

버섯농장 데이터 관리 시스템에 관한 연구

*황선명, 박성욱, 양석우, 손민수, 임동섭, 김정섭, 김상규
*대전대학교 컴퓨터공학과 교수
대전대학교 컴퓨터공학과
e-mail:pppforte@gmail.com

Farm Data Management System

Sun-Myung Hwang, Seong-Uk Park, Suk-Woo Yang, Min-Soo Son,
Dong-Sub Lim, Jeung-Seop Kim, Sang-Kyu Kim

*Professor, Dept of Computer Engineering, Dae-jeon University
Dept of Computer Engineering, Dae-jeon University

요 약

농촌의 많은 농장에서는 데이터 처리 및 분석 시스템에 대한 필요성을 느끼면서 지내오고 있다. 본 논문에서는 데이터베이스 시스템을 접목시켜서 기존 엑셀에서 간단한 계산만 이루어지던 농장 데이터 관리 시스템을 효율적으로 개편하여 데이터 분석 및 확인 시스템을 개발하였다. 본 연구에서 개발한 데이터 시스템을 실제 버섯농장에 설치 후 실험을 통하여 기존 방법 대비의 정확성, 실용성 그리고 신뢰할수 있는 분석 시스템을 구축하였다.

1. 서론

오늘날 농경사회는 많은 변화를 맞이하고 있는데, 기존 7080세대들의 귀농으로 인한 농장, 농업 등 농경에 대한 관심도가 높아지고 있고 귀농 페스티벌이 자주 열리는 등 농경에 대한 관심도는 날로 높아지고 있으며, 이로 인하여 자연스럽게 농경과 IoT의 접목 시스템은 사회의 중요 화두가 되었다. 이런 현상에 비해 현재 농경사회는 농업에서 나오는 각종 데이터들을 관리하고 분석하는 시스템이 많이 부족한 상황이다.

최근 IoT와 함께 사회의 주목을 받고 크게 성장해 나가고 있는 사업인 데이터 관리 시스템은, 빅 데이터 라는 키워드와 함께 현대 사회에 있어서 분석 등에 데이터 관련 처리 관리에 있어서 2013년 10대 IT 전략 기술 중 하나로 꼽히기도 하였다.[1] 이런 빅 데이터 등의 데이터 관리 시스템의 핵심은 신뢰성에 기반 하여야 하는데 데이터의 신뢰성이 보장되지 않는다면 그 데이터 시스템은 완전히 필요성이 상실된 데이터 시스템이라고 볼 수 있다. 데이터를 사용하는 클라이언트의 입장에서 신뢰할 수 없다면 그로인한 데이터 분석 등 모든 면에서 사업적인 부분이 지장을 받기 때문이다. 이런 것에 있어서 데이터의 신뢰성을 높이는 방법으로는 데이터들의 유실 방지 및 효율적인 프로그램 데이터베이스 접근을 위한 인터넷 데이터 서버 장치가 필요하였다. 기본 데이터베이스를 로컬 시스템으로 구축하지 않고 NAS

서버를 이용하여 그 안에 데이터를 저장 하여 어느 곳에서도 프로그램을 설치하면 데이터 입력 및 관리가 가능케 하여서 프로그램에 대한 접근성을 더욱 높였다. 이는 프로그램이 특정 컴퓨터에서 작동 시에 해당 컴퓨터의 오류 등으로 인한 불상사 등, 여러 대처상황에 유연하게 대처할 수 있었다.

본 논문에서는 이런 농업중 하나인 버섯농장에 데이터 분석 처리 시스템을 개발하여 기존 수기식 및 엑셀 식으로 하던 방법 보다 편의성 및 신뢰성을 높여주는 분석 시스템을 개발하였다.

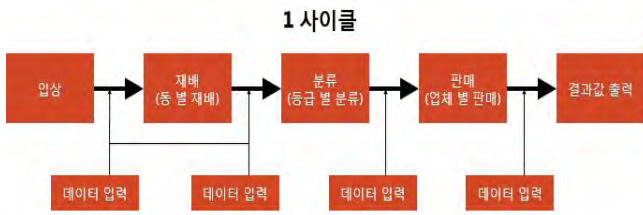
2. 버섯농장의 버섯 재배 시스템

본 논문에서 연구 대상으로 삼은 버섯 농장에서는 그림 1과 같은 사이클을 가지고 있는 상태이다.



[그림 1] 버섯농장의 버섯 재배 방식 사이클

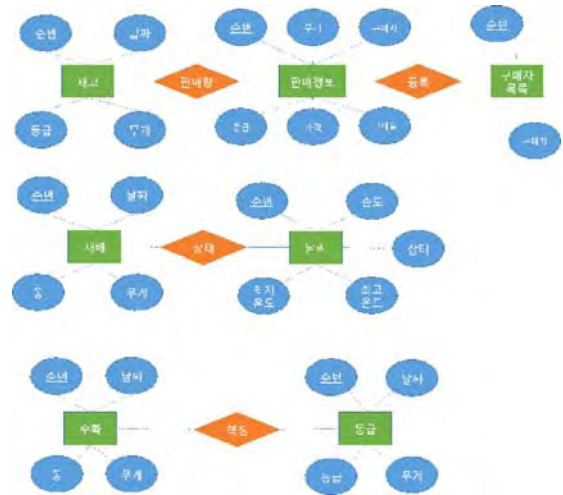
이러한 방식을 통하여 버섯을 재배하게 되는 버섯농장은 버섯을 재배하는데 있어서 그림 2와 같은 전체 사이클을 가지고 있고 그 사이사이 데이터가 발생되어 그 데이터들을 입력 하는 방식 이었다.



[그림 2] 버섯농장의 전체적인 생산 흐름 이런 방식에 있어서 데이터들을 수집하는 방식은 엑셀과 손으로 작성하여 노트에 적는 식으로 되어 있었고, 해당 버섯농장은 1년 총 생산량 통계나 분석 등을 통해 몇 월쯤에 가장 수요가 많고 적은지, 가격 변동 폭은 어떻게 되는지에 대한 빅 데이터 분석을 필요로 하였고 이는 해당 연구에 적합하다고 판단, 해당 농장에 맞는 데이터 시스템을 제작하여 연구에 사용하기로 하였다. 먼저, 그림 1의 재배 에 대한 사이클로는 현재 연구 대상으로 잡은 농장의 18개 동에서 나오는 데이터들로서 원목 1개당 보통 3회의 재배를 하고 폐상(원목에 있는 버섯 효모가 소모된 상태)이 되고 난 뒤 다시 재배를 위해 효모를 심어주고 재배를 하는 형태인데, 이런 식의 1회 사이클당 총 3회의 수확 데이터가 나오게 된다. 버섯의 수확량에 대한 데이터 뿐만 아니라, 현재 재배하는 동 의 상태(1동부터 18동까지의)를 메인화면 및 데이터 분석용으로 수집하기 때문에, 이런 데이터들을 입력 받는다. 현재 동이 수확 시기가 됐는지 물을 주는 침봉 시기가 됐는지 등에 대한 데이터 또한 재배과정에서 수집이 된다. 이런 재배 과정에서 중요한 데이터중 하나는 날씨에 관련된 데이터인데, 이 날씨에 따라 수확량, 제품의 등급 상태등이 나오기 때문에 날씨 또한 좋은 데이터 분석의 참고 상황이 된다. 수확을 하게 돼서 데이터에 입력을 시키면 수확날을 기준으로 기상청 날씨 API를 통해, 날씨에 대한 데이터를 파싱 해와 데이터베이스에 넣어준다. 이때 들어가는 데이터들은 최고온도, 최저온도, 습도 등이 해당된다. 수확을 하고 난 뒤에는 현재 총 수확량이나 각 동별 수확량, 시기별 수확량을 체크하기 위해 수확량에 대한 데이터베이스가 존재하며 이 데이터들과 연계하여 등급별로 나눌 수 있게 등급별 데이터베이스 또한 존재한다. 수확 후 바로 등급을 지정 하는 게 아닌 출고 시에 분류하여 등급을 지정하는 방식이기에 수확 데이터베이스에 넣지 않고 따로 테이블을 만들어 데이터베이스 관리를 해주는 식으로

기획하였다. 해서 등급별 판매량이나 등급별 가격 등에 대한 데이터관리를 가능케 하였다. 그리고 판매 업체에 대한 판매 업체 리스트 테이블과, 판매 업체에 대한 구입량 등에 대한 정보를 수집하고 관리하기 위한 구매자 테이블을 별도 구성하여 사용자들의 테이블을 구성하였다.

이를 통해 프로젝트의 기초적인 E-R 다이어그램은 다음과 같다.



[그림 3] 시스템의 E-R 다이어그램

3. 시스템 구현

3.1 전체 시스템 아키텍처



[그림 4] 현재 시스템의 전체적인 아키텍처

수확 분석 모듈에서는 현재 수확 중인 각 동 의 입력 데이터를 바탕으로 현재 상태와 해야 할 일을 보여 주고 작년 대비 동 별 수확량을 비교한다. 만약, 작년 데이터가 없으면 이번 년도 데이터만을 보여준다. 벚섯 농장은 수확 시작 후 보통 14일 후 침봉을 하게 되는데 상황에 따라서 14일 이후에 침봉을 하는 경우도 있기에 사용자가 침봉을 입력 하기 전 까지 계속 침봉 상태를 보여주게 된다. 작년 대비 동 별 수확량은 입력 각 동 의 받은 수확량을 동 별로 더해 그래프로 보여주어 작년과 올해 각 동 별 수확량을 비교 할 수 있게 만들었다.

분류 분석 모듈에서는 작년 대비 등급별 수확량을 보여주게 되는데 작년 데이터가 없으면 올해 데이터만 보여주게 된다. 입력 작년과 올해 입력 된 각 등급별 수확량을 더해 그래프로 보여주어 작년과 올해 각 등급별 수확량을 비교 할 수 있게 만들었다.

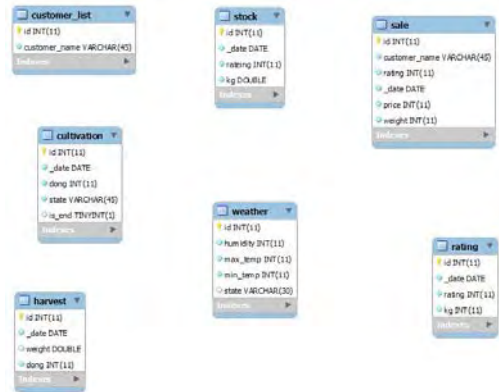
구매 분석 모듈에서는 구매 업체 별 누적 구매 량, 가장 많이 거래한 업체, 업체 별 평균 구매가, 업체 별 총 구매가, 작년 대비 등급별 판매 량과 판매 금액을 분석해 비교해 주는 기능이 있다. 각 업체마다 거래한 등급이 다를 경우가 있으므로 어떤 기업이 몇 등급을 평균적으로 얼마에 구매 했으며 총 구매 한 액수는 얼마인지 한눈에 보기 편하게 하기 위해 데이터 시트 형식으로 값을 보여준다. 등급 별 판매 량은 업체에 상관 없이 작년과 올해의 총 등급별 판매 량을 더해 그래프로 비교 할 수 있으며, 등급 별 판매 금액 역시 업체에 상관 없이 등급 별 판매 금액을 더해 그래프로 작년 과 올해 상황을 비교 할 수 있게 했다.

통계 파트에는 기간 별 각 동 의 수확 상태, 각 업체의 구매 량 상태, 각 등급의 판매 량 상태의 통계를 보여주게 된다. 이 통계는 모두 데이터 시트 형식으로 보여주게 된다. 각 동 의 수확 상태는 각 동에서 사용자가 지정한 날짜 범위 내에서 가장 마지막으로 넣은 데이터의 값과 해당 동 의 수확량 전체를 더한 값과 평균 값을 구해 각각 최근 수확량, 수확량 합계, 수확량 평균을 보여주게 된다. 각 업체의 구매 량 상태는 사용자가 지정한 날짜 범위 내에서 각 업체 별 가장 마지막으로 저장한 구매 데이터를 얻어 오고 해당 업체의 기간 내 평균 구매 가격, 총 합 구매 가격을 얻어와 작년과 비교해 등락률을 구하며 최종적으로 최근 구매 량, 최근 구매 가격, 평균 구매 량, 평균 구매가, 최고 구매가, 최저 구매가, 등락률, 총 구매 량, 총 구매액을 보여준다. 각 등급의

판매 량 상태는 지정한 기간 내에 등급의 마지막으로 입력한 분류 데이터를 얻어오고 해당 등급의 기간 내 평균 판매 량, 평균 판매가, 최고 판매가, 최저 판매가, 등락률, 총 판매 량, 총 판매액을 보여준다.

3.2 데이터 관리 시스템

2절에서 분석한 E-R 다이어그램을 토대로, 데이터베이스를 다음 그림과 같이 실제로 구성한 후 도입시켰다.



[그림 5] MySQL로 구현한 데이터베이스 스키마 외래 키가 없는 것이 특징인데, 데이터베이스를 참조 시스템으로 만들어 자동화 시스템 구현 보다 해당 시스템은 사용자가 개별적으로 입력을 시킨다는 부분에 포커스를 맞추어 서로 연관 되어있는 데이터베이스 구축이 아닌 개별 데이터베이스로 움직여 사용자의 개별 입력에 맞게 제작하였다.

3.3 데이터 분석 시스템

```

DELIMITER $$

CREATE DEFINER='root'@'%' PROCEDURE
    'get_dong'(IN max_dong INT, IN dateFrom DATE, IN dateTo DATE)
BEGIN
    START TRANSACTION;

    DROP TEMPORARY TABLE IF EXISTS test_table;
    CREATE TEMPORARY TABLE test_table (
        dong int,
        weight double,
        sum_weight double,
        avg_weight double
    ) engine = memory;

    SET @dong = 1;
    WHILE(@dong <= 10) DO
        INSERT INTO test_table (dong, weight, sum_weight, avg_weight)
        SELECT tmp.dong, tmp.weight, tmp2.sum_weight, tmp2.avg_weight FROM
        (SELECT dong, weight
        FROM harvest
        WHERE dong=@dong and
        dateFrom <= .date and
        .date <= dateTo
        ORDER BY id DESC LIMIT 1)
        AS tmp,
        (SELECT SUM(weight) AS sum_weight, AVG(weight) AS avg_weight
        FROM harvest
        WHERE dong=@dong and
        dateFrom <= .date and .date <= dateTo)
        AS tmp2;

        SET @dong = @dong + 1;
    END WHILE;

    select * from test_table;
END
    
```

[그림 6] 각 동 의 평균 수확량 및 총 수확량을 가져오는 프로시저.

사용자가 입력한 데이터를 각각의 수확, 재배, 등급, 고객 리스트별로 나뉘어진 각기 다른 테이블에 저장한다. 이렇게 세부적으로 나뉘어진 테이블에 의해 데이터를 검색하게 되는데 세부적으로 나누어진 테이블마다 쿼리를 보내게 된다. 여기서 이 논문에서 실험한 버섯 농장처럼 재배 장소가 18개의 동으로 나뉘어져 있을 경우 동 별의 상황을 얻기 위해서는 18개의 동의 정보를 얻어올 필요가 있어 이 때 DBMS에 보내는 쿼리의 양이 동의 갯수인 18번이게 된다. 이 방식은 DBMS에 부하를 줄 뿐만 아니라 처리 속도도 느리기 때문에 한번의 쿼리로 데이터를 검색해 찾을 수 있는 MySQL의 프로시저(Procedure) 기능을 이용했다. 이 프로시저의 기능을 이용하면 DB서버에 임시로 데이터를 저장할 테이블을 만들고 그 안에 18개의 동 정보를 담은 후 한번에 클라이언트로 전송함으로써, 쿼리의 수를 18번에서 1번으로 줄이는데 가능하며 임시 테이블로 메모리를 사용하기 때문에 임시로 데이터를 저장시 발생할 수 있는 오버헤드가 적다. 또한 쿼리를 처리하는 동안 네트워크를 사용하지 않으므로 네트워크 통신에 의한 지연 현상도 적어진다.

3.4 시스템 구현 화면

기본적으로 프로그램 첫 화면에서는 18개 동에 대해서 현재 상황과 입상일 등, 각 동에 대한 정보를 눈에 보여주며 각 메뉴로 접근하여, 데이터 분석 및 저장관리를 한다. 입력 화면에서는 각 동의 현재 상황이나, 수확량 등에 대한 상태를 입력하게 되는데 이는 입력 시 NAS에 연결된 데이터베이스에 저장되며, 그림 6와 같은 쿼리문을 통하여 작업이 이루어진다. 그 후 데이터에 관한 분석 창을 통해 데이터에 대한 통계 그래프를 확인하여 통계자료로 사용이 가능하다.



[그림 7] 메인화면

[그림 8] 입력화면



[그림 9] 등급별 판매량 평균치 분석 그래프

4. 결론과 향후 발전방향

현 시스템의 구현상 데이터가 자동으로 입력되지 않고 생산 시마다 관리자가 일일이 입력을 해 주어야 한다는 것에 있어서 불편한 점이 있다. 이 점은 추후 센서를 개발 후 적용이나, 농장에 맞는 공정시스템을 도입 후 자동입력 방식으로 변환한다면 이는 농장에 있어서 더욱 더 효율이 높고 자동화되어 좋은 시스템으로 발전할 가능성이 있다.

감사의 글

본 연구는 중소기업청에서 지원하는 2014년도 산학연협력 기술개발사업의 연구수행으로 인한 결과입니다.

참고문헌

[1] "Gartner"
<http://www.gartner.com/technology/research/top-10-technology-trends/>

[2] 찰스 페졸드, "C#과 XAML을 이용한 .NET 3.0 윈도우 프로그래밍 찰스 페졸드의 WPF" 2007.09

[3] 박상현, "너를 자극하는 C# 4.0 프로그래밍" 2011.08

[4] MSDN
<http://msdn.microsoft.com/en-us/default.aspx>