jinyPHP 2권 - 17. 데이터수정

**17. 데이터수정**

========================

지금까지 우리는 테이블에 새로운 데이터의 삽입과 결과를 조회해 보았습니다. 이번에는 기존에 데이터를 수정하는 쿼리에 대해서 학습해 보도록 하겠습니다.

**17.1 전체 수정**

========================

새로운 쿼리 명령UPDATE 는 테이블의 데이터를 수정을 할 수 있습니다. 다음은 UPDATE의 기본 문법입니다.

**쿼리문법)**

|  |
| --- |
| UPDATE 테이블명 SET 컬럼명=값, 컬럼명=값, 컬럼명=값, …. ; |

테이블의 데이터를 수정하는 쿼리는 기존의 작성 방식과 약간 틀린 점이 있습니다. 변경하고자 하는 컬럼을 컬럼명 = 값 형태로 한쌍씩 작성합니다. 각각의 컬럼은 콤마(,)로 분리하시면 됩니다.

위와 같이 쿼리 명령을 작성하게 되면 테이블의 모든 데이터가 지정한 값을 변경이 됩니다. 테이블의 전체 데이터를 변경하는 것은 매우 흔하지 않는 일입니다. 또한 이러한 명령은 실수로 데이터를 모두 잃어 버릴 수 있는 매우 위험한 쿼리 명령입니다.

**17.1.1 쿼리 실습**

========================

전체 수정 실습을 위해서 기존 테이블에 새로운 컬럼을 하나 추가해 봅니다.

|  |
| --- |
| // 새로운 컬럼을 추가합니다.  mysql> alter table members add manager varchar(100);  Query OK, 0 rows affected (0.06 sec)  Records: 0 Duplicates: 0 Warnings: 0 |

컬럼을 새롭게 추가하게 되면 데이터는 기본적으로 NULL로 설정이 됩니다.

|  |
| --- |
| // 새롭게 등록된 컬럼은 모두 NULL 입니다.  mysql> select \* from members;  +----+----------+-----------+-------------+-------+---------+---------+  | Id | LastName | FirstName | Address | City | Country | manager |  +----+----------+-----------+-------------+-------+---------+---------+  | 1 | hojin | lee | shinchon | seoul | korea | NULL |  | 2 | jiny | NULL | NULL | NULL | korea | NULL |  | 3 | james | kim | youngdun-po | seoul | korea | NULL |  | 4 | eric | jung | seocho-gu | busan | korea | NULL |  +----+----------+-----------+-------------+-------+---------+---------+  4 rows in set (0.00 sec) |

NULL로 설정되어 있는 컬럼에 새로운 값을 모두 변경해 보도록 하겠습니다.

**update members set manager = 'infohojin';**

|  |
| --- |
| // 전체 manager 컬럼필드의 값을 수정합니다.  mysql> update members set manager = 'infohojin';  Query OK, 4 rows affected (0.01 sec)  Rows matched: 4 Changed: 4 Warnings: 0 |

수정 쿼리가 정상적으로 처리가 되었습니다. 수정한 내용을 확인 하기 위해서 select 명령으로 테이블을 확인해 봅니다.

|  |
| --- |
| // 수정한 값을 확인할 수 있습니다.  mysql> select \* from members;  +----+----------+-----------+-------------+-------+---------+-----------+  | Id | LastName | FirstName | Address | City | Country | manager |  +----+----------+-----------+-------------+-------+---------+-----------+  | 1 | hojin | lee | shinchon | seoul | korea | infohojin |  | 2 | jiny | NULL | NULL | NULL | korea | infohojin |  | 3 | james | kim | youngdun-po | seoul | korea | infohojin |  | 4 | eric | jung | seocho-gu | busan | korea | infohojin |  +----+----------+-----------+-------------+-------+---------+-----------+  4 rows in set (0.00 sec) |

새롭게 추가한 manager 컬럼에 설정한 값이 모두 변경이 되었습니다.

**17.1.2 PHP 실습**

========================

PHP 코드를 통하여 전체 수정을 실습해 봅니다. 사실상 수정 쿼리의 작업은 유용하면서도 매우 위험한 명령 입니다. 실수로 전체 데이터를 삭제하기는 것을 방지하기 위해서 이렇게 별도의 메소드 형태로 만들어 호출하게 되면 오류를 사전에 방지 할 수 있습니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // arr배열 데이터를 기분으로 테이블 전체 컬럼필드를 업데이트 합니다.  public function updateAll($tbname,$arr)  {  if ($tbname) {  $queryString = $this->arrQueryUpdate($tbname, $arr);    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE) {  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 업데이트!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  } |

**예제파일) update-01.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  $tbname = "members";  $arr = array('Country' => 'Korea');  $db->updateAll($tbname, $arr);  ?> |

출력화면)

mysql connected!

Country

쿼리성공] UPDATE members SET `Country`='Korea'

데이터 업데이트!

**17.2 조건 데이터수정**

========================

데이터를 수정하는 UPDATE 쿼리는 테이블 컬럼의 전체를 수정하게 됩니다. 전체의 수정을 할 필요성도 있겠지만, 이런 작업들은 많이 않습니다. 즉, 전체를 수정한다는 것은 다른 데이터들을 잊어버리는 손실이 발생될 수 있습니다. 수정된 작업은 다시 되돌릴 수 없습니다.

일반적으로 UPDATE 문법은 조건 WHERE 문과 함께 사용을 합니다. 특정 데이터를 조건으로 찾아서 데이터를 변경을 합니다.

**쿼리문법)**

|  |
| --- |
| UPDATE 테이블명 SET 컬럼명=value, 컬럼명=value,... WHERE 조건컬럼명=조건값 |

**17.2.1 쿼리 실습**

========================

테이블의 데이터를 수정을 할 때는 선택적으로 데이터 열을 지정해야 합니다. 데이터의 열을 지정할 때 많이 이용하는 것이 테이블의 프라이머리 키 입니다. 프라이머리 키는 테이블의 각각의 데이터로 중복되지 않은 고유의 키 입니다.

**update members set City='busan' where Id=4;**

위의 쿼리 처럼 Id 프라이머리 키를 이용하면 무난하게 데이터 한 개의 열을 지정할 수 있습니다.

|  |
| --- |
| mysql> **update members set City='busan' where Id=4;**  Query OK, 1 row affected (0.01 sec)  Rows matched: 1 Changed: 1 Warnings: 0 |

UPDATE를 통하여 하나 하나 데이터를 변경할때는 Primary Key 값인 Id 를 사용하면 좋습니다. 만일 , 다수의 데이터를 변경할때는 다른 조건값을 같이 사용하면 편리합니다.

|  |
| --- |
| mysql> **select \* from members;**  +----+----------+-----------+-------------+-------+---------+  | Id | LastName | FirstName | Address | City | Country |  +----+----------+-----------+-------------+-------+---------+  | 1 | hojin | lee | shinchon | seoul | korea |  | 2 | jiny | NULL | NULL | NULL | korea |  | 3 | james | kim | youngdun-po | seoul | korea |  | 4 | eric | jung | seocho-gu | busan | korea |  +----+----------+-----------+-------------+-------+---------+  4 rows in set (0.00 sec) |

위의 예제는 Id =4 인 회원의 도시를 변경하는 것입니다.

**17.2.2 PHP 실습**

========================

PHP 코드를 통하여 데이터를 변경하는 실습을 해봅니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // array 배열기반 update 쿼리를 생성합니다.  public function arrQueryUpdate($tbname, $arr)  {  $queryString = "UPDATE $tbname SET ";  $arrKey = array\_keys($arr);  for ($i=0;$i<count($arr);$i++) {  // $key = $arr[$i];  echo $arrKey[$i]."<br>";  $key = $arrKey[$i];  $queryString .= "`".$key."`='".$arr[$key]."',";  }  $queryString .= ";";  $queryString = str\_replace(",;"," ",$queryString);  return $queryString;  }  public function updateID($tbname,$arr,$Id)  {  if ($tbname) {  $queryString = $this->arrQueryUpdate($tbname, $arr);  $queryString .= "where Id='$Id'";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE) {  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 $Id 업데이트!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  } |

**예제파일) update-02.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  $tbname = "members";  $arr = array('FirstName' => 'hojin', 'LastName' => 'lee');  $db->updateID($tbname, $arr, 2);  ?> |

출력화면)

mysql connected!

FirstName

LastName

쿼리성공] UPDATE members SET `FirstName`='hojin',`LastName`='lee' where Id='2'

데이터 2 업데이트!

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // 조건 업데이트  public function updateWhere($tbname,$arr,$where){  if ($tbname) {  $queryString = $this->arrQueryUpdate($tbname, $arr);  $queryString .= "where $where";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE) {  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 업데이트!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  } |

**예제파일) update-03.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  $tbname = "members";  $arr = array('Country' => 'Korea');  $db->updateWhere($tbname, $arr, "Id>='2'");  ?> |

출력화면)

mysql connected!

Country

쿼리성공] UPDATE members SET `Country`='Korea' where Id>='2'

데이터 업데이트!

**17.3 정렬 수정**

========================

데이터의 조건을 설정하여 데이터를 수정하는 방법 이외에, 데이터를 정렬후에 LIMIT 를 통하여 데이터를 수정할 수 있습니다.

즉, 데이터를 수정할때는 수정하는 데이터의 범위를 지정하는 것이 중요합니다. 프라이머리 키 Id와 같이 데이터를 1개만 지정을 할 수도 있고, 그룹 등 다양한 방법으로 범위를 지정할 수도 있습니다.

**17.3.1 쿼리 실습**

========================

다음은 주문테이블을 프라이머리 키로 정렬하여 상위 2개의 데이터만 수정을 합니다.

**update members set manager='info' order by Id desc limit 2;**

|  |
| --- |
| mysql> **update members set manager='info' order by Id desc limit 2;**  Query OK, 2 rows affected (0.01 sec)  Rows matched: 2 Changed: 2 Warnings: 0  mysql> **select \* from members;**  +----+----------+-----------+-------------+-------+---------+-----------+  | Id | LastName | FirstName | Address | City | Country | manager |  +----+----------+-----------+-------------+-------+---------+-----------+  | 1 | hojin | lee | shinchon | seoul | korea | infohojin |  | 2 | jiny | NULL | NULL | NULL | korea | infohojin |  | 3 | james | kim | youngdun-po | seoul | korea | info |  | 4 | eric | jung | seocho-gu | busan | korea | info |  +----+----------+-----------+-------------+-------+---------+-----------+  4 rows in set (0.00 sec) |

**17.3.2 PHP 실습**

========================

PHP 코드를 통하여 실습을 해봅니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // 최상의 데이터 몇개만 업데이트합니다.  public function updateLimit($tbname, $arr, $num, $orderby)  {  if ($tbname){  if($num){  $queryString = $this->arrQueryUpdate($tbname, $arr);  if ($orderby){  $queryString .= " ORDER BY $orderby LIMIT $num";  } else {  $queryString .= " $tbname LIMIT $num";  }    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 업데이트!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 업데이트할 테이터 갯수가 없습니다.");  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  } |

**예제파일) update-04.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  $tbname = "members";  $arr = array('Country' => 'Korea');  // 최상위 데이터 2개만 갱신합니다.  $db->updateLimit($tbname, $arr, 2);  // LastName 정렬후에 데이터 2개만 갱신합니다.  $db->updateLimit($tbname, $arr, 2, "LastName desc");  ?> |

출력화면)

mysql connected!

Country

Error] UPDATE members SET `Country`='Korea' members LIMIT 2

Country

쿼리성공] UPDATE members SET `Country`='Korea' ORDER BY LastName desc LIMIT 2

데이터 업데이트!