오늘의 예제 (**연예인성적표**)

데이터베이스를 이용하여 다음과 같이 동작하는 프로그램을 구현하시오.

1을 누르면 데이터 입력

이름, 직업명, 국어, 영어, 수학점수를 입력받아

데이터 베이스에 번호를 포함하여 입력한다.

번호는 시퀀스를 이용하여 순차적으로 입력한다. 데이터베이스의 내용은 다음과 같다

번호(pNO) 이름(pNAME) 직업(jno) 국어(kor) 영어(eng) 수학(mat)

────────────────────────────────────

1 정우성 10 90 80 81

2 박세영 10 80 90 80

3 배수지 20 20 90 90

2를 누르면 원하는 직업명을 입력 받아

직업별 조회 후 총점을 추가하여 총점이 높은 순으로 이름(번호)로 출력

등수 이름(pNO) 직업 국어(kor) 영어(eng) 수학(mat) 총점

───────────────────────────────────────

1등 정우성(1번) 배우 90 80 81 251

2등 박세영(2번) 배우 80 90 80 250

3을 누르면

데이터베이스에 입력된 사람 전체를 조회 후 총점을 추가하여

총점이 높은 순으로 출력한다

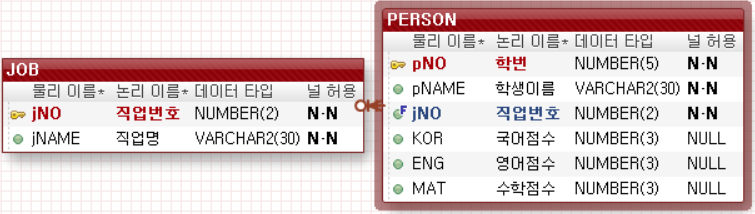
등수 이름(pNO) 직업 국어(kor) 영어(eng) 수학(mat) 총점

────────────────────────────────────

1등 정우성(1) 배우 90 80 81 251

2등 박세영(2) 배우 80 90 80 250

3등 배수지(3) 가수 20 90 90 200



0. 패키지를 만든다

1. 패키지 안에 erd를 만든다

2. DDL, 더미데이터 입력, 필요한 sql문 작성

3. java 구현

DROP TABLE PERSON;

DROP TABLE JOB;

CREATE TABLE JOB(

JNO NUMBER(2) PRIMARY KEY,

JNAME VARCHAR2(30) NOT NULL);

CREATE TABLE PERSON (

PNO NUMBER(5) PRIMARY KEY,

PNAME VARCHAR2(30) NOT NULL,

JNO NUMBER(2) REFERENCES JOB(JNO) NOT NULL,

KOR NUMBER(3),

ENG NUMBER(3),

MAT NUMBER(3) );

DROP SEQUENCE PERSON\_NO\_SQ;

CREATE SEQUENCE PERSON\_NO\_SQ MAXVALUE 99999 NOCACHE NOCYCLE;

INSERT INTO JOB VALUES (10, '배우');

INSERT INTO JOB VALUES (20, '가수');

INSERT INTO JOB VALUES (30, '엠씨');

SELECT \* FROM JOB;

-- 1. 입력

INSERT INTO PERSON VALUES (PERSON\_NO\_SQ.NEXTVAL, '정우성', (SELECT JNO FROM JOB WHERE JNAME='배우'), 90,80,81);

INSERT INTO PERSON VALUES

(PERSON\_NO\_SQ.NEXTVAL, '박세영',

(SELECT JNO FROM JOB WHERE JNAME='배우'), 80,80,81);

INSERT INTO PERSON VALUES

(PERSON\_NO\_SQ.NEXTVAL, '배수지',

(SELECT JNO FROM JOB WHERE JNAME='가수'), 100,90,91);

SELECT \* FROM PERSON;

COMMIT;

-- 2번 기능 1 박세영(2번) 배우 80 80 81 241(총점순 내림차순)

SELECT PNAME||'('||PNO||'번)' PNAME, JNAME, KOR, ENG, MAT, KOR+ENG+MAT SUM

FROM PERSON P, JOB J

WHERE P.JNO=J.JNO AND JNAME='배우'

ORDER BY SUM DESC;

SELECT ROWNUM RANK, S.\*

FROM (SELECT PNAME||'('||PNO||'번)' PNAME, JNAME, KOR, ENG, MAT, KOR+ENG+MAT SUM

FROM PERSON P, JOB J

WHERE P.JNO=J.JNO AND JNAME='배우'

ORDER BY SUM DESC) S;

-- 3번 기능

SELECT ROWNUM RANK, S.\*

FROM (SELECT PNAME||'('||PNO||'번)' PNAME, JNAME, KOR, ENG, MAT, KOR+ENG+MAT SUM

FROM PERSON P, JOB J

WHERE P.JNO=J.JNO

ORDER BY SUM DESC) S;

-- 콤보 박스에 넣을 직업 리스트

SELECT JNAME FROM JOB;

COMMIT;

delete from person where jno=30;

**public** **class** PersonDto {

**private** **int** rank;

**private** String pname;

**private** String jname;

**private** **int** kor;

**private** **int** eng;

**private** **int** mat;

**private** **int** sum;

// insert용

**public** PersonDto(String pname, String jname, **int** kor, **int** eng, **int** mat) {

**this**.pname = pname;

**this**.jname = jname;

**this**.kor = kor;

**this**.eng = eng;

**this**.mat = mat;

}

// select 용

**public** PersonDto(**int** rank, String pname, String jname, **int** kor, **int** eng, **int** mat, **int** sum) {

**this**.rank = rank;

**this**.pname = pname;

**this**.jname = jname;

**this**.kor = kor;

**this**.eng = eng;

**this**.mat = mat;

**this**.sum = sum;

}

@Override

**public** String toString() {

**return** rank + "\t" + pname + "\t" + jname + "\t" + kor + "\t" + eng

+ "\t" + mat + "\t" + sum;

}

**public** **int** getRank() { **return** rank; }

**public** String getPname() { **return** pname; }

**public** String getJname() { **return** jname; }

**public** **int** getKor() { **return** kor; }

**public** **int** getEng() { **return** eng; }

**public** **int** getMat() { **return** mat; }

**public** **int** getSum() { **return** sum; }

}

**public** **class** PersonDao {

**private** String driver = "oracle.jdbc.driver.OracleDriver";

**private** String url = "jdbc:oracle:thin:@localhost:1521:xe";

**public** **static** **final** **int** ***SUCCESS*** = 1;

**public** **static** **final** **int** ***FAIL*** = 0;

**private** **static** PersonDao *INSTANCE*;

**public** **static** PersonDao getInstance() {

**if**(*INSTANCE* == **null**) {

*INSTANCE* = **new** PersonDao();

}

**return** *INSTANCE*;

}

**private** PersonDao() {

**try** {

Class.*forName*(driver);

} **catch** (ClassNotFoundException e) {

System.***out***.println(e.getMessage());

}

}

// 1번 입력

**public** **int** insertPerson(PersonDto dto) {

**int** result = ***FAIL***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "INSERT INTO PERSON VALUES (PERSON\_NO\_SQ.NEXTVAL, ?, (SELECT JNO FROM JOB WHERE JNAME=?), ?, ?, ?)";

**try** {

conn = DriverManager.*getConnection*(url, "Scott","tiger");

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, dto.getPname());

pstmt.setString(2, dto.getJname());

pstmt.setInt(3, dto.getKor());

pstmt.setInt(4, dto.getEng());

pstmt.setInt(5, dto.getMat());

result = pstmt.executeUpdate();

}**catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (SQLException e2) {

// **TODO**: handle exception

}

}

**return** result;

}

// 2번

**public** ArrayList<PersonDto> selectJname(String jname){

ArrayList<PersonDto> dtos = **new** ArrayList<PersonDto>();

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT ROWNUM RANK, S.\* " +

" FROM (SELECT PNAME||'('||PNO||'번)' PNAME, JNAME, KOR, ENG, MAT, KOR+ENG+MAT SUM " +

" FROM PERSON P, JOB J" +

" WHERE P.JNO=J.JNO AND JNAME=? " +

" ORDER BY SUM DESC) S";

**try** {

conn = DriverManager.*getConnection*(url,"scott","tiger");

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, jname);

rs = pstmt.executeQuery();

**while**(rs.next()) {

**int** rank = rs.getInt("rank");

String pname = rs.getString("pname");

**int** kor = rs.getInt("kor");

**int** eng = rs.getInt("eng");

**int** mat = rs.getInt("mat");

**int** sum = rs.getInt("sum");

dtos.add(**new** PersonDto(rank, pname, jname, kor, eng, mat, sum));

}

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e2) {

// **TODO**: handle exception

}

}

**return** dtos;

}

// 3번

**public** ArrayList<PersonDto> selectAll(){

ArrayList<PersonDto> dtos = **new** ArrayList<PersonDto>();

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT ROWNUM RANK, S.\* " +

" FROM (SELECT PNAME||'('||PNO||'번)' PNAME, JNAME, KOR, ENG, MAT, KOR+ENG+MAT SUM " +

" FROM PERSON P, JOB J" +

" WHERE P.JNO=J.JNO " +

" ORDER BY SUM DESC) S";

**try** {

conn = DriverManager.*getConnection*(url,"scott","tiger");

pstmt = conn.prepareStatement(sql);

rs = pstmt.executeQuery();

**while**(rs.next()) {

**int** rank = rs.getInt("rank");

String pname = rs.getString("pname");

String jname = rs.getString("jname");

**int** kor = rs.getInt("kor");

**int** eng = rs.getInt("eng");

**int** mat = rs.getInt("mat");

**int** sum = rs.getInt("sum");

dtos.add(**new** PersonDto(rank, pname, jname, kor, eng, mat, sum));

}

} **catch** (SQLException e) {

System.***out***.println("3번기능"+e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e2) {

// **TODO**: handle exception

}

}

**return** dtos;

}

**public** ArrayList<String> jnamelist(){

ArrayList <String> jnames = **new** ArrayList<String>();

jnames.add("");

// 직업등 리스트를 jnames에 add 하기

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT JNAME FROM JOB";

**try** {

conn = DriverManager.*getConnection*(url, "scott","tiger");

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);

**while**(rs.next()) {

jnames.add(rs.getString("jname"));

}

} **catch** (SQLException e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(stmt!=**null**) stmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (SQLException e2) {

// **TODO**: handle exception

}

}

**return** jnames;

}

}class

**public** **class** TestMain { // 테스트해보기

**public** **static** **void** main(String[] args) {

PersonDto dto = **new** PersonDto("홍길동", "엠씨", 99, 99, 99);

PersonDao dao = PersonDao.*getInstance*();

**int** result = dao.insertPerson(dto);

System.***out***.println(result==PersonDao.***SUCCESS***? "입력성공":"입력실패");

System.***out***.println("---------2번 test ---------------");

ArrayList<PersonDto> dtos = dao.selectJname("배우");

**if**(dtos.size()==0) {

System.***out***.println("배우가 없습니다.");

}**else** {

**for**(PersonDto d : dtos)

System.***out***.println(d);

}

System.***out***.println("------------3번 test ------------");

dtos = dao.selectAll();

**if**(dtos.isEmpty()) {

System.***out***.println("등록된 사람이 없습니다.");

}**else** {

**for**(PersonDto d : dtos)

System.***out***.println(d);

}

}

}

**public** **class** PersonMngUseDao {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

PersonDao dao = PersonDao.*getInstance*();

String fn;

ArrayList<PersonDto> person;

**do** {

System.***out***.println("1:입력 | 2:직업별출력 | 3:전체출력 | 그외:종료 >");

fn = scanner.next();

**switch**(fn) {

**case** "1": // 이름, 직업명, 국영수 입력받아 dao.insertPerson()호출

System.***out***.print("입력할 이름 ?");

String name = scanner.next();

System.***out***.print("입력할 직업 ?");

String jname = scanner.next();

System.***out***.print("국어 점수 ?");

**int** kor = scanner.nextInt();

System.***out***.print("영어 점수 ?");

**int** eng = scanner.nextInt();

System.***out***.print("수학 점수 ?");

**int** mat = scanner.nextInt();

PersonDto newPerson = **new** PersonDto(name, jname, kor, eng, mat);

**int** result = dao.insertPerson(newPerson); // 입력 끝

System.***out***.println(result==PersonDao.***SUCCESS*** ? "입력성공":"입력실패");

**break**;

**case** "2" : // 직업명 입력받아 dao.selectJname() 호출하여 결과 출력

System.***out***.print("조회할 직업명은(배우 | 가수 | 엠씨)? ");

jname = scanner.next();

person = dao.selectJname(jname);

**if**(person.size()!=0) {

System.***out***.println("rank\t이름\t직업\t국어\t영어\t수학\t총점");

**for**(PersonDto p : person) {

System.***out***.println(p);

}

}**else** {

System.***out***.println("해당 직업명의 인원이 없습니다");

}

**break**;

**case** "3" : // dao.selectAll() 호출하여 결과 출력

person = dao.selectAll();

**if**(person.size()!=0) {

System.***out***.println("rank\t이름\t직업\t국어\t영어\t수학\t총점");

**for**(PersonDto p : person) {

System.***out***.println(p);

}

}**else** {

System.***out***.println("해당 직업명의 인원이 없습니다");

}

**break**;

}

}**while**(fn.equals("1")||fn.equals("2")||fn.equals("3"));

System.***out***.println("BYE");

}

}