데이터베이스 TERM

MySQL을 사용하여 홈페이지를 만들고 query 사용하기

제출일	2017.06.24	**전공**	컴퓨터학과
과목	데이터베이스	**학번**	2015410108
담당교수	정재화 교수님	**이름**	도지윤

서론

나는 컴퓨터학과지만 php나 http 같은 웹페이지를 만드는 언어에 대해 배운적이 없었다. 그래서 텀 프로젝트 계획이 나오자마자 조금씩 진행하였다.

내가 만든 과제는 '지윤톡'이다. '지윤톡'은 '카카오톡'을 모티브로 만든 홈페이지로, '카카오톡'의 대표적인 기능인 메세지 전달, 카카오페이와 기프티콘을 주고 받는 것을 할 수 있도록 하였다. 처음 계획서를 낼 때 정해놓은 entity와 relation에서 수정 과정이 있었고, 최종으로 결과의 entity와 relation은 다음과 같다.

Entity

user	'지윤톡'에 가입한 회원들
pfood	'지윤톡'에서 살 수 있는 음식 기프티콘
pflower	'지윤톡'에서 살 수 있는 꽃 기프티콘
message	'지윤톡'에서 전송할 수 있는 메세지
receipt	'지윤톡'에서 기프티콘을 구입한 영수증
foraccount	'지윤톡'에 돈을 충전한 영수증

Relation

user – message	사용자가 메세지를 주고 받는다.
user – receipt	사용자가 기프티콘을 다른 상대방에게 보낸다.
user – receipt	사용자가 기프티콘을 자신에게 보낸다.
receipt – pfood	사용자가 음식 기프티콘을 구매한다.
receipt – pfood	사용자가 음식 기프티콘을 선물한다.
receipt – pflower	사용자가 꽃 기프티콘을 구매한다.
receipt – pflower	사용자가 꽃 기프티콘을 선물한다.
user – foraccount	사용자가 기프티콘을 사기 위해서 필요한 돈을 충전한다.
user – foraccount	관리자는 모든 사용자들의 입금 내역을 볼 수 있다.
user – receipt	관리자는 모든 사용자들의 기프티콘 구매내역을 볼 수 있다.
user – message	관리자는 모든 사용자들의 메세지를 볼 수 있다.

이렇게 총 6개의 entity와 11개의 relation이 탄생하였다.

이 엔티티들의 schema 정의 코드는 다음과 같다.

```
CREATE TABLE foraccount (
  receiptID int(11) NOTNULL,
  userID char(10) NOTNULL,
  account int(11) NOTNULL,
  date datetime NOTNULL DEFAULT CURRENT_TIMESTAMP
)
CREATE TABLE message (
  mID int(10) NOTNULL,
  date datetime NOTNULL DEFAULT CURRENT_TIMESTAMP,
  post varchar(40) NOTNULL,
  rID char(10) NOTNULL,
  sID char(10) NOTNULL
)
```

```
CREATE TABLE pdoll (
  pID char(10) NOTNULL,
  money int(11) NOTNULL,
  count int(11) NOTNULL,
  type char(10) DEFAULT NULL
)
CREATE TABLE pflower (
  pID int(11) NOTNULL,
  money int(11) NOTNULL,
  count int(11) NOTNULL,
  type char(10) DEFAULT NULL
)
CREATE TABLE pfood (
  pID int(11) NOTNULL,
  money int(11) NOTNULL,
  count int(11) NOTNULL,
  type char(10) DEFAULT NULL
)
CREATE TABLE receipt (
  receiptID int(11) NOTNULL,
  rID char(10) NOTNULL,
  date datetimeNOTNULL DEFAULT CURRENT_TIMESTAMP,
  money int(11) NOTNULL,
  paid tinyint(1) NOTNULL,
  pID1 char(10) NOTNULL,
  sID char(10) NOTNULL
)
```

그리고, 이 table들을 Xampp 커널에서 join 시켜보았더니 결과는 다음과 같았다.

```
select * from user U inner join (receipt R inner join pfood F on R.pID1=F.pID) on U.userID=R.rID where U.userID='dbtest'
```

userID	PW	name	address	phone	account	receiptID	rID	date	money	paid	pID1	sID	pID	money	count	type
dbtest	test	test	test	test	78000	33	dbtest	2017-06-23 22:21:48	10000	1	11	dbtest	11	10000	99	복분자

이 query의 결과로 dbtest라는 아이디를 가진 user가 어떤 음식 기프티콘을 선물받았는 지를 알 수 있다. 이 결과를 도출하기 위해서 user, receipt, pfood의 3개의 table을 join 시켰다.

본론

이번 과제의 check list는 아래와 같았다.

CHECK LIST

회원 관리(가입) 기능이 포함되어야 하며 회원의 유형에는 관리자와 일반사용자가 있다.	완료
로그인 후 사용되는 기능들에서는 cookie를 이용하여 회원 정보를 활용할 수 있도록 해야 함.	완료
구현되는 시스템은 관리자 기능과 일반 사용자 기능을 포함해야 한다.	완료
위 기능은 관리자/일반사용자에 따라 기능 및 화면을 달리 구성해야 한다.	완료

구현되는 시스템은 핵심 데이터의 삽입, 수정, 삭제, 검색 기능이 반드시 포함되어야 한다.	완료
특정 정보의 검색 결과는 두 단계(검색 결과 전체에 대한 간략 정보와 특정 검색 결과에 대한 상세정보)로 볼 수 있도록 구현해야 한다.	완료
구현되는 기능에서 필요한 정보이지만 시스템 구현 범위 내에 해당 데이터 생성(insert) 기능이 포함되어 있지 않는 경우 해당 테이블의 데이터는 DBMS에서 직접 생성하여 사용한다.	완료
시스템 구현 시 둘 이상의 테이블이 조인되어 처리되어야 하는 기능이 반드시 포함되어야 한다.	완료
시스템 구현 시 transaction 개념을 적용시켜 개발해야 한다.	완료

모든 요구 조건을 충족하였으며, 로그인 후 사용되는 기능은 cookie 대신 session을 사용하였다.

모든 코드는 config.php 파일을 참조하도록 되어있는데, config.php 파일은 db와 서버를 연결시키는 코드이다. Github에 올렸는데, 비밀번호 부분만 ****로 수정하여 업로드 하였다.

먼저, 회원가입을 하는 부분인 registrater.php 이다.

```
        <article class="container">
          <div class="page-header">
            <h1>회원가입 <small>Sign up</small></h1>
          </div>
          <div class="col-md-6 col-md-offset-3">
              <div class="form-group">
                <label for="userID">아이디</label>
                <input type="text" class="form-control" id="userID" placeholder="아이디">
              </div>

              <div class="form-group">
                <label for="PW">비밀번호</label>
                <input type="password" class="form-control" id="PW" placeholder="비밀번호">
              </div>

              <div class="form-group">
                <label for="PW1">비밀번호 확인</label>
                <input type="password" class="form-control" id="PW1" placeholder="비밀번호 확
인">
                <p class="help-block">비밀번호 확인을 위해 다시 한번 입력 해 주세요.</p>
              </div>
              <div class="form-group">
                <label for="name">이름</label>
                <input type="text" class="form-control" id="name" placeholder="이름을 입력해 주
세요">
              </div>
              <div class="form-group">
                <label for="address">주소</label>
                <input type="text" class="form-control" id="address" placeholder="주소를 입력해
```

```
주세요">
                </div>
                <div class="form-group">
                  <label for="phone">전화 번호</label>
                  <input type="text" class="form-control" id="phone" placeholder="-를 제외한 전화 번호를 입력해 주세요">
                </div>

                <div class="form-group text-center">
                  <button type="submit" class="btn btn-info" onclick="RegisterUser()">회원가입<i class="fa fa-check spaceLeft"></i></button>
                  <button type="submit" class="btn btn-warning" onclick="Return()">가입취소<i class="fa fa-times spaceLeft"></i></button>
                </div>
            </div>
        </article>
    <script>
    function Return() {
            if (confirm("회원가입을 취소할까요?") ==true)
                    {    window.location ="index.php";}
            else {    return;      }
        }
    function RegisterUser() {
        if (document.getElementById("PW").value == document.getElementById("PW1").value) {
            var request = new XMLHttpRequest();
            var params ="?userID="+ document.getElementById("userID").value +"&PW="+ document.getElementById("PW").value
                +"&name="+ document.getElementById("name").value +"&address="+ document.getElementById("address").value
                +"&phone="+ document.getElementById("phone").value;
            request.open("GET", "memberSave.php"+ params, true);
            request.send(null);
            alert("Success!");
            location.replace("index.php");
        }
        else {
            alert("Passwords not same");
        }
    }
</script>
```

이 부분은 memberSave.php 라는 새로운 파일을 이용하여 registrater.php로 받은 사용자의 가입 정보를 db에 저장하도록 하였다.

```
<?php
    $conn = mysqli_connect("127.0.0.1","root","****","2017dbterm");
    $data_stream="".$_GET['userID']."','".$_GET['PW']."','".$_GET['name']."','".$_GET['address']."','".$_GET['phone']."','".$_GET['account']."";
    $query ="insert into User(userID, PW, name, address, phone, account) values (".$data_stream.")";
    $result = mysqli_query($conn, $query);
    if($result) {
      echo "1";
      header("location: index.php");
  }
    else
      echo "-1";
```

```
    mysqli_close($conn);
?>
```

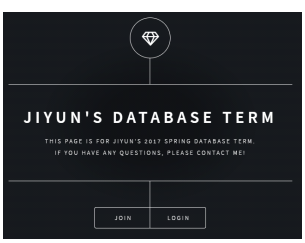
그 후 저장이 되면 index.php 로 들어가서 로그인을 할 수 있도록 location을 지정해 주었다.

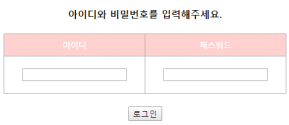
다음으로 login 부분이다.

```
<?php
    include("config.php");  //DB연결을 위한 config.php를 로딩합니다.
    session_start();    //세션의 시작
    if($_SERVER["REQUEST_METHOD"] =="POST"){
    $myusername=addslashes($_POST['username']);
    $mypassword=addslashes($_POST['password']);

    if ($myusername=='admin'&& $mypassword=='pass') {
        $_SESSION['login_user']=$myusername;

        header("location: index3.php");
    }
    else {
        $sql="SELECT userID FROM 2017dbterm.user WHERE userID='$myusername' and PW='$mypassword'";
    //echo $sql;
        $result=mysql_query($sql);
        $count=mysql_num_rows($result);
        //echo $count;
        if($count==1)  //count가 1이라는 것은 아이디와 패스워드가 일치하는 db가 하나 있음을 의미합니다.
        {
            $_SESSION['login_user']=$myusername;

            header("location: welcome.php");  // welcome.php 페이지로 넘깁니다.
        }
        else
        {
            echo "<script>alert(₩"아이디 또는 비밀번호가 잘못되었습니다.₩");</script>";
        }
    }
}
?>
```

간단한 웹페이지라서 비밀번호를 암호화하지는 않았다. 그리고, 관리자와 일반 사용자를 구별하기 위해서 관리자 계정은 admin/pass 라고 하나를 설정해두었고, 그 외는 다 일반 사용자로 분류하였다. 그래서 일반 사용자가 로그인을 하면 welcome.php 화면으로 넘어가고, 관리자가 로그인을 하면 index3.php로 넘어가게 된다.

회원가입과 로그인 과정을 화면으로 보면 다음과 같다.

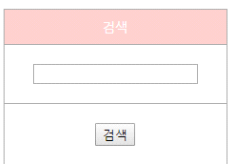




먼저, '**일반 사용자**' 모드로 접속을 하면 보이는 화면은 다음과 같다.

사용자의 아이디를 먼저 보여주고, 홈페이지로 이동할지 로그아웃을 할지 선택하도록 한다. 로그아웃을 하면 세션이 삭제되고 처음 화면으로 돌아간다. 따라서, 홈페이지로 이동을 하도록 하겠다.

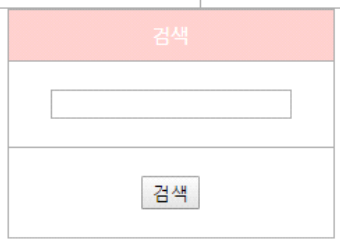
처음 회원가입과 로그인을 하던 홈과는 변화가 있다. 메뉴가 훨씬 늘었는데 선물(꽃), 선물(음식) 메뉴에서는 어떤 기프티콘이 있는지 확인할 수 있고 충전하기 메뉴를 통해 돈을 충전할 수 있다. 그리고 선물하기에서 실제로 돈을 사용하여 상대방에게 선물을 보내거나 무료로 메세지를 보낼 수 있다. 내 정보에서는 내가 입금했던 내역, 선물한 내역, 받은 내역, 문자 메세지 내역을 확인할 수 있다.



선물 번호	음식 이름	가격	남은 수량	선물하기
1	포테이토피자	12000	50	선물하기
2	치즈피자	12000	50	선물하기
3	콤비네이션피자	12000	50	선물하기
4	황금올리브치킨	21000	100	선물하기
5	황금양념치킨	22000	100	선물하기

선물(음식) 메뉴를 클릭하면 위와 같은 화면이 나온다. 이 때 '피자'를 검색하면,

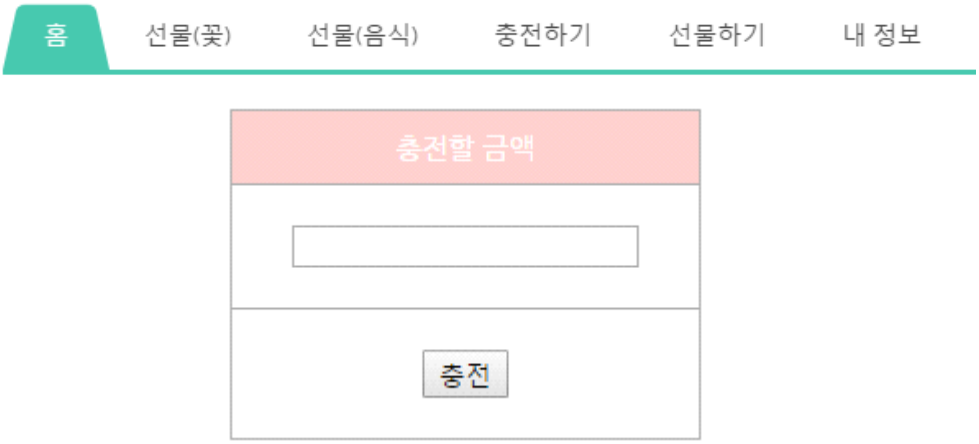
선물 번호	음식 이름	가격	남은 수량
1	포테이토피자	12000	50
2	치즈피자	12000	50
3	콤비네이션피자	12000	50



선물 중 '피자'라는 단어가 포함된 결과를 가져온다. 검색은

```
SELECT*FROM 2017dbterm.pfood WHERE type LIKE'%$key%' or pID like'%$key%'
```

위의 쿼리를 이용하였는데, 선물 번호와 선물의 이름을 가지고 검색을 할 수 있도록 하였다. 그리고 모든 메뉴 화면에 네비게이션 바를 두어 모든 페이지에 접근이 용이하도록 설정했다.



'지윤톡'에서 사용할 돈을 충전하는 페이지는 이런 모양이다. 박스에 원하는 액수를 입력한 뒤 충전을 누르면

```
$money=addslashes($_POST['money']);
$insql="INSERT INTO 2017dbterm.foraccount (receiptID, userID, account) VALUES (NULL, '$login_session', '$money')";
mysql_query($insql);
$sql="UPDATE 2017dbterm.user SET account=account+$money WHERE userID='$login_session'";
mysql_query($sql);
echo "<script>alert(₩"충전이 완료되었습니다.₩");</script>";
```

위 쿼리문이 동작하고 foraccount에 충전 내역에 관한 record가 insert되고, user는 새로운 account로 update된다.

다음으로, 선물하기 메뉴이다.

정보	입력
받는 사람 아이디	
선물 번호	
메세지	
선물하기	선물

선물 번호	선물 이름	가격	남은 수량
1	포테이토피자	12000	50
2	치즈피자	12000	50

선물하기 메뉴에선 종류에 상관없는 모든 기프티콘을 볼 수 있다.

선물을 보내든, 메세지를 전송하든 간에 받는 사람의 아이디를 꼭 적어야 한다. 그리고 이 user가 선물을 보내고자 하는지, 메세지를 보내고자 하는지 구별은 선물 번호로 하게 되어있다.

만약 선물 번호에 input이 있다면, 선물을 보내게 되고, 선물 번호에 input이 없고 null이면 메세지를 전송하게 된다. 선물의 번호를 통해서 꽃인지 음식인지 구별을 하게 되는데, 선물 번호가 50000보다 작으면 음식이고 크면 꽃 선물이다. 선물을 보낼 땐 선물의 가격만큼 account도 깎이고, 선물의 남은 수량도 감소한다. 또한, receipt에 결제 내역이 추가된다. 그러나, 선물의 가격이 내가 가진 돈보다 비싸면 구매할 수 없도록 충전을 먼저 하라는 알림이 뜬다.

이 부분은 다음 쿼리문으로 동작한다.

```
if($_SERVER["REQUEST_METHOD"] == "POST"){

        $sID=addslashes($_POST['sID']);
        $pID=addslashes($_POST['pID']);
        $msg=addslashes($_POST['msg']);
        if ($pID>0) {
                $acc = mysql_query("SELECT * FROM 2017dbterm.user WHERE
userID='$login_session'");
                $acco = mysql_fetch_array($acc);
                $account = $acco['account'];
                //echo $account;
                  //echo "선물전송";
                  if ($pID<50000) {
                        $sql="UPDATE 2017dbterm.pfood SET count=count-1 WHERE
pID='$pID'";
                        $money = mysql_query("SELECT * FROM 2017dbterm.pfood
WHERE pID='$pID'");
                        $a = mysql_fetch_array($money);
                        $money1=$a['money'];
                  } else {
                        $sql="UPDATE 2017dbterm.pflower SET count=count-1
```

```
WHERE pID='$pID'";
                    $money = mysql_query("SELECT * FROM 2017dbterm.pflower
WHERE pID='$pID'");
                    $a = mysql_fetch_array($money);
                    $money1=$a['money'];
                }
                if ($money1<=$account) {
                    mysql_query($sql);
                    $sql1="INSERT INTO 2017dbterm.receipt (receiptID, rID,
money, paid, pID1, sID) VALUES (NULL, '$sID', '$money1', 1, '$pID', '$login_session')";
                mysql_query($sql1);
                //echo "영수증 생성 완료";
                 $sql2="UPDATE 2017dbterm.user SET account=account-$money1 WHERE
userID='$login_session'";
                mysql_query($sql2);
                //echo "돈 빼는것도 완료";
                echo "<script>alert(₩"선물 전송이 완료되었습니다.₩");</script>";
            } else {
                echo "<script>alert(₩"잔액이 부족합니다. 충전 후 이용해주세
요.₩");</script>";
                }

        } else {
                //echo "메세지전송";
                  $sql1="INSERT INTO 2017dbterm.message (mID, rID, sID, post)
VALUES (NULL, '$sID', '$login_session', '$msg')";
                mysql_query($sql1);
                echo "<script>alert(₩"메세지 전송이 완료되었습니다.₩");</script>";
        }
    }
```

그리고, 친구가 나에게 보낸 기프티콘을 확인하거나, 나의 '지윤톡' 사용내역을 보기 위해 내 정보라는 메뉴가 존재한다.

내 정보 메뉴에서 알 수 있는 정보들은 다음 그림을 통해 제시하겠다.

로그인정보

정보	내 정보
아이디	dbtest
이름	test
주소	test
연락처	test
계좌	78000

받은 메세지

받은 날짜	내용	보낸 사람
2017-06-23 22:18:33	안녕이야	dbtest
2017-06-23 22:23:22	잘 받았어. 고마워. ^^	kalmpink

보낸 메세지

보낸 날짜	내용	받는 사람
2017-06-23 22:18:33	안녕이야	dbtest

받은 선물함

받은 날짜	받은 선물	수량	가격	보낸 사람
2017-06-23 22:21:48	복분자	99	10000	dbtest
2017-06-23 22:24:13	장미	1000	1000	kalmpink

충전내역

날짜	금액
2017-06-23 22:21:27	100000

보낸 선물함

보낸 날짜	보낸 선물	가격	받는 사람
2017-06-23 22:21:48	복분자	10000	dbtest
2017-06-23 22:22:03	포테이토피자	12000	kalmpink

이렇게 총 6개의 표가 차례대로 제시된다.

이 표들을 통해 홈페이지에서 이루어졌던 user의 모든 행동들을 확인할 수 있다.

다음으로, '**관리자 모드**'로 수행을 해보도록 하겠다. 관리자 모드로 로그인을 하면 index3.php 화면으로 넘어가게 된다. 이 화면은 다음과 같다.

이번엔, 모든 메뉴가 '관리'와 관련되어 있는 것을 확인할 수 있다.

유저

아이디	비밀번호	이름	핸드폰번호	주소	잔액
admin	pass	관리자	0234714106	서울특별시 서초구 방배로20길 5 (방배동, 5층)	0
dbtest	test	test	test	test	78000
kalmpink	wldbsdl1	도지윤	01020503417	경기도	0
rkdlq	rkdlq	가입	01020303417	가입한다	0

유저관리 부분은 user가 가입하면서 적었던 모든 부분과, user가 '지윤톡'을 사용하면서 충전한 잔액 중 남은 잔액을 같이 표시하도록 하였다.

상품관리 부분은 다음과 같다.

상품(꽃) 등록

꽃 번호	꽃 이름	가격	남은 수량

추가

상품(꽃) 수정

선물 번호	음식 이름	가격	남은 수량	선물하기
50001	튤립	1700	10000	삭제 수정
50002	안개꽃	15000	1000	삭제 수정

상품관리 부분에서는 insert, delete, update가 모두 가능하다. Insert는 맨 위의 테이블에 모든 정보를 적고 '추가'버튼을 누르면 insert가 수행되며, 삭제와 수정은 각 record 별로 이루어지도록 별개의 버튼을 두었다. 삭제를 누르면 '정말 삭제하시겠습니까?' 라는 확인 메세지가 뜨고 이 때 확인을 누르면 정말 레코드가 삭제된다.

수정은 다음과 같이 이루어진다.

상품을 수정합니다.

	상품번호	이름	가격	수량
수정전	50001	튤립	1700	10000
수정후				

수정

수정하기 전의 원래 record를 보여주고, 어떻게 수정할 것인지 관리자가 직접 적은 뒤 수정을 누르면 수정이 완료된다.

수정을 하는 과정에서

이 방법을 통해 modify.php라는 새로운 페이지에 정보를 전달하고, 이 페이지에서 그 특정 record 부분만 자세하게 다루도록 구현하였다. 이렇게 한 이유는 **정보의 검색 결과를 두 단계로 나누는 부분(check list)** 때문이다. 이번 텀에 내가 사용한 entity는 검색을 통해 간단한 정보를 나타내거나 자세한 정보를 나타낸다고 나누기 어려울 정도로 필수적인 정보만 가지고 있는 단순한 entity기 때문에 이 조건을 사용할 다른 방법을 생각하던 중 수정 과정에서 특정 record를 가지고 새로운 페이지에 접속하도록 하여 그 record에 대한 정보를 통해 일을 진행하도록 처리하였다.

데이터의 삽입, 수정, 삭제 부분을 flower_admin.php, food_admin.php 파일이 진행하고 동작하는 쿼리문은 다음과 같다.

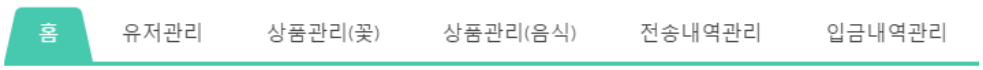
INSERT

```
$sql1="INSERT INTO 2017dbterm.pflower (pID, money, count, type) VALUES ('$pID1', '$money1', '$count1', '$name1')";
```

```
        mysql_query($sql1);

DELETE
 $sql = "DELETE FROM 2017dbterm.pflower WHERE pID = '$pID'";
 mysql_query($sql);

UPDATE
 $sql1="UPDATE 2017dbterm.pfood SET type='$type' WHERE pID='$pID1'";
 mysql_query($sql1);
 $sql2="UPDATE 2017dbterm.pfood SET money='$money' WHERE pID='$pID1'";
 mysql_query($sql2);
 $sql3="UPDATE 2017dbterm.pfood SET count='$count' WHERE pID='$pID1'";
 mysql_query($sql3);
 $sql="UPDATE 2017dbterm.pfood SET pID='$pID' WHERE pID='$pID1'";
 mysql_query($sql);
```

Update를 각 attribute 별로 구별한 이유는 특정 attribute만 수정하는 것을 대비하기 위해서이고, pID를 가장 마지막에 바꿔주어서 다른 record를 수정하는 일이 없도록 하였다. 즉, Insert, delete, update는 상품 table을 중심으로 이루어지는데 pID가 super key이기 때문에 다른 attribute들은 겹칠 수 있으므로 pID를 기준으로 먼저 구별을 하게 고, pID가 먼저 바뀌면 다른 record를 나타낼 수 있기 때문에 pID를 가장 마지막에 수정하도록 한 것이다.

상품관리는 이렇게 이루어지고, 전송내역관리와 입금내역관리는 단순히 사용자들의 내역을 보여주는 형식으로 나타난다.



선물 전송내역

영수증번호	날짜	발신자	수신자	상품번호	상품명	상품가격
33	2017-06-23 22:21:48	dbtest	dbtest	11	복분자	10000
34	2017-06-23 22:22:03	dbtest	kalmpink	1	포테이토피자	12000
36	2017-06-23 22:24:13	kalmpink	dbtest	100004	장미	1000

메세지 전송내역 관리

메세지번호	날짜	내용	발신자	수신자
1	2017-06-23 22:18:33	안녕이야	dbtest	dbtest
2	2017-06-23 22:23:22	잘 받았어. 고마워. ^^	kalmpink	dbtest

입금내역

영수증번호	날짜	입금인	입금액
11	2017-06-23 22:21:27	dbtest	100000
12	2017-06-23 22:23:38	kalmpink	1000

이 부분을 통해 단순하지만 '지윤톡'에서 이뤄지는 모든 일을 관리자가 볼 수 있게 만들었다. 마찬가지로 네비게이션 바를 두어 페이지의 원활한 이동이 가능하도록 설정하였다.

마지막으로, 모든 php 파일에 들어가는 config.php 파일의 내용은 다음과 같다.

```
<?php
$db_host = "127.0.0.1";
$db_user = "root";
$db_passwd = "****";
$db_name = "2017dbterm";
$conn = mysqli_connect($db_host, $db_user, $db_passwd, $db_name);
if (mysqli_connect_errno($conn)) {
        echo "fail to connect DB :" . mysqli_connect_error();
} else {

}
?>
```

데이터베이스의 비밀번호는 ****으로 수정하였다.

소감

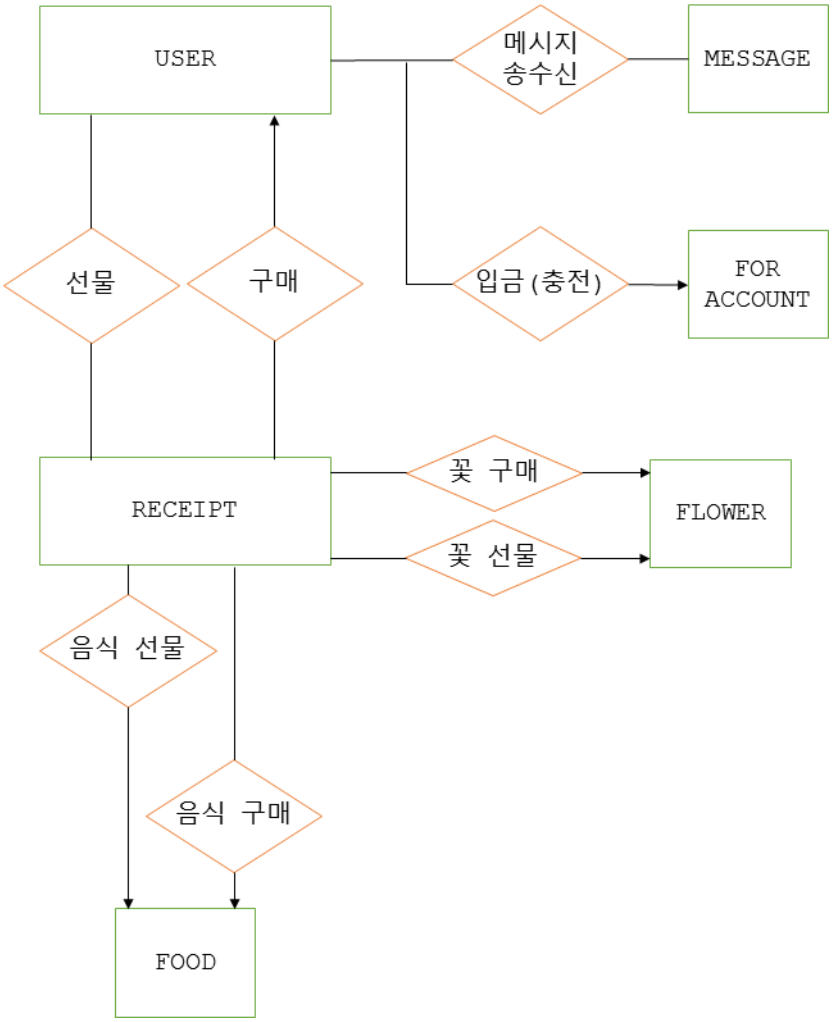
이번 학기에는 전공을 5개 들어서 중간 텀까지 거의 7개의 텀 프로젝트가 있었는데, 데이터베이스 텀이 가장 재미있었다. 생소한 부분이라서 접근은 어려웠으나, 실제로 로그인과 회원가입 등을 구현하여 내가 직접 만든 페이지가 생기기도 하고 빨리 결과를 확인할 수 있기 때문에 코드를 컴파일 하거나 컴파일을 위한 용량이 큰 프로그램을 까느라 소모했던 시간을 절약할 수 있었다.

그리고, 중간고사 때 query문이 시험 범위였는데 그 때 공부하던 것 보다 실제로 하면서 진행하니 덜 지치고 좋았다.

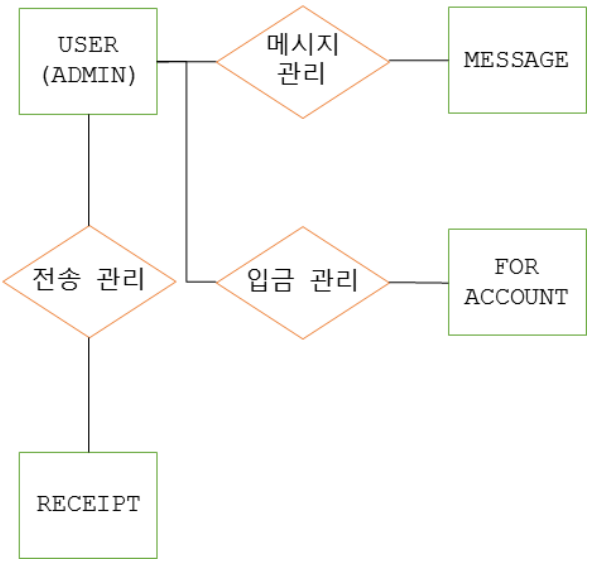
모든 코드와 파일들은 ('https://github.com/yunmap/2017DBterm')에 올려두었다. 지윤톡의 ER 다이어그램은 부록과 같고 USER를 관리자와 일반 사용자로 구분하여 나타냈다.

부록

-일반사용자-


USER
메시지
송수신
MESSAGE
선물
구매
입금 (충전)
FOR
ACCOUNT
RECEIPT
꽃 구매
꽃 선물
FLOWER
음식 선물
음식 구매
FOOD

-관리자-


USER
(ADMIN)
메시지
관리
MESSAGE
전송 관리
입금 관리
FOR
ACCOUNT
RECEIPT