jinyPHP 2권 - 004. 데이터베이스 생성

**04. 데이터베이스**

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

MYSQL과 같은 데이터베이스 시스템을 사용하기 위해서는 데이터의 그룹의 역할과 같은 데이터베이스 이름을 지정하여야 합니다. 데이터베이스 이름을 통하여 MYSQL은 다수의 데이터베이스 그룹을 관리할 수 있습니다. 데이터베이스 그룹에는 여러 개의 테이블을 포함하고 있으며, 테이블은 다시 컬럼 정보와 데이터를 포함하고 있습니다.

만일 데이터베이스 시스템이 한 개의 데이터베이스 이름만 생성이 가능하다고 하면 매우 비효율 적일 것입니다. 마치 컴퓨터에서 파일들을 관리하는 폴더가 없이 수많은 파일들이 시스템에 뒤죽박죽으로 섞여저 있는 상태와 같을 것입니다.

MYSQL은 데이터베이스명을 기준으로 테이블을 구분합니다. 또한 테이터베이스 명 없이 테이블을 생성하고 관리를 할 수 없습니다.

**04.1 생성**

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

데이터베이스 이름은 다수의 테이블을 담고 있는 폴더와 같습니다. 실제적으로 데이터베이스 명을 생성하게 되면 MYSQL은 해당 데이터 베이스명과 동일한 폴더를 내부적으로 하나 생성을 하게 됩니다.

**04.1.1 쿼리실습**

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

데이터베이스 명의 생성은 CREATE DATABASE 명령을 실행할 수 있습니다. SQL 쿼리 구분을 만들어 MySQL 서버에 전송을 해주면 됩니다.

**쿼리문법)**

|  |
| --- |
| CREATE DATABASE 데이터베이스명; |

데이터베이스 생성 문법은 2개의 키워드를 같이 사용하여 작성합니다. CREATE + DATABASE 로 지정한 이름으로 데이터 베이스를 생성한다는 의미 입니다.

"jiny" 이름이 새로운 데이터베이스를 만들어 봅니다.

예제)

|  |
| --- |
| CREATE DATABASE jiny; |

콘솔 출력화면)

|  |
| --- |
| mysql> create database jiny;  Query OK, 1 row affected (0.01 sec) |

콘솔창을 확인해 보면 Query OK 라고 정상적으로 처리가 되었다는 메세지가 출력됩니다. 또한 처리 쿼리의 갯수와 실행시간을 같이 표시합니다.

**04.1.2 PHP 실습**

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

실제적으로 SQL쿼리와 PHP 연동 소스를 통하여 MYSQL에 새로운 데이터베이스명을 하나 추가해 보도록 하겠습니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다. 데이터베이스 생성 쿼리를 만들어 내는 메서드 입니다.

|  |
| --- |
| // 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.");  }    } |

데이터베이스 명이 있는지 확인하는 추가 메서드도 하나 더 만들어 봅니다.

|  |
| --- |
| // 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.");  }  } |

추가한 클래스 메서드를 통하여 실제적인 데이터베이스 생성 코드를 실험해 보도록 합니다.

**예제파일) sql-06.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  $dbname = "jiny123";  if ($db->isDatabases($dbname)){  echo $dbname." 존재하는 데이터베이스 이름입니다.";  } else {  $db->createDatabase($dbname);  }  ?> |

화면출력)

mysql connected!

show databases like 'jiny123'

jiny123이 없습니다. 생성이 가능합니다.

CREATE DATABASE jiny123;

쿼리성공] CREATE DATABASE jiny123;

데이터베이스 생성!

**04.1.3 중복이름**

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

MYSQL에서 데이터베이스 이름은 중복으로 사용을 할 수 없습니다. 데이터베이스명은 1개로 유일한 이름을 가지고 있어야 합니다. 따라서 중복된 이름의 데이터베이스 생성을 지정하면, DB 시스템은 오류를 발생합니다.

한번 실수로 기존에 존재하는 동일한 데이터베이스 이름으로 데이터베이스를 생성하는 SQL 명령을 입력해 봅니다.

**콘솔 출력화면)**

|  |
| --- |
| mysql> create database jiny;  ERROR 1007 (HY000): Can't create database 'jiny'; database exists |

위처럼 데이터베이스 이름이 중복되어 생성을 할 수 없다는 에러 메시지를 출력합니다.

**04.1.4 대소문자 구별**

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

SQL 키워드는 대소문자를 구별하지 않지만, 데이터베이스 이름은 운영체제별로 차이가 있습니다.

Windows 운영체제는 데이터베이스 이름에 대해서 대소문자를 구분하지 않습니다. 하지만 Linux 계열의 운영체제는 데이터베이스 이름을 대소문자로 구분을 합니다. 이는 리눅스 환경의 특성 이라고도 할 수 있습니다.

**04.2 데이터베이스 목록**

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

데이터베이스의 이름은 중복하여 사용을 할 수 없습니다. 또한 MYSQL은 데이터베이스명 이라는 그룹을 통하여 여러 개의 데이터베이스를 관리할 수 있습니다.

여러 개의 데이터베이스 명을 포함한 MYSQL 시스템은 아주 흔합니다. 한 개의 서버에 한 개의 DB 명만 가지고 있는 경우는 없을 것입니다. 호스팅과 같은 회사는 한 개의 MYSQL 시스템에 수백개의 DB명을 가지고 있고, 이를 각각의 사용자에게 할당하여 호스팅을 제공하고 있습니다.

**04.2.1 쿼리실습**

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

MYSQL은 자신이 가지고 있는 데이터베이스의 명의 목록을 출력해 주는 SHOW DATABASE 명령어를 지원합니다.

**쿼리문법)**

|  |
| --- |
| SHOW DATABASE; |

SHOW DATABASE; 문법은 현재 생성되어 있는 데이터베이스의 이름 목록 전체를 출력합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> **show databases;**  +--------------------+  | Database |  +--------------------+  | information\_schema |  | jiny |  | mysql |  | performance\_schema |  | test |  +--------------------+  5 rows in set (0.01 sec) |

위처럼 방금 생성한 jiny 데이터베이스 명도 같이 포함된 전체 목록을 확인하실 수 있습니다.

**04.2.2 PHP 실습**

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

실제적으로 SQL쿼리와 PHP 연동 소스를 통하여 MYSQL에 데이터베이스 목록을 출력해 보도록 하겠습니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다. 새로 추가한 메소드는 데이터베이스 목록을 출력해는 쿼리를 생성하여 결과를 반환합니다.

|  |
| --- |
| // 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;  }  } |

**예제파일) sql-07.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  if ($rowss = $db->showDatabases()) {  echo "db Count = ". count($rowss) . "<br>";  for ($i=0;$i<count($rowss);$i++) {  echo $i."=";  echo $rowss[$i]->Database;  echo "<br>";  print\_r($rowss[$i]);  echo "<br>";  }  }  ?> |

화면출력)

mysql connected!

show databases

db Count = 6

0=information\_schema

stdClass Object ( [Database] => information\_schema )

1=jiny

stdClass Object ( [Database] => jiny )

2=jiny123

stdClass Object ( [Database] => jiny123 )

3=mysql

stdClass Object ( [Database] => mysql )

4=performance\_schema

stdClass Object ( [Database] => performance\_schema )

5=test

stdClass Object ( [Database] => test )

**04.2.3 기본 데이터베이스**

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

SHOW DATABASE; 를 통하여 목록을 확인해 보면 직접 생성된 것 이외에도 기본적으로 MYSQL에서 미리 가지고 있는 데이터베이스명이 있습니다. MYSQL은 자체적으로 권한 및 모델구조들을 유지 관리하기 위해서 사용하는 데이터베이스 그룹들이 있습니다.

* performance\_schema
* information\_schema
* mysql
* test

위의 목록 중에 test 데이터베이스는 삭제하셔도 됩니다. 기본적으로 MYSQL에서 test 목적을 위해서 생성해 놓은 임시 폴더입니다.

mysql 데이터베이스명 에는 MYSQL의 사용자, 접속, DB 정보등을 가지고 있는 중요 정보 테이블이 들어가 있습니다.

* user 데이블
* db 테이블
* 그외 여러 파일들.

**04.3 데이터베이스 전환**

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

MYSQL은 앞서 설명한 것과 같이 다수의 데이터베이스 그룹명을 이용하여 데이터베이스를 관리합니다. 즉, 한 개의 데이터베이스 명이 있는 것이 아니기 때문에 그중 선택하여 사용을 할 수 있습니다.

**04.3.1 use 명령어**

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

데이터베이스에 작업을 하기 위해서는 먼저 작업하고자 하는 데이터베이스를 선택하여야 합니다. MYSQL은 처음 실행할 때 여러 개의 데이터베이스 명 그룹 중에 어느 것을 사용할지 모르기 때문입니다.

데이터베이스를 선택하는 SQL 키워드는 use 입니다.

**쿼리문법)**

|  |
| --- |
| USE 데이터베이스명; |

USE 다음에 데이터베이스명을 입력하면, 해당 데이터베이스명으로 작업 영역을 변경합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> use **jiny**;  Database changed |

위의 콘솔창을 확인해 보면 jiny 데이터베이스로 변경이 되었다는 메시지(Database Changed)를 확인해 볼 수 있습니다.

데이터베이스 그룹명을 선택하는 것은 한번만 하면 됩니다. 또한, 이전에 데이터베이스 그룹명을 미리 선언을 한 경우에는 생략하고 시작을 할 수 있습니다.

**04.3.2 \u 키워드**

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

use 명령문 대신에 약자 기호인 \u 를 사용할 수도 있습니다.

콘솔 출력화면)

|  |
| --- |
| mysql> \u test  Database changed |

**04.3.3 PHP 실습**

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

데이터베이스 그룹을 변경은 직접 SQL 실행 명령문을 작성하여 전송하여도 됩니다. 하지만, PHP는 데이터베이스를 선택할 수 있는 내부함수 mysqli\_select\_db()를 별도로 제공합니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| public function changeDatabase($dbname){  $this->msgEcho($dbname." 으로 작업 데이터베이스를 변경합니다.");  mysqli\_select\_db($this->dbcon, $dbname);  } |

**예제파일) sql-08.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  // 데이터베이스명을 변경합니다.  $dbname = "jiny";  $db->changeDatabase($dbname);  ?> |

화면출력)

mysql connected!

jiny 으로 작업 데이터베이스를 변경합니다.

**04.4 현재 데이터베이스**

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

모든 SQL의 실행 명령어는 선택한 데이터베이스명 내에서만 작업이 이루어 집니다. 만일 SQL 작업이 많아지거나 여러 개의 데이터베이스를 전환하다 보면 현재의 데이터베이스명을 잊어 버리는 경우도 있습니다.

**04.4.1 쿼리실습**

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

MYSQL은 현재 선택되어 있는 데이터베이스명을 확인할 수 있는 SQL 함수를 제공합니다.

**쿼리문법)**

|  |
| --- |
| SELECT DATABASE(); |

SELECT DATABASE(); 는 현재 USE로 선택되어 있는 데이터베이스 명을 출력합니다. DATABASE 명령와 같은 글자 뒤에 소괄호 () 가 더 있는 것을 볼 수 있습니다. 이처럼 데이터베이스에서도 함수와 같은 형태의 명칭이 사용 되기도 합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> **select database();**  +------------+  | database() |  +------------+  | test |  +------------+  1 row in set (0.00 sec) |

use 로 선택한 현재의 데이터베이스 명을 출력합니다.

**04.4.2 PHP 실습**

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

PHP소스를 통하여 현재 MYSQL에 있는 데이터베이스 목록을 출력해 보도록 하겠습니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // 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>";  }    } |

**예제파일) sql-09.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  // 현재 데이터베이스 이름을 확인합니다.  echo "현재 데이터베이스 = ".$db->currentDatabase();  ?> |

출력화면)

mysql connected!

SELECT DATABASE()

현재 데이터베이스 = jiny

**04.4.3 사전 지정**

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

매번 mysql 콘솔을 실행후에 use 명령어로 데이터베이스명을 선언하는 것이 불편할 수 있습니다.

이런경우 mysql 콘솔을 시작할때 미리 데이터베이스명을 직접 지정하여 실행을 할 수도 있습니다.

다음과 같이 콘솔 접속시

**mysql 데이터베이스명 -u계정명 -p**

데이터베이스명을 같이 입력을 하시면 콘솔 접속과 데이터베이스 선택을 같이 하실 수 있습니다.

**04.4.4 데이터베이스 지정**

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

USE 를 통하여 작업 데이터베이스 그룹을 변경한 후에는 데이터 베이스이름을 생략한 쿼리 생성이 가능합니다.

또는 현재 선택된 데이터베이스을 USE로 변경하지 않고 다른 데이터베이스를 간접적으로 선택할 경우 테이블명 앞에 데이터베이스 명을 지정하면 됩니다.

**쿼리문법)**

|  |
| --- |
| SELECT \* from 데이터베이스.테이블; |

데이터베이스 이름과 테이블명 사이에 점(.) 으로 연결하여 사용하면 됩니다.

콘솔 출력화면)

|  |
| --- |
| mysql> use test;  Database changed  mysql> **select \* from jiny.members;**  +----+----------+-----------+-------------+-------+---------+-----------+-------------------+  | Id | LastName | FirstName | Address | City | Country | manager | email |  +----+----------+-----------+-------------+-------+---------+-----------+-------------------+  | 1 | hojin | lee | shinchon | seoul | korea | infohojin | hojin@jinyphp.com |  | 2 | jiny | NULL | NULL | NULL | korea | infohojin | jiny@jinyphp.com |  | 3 | james | kim | youngdun-po | seoul | korea | info | james@jinyphp.com |  +----+----------+-----------+-------------+-------+---------+-----------+-------------------+  3 rows in set (0.01 sec) |

위의 예를 보면 데이터베이스 선택은 test로 되어 있으나, select 문은 jiny 데이터베이스의 members 테이블을 선택합니다.

**04.5 데이터베이스 삭제**

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

생성한 데이터베이스 명 그룹은 삭제할 수 있습니다. 시스템에서 데이터베이스명을 삭제할 때는 매우 주의해야 합니다.

**04.5.1 삭제쿼리**

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

생성한 데이터베이스 그룹명을 삭제할 때는 DROP DATABSE 명령을 사용합니다.

**쿼리문법)**

|  |
| --- |
| DROP DATABASE 삭제할데이터베이스명; |

데이터베이스명을 삭제를 하게 되면 안에 들어있는 테이블과 모든 데이터가 같이 삭제가 됩니다. 또한 삭제된 데이터들을 복구가 불가능하기 때문에 매우 신중하게 사용을 해야 합니다.

예제)

|  |
| --- |
| DROP DATABASE jiny; |

콘솔 출력화면)

|  |
| --- |
| mysql> drop database jiny;  Query OK, 0 rows affected (0.00 sec) |

기존에 생성되어 있는 "jiny" 데이터베이스 명을 drop 명령어를 통하여 삭제가 성공되었다는 메시지를 확인할 수 있습니다.

만일 데이터베이스가 존재하는 여부를 확인한 후에 삭제를 하려고 할 때는 IF EXISTS 키워드를 추가할 수도 있습니다.

**쿼리문법)**

|  |
| --- |
| DROP DATABASE IF EXISTS 삭제할데이터베이스명; |

**04.5.2 PHP 실습**

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

PHP 소스를 통하여 데이터베이스를 삭제하는 명령을 실행해 보도록 하겠습니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| 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.");  }  } |

**예제파일) sql-10.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  // 데이터베이스 삭제합니다.  $dbname = "jiny123";  $db->dropDatabase($dbname);  ?> |

출력화면)

mysql connected!

DROP DATABASE jiny123;

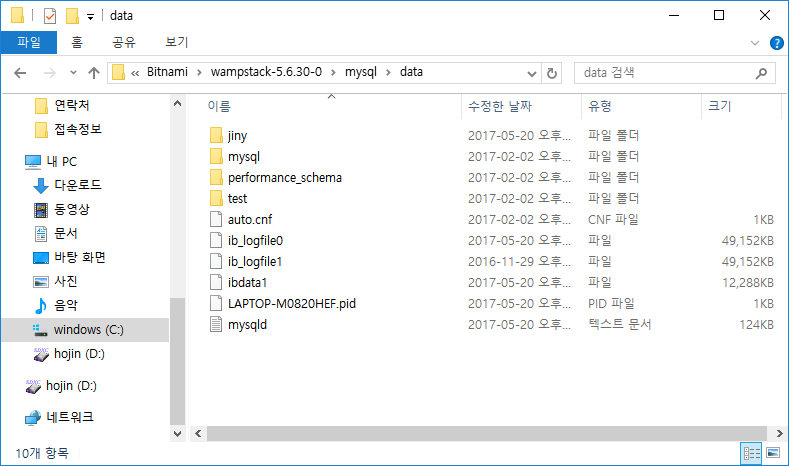
쿼리성공] DROP DATABASE jiny123;

데이터베이스 삭제!

**04.6 데이터베이스 폴더**

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

실제적으로 데이터베이스 생성명령을 통하여 생성을 하면은 MYSQL은 그와 관련된 실제적인 폴더를 하나 생성합니다.



위의 예처럼 mysql> data 폴더안에 생성한 jiny 데이터베이스에 대한 폴더가 생성된 것을 확인해 볼 수 있습니다. 하지만 DB 시스템을 통항 SQL명령문을 통하여 직접 접근을 해야지 임의 적인 접근 또는 수정을 한 경우에는 데이터 시스템에 문제가 발생 될 수 있습니다.