

나 . 데이터 입출력 요건에 맞게 테이블생성 SQL문장을 작성하고 데이터베이스 시스템에 테이블을 생성하시오.

- 오디션지원자정보 테이블 명세서 (테이블 명 : tbl_artist_201905))

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	artist_id	참가번호	char	4	NOT NULL	Primary Key
2	artist_name	이름	varchar2	20		
3	artist_gender	성별	char	1		남자 : M 여자 : F
4	artist_birth	생년월일	char	8		
5	talent	특기	char	1		1 : 댄스 2 : 랩 3: 노래
6	agency	소속사	varrchar2	20		

```
create table tbl_artist_201905 (  
  artist_id char(4) not null,  
  artist_name varchar2(20),  
  artist_gender char(1),  
  artist_birth char(8),  
  talent char(1),  
  agency varchar2(20),  
  constraint artist_pk primary key (artist_id)  
);
```

- 샘플 데이터

순서	참가번호	이름	성별	생년월일	특기	소속사
1	A001	김스타	F	19970101	1	A엔터테인먼트
2	A002	조스타	M	19980201	2	B엔터테인먼트
3	A003	왕스타	M	19990301	3	C엔터테인먼트
4	A004	정스타	M	20000401	1	D엔터테인먼트
5	A005	홍스타	F	20010501	2	E엔터테인먼트

샘플데이터 5개 입력

```
insert into tbl_artist_201905 (artist_id,artist_name,artist_genger,artist_gender,artist_birth,talent,agency)
values ('A001','김스타','F','19970101','1','A엔터테인먼트');
```

- 멘토정보 테이블 (테이블 명 : tbl_mento_201905

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	mento_id	멘토 아이디	char	4	NOT NULL	Primary Key
2	mento_name	멘토 이름	varchar2	20		

- 샘플 데이터

순서	멘토번호	이름
1	J001	함멘토
2	J002	박멘토
3	J003	장멘토

```
create table tbl_mento_201905 (  
  mento_id char(4) not null,  
  mento_name varchar2(20),  
  constraint mento_pk primary key (mento_id)  
);
```

```
insert into tbl_mento_201905 values ('J001','함멘토');
```

컬럼이 만들어진 순서대로 데이터를 입력한다면 컬럼명을 생략할 수 있다

- 심사위원의 채점결과 정보 테이블(테이블명 : tbl_point_201905)

순 서	컬럼ID	컬럼명	형태	길 이	NULL	비고
1	serial_no	채점번호	number	8	NOT NULL	Primary Key
4	artist_id	참가번호	char	4		FK
5	mento_id	멘토아이디	char	4		FK
6	point	점수	number	3		

```
create table tbl_point_201905 (  
  serial_no number(8) not null,  
  artist_id char(4),  
  mento_id char(4),  
  point number(3),  
  constraint point_pk primary key (serial_no),  
  constraint point_fk foreign key (artist_id) references tbl_artist_201905 (artist_id),  
  constraint mento_fk foreign key (mento_id) references tbl_mento_201905 (mento_id)  
);
```

- 샘플 데이터

순서	채점번호	참가번호	멘토아이디	점수
1	2019001	A001	J001	78
2	2019002	A001	J002	76
3	2019003	A001	J003	70
4	2019004	A002	J001	80
5	2019005	A002	J002	72
6	2019006	A002	J003	78
7	2019007	A003	J001	90
8	2019008	A003	J002	92
9	2019009	A003	J003	88
10	2019010	A004	J001	96
11	2019011	A004	J002	90
12	2019012	A004	J003	98
13	2019013	A005	J001	88
14	2019014	A005	J002	86
15	2019015	A005	J003	86

샘플데이터 입력

테이블을 생성후 자바언어와 연동을 하여 CRUD 작업을 처리하기 위해서는 어떤 설계도가 필요한가?

- 패키지이름 : javaJDBCotion
- 싱글톤 패턴으로 데이터베이스와 연동하는 설계도 : DBManager
- DBManager 클래스를 테스트할 설계도 : DBTest

▼ javaJDBCotion
> DBManager.java

```
public class DBManager {  
  
    private static DBManager instance = new DBManager();  
    private DBManager() {}  
  
    public static DBManager getInstance() {  
        return instance;  
    }  
  
    public Connection getDBmanager() {  
        Connection conn = null;  
        String driver = "oracle.jdbc.driver.OracleDriver";  
        String url = "jdbc:oracle:thin:@localhost:1521:xe";  
        String id="jsl18";  
        String pw="1234";  
        try {  
            Class.forName(driver);  
            conn = DriverManager.getConnection(url,id,pw);  
        }catch(Exception e) {  
            e.printStackTrace();  
        }  
        return conn;  
    }  
}
```

문제1> DB Test 하시오

▼ javaJDBCodition
> DBManager.java
> DbTest.java

```
public class DbTest {  
  
    public static void main(String[] args) {  
  
        Connection conn = DBManager.getInstance().getDBmanager();  
        if(conn != null) {  
            System.out.println("접속완료!");  
        }  
    }  
}
```

문제2> 작업선택 메뉴에서 1번을 선택하면 아래와 같이 데이터를 입력받아 DB 테이블에 저장하시오

작업선택 : [1]오디션등록[2]참가자 목록 조회[3]멘토 점수 목록 조회[4]참가자등수조회[5]종료 1
오디션 등록
참가번호 : A005
참가자명 : 홍스타
생년월일 : 20010501
성별 : [남자:M,여자:F] F
특기 : [1]댄스[2]랩[3]노래 2
소속사 : E엔터테인먼트
등록하시겠습니까?[y/n]y

등록여부에서 대문자 Y 또는 소문자 y 가 입력되면 테이블에 저장후 작업선택 메뉴로 돌아 간다
N 또는 n이 입력되면 작업선택 메뉴로 돌아 간다

등록처리 순서

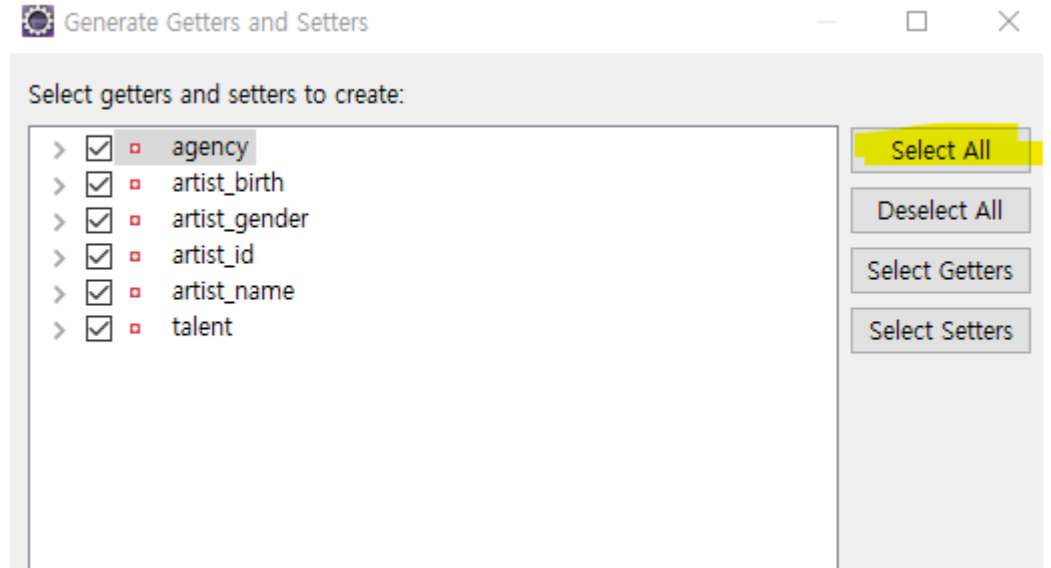
- 1) 어떤 클래스(설계도)가 필요할까?
 - 1) 메뉴화면 설계도 – OditionMain
 - 2) 객체로 데이터 처리를 위한 설계도 – OditionVo
 - 3) 데이터 처리(CRUD)위한 설계도 – OditionDao

테이블에 등록 작업을 하기 위한 절차는 어떻게 진행하면 될까?

- 1) 키보드로 부터 처리하고자 데이터를 입력받아 Vo 객체에 저장후 Dao 메서드로 보낼 설계를 만든다 – OditionVo
- 2) Vo 설계를 만들때는 테이블 만들때 컬럼이름과 타입을 보고 컬럼이름과 똑같이 멤버변수를 만든다
- 3) 멤버변수의 getter 메서드와 setter 메서드를 만든다 – Source > Generate Getter and Setter

▼ javaJDBCodition
> DBManager.java
> DbTest.java
> OditionDao.java
> OditionVo.java

```
public class OditionVo {  
  
    private String artist_id;  
    private String artist_name;  
    private String artist_gender;  
    private String artist_birth;  
    private String talent;  
    private String agency;  
  
    //getter, setter 메서드 만들기  
  
}
```



1) 테이블과 연동하여 CRUD 작업을 처리하기 위한 설계를 만든다 – OditionDao

```
▼ javaJDBCodition
  > DBManager.java
  > DbTest.java
  > OditionDao.java
```

2) 테이블에 저장하기 위해 Vo 객체에 저장된 데이터를 insert 하기 위한 메서드를 만든다

```
public class OditionDao {

    public void setOdiInsert(OditionVo vo) {
        Connection conn = null;
        PreparedStatement pstmt=null;
        String sql = "insert into tbl_artist_201905 (artist_id,artist_name,artist_gender,artist_birth,talent,agency) "
            + " values (?,?,?,?,?,?)";
        try {
            conn = DBManager.getInstance().getDBmanager();
            pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, vo.getArtist_id()); //1번째 물음표에 vo.getArtist_id()메서드를 이용하여 artist_id 멤버변수 값을 읽어와 대입한다
            pstmt.setString(2, vo.getArtist_name());
            pstmt.setString(3, vo.getArtist_gender());
            pstmt.setString(3, vo.getArtist_birth());
            pstmt.setString(4, vo.getTalent());
            pstmt.setString(5, vo.getAgency());
            pstmt.executeUpdate();
        }catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

메뉴 만들고 1번 입력 됐을 때 테이블에 저장 알고리즘 코딩하기

```
✓ javaJDBCodition
> DBManager.java
> DbTest.java
> OditionDao.java
> OditionMain.java
> OditionVo.java
```

```
public class OditionMain {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        OditionDao dao = new OditionDao();
        while(true) {
            System.out.print("메뉴선택 : [1]오디션등록[2]참가자목록조회[3]멘토점수목록조회[4]참가지등수조회[5]종료");
            int sel=sc.nextInt();
            if(sel==1) {
                System.out.print("참가번호 : ");
                String artist_id = sc.next();
                System.out.print("이름 : ");
                String artist_name = sc.next();
                System.out.print("성별 : [남자:M, 여자:F] ");
                String artist_gender = sc.next();
                System.out.print("생년월일 : ");
                String artist_birth = sc.next();
                System.out.print("특기 : [1:댄스,2:랩,3:노래] ");
                String talent = sc.next();
                System.out.print("소속사 : ");
                String agency = sc.next();

                OditionVo vo = new OditionVo();

                vo.setArtist_id(artist_id);
                vo.setArtist_name(artist_name);
                vo.setArtist_gender(artist_gender);
                vo.setArtist_birth(artist_birth);
                vo.setTalent(talent);
                vo.setAgency(agency);

                dao.setOdiInsert(vo);
            }
            else if(sel==2) {
            }else if(sel==3) {
            }else if(sel==4) {
            }else if(sel==5) {
                break;
            }
        }
    }
}
```

테이블에 저장하기 위한 데이터를
키보드로 부터 입력받아 Vo 객체의
멤버변수에 저장을 시키고
Dao 클래스 의 저장 메서드를 호출하여
Vo 객체를 넘겨준다

문제3> 2번 적업 선택시 아래와 같이 출력하는 프로그램을 완성하시오

메뉴선택 : [1]오디션등록[2]참가자목록조회[3]멘토점수목록조회[4]참가지등수조회[5]종료2

A001	김스타	1997년01월01일	여성	댄스	A엔터테인먼트
A002	조스타	1998년02월01일	남성	노래	B엔터테인먼트
A003	왕스타	1999년03월01일	남성	랩	C엔터테인먼트
A004	정스타	2000년04월01일	남성	댄스	D엔터테인먼트
A005	홍스타	2001년05월01일	여성	노래	E엔터테인먼트
A006	홍길동	2023년11월29일	남성	댄스	null

- ① 생년월일은 yyyy년mm월dd일의 형식으로 출력하시오
- ② 지원자의 성별은 남, 여로 출력하시오
- ③ 특기 코드에 맞는 특기를 출력하시오(1, 댄스 / 2, 랩 / 3, 노래)

풀이

1) SQL Developer CLI 창에서 SQL 문을 실행해 본후 시험지와 같은 결과가 나오는지 확인해 본다

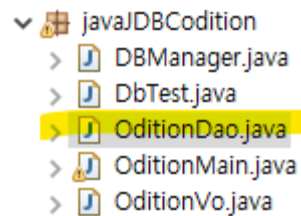
```
select * from tbl_artist_201905;
```

질의 결과 x

SQL | 인출된 모든 행: 6(0,005초)

	ARTIST_ID	ARTIST_NAME	ARTIST_GENDER	ARTIST_BIRTH	TALENT	AGENCY
1	A001	김스타	F	19970101	1	A엔터테인먼트
2	A002	소스타	M	19980201	2	B엔터테인먼트
3	A003	왕스타	M	19990301	3	C엔터테인먼트
4	A004	성스타	M	20000401	1	D엔터테인먼트
5	A005	홍스타	F	20010501	2	E엔터테인먼트

2) OditionDao 클래스를 열고 select를 처리하기 위한 메서드를 추가한다



```
public List<OditionVo> getPartList() {  
    Connection conn = null;  
    PreparedStatement pstmt = null;  
    ResultSet rs = null;  
  
    String sql = "select * from tbl_artist_201905";  
  
    List<OditionVo> list = new ArrayList<OditionVo>();  
  
    try {  
        conn = DBManager.getInstance().getDBmanager();  
        pstmt = conn.prepareStatement(sql);  
        rs = pstmt.executeQuery();  
        while(rs.next()) {  
            OditionVo vo = new OditionVo();  
            vo.setArtist_id(rs.getString("artist_id"));  
            vo.setArtist_name(rs.getString("artist_name"));  
            vo.setArtist_birth(rs.getString("artist_birth"));  
            vo.setArtist_gender(rs.getString("artist_gender"));  
            vo.setTalent(rs.getString("talent"));  
            vo.setAgency(rs.getString("agency"));  
            list.add(vo);  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return list;  
}
```

3) OditionMain 클래스에서 2번 메뉴 선택시 OditionDao 클래스의 getPartList() 메서드를 호출하여 출력하는 프로그램을 완성한다

```
}else if(sel == 2) {  
    List<OditionVo> list = dao.getPartList();  
    for(OditionVo vo : list) {  
        System.out.print(vo.getArtist_id()+"\t");  
        System.out.print(vo.getArtist_name()+"\t");  
        System.out.print(vo.getArtist_birth()+"\t");  
        if(vo.getArtist_gender().equals("M")) {  
            System.out.print("남성"+"\\t");  
        }else {  
            System.out.print("여성"+"\\t");  
        }  
        if(vo.getTalent().equals("1")) {  
            System.out.print("댄스"+"\\t");  
        }else if(vo.getTalent().equals("2")) {  
            System.out.print("노래"+"\\t");  
        }else {  
            System.out.print("랩"+"\\t");  
        }  
  
        System.out.print(vo.getAgency()+"\\t");  
        System.out.println();  
    }  
}
```


문제4> 3번 적업 선택시 아래와 같이 출력하는 프로그램을 완성하시오

메뉴선택 : [1]오디션등록[2]참가자목록조회[3]멘토점수목록조회[4]참가지등수조회[5]종료3

2019001	A001	김스타	1997년01월01일	78	C	함멘토
2019002	A001	김스타	1997년01월01일	76	C	박멘토
2019003	A001	김스타	1997년01월01일	70	C	장멘토
2019004	A002	조스타	1998년02월01일	80	B	함멘토
2019005	A002	조스타	1998년02월01일	72	C	박멘토
2019006	A002	조스타	1998년02월01일	78	C	장멘토
2019007	A003	왕스타	1999년03월01일	92	A	함멘토
2019009	A003	왕스타	1999년03월01일	88	B	장멘토
2019010	A004	정스타	2000년04월01일	96	A	함멘토
2019011	A004	정스타	2000년04월01일	90	A	박멘토
2019012	A004	정스타	2000년04월01일	98	A	장멘토
2019013	A005	홍스타	2001년05월01일	86	B	함멘토
2019015	A005	홍스타	2001년05월01일	86	B	장멘토

- ① 생년월일은 yyyy년mm월dd일의 형식으로 출력
- ② 평점은 점수가 90이상 A, 80이상 B, 70이상 C, 60이상 D, 60미만은 F

1) 먼저 3개의 테이블을 이용하여 조인하는 SQL 문 이므로 아래와 같이 코딩한다

```
select  
from  
where  
order by  
;
```

2) 결과문을 보고 어떤 테이블이 사용되는지 분석한 후 from 뒤에 사용하는 테이블을 코딩한다

```
select  
from tbl_point_201905 p, tbl_artist_201905 a, tbl_mento_201905 m  
where  
order by  
;
```

3) 출력 결과를 보고 어떤 테이블에 어떤 컬럼을 출력할 것인지 결정하고 코딩한다

```
select p.serial_no,a.artist_id,a.artist_name,a.artist_birch,p.point,m.mento_name  
from tbl_point_201905 p, tbl_artist_201905 a, tbl_mento_201905 m  
where  
order by  
;
```

4) 출력 결과를 보고 어떤 테이블에 어떤 컬럼을 이용하여 조인 할 것인지 오라클 조인한다

```
select p.serial_no,a.artist_id,a.artist_name,a.artist_birch,p.point,m.mento_name  
from tbl_point_201905 p, tbl_artist_201905 a, tbl_mento_201905 m  
where p.artist_id = a.artist_id and m.mento_id = p.mento_id  
order by ;
```

5) 정렬 조건을 추가한다

```
select p.serial_no,a.artist_id,a.artist_name,a.artist_birch,p.point,m.mento_name  
from tbl_point_201905 p, tbl_artist_201905 a, tbl_mento_201905 m  
where p.artist_id = a.artist_id and m.mento_id = p.mento_id  
order by p.serial_no;
```

6) 완성된 select 문을 실행시켜서 시험지와 같이 출력 결과 나오는지 확인한다

```
343 select p.serial_no,a.artist_id,a.artist_name,a.artist_birth,p.point,m.mento_name
344 from tbl_point_201905 p, tbl_artist_201905 a, tbl_mento_201905 m
345 where p.artist_id = a.artist_id and m.mento_id = p.mento_id
```

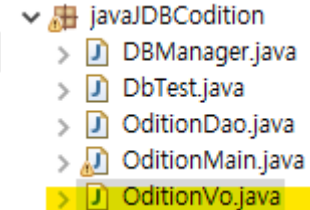
질의 결과 x

SQL | 인출된 모든 행: 13(0.007초)

	SERIAL_NO	ARTIST_ID	ARTIST_NAME	ARTIST_BIRTH	POINT	MENTO_NAME
1	2019001	A001	김스타	19970101	78	함멘노
2	2019002	A001	김스타	19970101	76	박멘노
3	2019003	A001	김스타	19970101	70	상멘노
4	2019004	A002	소스타	19980201	80	함멘노
5	2019005	A002	소스타	19980201	72	박멘노
6	2019006	A002	소스타	19980201	78	상멘노
7	2019007	A003	왕스타	19990301	92	함멘노
8	2019009	A003	왕스타	19990301	88	상멘노
9	2019010	A004	성스타	20000401	96	함멘노
10	2019011	A004	성스타	20000401	90	박멘노
11	2019012	A004	성스타	20000401	98	상멘노
12	2019013	A005	웅스타	20010501	86	함멘노
13	2019015	A005	웅스타	20010501	86	상멘노

결과물을 확인해보고 시험지와 같이 추가된 컬럼을 확인해 보자. 노랑색으로 색칠한 컬럼이 추가 되었다 한 개의 레코드를 VO객체에 담기 위해서 노랑색으로 추가된 컬럼을 VO 클래스에 멤버변수를 추가하고 getter 메서드와 setter 메서드를 추가한다

	SERIAL_NO	ARTIST_ID	ARTIST_NAME	ARTIST_BIRTH	POINT	MENTO_NAME
1	2019001	A001	김스타	19970101	78	함멘노
2	2019002	A001	김스타	19970101	76	박멘노
3	2019003	A001	김스타	19970101	70	상멘노
4	2019004	A002	소스타	19980201	80	함멘노
5	2019005	A002	소스타	19980201	72	박멘노
6	2019006	A002	소스타	19980201	78	상멘노
7	2019007	A003	왕스타	19990301	92	함멘노
8	2019009	A003	왕스타	19990301	88	상멘노
9	2019010	A004	신스타	20000401	96	함멘노
10	2019011	A004	신스타	20000401	90	박멘노
11	2019012	A004	신스타	20000401	98	상멘노
12	2019013	A005	홍스타	20010501	86	함멘노
13	2019015	A005	홍스타	20010501	86	상멘노



```
public class OditionVo {

    private String artist_id;
    private String artist_name;
    private String artist_gender;
    private String artist_birth;
    private String talent;
    private String agency;

    //추가
    private String serial_no;
    private String mento_name;
    private int point;
    private String grade;

    //getter, setter 메서드 추가
```

참가자 목록 조회와 같이 select 결과를 List<OditionVo> 리턴타입을 갖는 메서드를 만든다.
만약 select 메서드 만드는 것이 이해가 되고 코딩할 수 있으면 참가자 목록 조회 메서드를 복사하여 수정하는 것이
코딩 시간을 단축 할 수 있다

```
//멘토점수목록 조회
public List<OditionVo> getMentoList() {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    String sql = "select
p.serial_no,a.artist_id,a.artist_name,a.artist_birth
,p.point,m.mento_name\r\n" +
"from tbl_point_201905 p, tbl_artist_201905 a,
tbl_mento_201905 m\r\n" +
"where p.artist_id = a.artist_id and m.mento_id =
p.mento_id\r\n" +
"order by p.serial_no";

    List<OditionVo> list = new ArrayList<OditionVo>();
```

SQL 문장은 SQL Developer 에서 복사하여
붙여넣기 한다
위 문장이 확실하게 연습이 될때 까지 반복하여 복습해 두기 바란다

```
try {
    conn = DBManager.getInstance().getDBmanager();
    pstmt = conn.prepareStatement(sql);
    rs = pstmt.executeQuery();
    while(rs.next()) {
        OditionVo vo = new OditionVo();
        vo.setSerial_no(rs.getString("serial_no"));
        vo.setArtist_id(rs.getString("artist_id"));
        vo.setArtist_name(rs.getString("artist_name"));
        vo.setArtist_birth(rs.getString("artist_birth"));
        vo.setPoint(rs.getInt("point"));
        vo.setMento_name(rs.getString("mento_name"));

        list.add(vo);
    }
} catch (Exception e) {
    e.printStackTrace();
}
return list;
}
```

OditionMain class 에 3번 메뉴를 선택하면 출력되는 프로그램을 추가한다

```
else if(sel==3) {
    List<OditionVo> list = dao.getMentoList();
    for(OditionVo vo : list) {
        System.out.print(vo.getSerial_no()+"\t");
        System.out.print(vo.getArtist_id()+"\t");
        System.out.print(vo.getArtist_name()+"\t");
        System.out.print(vo.getArtist_birth().substring(0,4)+"년");
        System.out.print(vo.getArtist_birth().substring(4,6)+"월");
        System.out.print(vo.getArtist_birth().substring(6)+"일+"\t");
        System.out.print(vo.getPoint()+"\t");
        if(vo.getPoint()>=90) {
            System.out.print("A+"\t");
        }else if(vo.getPoint()>=80) {
            System.out.print("B+"\t");
        }else if(vo.getPoint()>=70) {
            System.out.print("C+"\t");
        }else if(vo.getPoint()>=60) {
            System.out.print("D+"\t");
        }else {
            System.out.print("F+"\t");
        }
        System.out.print(vo.getMento_name()+"\t");
        System.out.println();
    }
}

else if(sel==4) {

}

else if(sel==5) {
    break;
}
```

문제5> 4번 적업 선택시 아래와 같이 참가자 등수 조회가 되도록 프로그램을 완성하시오

메뉴선택 : [1]오디션등록[2]참가자목록조회[3]멘토점수목록조회[4]참가지등수조회[5]종료4

A004	정스타	남성	284	94.67	1
A003	왕스타	남성	180	90.00	2
A005	홍스타	여성	172	86.00	3
A002	조스타	남성	230	76.67	4
A001	김스타	여성	224	74.67	5

- ① 평균은 소수점 둘째자리까지 출력하시오.
- ② 동점자는 없고, 평균이 나누어 떨어져도 소수점 둘째자리까지 표시하시오(90 – 90.00)
- ③ 성별은 남성 / 여성 로 표시하세요.
- ④ 순위를 표시하세요

멘토점수 목록조회 문제를 푸는 것과 같다. SQL문을 생각해 보고 만들어 보자. 출력결과를 보면 artist_id별 그룹을 지어서 정렬 하는 결과문 이다. 그럼 아래와 같이 먼저 코딩한다

```
select  
from  
where  
group by  
order by
```

출력결과문을 보고 사용하는 테이블을 파악해 보자

```
select  
from tbl_artist_201605 a, tbl_point_201905 p  
where  
group by  
order by
```

출력결과문을 보고 출력하는 컬럼을 분석하고 코딩한다

```
select a.artist_id,a.artist_name,a.artist_gender,sum(p.point) as tot,round(avg(p.point),2) as ave  
from tbl_artist_201905 a, tbl_point_201905 p  
where  
group  
order by  
;
```


출력결과문을 보고 where 절에 조인 조건을 입력한다

```
select a.artist_id,a.artist_name,a.artist_gender,sum(p.point) as tot,round(avg(p.point),2) as ave
from tbl_artist_201905 a, tbl_point_201905 p
where a.artist_id = p.artist_id
group by
order by
```

group by 절은 select 문의 컬럼만 복사 붙여 넣기 한다. 함수는 제외 시킨다

```
select a.artist_id,a.artist_name,a.artist_gender,sum(p.point) as tot,round(avg(p.point),2) as ave
from tbl_artist_201905 a, tbl_point_201905 p
where a.artist_id = p.artist_id
group by a.artist_id,a.artist_name,a.artist_gender
order by ;
```

마지막으로 평균순으로 정렬값을 입력한다

```
select a.artist_id,a.artist_name,a.artist_gender,sum(p.point) as tot,round(avg(p.point),2) as ave
from tbl_artist_201905 a, tbl_point_201905 p
where a.artist_id = p.artist_id
group by a.artist_id,a.artist_name,a.artist_gender
order by ave desc;
```

위 완성된 select 질의어를 실행하면 아래와 같이 콘솔창에 출력이 될 것이다. 그럼 실행 결과중 한 개의 레코드를 VO 객체에 담아야 한다. 추가된 컬럼은 tot와 ave 이다

	ARTIST_ID	ARTIST_NAME	ARTIST_GENDER	TOT	AVE
1	A004	성스타	M	284	94.67
2	A003	왕스타	M	180	90
3	A005	옹스타	F	172	86
4	A002	소스타	M	230	76.67
5	A001	김스타	F	224	74.67

OrdinationVo 클래스에 위 레코드를 담을 멤버변수를 추가한후 getter,setter 메서드를 만든다

javaJDBCodition
 > DBManager.java
 > DbTest.java
 > OditionDao.java
 > OditionMain.java
 > OditionVo.java

```
public class OditionVo {  
  
    private String artist_id;  
    private String artist_name;  
    private String artist_gender;  
    private String artist_birth;  
    private String talent;  
    private String agency;  
  
    private String serial_no;  
    private String mento_name;  
    private int point;  
    private String grade;  
  
    private int tot;  
    private double ave;
```

참가자 등수 조회를 처리할 메서드를 만든다. 멘토점수목록조회 메서드를 복사하여 붙여넣기 하고 수정해서 처리해 보도록 하자

// 참가자 등수 조회

```
public List<OditionVo> getRankList() {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    String sql = "select a.artist_id,a.artist_name,a.artist_gender,sum(p.point) as tot,round(avg(p.point),2) as ave " +
        "from tbl_artist_201905 a, tbl_point_201905 p " +
        "where a.artist_id = p.artist_id " +
        "group by a.artist_id,a.artist_name,a.artist_gender " +
        "order by ave desc";
    List<OditionVo> list = new ArrayList<OditionVo>();
    try {
        conn = DBManager.getInstance().getDBmanager();
        pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery();
        while (rs.next()) {
            OditionVo vo = new OditionVo();
            vo.setArtist_id(rs.getString("artist_id"));
            vo.setArtist_name(rs.getString("artist_name"));
            vo.setArtist_gender(rs.getString("artist_gender"));
            vo.setTot(rs.getInt("tot"));
            vo.setAve(rs.getDouble("ave"));
            list.add(vo);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return list;
}
```

이제 OditionMain class에서 4번 메뉴 선택시 출력하는 프로그램을 만들어 보자

```
else if(sel==4) {
    List<OditionVo> list = dao.getRankList();
    int rank=1;
    DecimalFormat df = new DecimalFormat("###.00");
    for(OditionVo vo : list) {
        System.out.print(vo.getArtist_id()+"\t");
        System.out.print(vo.getArtist_name()+"\t");
        if(vo.getArtist_gender().equals("M")) {
            System.out.print("남성"+"\\t");
        }else {
            System.out.print("여성"+"\\t");
        }
        System.out.print(vo.getTot()+"\\t");
        System.out.print(df.format(vo.getAve()+"\\t");
        System.out.print(rank+"\\t");
        rank++;
        System.out.println();
    }
}
```

이 프로젝트를 10번 코딩 하시오