력한다.(424)
* X 입력 시: 이전화면 (434)
* 전화번호 입력 시: UPDATE를 이용해 해당 매니저의 전화번호 수정 (436)

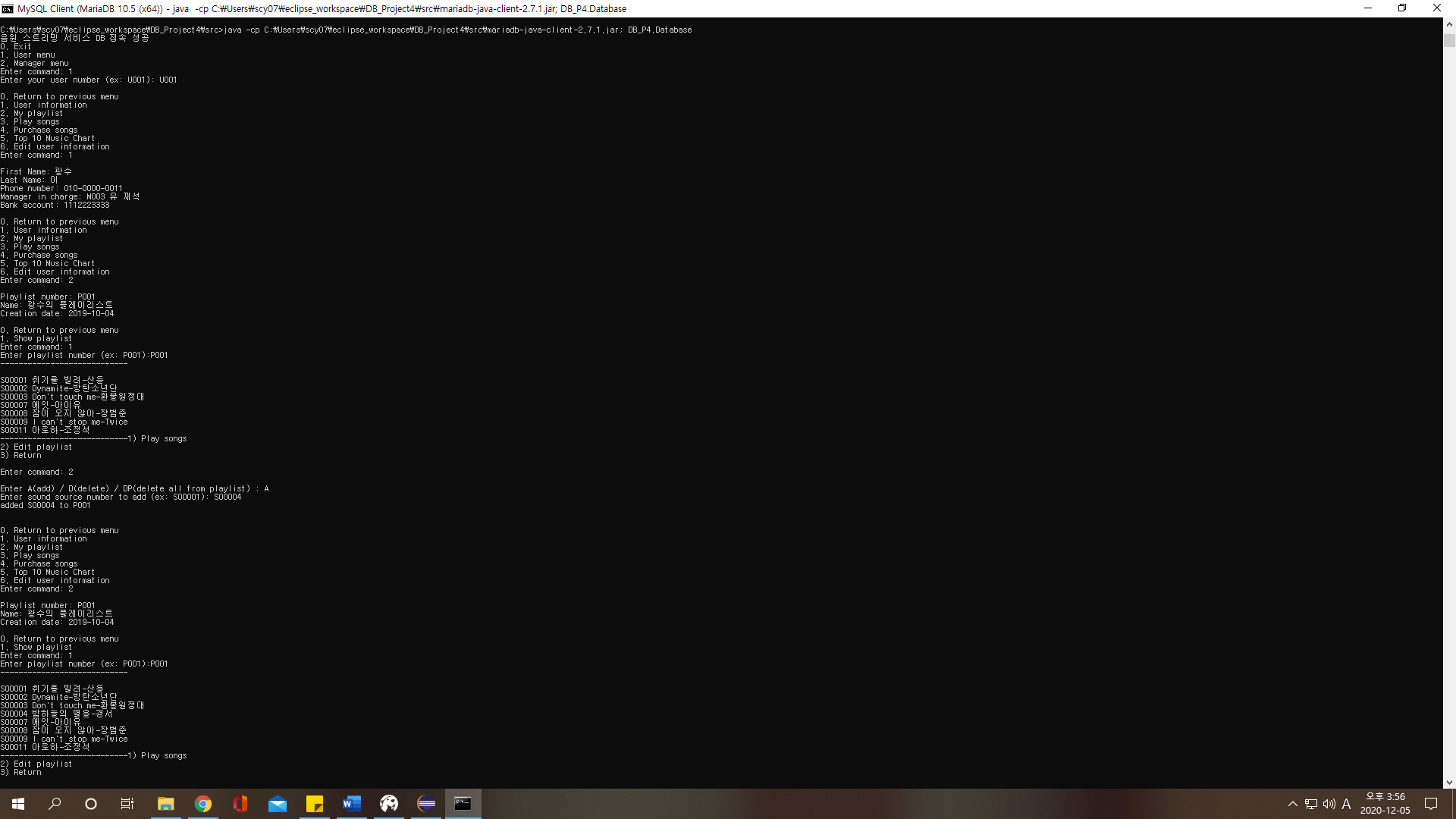
0,1,2가 모두 아닐 때 에러메세지 출력

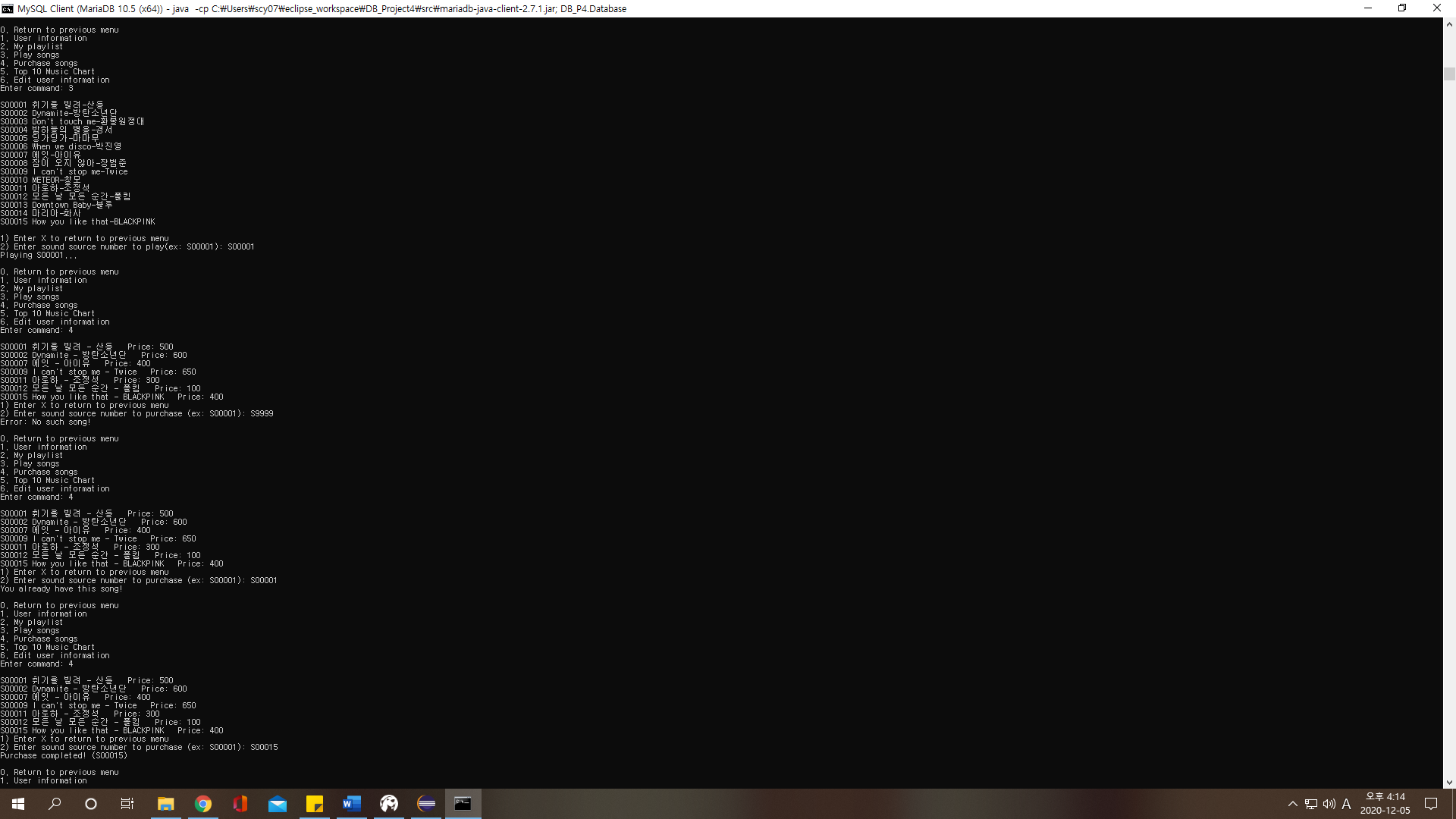
\*\*실제 제출 프로그램에는 조금의 수정 사항이 있으나 출력 형식이나 크지 않은 변화이므로 코드 해설에서 따로 수정하지 않았습니다.

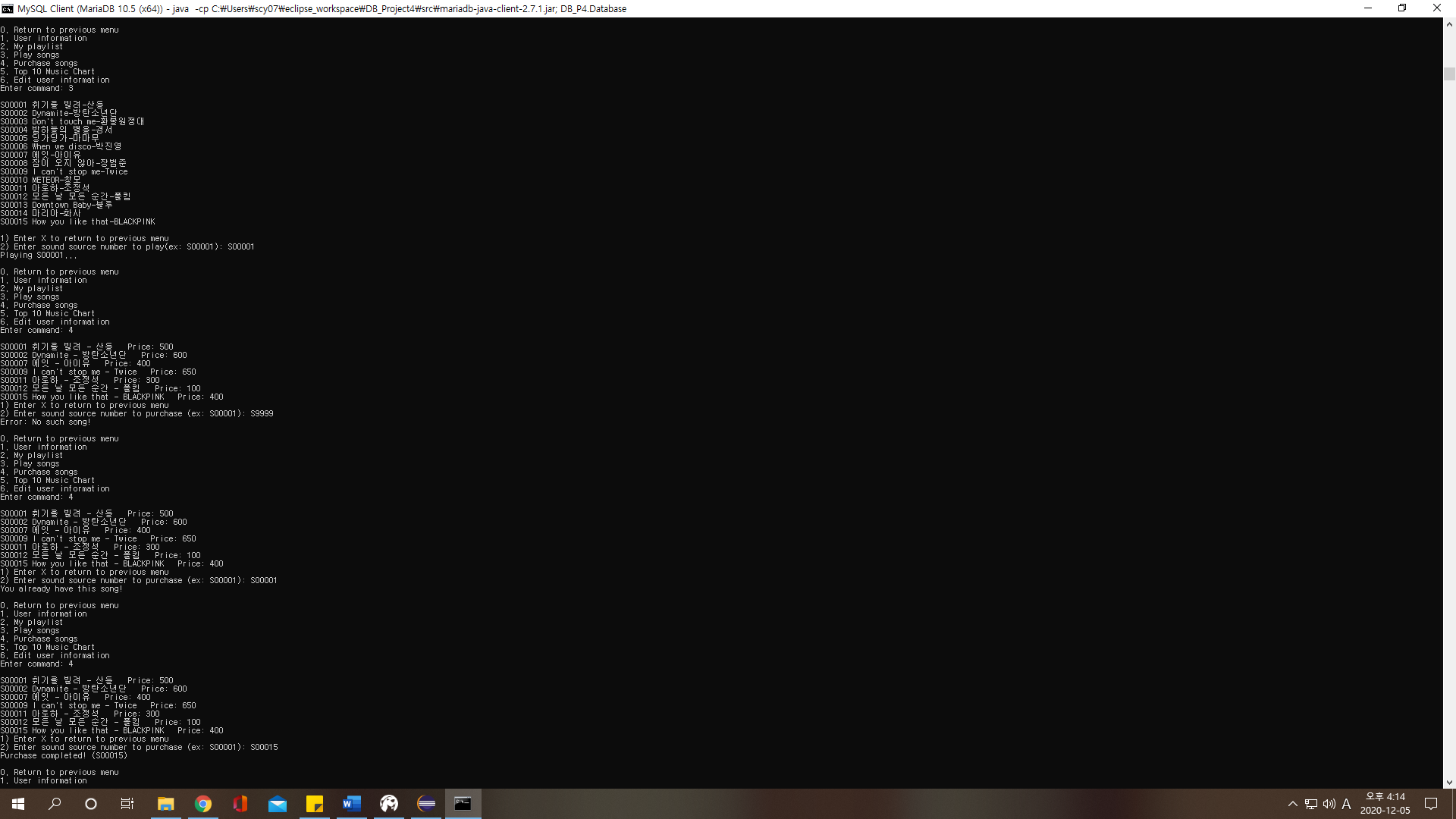
* 플레이리스트에서 노래를 삭제할 때 없는 노래일 경우 경고문 출력 (SELECT EXIST)
* 잘못된 명령어를 입력한 경우 에러 메시지 출력 (2 군데 누락)

**CMD창 실행 캡쳐**

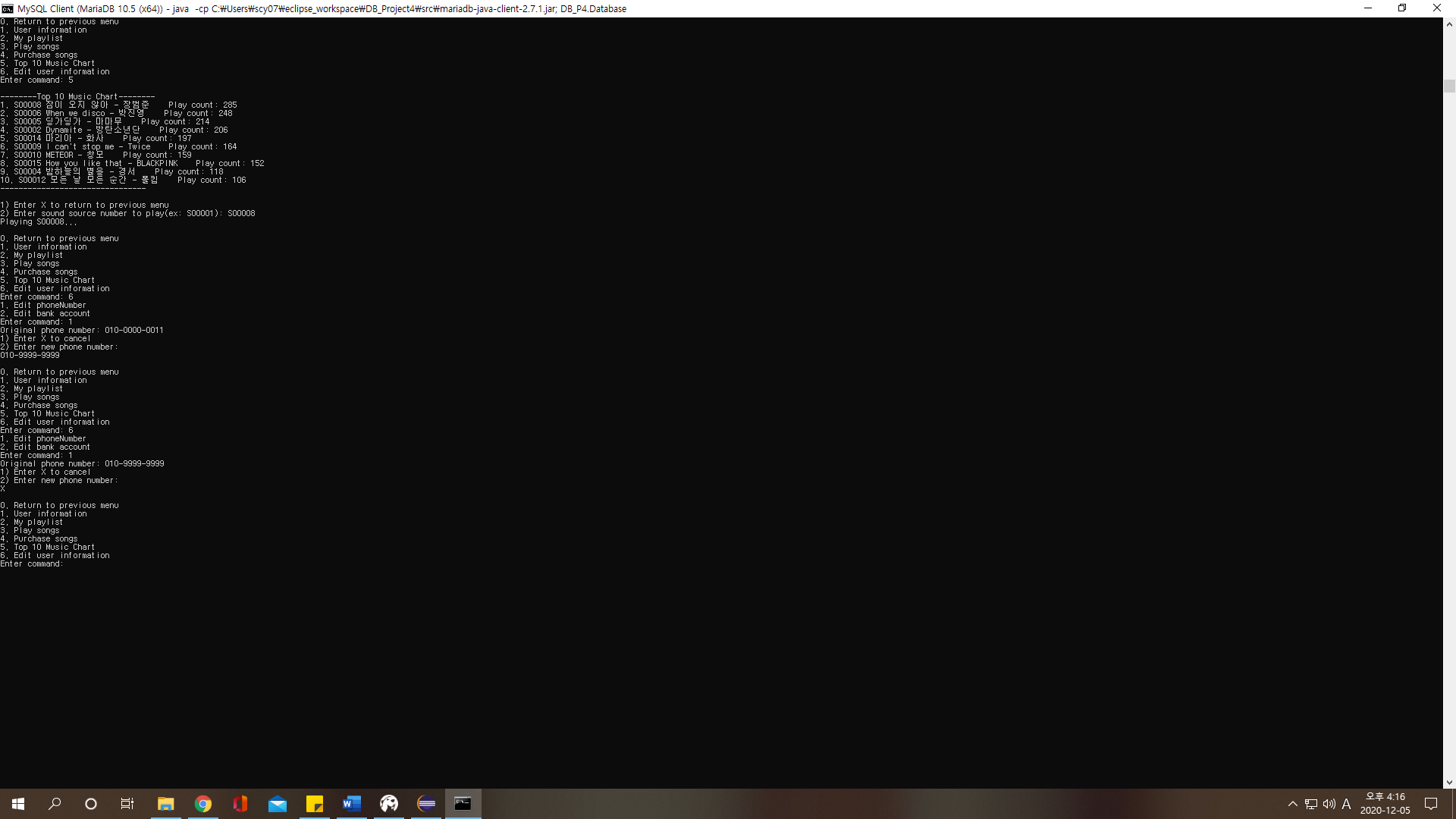
1. **프로그램 종료**
2. **유저메뉴**
   1. ****유저 정보
   2. 플레이리스트



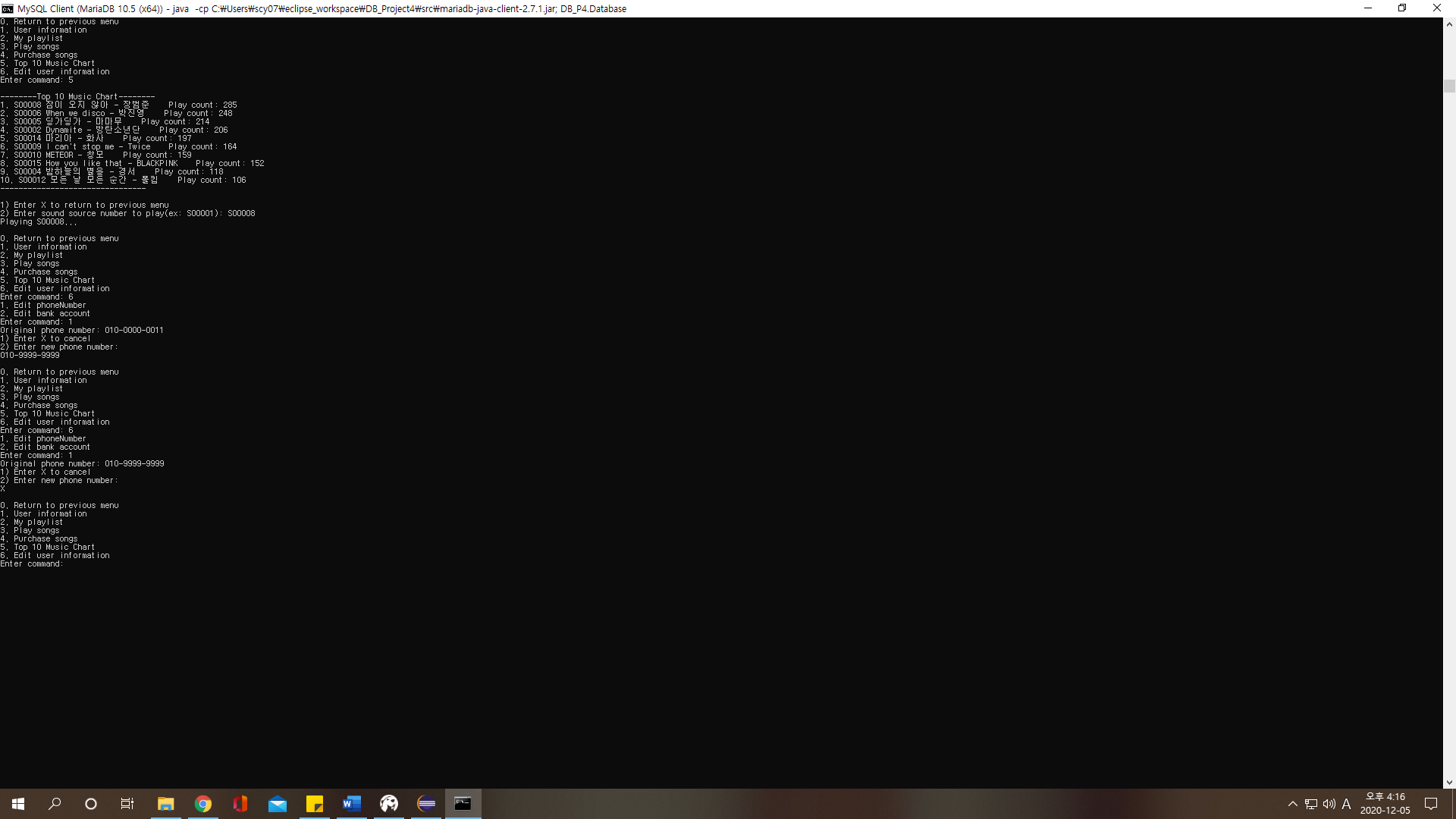
* 플레이리스트
* 노래 추가 (S00004)
* 추가된 모습
  1. 노래 재생
  2. 노래 구매



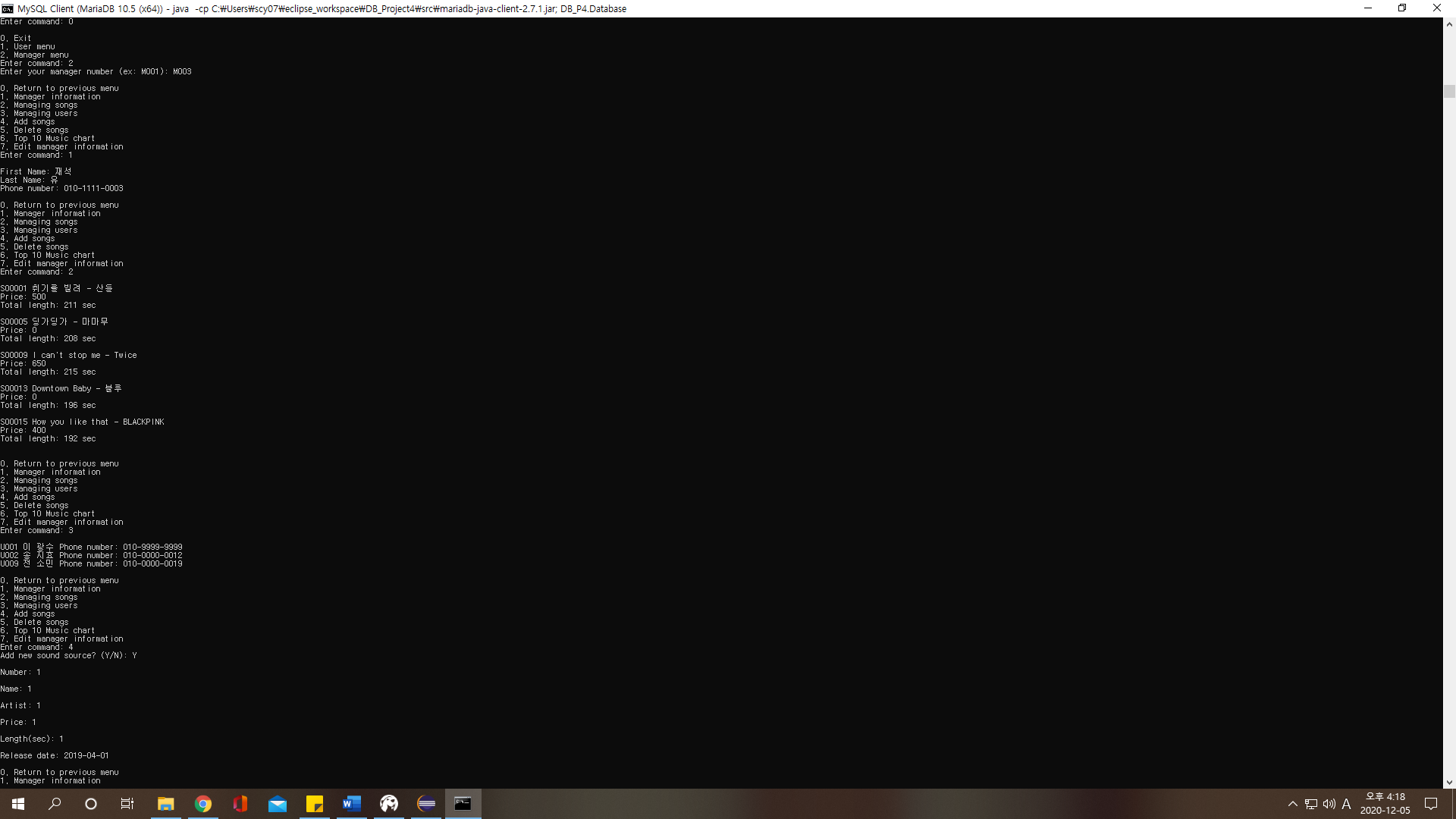
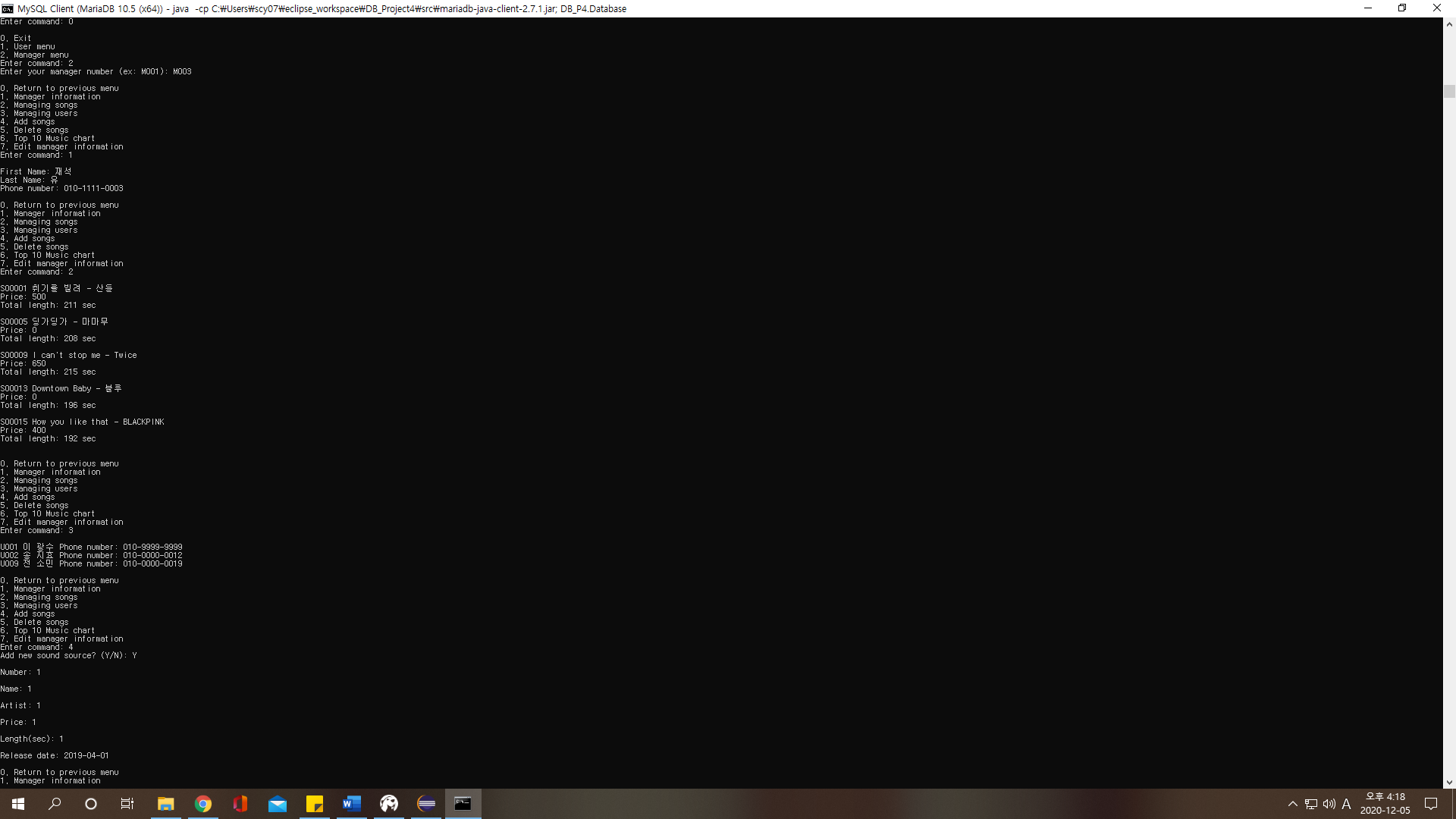
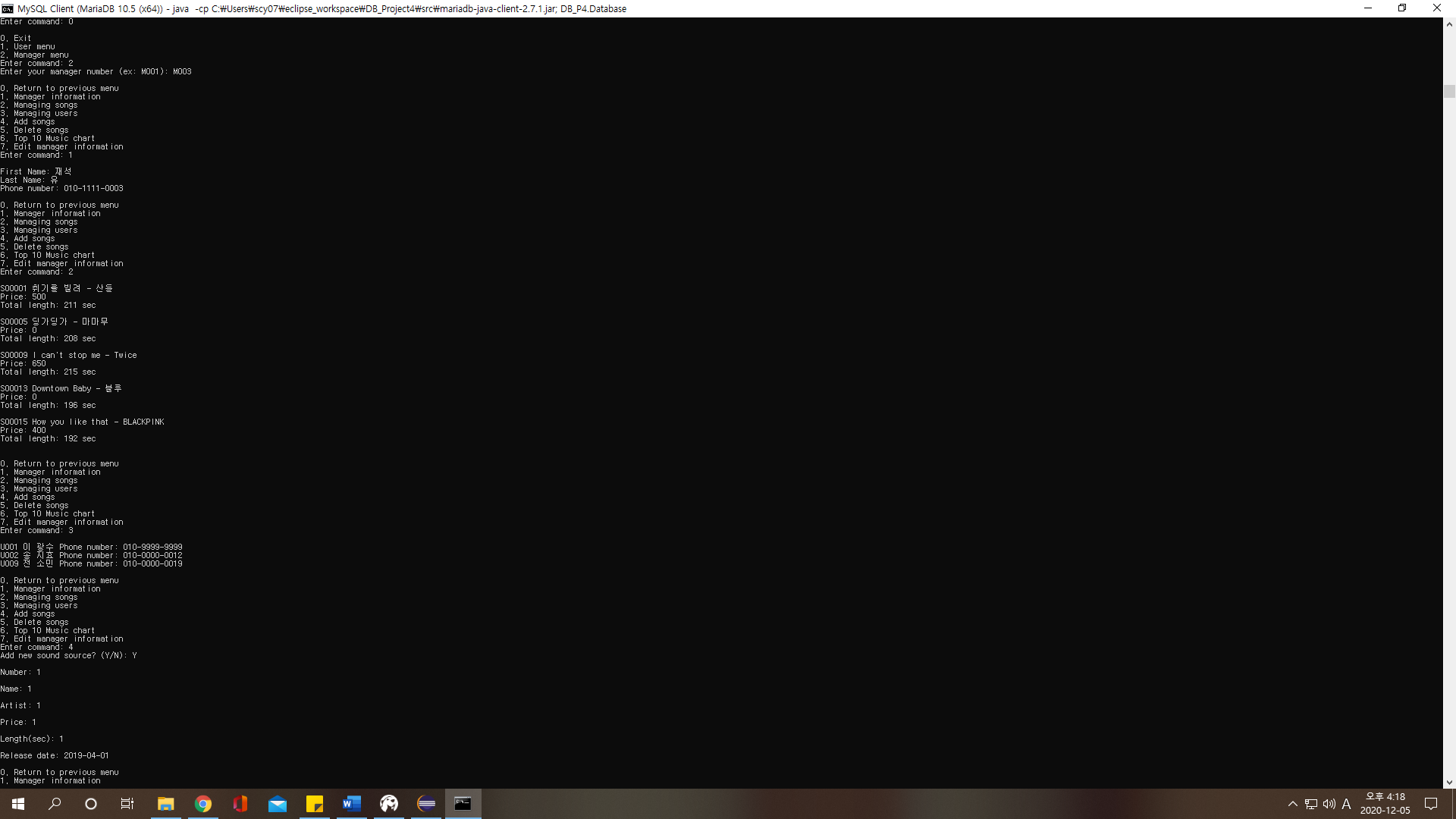
* 없는 노래일 경우
* 이미 구매한 노래일 경우
* 구매 성공
  1. TOP 10 차트

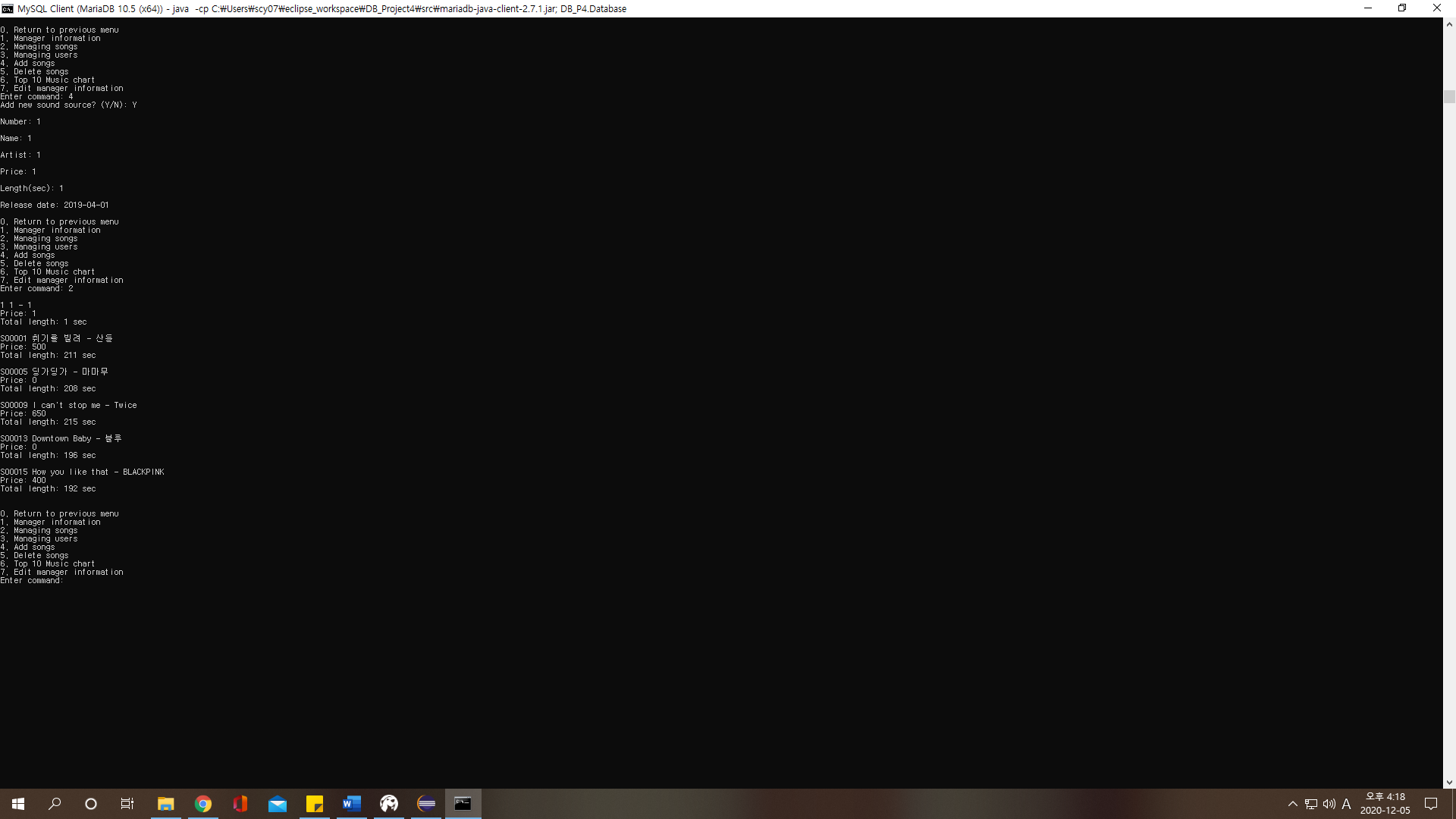


* 노래 재생
  1. 유저 정보 수정

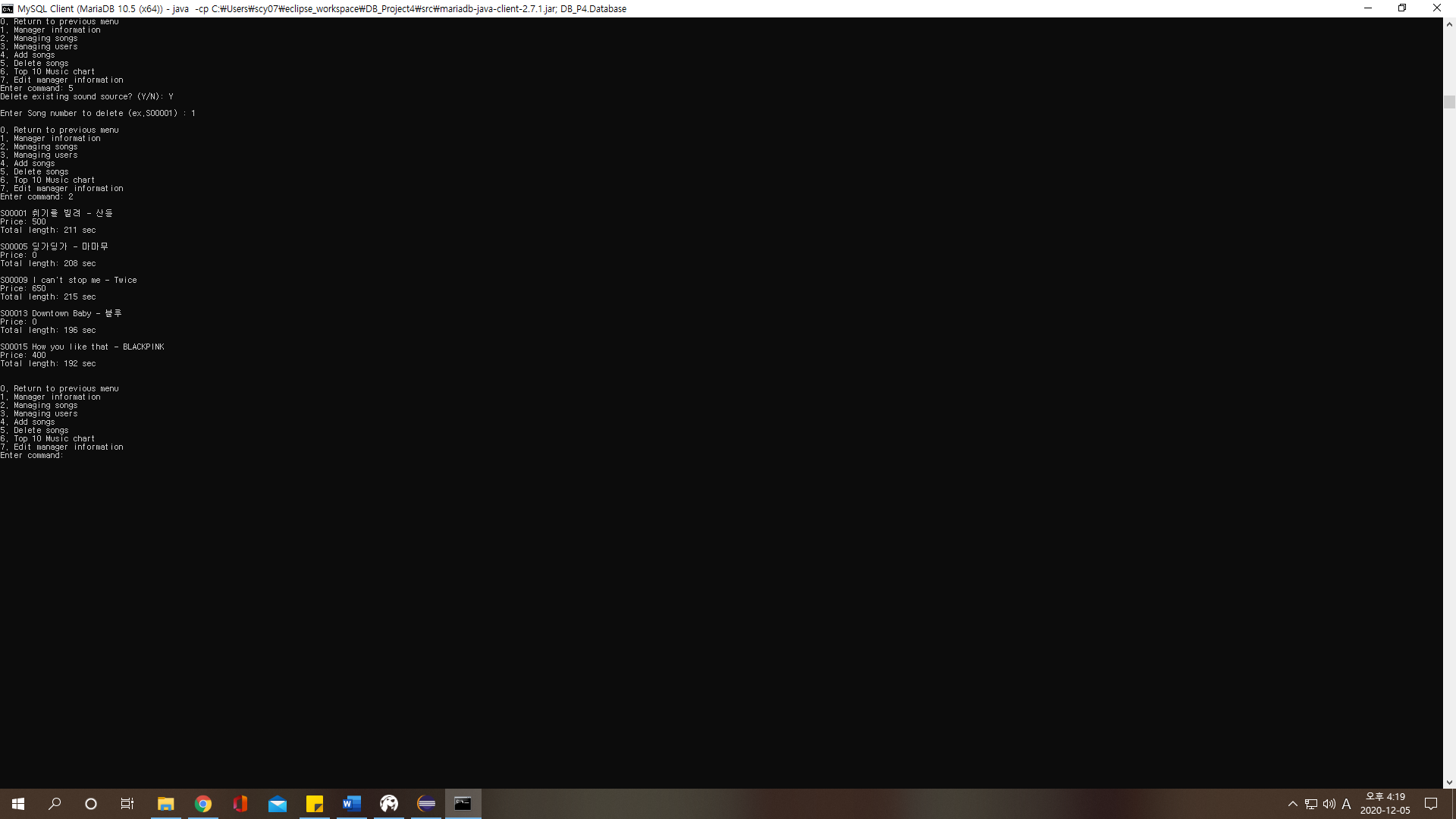


* 기존 번호
* 새 번호
* 변경된 모습

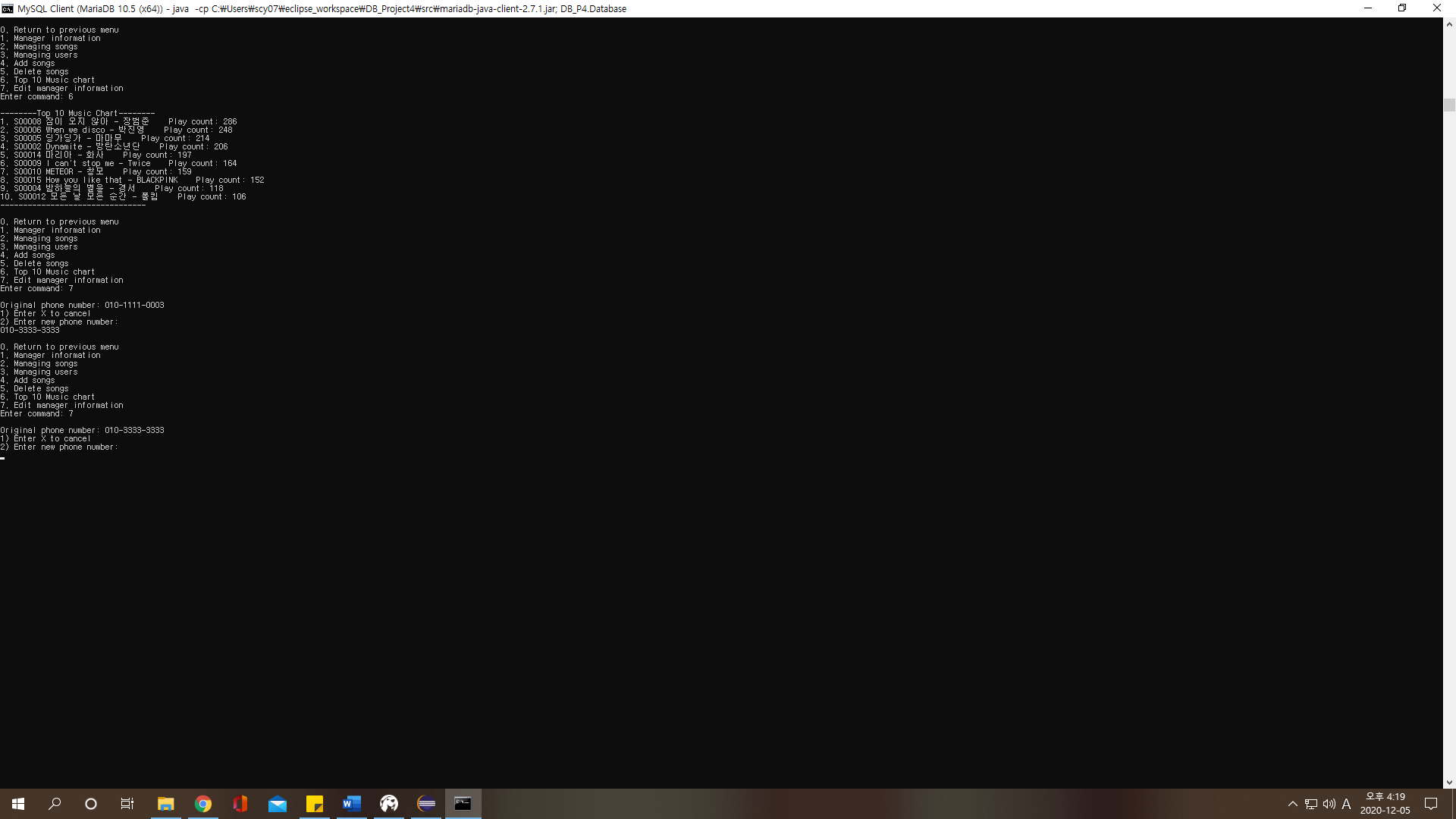
1. **매니저 메뉴**
   1. 매니저 정보
   2. 담당하는 음원들 정보
   3. 담당하는 유저들 정보
   4. 노래 추가



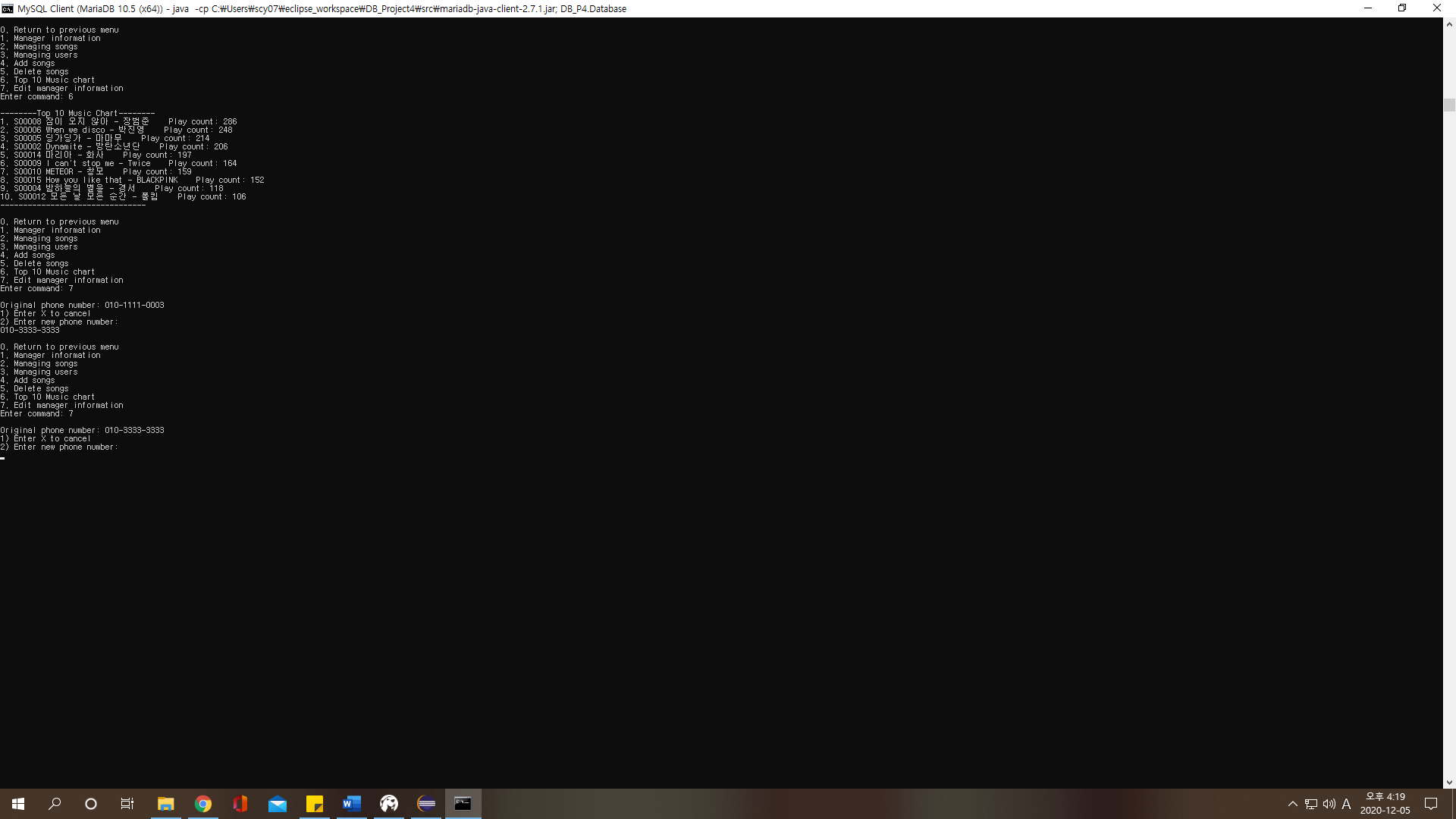
* 새 음원 정보 입력
* 추가된 모습
  1. 노래삭제



* 삭제 번호 입력
* 삭제된 모습
  1. TOP 10 차트



* 1. 매니저 정보 수정



* 새 번호 입력
* 변경된 모습

**Project 4 제출 목록(Git 제출):**

1. DB\_P4 자바 프로그램 (Class명 Database)
2. P4 보고서
3. “db4” 데이터스키마(샘플 포함) dump file