jinyPHP 2권 - 008. 테이블수정

**08. 테이블 수정**

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

테이블은 대부분 DB 모델링과 같이 처음 데이터의 구조를 설계할 때 생성이 됩니다. 처음에 데이터의 모델구조가 잘 설계가 되었다고 한다면 크게 테이블의 수정 작업들은 필요가 없을 것입니다.

하지만 시스템을 운영하다 보면 새로운 요구나 기능의 개선으로 테이블의 구조는 약간씩 수정이 필요하기 마련 입니다. 또한 정규화 작업, 역정규화 작업등으로 인하여도 테이블의 수정 작업이 필요로 합니다.

테이블은 생성된 후에도 수정이 가능합니다. 테이블을 수정을 할 때는 ALTER 쿼리명령을 사용합니다. ALTER 명령은 크게 정의를 변경하는 MODIFY, 추가하는 ADD, 이름을 변경하는 CHANGE, 삭제하는 DROP 동작으로 구분해 볼 수 있습니다.

**08.1 컬럼추가**

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

생성된 테이블에 새로운 컬럼을 추가할 수 있습니다. 컬럼을 추가할 때는 기존의 컬럼명과 중복이 되지 안도록 이름을 잘 정의하여야 합니다. 컬럼명이 중복이 되면 쿼리명령은 실행오류를 발생합니다.

테이블에 컬럼을 추가 할 때는 ADD 키워드를 추가 합니다.

**쿼리문법)**

|  |
| --- |
| ALTER TABLE 테이블명 **ADD** 컬럼이름 자료형; |

**08.1.1 쿼리실습**

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

새로운 컬럼을 하나 추가해 봅니다. 먼저 desc 쿼리명령을 통하여 기존 members 테이블의 구조를 확인해 봅니다.

|  |
| --- |
| mysql> desc members;  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  6 rows in set (0.02 sec) |

기존 members 테이블에 나이를 저장할 새로운 컬럼 age를 추가해 보도록 하겠습니다. 데이터 타입은 가변문자열 10자리 varchar(10) 으로 설정합니다.

alter table members add age varchar(10);

|  |
| --- |
| mysql> **alter table members add age varchar(10);**  Query OK, 0 rows affected (0.04 sec)  Records: 0 Duplicates: 0 Warnings: 0 |

컬럼을 추가하는 쿼리 명령이 정상적으로 잘 처리가 되었습니다. 다시한번 desc 쿼리명령을 통하여 새롭게 추가한 컬럼이 추가 되었는지 확인을 해봅니다.

|  |
| --- |
| mysql> desc members;  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  | age | varchar(10) | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  7 rows in set (0.02 sec) |

위의 결과를 보면 새로운 컬럼 age 가 마지막에 추가된 것을 확인 할 수 있습니다.

**08.1.2 PHP 실습**

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

PHP 코드를 통하여 테이블에 새로운 컬럼을 추가해 보도록 하겠습니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // 컬럼 데이터 타입을 수정합니다.  public function tableFieldADD($tbname, $field, $datatype, $pos)  {  if ($tbname) {  $queryString = "ALTER TABLE $tbname ADD $field $datatype";  if ($pos) {  if($pos == "first") $queryString .= " first";  else $queryString .= " after $pos";  }  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE) {  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 컬럼추가!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  } |

**예제파일) alter-01.php**

|  |
| --- |
| <?php    include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();    $tbname = "members";  $field = "address2";  $datatype = "varchar(255)";    // 필드를 추가합니다.  $db->tableFieldADD($tbname, $field, $datatype);    ?> |

출력화면)

mysql connected!

쿼리성공] ALTER TABLE members ADD address2 varchar(255)

컬럼추가!

**08.2 컬럼 위치변경**

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

테이블에 컬럼을 새롭게 추가하면 생성되는 컬럼은 테이블의 맨 마지막 부분에 생성이 됩니다. 추가적으로 등록한 컬럼순서는 데이터를 해석하는데 가독성이 불편할 수 있습니다. 이런 경우 쿼리 마지막에 위치를 결정하는 키워드를 추가하여 순서를 변경할 수 있습니다. 또는 생성시 자리를 지정할 수 있습니다.

**08.2.1 위치변경**

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

이미 생성된 컬럼의 위치를 변경할 때는 MODIFY 키워드를 이용합니다. 또한 ALTER 쿼리의 마지막에 위치를 정의합니다.

**쿼리문법)**

|  |
| --- |
| ALTER TABLE 테이블명 **MODIFY** 컬럼이름 자료형 **after** 컬럼위치; |

그럼 컬럼의 위치를 변경해 보는 실습을 해봅니다. 다음과 같이 desc 명령을 통하여 members 테이블의 구조를 확인해 봅니다.

|  |
| --- |
| mysql> **desc members;**  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  | age | varchar(10) | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  7 rows in set (0.02 sec) |

새롭게 추가한 컬럼(age)가 제일 마지막에 위치해 있습니다. age 컬럼을 FirstName 컬럼명 다음으로 이동해 봅니다.

ALTER TABLE members MODIFY COLUMN age varchar(10) after FirstName;

|  |
| --- |
| mysql> **ALTER TABLE members MODIFY COLUMN age varchar(10) after FirstName;**  Query OK, 0 rows affected (0.03 sec)  Records: 0 Duplicates: 0 Warnings: 0 |

쿼리 명령이 정상적으로 잘 수행이 되었습니다. 다시한번 desc 쿼리명령을 통하여 members 테이블의 구조를 확인해 봅니다.

|  |
| --- |
| mysql> **desc members;**  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | age | varchar(10) | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  7 rows in set (0.02 sec) |

위의 콘솔결과를 보면 컬럼 age의 위치가 컬럼 FirstName 뒤로 변경이 된 것을 확인 할 수 있습니다. 컬럼의 위치를 변경할 때는 after 키워드 뒤에 위치할 다른 컬럼명을 지정하면 됩니다.

만일 컬럼명을 제일 처음으로 변경을 할 때는 ‘after 컬럼명’ 대신에 FIRST 키워드로 사용하면 됩니다.

ALTER TABLE members MODIFY COLUMN age varchar(10) **first**;

|  |
| --- |
| mysql> **ALTER TABLE members MODIFY COLUMN age varchar(10) first;**  Query OK, 0 rows affected (0.04 sec)  Records: 0 Duplicates: 0 Warnings: 0  mysql> desc members;  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | age | varchar(10) | YES | | NULL | |  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  7 rows in set (0.02 sec) |

컬럼 age가 처음으로 이동이 되었습니다.

**08.2.2 PHP 실습**

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

PHP코드를 통하여 컬럼의 위치를 변경하는 메소드를 작성해서 처리해 보도록 해보겠습니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // 컬럼 데이터 타입을 수정합니다.  public function tableFieldModify($tbname, $field, $datatype, $pos)  {  if ($tbname) {  $queryString = "ALTER TABLE $tbname MODIFY $field $datatype";  if ($pos) {  if($pos == "first") $queryString .= " first";  else $queryString .= " after $pos";  }    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE) {  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 컬럼변경!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  } |

**예제파일) alter-02.php**

|  |
| --- |
| <?php    include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();    $tbname = "members";  $field = "address2";  $datatype = "varchar(255)";    // 필드를 추가합니다.  $db->tableFieldModify($tbname, $field, $datatype, "address");    ?> |

출력화면)

mysql connected!

쿼리성공] ALTER TABLE members MODIFY address2 varchar(255) after Address

컬럼변경()!

**08.2.3 생성위치**

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

기존 생성된 테이블의 컬럼을 수정을 해보았습니다. 또는 새로운 컬럼을 추가할 때도 컬럼의 위치를 지정하여 등록을 할 수 있습니다. 컬럼의 위치를 결정하는 FIRST나 ‘alter 컬럼명’ 은 컬럼을 추가하는 ADD 키워드와 같이 사용이 가능합니다.

**쿼리문법)**

|  |
| --- |
| ALTER TABLE 테이블명 **ADD** 컬럼이름 자료형 **after** 컬럼위치; |

새로운 컬럼을 추가해 보도록 합니다. 컬럼을 추가할 때 위치를 같이 지정하여 추가합니다. 컬럼의 위치결정 하는 쿼리를 같이 넣어주면, 나중에 별도로 위치를 변경할 필요가 없습니다. desc 명령을 통하여 테이블의 구조를 확인합니다.

|  |
| --- |
| mysql> **desc members;**  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | age | varchar(10) | YES | | NULL | |  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  7 rows in set (0.02 sec) |

새로운 컬럼 birth를 추가합니다. birth의 위치는 FirstName 다음에 위치합니다.

alter table members add birth datetime after FirstName;

|  |
| --- |
| mysql> alter table members add birth datetime after FirstName;  Query OK, 0 rows affected (0.04 sec)  Records: 0 Duplicates: 0 Warnings: 0 |

쿼리가 정상적으로 잘 수행이 되었습니다. 다시한번 desc 쿼리 명령을 통하여 테이블의 구조를 확인해 봅니다.

|  |
| --- |
| mysql> desc members;  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | age | varchar(10) | YES | | NULL | |  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | birth | datetime | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  8 rows in set (0.02 sec) |

정상적으로 새롭게 추가한 birth 컬럼이 FirstName 컬럼 다음에 위치하여 생성이 되었습니다.

**08.2.5 PHP 실습**

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

두번째 위치 컬럼 추가 쿼리를 PHP를 통하여 실습해 봅니다.

**예제파일) alter-03.php**

|  |
| --- |
| <?php    include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();    $tbname = "members";  $field = "birth";  $datatype = "datetime";    // 필드를 추가합니다.  $db->tableFieldADD($tbname, $field, $datatype,"FirstName");    ?> |

출력화면)

mysql connected!

쿼리성공] ALTER TABLE members ADD birth datetime after FirstName

컬럼추가!

**08.3 자료형 변경**

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

테이블의 컬럼을 변경 외에도 컬럼의 데이터타입 자료형도 같이 변경이 가능합니다. 하지만 컬럼의 데이터 타입을 변경을 할 때에는 주의할 사항들이 많이 있습니다.

데이터의 타입을 변경할 때에는 변경하고자 하는 데이터의 타입이 자료를 같이 유지할 수 있는 호한적인 타입이어야 합니다. 만일 잘못된 자료형으로 변경을 하게 되면 기존 데이터에 문제가 발생 됩니다. 저장되어 있는 데이터가 삭제되거나, 이상한 값으로 변경될 수도 있습니다.

**쿼리문법)**

|  |
| --- |
| ALTER TABLE 테이블명 **MODIFY** 컬럼이름 자료형; |

**08.3.1 쿼리실습**

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

desc 쿼리명령을 통하여 members 의 테이블의 구조를 확인해 봅니다. members 테이블의 age 칼럼은 데이터 타입이 가변문자열 10자리 varchar(10) 으로 되어 있습니다.

|  |
| --- |
| mysql> **desc members;**  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  | age | **varchar(10)** | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  7 rows in set (0.02 sec) |

기존 age 컬럼의 데이터 타입을 정수형 타입 int(8) 형태로 변경해 보도록 하겠습니다.

alter table members modify age int(8);

|  |
| --- |
| mysql> **alter table members modify age int(8);**  Query OK, 0 rows affected (0.04 sec)  Records: 0 Duplicates: 0 Warnings: 0 |

쿼리가 정상적으로 잘 수행이 되었습니다. desc 쿼리명령을 통하여 다시한번 테이블의 구조를 확인해 보도록 합니다.

|  |
| --- |
| mysql> **desc members;**  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  | age | **int(8)**  | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  7 rows in set (0.02 sec) |

테이블의 구조를 확인해 보면 컬럼 age의 데이터 타입이 변경이 된 것을 확인해 보실 수 있습니다.

**08.3.2 PHP 실습**

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

PHP 코드를 통하여 컬럼의 데이터 타입을 변경해 보도록 하겠습니다.

**예제파일) alter-04.php**

|  |
| --- |
| <?php    include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();    $tbname = "members";  $field = "birth";  $datatype = "varchar(20)";    // 필드를 추가합니다.  $db->tableFieldModify($tbname, $field, $datatype);    ?> |

출력화면)

mysql connected!

쿼리성공] ALTER TABLE members MODIFY birth varchar(20)

컬럼변경!

**08.4 컬럼이름 변경**

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

생성된 테이블의 컬럼의 이름을 변경할 수 있습니다. 컬럼의 이름은 데이터베이스에서 쉽게 변경을 할 수 있습니다. 하지만, 데이터베이스와 연결된 응용프로그램에서는 데이터를 접근할 때 컬럼명을 많이 사용합니다. 컬럼명 변경과 프로그램 코드를 잘 반영하여 적절하게 수정을 하는 것이 중요합니다.

컬럼명을 변경할 때는 CHANGE 키워드를 사용합니다.

**쿼리문법)**

|  |
| --- |
| ALTER TABLE 테이블명 **CHANGE** 변경전\_컬럼이름 변경후\_컬럼이름 자료형; |

CHANGE는 컬럼명 변경과 함께 자료형도 같이 변경을 할 수 있습니다.

**08.4.1 쿼리 실습**

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

기존 테이블의 컬럼명을 변경해 보도록 실습해 봅니다. desc 쿼리명령을 통하여 기존 테이블의 구조를 확인합니다.

|  |
| --- |
| mysql> **desc members;**  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | age | varchar(10) | YES | | NULL | |  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | birth | datetime | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  8 rows in set (0.02 sec) |

기존 birth 컬럼명을 새로운 컬럼명 regdate 로 이름을 변경해 보겠습니다. 또한 자료형도 datetime 에서 timestamp 타입으로 함께 변경을 해봅니다.

alter table members change birth regdate timestamp;

|  |
| --- |
| mysql> **alter table members change birth regdate timestamp;**  Query OK, 0 rows affected (0.04 sec)  Records: 0 Duplicates: 0 Warnings: 0 |

쿼리 명령이 잘 실행되었습니다. desc 쿼리명령을 통하여 다시한번 테이블의 구조를 확인해 봅니다.

|  |
| --- |
| mysql> desc members;  +-----------+--------------+------+-----+-------------------+-----------------------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+-------------------+-----------------------------+  | age | varchar(10) | YES | | NULL | |  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | regdate | timestamp | NO | | CURRENT\_TIMESTAMP | on update CURRENT\_TIMESTAMP |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  +-----------+--------------+------+-----+-------------------+-----------------------------+  8 rows in set (0.01 sec) |

**08.4.2 PHP 실습**

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

컬럼명 변경을 PHP코드로 실습을 해보도록 합니다. 새로운 메서드를 추가하여 실습 코드를 작성을 합니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // 테이블 컬럼필드를 수정합니다.  public function tableFieldChange($tbname, $src, $dst, $datatype)  {  if ($tbname){  $queryString = "ALTER TABLE $tbname CHANGE $src $dst $datatype";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 컬럼명 변경");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  } |

**예제파일) alter-05.php**

|  |
| --- |
| <?php    include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();    $tbname = "members";  $src = "birth";  $dst = "regdate";  $datatype = "varchar(20)";    // 필드를 추가합니다.  $db->tableFieldChange($tbname, $src, $dst, $datatype);    ?> |

출력화면)

mysql connected!

쿼리성공] ALTER TABLE members CHANGE birth regdate varchar(20)

컬럼명 변경!

**08.5 컬럼 삭제**

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

테이블의 컬럼을 삭제할 수 있습니다. 컬럼을 삭제할 때는 매우 조심스러운 주의가 필요합니다. 컬럼을 삭제하게 되면 컬럼안에 있는 데이터 모두 같이 삭제가 됩니다. 또한, 삭제된 컬럼의 자료들은 복구가 되지 않습니다.

컬럼을 삭제할 때는 DROP 키워드를 같이 사용을 합니다.

**쿼리문법)**

|  |
| --- |
| ALTER TABLE 테이블명 **DROP** 컬럼이름; |

**08.5.1 쿼리 실습**

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

테이블에 존재하는 컬럼 하나를 삭제해 보도록 합니다. desc 쿼리명령을 통하여 테이블의 구조를 확인해 봅니다.

|  |
| --- |
| mysql> **desc members;**  +-----------+--------------+------+-----+-------------------+-----------------------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+-------------------+-----------------------------+  | age | varchar(10) | YES | | NULL | |  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | regdate | timestamp | NO | | CURRENT\_TIMESTAMP | on update CURRENT\_TIMESTAMP |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  +-----------+--------------+------+-----+-------------------+-----------------------------+  8 rows in set (0.01 sec) |

테이블의 컬럼 regdate 를 삭제해 봅니다.

alter table members drop regdate;

|  |
| --- |
| mysql> alter table members drop regdate;  Query OK, 0 rows affected (0.04 sec)  Records: 0 Duplicates: 0 Warnings: 0 |

쿼리가 정상적으로 잘 수행이 되었습니다. desc 쿼리 명령을 통하여 테이블의 구조를 다시 확인해 봅니다.

|  |
| --- |
| mysql> desc members;  +-----------+--------------+------+-----+---------+----------------+  | Field | Type | Null | Key | Default | Extra |  +-----------+--------------+------+-----+---------+----------------+  | age | varchar(10) | YES | | NULL | |  | Id | int(11) | NO | PRI | NULL | auto\_increment |  | LastName | varchar(255) | YES | | NULL | |  | FirstName | varchar(255) | YES | | NULL | |  | Address | varchar(255) | YES | | NULL | |  | City | varchar(255) | YES | | NULL | |  | Country | varchar(255) | YES | | NULL | |  +-----------+--------------+------+-----+---------+----------------+  7 rows in set (0.02 sec) |

테이블의 regdate 컬럼이 삭제된 것을 확인 할 수 있습니다.

**08.5.2 PHP 실습**

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

PHP 코드를 통하여 테이블의 컬럼을 삭제해 봅니다. 관련 메소드를 하나더 추가하여 실습해 해보도록 합니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // 테이블 컬럼필드를 삭제합니다.  public function tableFieldDrop($tbname,$field)  {  if ($tbname) {  $queryString = "ALTER TABLE $tbname DROP $field";  // 쿼리를 전송합니다.    if (mysqli\_query($this->dbcon, $queryString)=== TRUE) {  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 컬럼삭제!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  } |

**예제파일) alter-06.php**

|  |
| --- |
| <?php    include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();    $tbname = "members";  $field = "regdate";    // 필드를 삭제합니다.  $db->tableFieldDrop($tbname, $field);    ?> |

출력화면)

mysql connected!

쿼리성공] ALTER TABLE members DROP regdate

컬럼삭제!