

1. 서론

1.1 프로젝트 개요

부산 대기오염 조회 프로그램은 Oracle DBMS를 활용하여 각 거주지의 오염단계를 효율적으로 갱신하고, 데이터를 구축한 뒤 JDBC 기반 자바 프로그램을 통해 다음 기능을 제공한다.

1. 회원이 원하는 지역의 오염수치 및 오염단계 조회
2. 관리자 로그인을 통한 오염 측정값 변경 및 회원 관리
3. 데이터베이스 관리 효율성을 높이기 위해 저장 프로시저와 트리거 사용

1.2 개발 개요

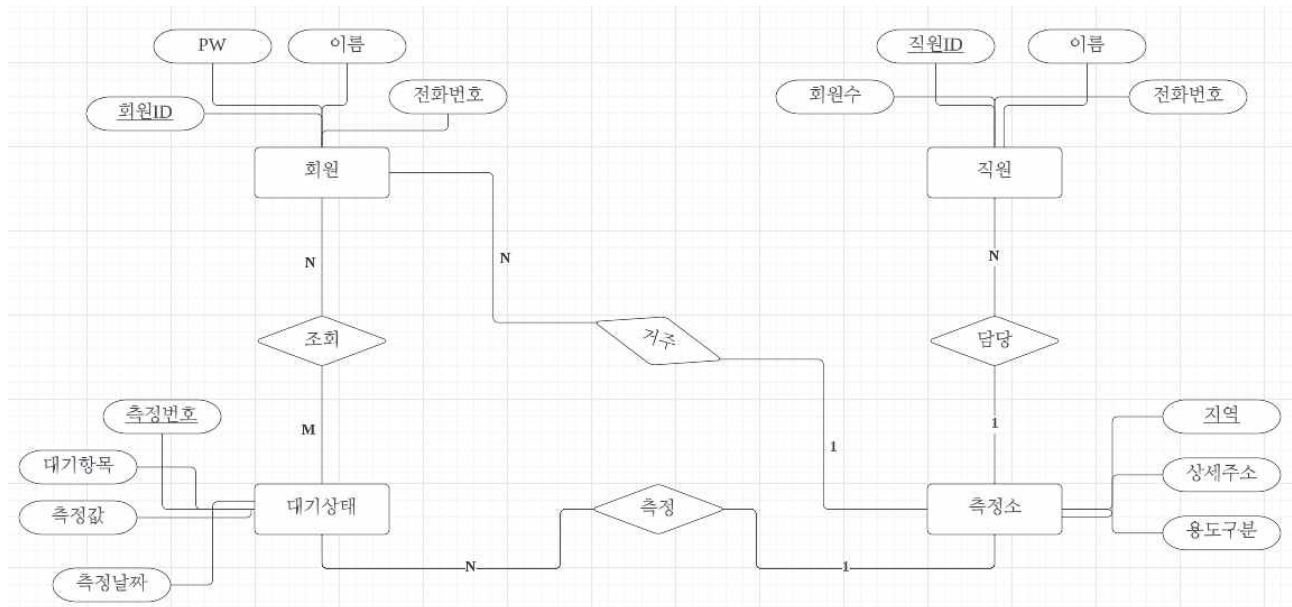
- » 프로젝트 명 : 부산 대기오염 조회 프로그램
- » 기술 스택: Oracle, Java (JDBC), Swing
- » 프로젝트 기간 : 2022. 11. 02 ~ 2022. 12. 08 (약 1개월)
- » 개발 인원: 3명

1.3 담당 역할

1. 측정값이 수정되면 오염단계가 자동으로 수정되는 트리거
2. 지역별 현재 날짜에 맞는 대기상태 검색 및 전체 선택 시 모든 대기상태 검색
3. 대기항목 중 오염단계가 '나쁨'인 지역을 검색

2. 본론

2.1 E-R 다이어그램



2.2 릴레이션 스키마

회원 (회원ID, PW, 이름, 거주지, 전화번호)

직원 (직원ID, 이름, 전화번호, 담당측정소, 회원수, 지역)

측정소 (지역, 상세주소, 용도구분, 직원ID)

대기상태 (대기항목, 측정날짜, 측정시간, 측정값, 지역)

확인 (회원ID, 대기항목)

- 외래키는 이탤릭체로 표기 -

2.3 구현 결과

- 측정값에 따라 오염단계 갱신하는 트리거

```
--측정값 변경하면 트리거 발동
create or replace NONEDITIONABLE TRIGGER update_plt
AFTER UPDATE of 측정값 ON 대기상태
declare
station VARCHAR(20); --지역
air_value NUMBER(10,3); --측정값
item VARCHAR(20); --대기항목

--입력받은 지역의 대기항목과 측정값을 커서c에 저장
CURSOR C IS SELECT 측정지역, 대기항목, 측정값 FROM 대기상태 WHERE 측정날짜 = TO_CHAR(SYSDATE);
BEGIN
OPEN C; --커서열기
LOOP
FETCH C INTO station, item, air_value; --데이터 인출
EXIT WHEN C%NOTFOUND; --데이터가 없으면 LOOP 종료

CASE
WHEN item = '초미세먼지' THEN
BEGIN
IF air_value BETWEEN 0 AND 15 then
UPDATE 대기상태 SET 오염단계 = '좋음' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSIF air_value BETWEEN 16 AND 35 then
UPDATE 대기상태 SET 오염단계 = '보통' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSIF air_value BETWEEN 36 AND 75 then
UPDATE 대기상태 SET 오염단계 = '나쁨' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSE
UPDATE 대기상태 SET 오염단계 = '매우나쁨' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
END IF;
END;
WHEN item = '미세먼지' THEN
BEGIN
IF air_value BETWEEN 0 AND 30 then
UPDATE 대기상태 SET 오염단계 = '좋음' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSIF air_value BETWEEN 31 AND 80 then
UPDATE 대기상태 SET 오염단계 = '보통' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSIF air_value BETWEEN 81 AND 150 then
UPDATE 대기상태 SET 오염단계 = '나쁨' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSE
UPDATE 대기상태 SET 오염단계 = '매우나쁨' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
END IF;
END;
WHEN item = '오존' THEN
BEGIN
IF air_value BETWEEN 0 AND 0.030 then
UPDATE 대기상태 SET 오염단계 = '좋음' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSIF air_value BETWEEN 0.031 AND 0.090 then
UPDATE 대기상태 SET 오염단계 = '보통' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSIF air_value BETWEEN 0.091 AND 0.150 then
UPDATE 대기상태 SET 오염단계 = '나쁨' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSE
UPDATE 대기상태 SET 오염단계 = '매우나쁨' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
END IF;
END;
WHEN item = '일산화탄소' THEN
BEGIN
IF air_value BETWEEN 0 AND 2.00 then
UPDATE 대기상태 SET 오염단계 = '좋음' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSIF air_value BETWEEN 2.01 AND 9.00 then
UPDATE 대기상태 SET 오염단계 = '보통' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSIF air_value BETWEEN 9.01 AND 15 then
UPDATE 대기상태 SET 오염단계 = '나쁨' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
ELSE
UPDATE 대기상태 SET 오염단계 = '매우나쁨' WHERE 대기상태.대기항목 = item AND 대기상태.측정지역 = station AND 측정날짜 = TO_CHAR(SYSDATE);
END IF;
END;
END CASE;
END LOOP;
CLOSE C;
END;
```

기존 : 초미세먼지(나쁨), 미세먼지(매우나쁨), 오존(좋음), 일산화탄소(나쁨)

트리거 동작 후 : 초미세먼지(좋음), 미세먼지(좋음), 오존(매우나쁨), 일산화탄소(보통)

측정값 변경

덕천동 2022-12-08

직원ID 관리자

초미세먼지 3

미세먼지 3

오존 3

일산화탄소 3

환영합니다 거주지 : 덕천동

날짜	대기항목	측정값	오염단계
	초미세먼지	38.0	나쁨
	미세먼지	152.0	매우나쁨
	오존	0.02	좋음
	일산화탄소	10.69	나쁨

JDK님 환영합니다 거주지 : 덕천동

측정날짜	대기항목	측정값	오염단계
2022-12-08	초미세먼지	3.0	좋음
2022-12-08	미세먼지	3.0	좋음
2022-12-08	오존	3.0	매우나쁨
2022-12-08	일산화탄소	3.0	보통

자바 GUI 프로그램에서 Statement 수행 기능

(1) 대기상태 테이블을 select 하여 Jtable을 채운다.

① 메인화면 초기에는 현재 날짜 내용을 표현

```

//대기상태 정보 DB에서 받아옴
public void tableDb() throws SQLException {
    LocalDate now = LocalDate.now();
    int i=0;
    if(dbOk==0) {
        //현재날짜만
        String query = "select 대기항목, 측정값, 측정날짜, 오염단계 from 대기상태 where 측정지점='"+address+"' and 측정날짜='"+Date.valueOf(now)+"' order by 측정날짜 asc";
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        try {
            while (rs.next()) {
                String itemers.getString("대기항목");
                double values.getDouble("측정값");
                Date Mdateers.getDate("측정날짜");
                String pollution = rs.getString("오염단계");
                Object data[] = {Mdate, item, value, pollution};
                main.setTable(data);
                main.setVal(i, value);
                i++;
            }
            if(i==4) i=0;
        } catch (SQLException e) { e.printStackTrace(); }
    } else {
        //모든 날짜
        String query = "select 대기항목, 측정값, 측정날짜, 오염단계 from 대기상태 where 측정지점='"+address+"' order by 측정날짜 asc";
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        try {
            while (rs.next()) {
                String itemers.getString("대기항목");
                double values.getDouble("측정값");
                Date Mdateers.getDate("측정날짜");
                String pollution = rs.getString("오염단계");
                Object data[] = {Mdate, item, value, pollution};
                main.setTable(data);
                main.setVal(i, value);
                i++;
            }
            if(i==4) i=0;
        } catch (SQLException e) { e.printStackTrace(); }
    }
}

```

2022-12-08 광복동

초미세먼지 25.0 미세먼지 45.0

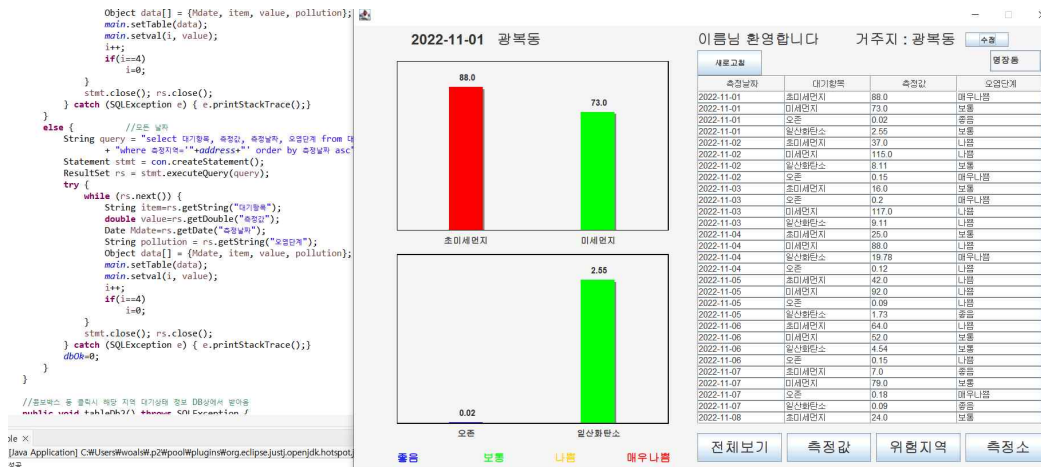
오존 0.09 일산화탄소 15.84

좋음
보통
나쁨
매우나쁨

이름님 환영합니다 거주지 : 광복동

측정날짜	대기항목	측정값	오염단계
2022-12-08	초미세먼지	25.0	보통
2022-12-08	미세먼지	45.0	보통
2022-12-08	오존	0.09	보통
2022-12-08	일산화탄소	15.84	매우나쁨

②전체보기를 누를 시 해당 모든 내용이 나오게 된다.



(2)대기상태 테이블을 select하되 오염단계가 나쁨 or 매우나쁨만 가져와 Jtable을 채운다.

```
try {
    stmt.close();
} catch (SQLException e) { e.printStackTrace(); }

//위험지역 DB상에서 받아옴
public void dangerDb() throws SQLException {
    String query = "select 측정지역, 대기질목, 측정날짜, 오염단계 from 대기상태 "
        + "where 오염단계 = '매우나쁨' or 오염단계 = '나쁨' order by 측정날짜 asc";
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(query);
    try {
        while(rs.next()) {
            String region = rs.getString("측정지역");
            String item = rs.getString("대기질목");
            Date date = rs.getDate("측정날짜");
            String pol = rs.getString("오염단계");
            Object data[] = {date, item, region, pol};
            dg.setTable(data);
        }
    } catch (SQLException e) { e.printStackTrace(); }
}

//측정소 정보 DB상에서 받아옴
public void placeDb() throws SQLException {
    String query = "select 이름, 지역, 좌표를 from 측정소";
    PreparedStatement stmt = con.prepareStatement(query);
    ResultSet rs = stmt.executeQuery(query);
    try {
        while (rs.next()) {
            String name=rs.getString("이름");
            String region=rs.getString("지역");
        }
    }
}
```

날짜	대기질목	위험지역	오염단계
2022-11-01	오존	창원동	나쁨
2022-11-01	일산화탄소	창원동	나쁨
2022-11-01	초미세먼지	창원동	매우나쁨
2022-11-01	미세먼지	창원동	나쁨
2022-11-01	초미세먼지	삼척동	매우나쁨
2022-11-01	미세먼지	화랑동	나쁨
2022-11-01	초미세먼지	용호동	매우나쁨
2022-11-01	오존	용호동	나쁨
2022-11-01	일산화탄소	용호동	나쁨
2022-11-01	일산화탄소	본천동	나쁨
2022-11-01	일산화탄소	영지동	매우나쁨
2022-11-01	초미세먼지	화랑동	매우나쁨
2022-11-01	미세먼지	화랑동	나쁨
2022-11-01	일산화탄소	화랑동	매우나쁨
2022-11-01	오존	재송동	매우나쁨
2022-11-01	미세먼지	창학동	매우나쁨
2022-11-01	오존	창학동	매우나쁨
2022-11-01	일산화탄소	창학동	나쁨
2022-11-01	초미세먼지	창학동	나쁨

달기

3. 결론

지금은 공공데이터포털에 있는 자료 기반으로 날짜별로 넣어줬는데, 오픈 API와 같은 자료를 이용해서 시간별로 실시간 업데이트 해주는 프로그램을 만든다면 더 유용할 것이다.