jinyPHP 2권 – 부록A

**부록. A**

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

**mysql.class.php**

|  |
| --- |
| <?php    class JinyMysql  {  public $dbcon;  public $dbname;    // 데이터베이스 초기 연결  function \_\_construct()  {    $this->dbcon = mysqli\_connect($GLOBALS['mysql\_host'], $GLOBALS['mysql\_user'], $GLOBALS['mysql\_password'], $GLOBALS['mysql\_database']);  if (mysqli\_connect\_errno()) {  printf("Connect failed: %s\n", mysqli\_connect\_error());  exit();  } else {    $this->msgEcho("mysql connected!");  mysqli\_query($this->dbcon, "set names utf8");    }  }    public function msgEcho($msg)  {  echo $msg."<br>";  }  // ======================================================  // 테이블 잠금  // 매개인자: 데이블명, 옵션 = 옵션이 없을 경우 read로 설정합니다.  // 반한값 : 객체 this를 반환 합니다.    public function tableLock($tableName, $option="read")  {    $queryString = "LOCK TABLES ";  if ($tableName){    $queryString .= $tableName;    // 옵션이 없는 경우에는 READ로 기본설정 합니다.  $queryString .= " ".$option;    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho($tableName." LOCK!");    } else {  $this->msgEcho("Error] ".$queryString);  }    // 객체 반환, 매서드체인  return $this;    }    }    // 테이블 잠금해제  public function tableUnLock()  {    $queryString = "UNLOCK TABLES ";    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" TABLE UNLOCK!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }      }  // ======================================================  // 데이터베이스 생성  public function createDatabase($dbname)  {  if ($dbname){  $queryString = "CREATE DATABASE $dbname;";  $this->msgEcho($queryString);  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터베이스 생성!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 데이터베이스 이름이 없습니다.");  }    }  // 데이터베이스명 존재여부 확인  public function isDatabases($dbname)  {  if ($dbname){  $queryString = "show databases like '".$dbname."'";  $this->msgEcho($queryString);  // 쿼리를 전송합니다.  if ($result = mysqli\_query($this->dbcon, $queryString)){  if (mysqli\_num\_rows($result)) {  $row = mysqli\_fetch\_object($result);  $result->free();  $this->msgEcho($dbname."은 존재합니다.");  return true;  } else {  $this->msgEcho($dbname."이 없습니다. 생성이 가능합니다.");  return false;  }  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 데이터베이스 이름이 없습니다.");  }  }  // 데이터베이스 목록을 읽어 옵니다  public function showDatabases()  {  $queryString = "show databases";  $this->msgEcho($queryString);  if($result = mysqli\_query($this->dbcon, $queryString)){  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0;$i<$row\_cnt;$i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  }  }  public function changeDatabase($dbname){  $this->msgEcho($dbname." 으로 작업 데이터베이스를 변경합니다.");  mysqli\_select\_db($this->dbcon, $dbname);  }  // 현재 데이터베이스명을 리턴합니다.  public function currentDatabase()  {  $queryString = "SELECT DATABASE()";  $this->msgEcho($queryString);  // 쿼리를 전송합니다.  if ($result = mysqli\_query($this->dbcon, $queryString)){  $row = mysqli\_fetch\_object($result);  $result->close();  $key = "DATABASE()";  return $row->$key;  } else {  echo "선택된 데이터베이스가 없습니다. <br>";  }    }  public function dropDatabase($dbname){  if ($dbname){  $queryString = "DROP DATABASE $dbname;";  $this->msgEcho($queryString);  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터베이스 삭제!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 데이터베이스 이름이 없습니다.");  }  }  // ======================================================  public function isTables($tbname)  {  if ($tbname){  $dbname = $this->dbname;  $queryString = "show tables from $dbname like '$tbname'";  $this->msgEcho($queryString);  // 쿼리를 전송합니다.  if ($result = mysqli\_query($this->dbcon, $queryString)){  if (mysqli\_num\_rows($result)) {  $row = mysqli\_fetch\_object($result);  $result->free();  $this->msgEcho($tbname."은 존재합니다.");  return true;  } else {  $this->msgEcho($tbname."이 없습니다. 생성이 가능합니다.");  return false;  }  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블이 존재합니다.");  }  }  // 테이블 생성  public function createTable($queryString){  $this->msgEcho($queryString);  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 테이블 생성!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  }  public function dropTable($tbname)  {  if ($tbname){  $queryString = "DROP TABLES $tbname;";  $this->msgEcho($queryString);  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 테이블 삭제!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  public function clearTable($tbname)  {  if ($tbname){  $queryString = "TRUNCATE TABLE $tbname;";  $this->msgEcho($queryString);  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 테이블 초기화!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 테이블 정보를 읽어 옵니다  public function descTable($tbname)  {  $queryString = "desc $tbname";  $this->msgEcho($queryString);  if($result = mysqli\_query($this->dbcon, $queryString)){  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0;$i<$row\_cnt;$i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  }  }  // 테이블 목록을 읽어 옵니다  public function showTables()  {  $queryString = "show tables";  $this->msgEcho($queryString);  if($result = mysqli\_query($this->dbcon, $queryString)){  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0;$i<$row\_cnt;$i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  }  }  // ====================================  // 컬럼 데이터 타입을 수정합니다.  public function tableFieldADD($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 {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 컬럼 데이터 타입을 수정합니다.  public function tableFieldModify($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 {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 테이블 컬럼필드를 수정합니다.  public function tableFieldChange($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 {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 테이블 컬럼필드를 삭제합니다.  public function tableFieldDrop($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 {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  public function constraintNOTNULL($tbname, $field, $datatype)  {  if ($tbname){  $queryString = "ALTER TABLE $tbname CHANGE COLUMN `$field` `$field` $datatype NOT NULL";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" NOT NULL 설정!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }    } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  return false;  }  }  public function constraintNULL($tbname, $field, $datatype)  {  if ($tbname){  $queryString = "ALTER TABLE $tbname CHANGE COLUMN `$field` `$field` $datatype NULL";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" NULL 설정!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }    } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  return false;  }  }  public function constraintDEFAULT($tbname, $field, $default)  {  if ($tbname){  if($default){  $queryString = "ALTER TABLE $tbname ALTER $field SET DEFAULT '$default'";  } else {  $queryString = "ALTER TABLE $tbname ALTER $field DROP DEFAULT";  }  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 기본값 설정!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }      } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  public function constraintUNIQUE($tbname, $field)  {  if ($tbname){  $queryString = "ALTER TABLE $tbname ADD UNIQUE ($field)";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 기본값 설정!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }    } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  return false;  }  }  public function constraintUNIQUE\_Drop($tbname, $field)  {  if ($tbname){  $queryString = "ALTER TABLE $tbname DROP INDEX `$field`";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 기본값 설정!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }    } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  return false;  }  }  public function autoIncrementClear($tbname,$inc=0)  {  if ($tbname){  $queryString = "ALTER TABLE $tbname AUTO\_INCREMENT=$inc;";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 자동증가값 초기화!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }    } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  return false;  }  }  // ====================================  // 테이블 엔진을 설정합니다.  public function setEngine($tbname,$engine="InnoDB")  {  if ($tbname){  $queryString = "ALTER TABLE $tbname ENGINE=$engine";    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 엔진변경($engine)!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // ======================================================  // array 배열기반 insert 쿼리를 생성합니다.  public function arrQueryInsert($tbname, $arr)  {  $arrKey = array\_keys($arr);  for($i=0;$i<count($arr);$i++){  // $key = $arr[$i];  echo $arrKey[$i]."<br>";  $key = $arrKey[$i];  $insert\_filed .= "`".$key."`,";  $insert\_value .= "'".$arr[$key]."',";  }  $queryString = "INSERT INTO ".$tbname." ($insert\_filed) VALUES ($insert\_value);";  $queryString = str\_replace(",)",")",$queryString);  return $queryString;  }  // 어레이데이터를 insert 쿼리로 삽입합니다.  public function arrInsert($tbname, $arr)  {  if ($tbname){  $queryString = $this->arrQueryInsert($tbname, $arr);  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 삽입!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 마지막 삽입된 Id 값 확인  public function insertID(){  return mysqli\_insert\_id($this->dbcon);  }  // ======================================================  // SELECT  public function selectRowss($queryString){  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  }  public function select($tbname,$field,$where){  if ($tbname){  $queryString = "SELECT ";    if($field){  $queryString .= "".$field . " FROM $tbname";  } else {  $queryString .= "\* FROM $tbname";  }  if($where){  $queryString .= " where ".$where.";";  } else {  $queryString .= ";";  }  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  public function selectDistinct($tbname,$field,$where){  if ($tbname){  $queryString = "SELECT DISTINCT ";    if($field){  $queryString .= "".$field . " FROM $tbname";  } else {  $queryString .= "\* FROM $tbname";  }  if($where){  $queryString .= " where ".$where.";";  } else {  $queryString .= ";";  }  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // ======================================================  public function \_IsEqual($a,$b)  {  return "$a='$b'";  }  public function \_IsNot($a,$b)  {  return "$a<>'$b'";  }  // 입력된 쿼리의 출력제한 limit를 설정합니다.  public function limit($query, $a, $b)  {  $queryString = $query . " limit $a ";  if ($b){  $queryString .= ",".$b;  }  return $queryString;  }  // 지정한 위치많큼 이동후, 갯수를 출력합니다.  public function offset($query, $a, $b)  {  $queryString = $query . " limit $b offset $a";  return $queryString;  }  public function orderBy($query,$by)  {  return $query . " ORDER BY " . $by;  }  public function groupBy($query,$by)  {  return $query . " GROUP BY " . $by;  }  public function groupByHaving($query,$by,$where)  {  return $query . " GROUP BY " . $by . " HAVING $where";  }  // ======================================================  // 선택한 데이터 Id를 삭제합니다.  public function deleteID($tbname, $Id)  {  if ($tbname){  if($Id){  $queryString = "DELETE FROM $tbname where Id = '$Id'";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 $Id 삭제!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 삭제할 Id값이 없습니다.");  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 조건에 맞는 데이터를 모두 삭제합니다.  public function deleteWhere($tbname, $where)  {  if ($tbname){  if($where){  $queryString = "DELETE FROM $tbname where $where";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 조건 삭제!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 삭제조건이 없습니다.");  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 데이터 전체를 삭제합니다.  public function deleteAll($tbname)  {  if ($tbname){  $queryString = "DELETE FROM $tbname";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 전체 삭제!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 최상의 데이터 몇개만 삭제합니다.  public function deleteLimit($tbname, $num, $orderby)  {  if ($tbname){  if($num){  if ($orderby){  $queryString = "DELETE FROM $tbname ORDER BY $orderby LIMIT $num";  } else {  $queryString = "DELETE FROM $tbname LIMIT $num";  }    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 $Id 삭제!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 삭제할 테이터 갯수가 없습니다.");  }  } else {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // ======================================================  // array 배열기반 update 쿼리를 생성합니다.  public function arrQueryUpdate($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 {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 조건 업데이트  public function updateWhere($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 {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // arr배열 데이터를 기분으로 테이블 전체 컬럼필드를 업데이트 합니다.  public function updateAll($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 {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // 최상의 데이터 몇개만 업데이트합니다.  public function updateLimit($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 {  $this->msgEcho("Error] 테이블 이름이 없습니다.");  }  }  // ======================================================  // 테이블 구조를 복사합니다.  public function copyTable($src,$dst)  {  if ($src){  if ($dst){  $dstName = $dst;  } else {  $dstName = $src."\_copy";  }  $queryString = "create table $dstName like $src";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 테이블 복사!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 원본 테이블명을 적어주세요.");  }    }  // 테이블 구조와 데이터 모두를 복사합니다.  public function copyTableWithAll($src,$dst)  {  if ($src){  if ($dst){  $dstName = $dst;  } else {  $dstName = $src."\_copy";  }  $queryString = "create table $dstName select \* from $src;";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 테이블 전체복사!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 원본 테이블명을 적어주세요.");  }    }  // select 데이터를 테이블로 복사합니다.  public function insertSelect($src,$dst,$fields)  {  if ($src){  if ($fields){  $queryString = "insert into $dst ($fields) select $fields from $src";  } else {  $queryString = "insert into $dst select \* from $src";  }    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 복사!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 원본 테이블명을 적어주세요.");  }  }  // select 데이터를 테이블로 복사합니다.  public function insertSelectWhere($src, $dst, $where, $fields)  {  if ($src){  if ($fields){  $queryString = "insert into $dst ($fields) select $fields from $src where $where";  } else {  $queryString = "insert into $dst select \* from $src where $where";  }  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 데이터 복사!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 원본 테이블명을 적어주세요.");  }  }  // ======================================================  // 입력된 select 쿼리를 기반으로 view를 셍성합니다.  public function createView($view, $selectQuery){  if ($view){  if ($selectQuery){  $queryString = "CREATE VIEW $view AS $selectQuery";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 뷰 생성!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }    } else {  $this->msgEcho("Error] 조건 select 쿼리를 입력해 주세요..");  }  } else {  $this->msgEcho("Error] 뷰 이름을 적어주세요.");  }  }  public function updateView($view, $selectQuery){  if ($view){  if ($selectQuery){  $queryString = "CREATE OR REPLACE VIEW $view AS $selectQuery";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 뷰 생성!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }    } else {  $this->msgEcho("Error] 조건 select 쿼리를 입력해 주세요..");  return false;  }  } else {  $this->msgEcho("Error] 뷰 이름을 적어주세요.");  return false;  }  }  public function alterView($view, $selectQuery){  if ($view){  if ($selectQuery){  $queryString = "ALTER VIEW $view AS $selectQuery";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 뷰 생성!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }    } else {  $this->msgEcho("Error] 조건 select 쿼리를 입력해 주세요..");  return false;  }  } else {  $this->msgEcho("Error] 뷰 이름을 적어주세요.");  return false;  }  }  public function dropView($view)  {  if ($view){  $queryString = "DROP VIEW IF EXISTS $view";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 뷰 삭제!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 뷰 이름을 적어주세요.");  }  }    // ======================================================  // 다수의 query를 union 연결 문장을 실행합니다.  // query 문장을 가진 배열을 입력받습니다.  public function union($queryARR)  {  for ($i=0;$i<count($queryARR);$i++){  $queryString .= "(".$queryARR[$i].") union ";  }  $queryString .= ";";  $queryString = str\_replace("union ;","",$queryString);  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  }  // 내부조인을 처리합니다.  public function innerJoin($tbname1, $tbname2, $onWhere, $fields)  {  $queryString = "SELECT ";  // 컬럼필드 선택여부를 지정  if ($fields){  $queryString .= $fields." ";  } else {  $queryString .= "\* ";  }  $queryString .= "from $tbname1 JOIN $tbname2 ON $onWhere";  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  }  // LEFT조인을 처리합니다.  public function leftJoin($tbname1, $tbname2, $onWhere, $fields)  {  $queryString = "SELECT ";  // 컬럼필드 선택여부를 지정  if ($fields){  $queryString .= $fields." ";  } else {  $queryString .= "\* ";  }  $queryString .= "from $tbname1 LEFT JOIN $tbname2 ON $onWhere";  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  }  // RIGHT 조인을 처리합니다.  public function rightJoin($tbname1, $tbname2, $onWhere, $fields)  {  $queryString = "SELECT ";  // 컬럼필드 선택여부를 지정  if ($fields){  $queryString .= $fields." ";  } else {  $queryString .= "\* ";  }  $queryString .= "from $tbname1 RIGHT JOIN $tbname2 ON $onWhere";  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  }  // FULL 조인을 처리합니다.  public function fullJoin($tbname1, $tbname2, $onWhere, $fields)  {  $queryString = "SELECT ";  // 컬럼필드 선택여부를 지정  if ($fields){  $queryString .= $fields." ";  } else {  $queryString .= "\* ";  }  $queryString = $queryString."from $tbname1 LEFT JOIN $tbname2 ON $onWhere"." union all ".$queryString."from $tbname1 RIGHT JOIN $tbname2 ON $onWhere";  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  }  // SELF 조인을 처리합니다.  public function selfJoin($tbname, $fields)  {  $queryString = "SELECT ";  // 컬럼필드 선택여부를 지정  if ($fields){  $queryString .= $fields." ";  } else {  $queryString .= "\* ";  }  $queryString .= "from $tbname as a JOIN $tbname as b";  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  }  // ==============================================  public function transaction()  {  $queryString = "START TRANSACTION";    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 트렌젝션 시작!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }  }  public function commit()  {  $queryString = "COMMIT";    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 작업내용 커밋!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }  }  public function rollback()  {  $queryString = "ROLLBACK";    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 작업내용 켄슬(롤백)!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }  }  public function autoCommit($mode)  {  if($mode == true){  $queryString = "SET AUTOCOMMIT=1;";  } else {  $queryString = "SET AUTOCOMMIT=0;";  }  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  if($mode == true){  $this->msgEcho(" 자동커밋 상태전환!");  } else {  $this->msgEcho(" 자동커밋 OFF!");  }    return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }  }  // ======================================================  public function delimiter($code)  {  if ($code){  $queryString = "DELIMITER $code";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" DELIMITER 변경!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] delimiter 키명을 적어주세요.");  }  }  // 프로시저를 생성합니다.  public function createProcedure($name, $queryArr){  $queryString = "  CREATE PROCEDURE $name()  BEGIN ";  for ($i=0;$i<count($queryArr);$i++){  $queryString .= $queryArr[$i].";";  }  $queryString .= " END";  // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 프로시저생성!");  // 객체 반환, 매서드체인  return $this;  } else {  $this->msgEcho("Error] ".$queryString);  }  }  public function callProcedure($name,$args)  {  if ($name){  $queryString = "CALL $name($args);";  if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 프로시저명을 적어주세요.");  }  }  public function isProcedure($name)  {  if ($name){  $queryString = "SHOW CREATE PROCEDURE $name";    if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 프로시저명을 적어주세요.");  }  }  // 생성한 프로시저를 삭제합니다.  public function dropProcedure($name)  {  if ($name){  $queryString = "DROP PROCEDURE $name";    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 프로시저 삭제!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }  } else {  $this->msgEcho("Error] 프로시저명을 적어주세요.");  }  }  // =================================================  public function isFunction($name)  {  if ($name){  $queryString = "SHOW CREATE FUNCTION $name";    if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }  } else {  $this->msgEcho("Error] 함수명을 적어주세요.");  }  }  // 생성한 함수를 삭제합니다.  public function dropFunction($name)  {  if ($name){  $queryString = "DROP FUNCTION $name";    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 함수 삭제!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }  } else {  $this->msgEcho("Error] 함수명을 적어주세요.");  }  }  public function createTrigger($queryTrigger)  {  $queryString = "DELIMITER //";  mysqli\_query($this->dbcon, $queryString);  $this->msgEcho($queryString );  $this->msgEcho($queryTrigger);  if (mysqli\_query($this->dbcon, $queryTrigger)=== TRUE){  $this->msgEcho("트리거 성공] ".$queryTrigger);  $queryString = "DELIMITER ;";  mysqli\_query($this->dbcon, $queryString);  $this->msgEcho($queryString );  return true;  } else {  $this->msgEcho("Error] 트리거 생성 실패!");  return false;  }  }  public function showTriggers()  {    $queryString = "SHOW TRIGGERS";    if($result = mysqli\_query($this->dbcon, $queryString)){  $this->msgEcho("쿼리성공] ".$queryString);  $rowss = "";  $row\_cnt = mysqli\_num\_rows($result);  for($i=0; $i<$row\_cnt; $i++){  $rowss[$i] = mysqli\_fetch\_object($result);  }  $result->free();  return $rowss;  } else {  $this->msgEcho("Error] ".$queryString);  }    }  // 생성한 트리거를 삭제합니다.  public function dropTrigger($name)  {  if ($name){  $queryString = "DROP TRIGGER $name";    // 쿼리를 전송합니다.  if (mysqli\_query($this->dbcon, $queryString)=== TRUE){  $this->msgEcho("쿼리성공] ".$queryString);  $this->msgEcho(" 트리거 삭제!");  return true;  } else {  $this->msgEcho("Error] ".$queryString);  return false;  }  } else {  $this->msgEcho("Error] 트리거명을 적어주세요.");  }  }  }    ?> |