

**[Term Project 2]**

**Development of Web DB application  
with transaction**

#2017320233 김채령

# System Refinement

1. Insert, Update, Delete가 발생하는 작업을 transaction 처리
2. campus, cafe 정보 수정 모듈 개선

# 1. Insert, Update, Delete가 발생하는 작업을 transaction 처리

## - transaction 처리 해당 모듈

campus\_insert .php  
campus\_modify .php  
campus\_delete .php

campus\_info\_insert .php  
campus\_info\_modify .php  
campus\_info\_delete .php

cafe\_insert. .php  
cafe\_modify .php  
cafe\_delete.php

cafe\_info\_insert .php  
cafe\_info\_modify. .php  
cafe\_info\_delete.php

event\_insert .php  
event\_modify .php  
event\_delete .php

# 1. Insert, Update, Delete가 발생하는 작업을 transaction 처리

수정한 부분: 데이터베이스에 insert, update, delete가 발생하는 작업을 transaction 처리

데이터베이스에 연결

관련 php에서 post로 정보 받아 오기

**1. auto commit 해제**

**2. isolation level을 serializable로 설정**

**3. transaction 시작**

insert, update, delete 쿼리 실행

if 쿼리 수행 실패 => **4-1. 수행 전으로 roll back**

if 쿼리 수행 성공 => **4-2. 수행 내역 commit**

# 1. Insert, Update, Delete가 발생하는 작업을 transaction 처리

(\*\* insert, modify, delete와 관련한 모듈은 동일한 수정을 했으므로, campus와 관련한 모듈을 예시로 설명합니다)

e.g. campus\_insert.php

```
$conn = dbconnect($host,$dbid,$dbpass,$dbname); -> 데이터베이스 연결

$campus_name = $_POST['campus_name'];
$building = $_POST['building'];
$floor = $_POST['floor'];
$campus_seat = $_POST['campus_seat'];
$campus_explain = $_POST['campus_explain'];

-> campus_form.php에서
입력한 정보 post로 받아옴

$double_query = "select * from campus where campus_name = '$campus_name'";
$double_res = mysqli_query($conn, $double_query);
$double_count = mysqli_num_rows($double_res);

if($double_count != 0){
    msg("이미 존재하는 교내 공간 이름입니다. 다른 교내 공간 이름을 입력하세요");
}
else{
    $ret = mysqli_query($conn, "insert into campus (campus_name, building, floor,
    -> 쿼리 실행
}

if(!$ret)
{
    echo mysqli_error($conn);
    msg('Query Error : '.mysqli_error($conn));
}
else
{
    s_msg ('성공적으로 입력 되었습니다');
    echo "<meta http-equiv='refresh' content='0;url=campus_list.php'>";
}
```



```
$campus_name = $_POST['campus_name'];
$building = $_POST['building'];
$floor = $_POST['floor'];
$campus_seat = $_POST['campus_seat'];
$campus_explain = $_POST['campus_explain'];

-> autocommit 해제, transaction level을 serializable로 설정, transaction 시작
mysqli_query($conn, "set autocommit = 0"); // autocommit 해제
mysqli_query($conn, "set transaction isolation level serializable"); // isolation
mysqli_query($conn, "begin"); // begins a transaction

/*added*/

$double_query = "select * from campus where campus_name = '$campus_name'";
$double_res = mysqli_query($conn, $double_query);
$double_count = mysqli_num_rows($double_res);

if($double_count != 0){
    msg("이미 존재하는 교내 공간 이름입니다. 다른 교내 공간 이름을 입력하세요");
}
else{
    $ret = mysqli_query($conn, "insert into campus (campus_name, building, floor,

if(!$ret)
{
    -> 쿼리 수행 실패: 수행 전으로 roll back
    mysqli_query($conn, "rollback"); // 교내 공간 등록 query 수행 실패, 수행 전으
    echo mysqli_error($conn);
    msg('Query Error : '.mysqli_error($conn));
}
else
{
    -> 쿼리 수행 성공: 수행 내역 commit
    mysqli_query($conn, "commit"); // 교내 공간 등록 query 수행 성공, 수행 내역
    s_msg ('성공적으로 입력 되었습니다');
    echo "<meta http-equiv='refresh' content='0;url=campus_list.php'>";
}
```

# 1. Insert, Update, Delete가 발생하는 작업을 transaction 처리

(\*\* insert, modify, delete와 관련한 모듈은 동일한 수정을 했으므로, campus와 관련한 모듈을 예시로 설명합니다)

e.g. campus\_modify.php

```
$conn = dbconnect($host,$dbid,$dbpass,$dbname); -> 데이터베이스 연결

$campus_id = $_POST['campus_id'];
$campus_name = $_POST['campus_name'];
$building = $_POST['building'];
$floor = $_POST['floor'];
$campus_seat = $_POST['campus_seat'];
$campus_explain = $_POST['campus_explain'];

/*modified*/
$double_query = "select * from campus where campus_name = '$campus_name'";
$double_res = mysqli_query($conn, $double_query);
$double_count = mysqli_num_rows($double_res);

if($double['campus_id'] != $campus_id && $double_count != 0){
    msg("이미 존재하는 교내 공간 이름입니다. 다른 교내 공간 이름을 입력하세요");
}
else{ -> 쿼리 실행
    $ret = mysqli_query($conn, "update campus set campus_name = '$campus_name', b
}

if(!$ret)
{
    msg('Query Error : '.mysqli_error($conn));
}
else
{
    s_msg ('성공적으로 수정 되었습니다');
    echo "<meta http-equiv='refresh' content='0;url=campus_list.php'>";
}
```



```
$conn = dbconnect($host,$dbid,$dbpass,$dbname);

$campus_id = $_POST['campus_id'];
$campus_name = $_POST['campus_name'];
$building = $_POST['building'];
$floor = $_POST['floor'];
$campus_seat = $_POST['campus_seat'];
$campus_explain = $_POST['campus_explain'];

-> autocommit 해제, transaction level을 serializable로 설정, transaction 시작
mysqli_query($conn, "set autocommit = 0"); // autocommit 해제
mysqli_query($conn, "set transaction isolation level serializable"); // isolation level 설정
mysqli_query($conn, "begin"); // begins a transaction
/*added*/

/*modified*/
$double_query = "select * from campus where campus_name = '$campus_name' and campus_id != $campus_id";
$double_res = mysqli_query($conn, $double_query);
$double_count = mysqli_num_rows($double_res);

if($double_count != 0){
    msg("이미 존재하는 교내 공간 이름입니다. 다른 교내 공간 이름을 입력하세요");
}
/*modified*/
else{
    $ret = mysqli_query($conn, "update campus set campus_name = '$campus_name', building = '$building', f
}

if(!$ret)
{ -> 쿼리 수행 실패: 수행 전으로 roll back
    /*added*/
    mysqli_query($conn, "rollback"); // 교내 공간 수정 query 수행 실패. 수행 전으로 rollback
    /*added*/
    msg('Query Error : '.mysqli_error($conn));
}
else
{ -> 쿼리 수행 성공: 수행 내역 commit
    /*added*/
    mysqli_query($conn, "commit"); // 교내 공간 수정 query 수행 성공. 수행 내역 commit
    /*added*/
    s_msg ('성공적으로 수정 되었습니다');
    echo "<meta http-equiv='refresh' content='0;url=campus_list.php'>";
}
```

# 1. Insert, Update, Delete가 발생하는 작업을 transaction 처리

(\*\* insert, modify, delete와 관련한 모듈은 동일한 수정을 했으므로, campus와 관련한 모듈을 예시로 설명합니다)

e.g. campus\_delete.php

```
$conn = dbconnect($host,$dbid,$dbpass,$dbname); -> 데이터베이스 연결
$campus_id = $_GET['campus_id']; -> campus_list.php에서
                                     삭제 버튼 누른 campus_id를 post로 받아옴
$ret_fk = mysqli_query($conn, "delete from campus_info where campus_id = $campus_id");
$ret = mysqli_query($conn, "delete from campus where campus_id = $campus_id");
-> 쿼리 실행
if(!$ret)
{
    msg('Query Error : '.mysqli_error($conn));
}
else
{
    s_msg('성공적으로 삭제 되었습니다');
    echo "<meta http-equiv='refresh' content='0;url=campus_list.php'>";
}
```



```
$conn = dbconnect($host,$dbid,$dbpass,$dbname);

$campus_id = $_GET['campus_id'];

/*-> autocommit 해제, transaction level을 serializable로 설정, transaction 시작
mysqli_query($conn, "set autocommit = 0"); // autocommit 해제
mysqli_query($conn, "set transaction isolation level serializable"); // isolation level
mysqli_query($conn, "begin"); // begins a transaction
*/

$ret_fk = mysqli_query($conn, "delete from campus_info where campus_id = $campus_id");
$ret = mysqli_query($conn, "delete from campus where campus_id = $campus_id");

if(!$ret)
{
    -> 쿼리 수행 실패: 수행 전으로 roll back
    /*added*/
    mysqli_query($conn, "rollback"); // 교내 공간 삭제 query 수행 실패. 수행 전으로 roll
    /*added*/
    msg('Query Error : '.mysqli_error($conn));
}
else
{
    -> 쿼리 수행 성공: 수행 내역 commit
    /*added*/
    mysqli_query($conn, "commit"); // 교내 공간 삭제 query 수행 성공. 수행 내역 commit
    /*added*/
    s_msg('성공적으로 삭제 되었습니다');
    echo "<meta http-equiv='refresh' content='0;url=campus_list.php'>";
}
```

## 2. campus, cafe 정보 수정 모듈 개선

- 수정한 모듈: campus\_modify.php, cafe\_modify.php

\*\* 교내 공간 이름과 카페 상호는 unique하다고 가정

수정 전)

- 교내 공간이나 카페 공간 정보를 수정하려면 **무조건 공간명을 바꿔줘야** 해당 relation의 다른 정보도 수정이 가능했음.
- **수정을 하는 해당 공간명을 포함한** 이미 등록된 (교내/카페) 공간명과 같은 공간명으로 수정할 수 없음.
- 수정을 할 때는 이미 등록된 공간명과 겹치면 중복된 공간이 생기니까 단순히 이미 등록된 공간명과 같을 수 없도록 구현. 수정을 하고자 하는 (교내/카페)공간도 이미 등록된 공간에 포함되는 것을 고려하지 못함.

수정 후)

- **공간명을 바꿔주지 않고도** 좌석수 등의 해당 relation의 다른 정보도 수정이 가능함.
- 단, **수정을 하는 해당 공간명을 제외한** 이미 등록된 다른 (교내/카페) 공간명과 같은 공간명으로 수정할 수 없음.
- 공간명은 바뀌지 않고 해당 공간에 대한 부수 정보가 바뀌는 경우를 추가 고려함.



## 2. campus, cafe 정보 수정 모듈 개선

campus\_modify.php

```
$double_query = "select * from campus where campus_name = '$campus_name'";  
$double_res = mysqli_query($conn, $double_query);  
$double_count = mysqli_num_rows($double_res);  
  
if($double['campus_id'] != $campus_id && $double_count != 0){ -> 수행이 잘 안됐음  
    msg("이미 존재하는 교내 공간 이름입니다. 다른 교내 공간 이름을 입력하세요");  
}  
else{  
    $ret = mysqli_query($conn, "update campus set campus_name = '$campus_name', build  
}
```



```
$double_query = "select * from campus where campus_name = '$campus_name' and campus_id != $campus_id";  
$double_res = mysqli_query($conn, $double_query);  
$double_count = mysqli_num_rows($double_res);  
  
if($double_count != 0){  
    msg("이미 존재하는 교내 공간 이름입니다. 다른 교내 공간 이름을 입력하세요");  
} /*modified*/  
else{  
    $ret = mysqli_query($conn, "update campus set campus_name = '$campus_name', building = '$building', f  
}
```

## 2. campus, cafe 정보 수정 모듈 개선

cafe\_modify.php

```
$double_query = "select * from cafe where cafe_name = '$cafe_name'";  
$double_res = mysqli_query($conn, $double_query);  
$double_count = mysqli_num_rows($double_res);  
  
if($double['cafe_id'] != $cafe_id && $double_count != 0){ -> 수행이  
    msg("이미 존재하는 카페명입니다. 다른 카페를 입력하세요"); 잘 안됐음  
}  
else{  
    $ret = mysqli_query($conn, "update cafe set cafe_name = '$cafe_name'  
}
```



```
$double_query = "select * from cafe where cafe_name = '$cafe_name' and cafe_id != $cafe_id";  
$double_res = mysqli_query($conn, $double_query);  
$double_count = mysqli_num_rows($double_res);  
  
if($double_count != 0){  
    msg("이미 존재하는 카페명입니다. 다른 카페를 입력하세요");  
} /*modified*/  
else{  
    $ret = mysqli_query($conn, "update cafe set cafe_name = '$cafe_name', location = '$location'  
}
```

# The end of the report.

Thank you😊