jinyPHP 2권 - 25. 내부함수

**25. 함수**

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

MYSQL은 쿼리 명령이외에 데이터를 처리할 수 있는 다양한 내장함수를 제공하고 있습니다. 내장함수들은 기본적으로 제공되는 함수들로 간단한 연산, 시스템 정보, 데이터 가공작업을 할 수 있습니다.

MYSQL 5.0 이상 버전에는 사용자가 직접 함수를 생성하고 사용을 할 수 있는 기능을 제공합니다.

**25.1 함수 생성**

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

내장함수들은 기본적으로 제공되는 함수들로서 모든 사용자들의 원하는 기능들을 충족하기에는 부족한 부분들이 있습니다. MYSQL은 기본적으로 제공하는 내장함수 이외에 직접 함수를 정의하여 생성을 할 수 있습니다.

CREATE FUNCTION명령은 사용자함수를 생성합니다. 생성된 함수는 데이터베이스 시스템에 저장되어서 재사용이 가능합니다. 함수를 선언할때는 함수명, 반환값타입, SQL실행후 RETRUN 값을 반환 홥니다.

**쿼리문법)**

|  |
| --- |
| CREATE FUNCTION 함수명(변수 자료형) RETURNS 반환\_자료형  BEGIN  SQL 쿼리1;  SQL 쿼리2;  RETURN 반환값  END |

**25.1.1 쿼리 실습**

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

간단한 사용자 함수를 하나 생성해 보도록 합니다. 다음 함수는 제곱을 처리하는 사용하 함수 입니다.

CREATE FUNCTION square(**a INT**) RETURNS **DOUBLE**

BEGIN

RETURN a \* a;

END

함수를 선언할때는 입력되는 변수와 타입, 반환되는 데이터 타입을 지정합니다. 직접 콘솔에서 입력하여 실습을 해봅니다.

|  |
| --- |
| mysql> DELIMITER //  mysql> CREATE FUNCTION square(a INT) RETURNS DOUBLE  -> BEGIN  -> RETURN a \* a;  -> END  -> //  Query OK, 0 rows affected (0.00 sec)  mysql> DELIMITER ;  mysql> SELECT square(5);  +-----------+  | square(5) |  +-----------+  | 25 |  +-----------+  1 row in set (0.00 sec) |

함수를 생성시 반환값의 타입설정 키워드는 복수형 RETURN**S** 입니다. 함수안에서의 반환 키워드는 단수형 RETURN 입니다. 두개의 철자를 구분하여 작업을 합니다.

**25.2 함수 실행**

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

사용자 정의 함수의 실행은 내장함수를 사용하는 것과 같습니다. SELECT 와 같은 쿼리 실행문 안에서 함수명을 사용하시면 됩니다. 또한, 인자값은 괄호 안에 넣어 주시면 됩니다.

|  |
| --- |
| SELECT 함수명(값); |

**25.2.1 쿼리 실습**

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

생성한 사용자 함수를 직접 콘솔에서 실행을 해보도록 합니다. 함수를 통하여 실행된 쿼리는 컬럼명이 없습니다. Alias 별칭을 통하여 함수에 대한 컬럼값을 지정할 수 있습니다.

|  |
| --- |
| mysql> SELECT square(5) as aaa;  +------+  | aaa |  +------+  | 25 |  +------+  1 row in set (0.00 sec) |

정상적으로 입력한 값 5에 대해서 제곱 25를 출력하였습니다.

**25.3 변수 선언**

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

사용자 정의 함수에서는 내부적으로 사용할 수 있는 변수를 정의할 수 있습니다. 함수 내부에서 변수를 정의할 때는 DECLARE 키워드를 사용합니다.

**쿼리문법)**

|  |
| --- |
| DECLARE 변수명 자료형; |

선언한 변수 값의 대입은 INTO 명령을 사용합니다.

**25.3.1 쿼리 실습**

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

변수의 기능을 통하여 사용자 정의 함수를 만들어 실행해 보도록 합니다.

CREATE FUNCTION test(a INT) RETURNS DOUBLE

BEGIN

**DECLARE s DOUBLE;**

SELECT a \* a INTO s;

RETURN s;

END

위이 예에서는 변수 **s** 를 하나 생성하였습니다. Select 에서 연산처리한 값을 **INTO s** 문을 통하여 변수 s에 저장을 합니다. 저장한 변수를 반환 값으로 리턴합니다.

|  |
| --- |
| mysql> DELIMITER //  mysql> CREATE FUNCTION test(a INT) RETURNS DOUBLE  -> BEGIN  -> DECLARE s DOUBLE;  -> SELECT a \* a INTO s;  -> RETURN s;  -> END  -> //  Query OK, 0 rows affected (0.00 sec)  mysql> DELIMITER ;  mysql> SELECT test(3);  +---------+  | test(3) |  +---------+  | 9 |  +---------+  1 row in set (0.01 sec)  mysql> |

정상적으로 제곱값이 출력이 되었습니다.

**25.4 함수 확인**

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

사용자 정의로 저장된 함수들은 데이터베이스 시스템에 저장됩니다. 저장된 함수들은 재사용이 가능합니다. 사용자 정의된 함수의 내용들을 SHOW CREATE FUNCTION 명령을 통하여 확인을 할 수 있습니다.

**쿼리문법)**

|  |
| --- |
| SHOW CREATE FUNCTION 함수명; |

**25.4.1 쿼리 실습**

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

작성한 사용자 함수의 내용을 확인해 보도록 합니다.

|  |
| --- |
| mysql> show create function test;  +----------+------------------------+------------------------------------------------------------------------------------------------------------------------------------+----------------------+----------------------+--------------------+  | Function | sql\_mode | Create Function | character\_set\_client | collation\_connection | Database Collation |  +----------+------------------------+------------------------------------------------------------------------------------------------------------------------------------+----------------------+----------------------+--------------------+  | test | NO\_ENGINE\_SUBSTITUTION | CREATE DEFINER=`root`@`localhost` FUNCTION `test`(a INT) RETURNS double  BEGIN  DECLARE s DOUBLE;  SELECT a \* a INTO s;  RETURN s;  END | utf8 | utf8\_general\_ci | utf8\_general\_ci |  +----------+------------------------+------------------------------------------------------------------------------------------------------------------------------------+----------------------+----------------------+--------------------+  1 row in set (0.00 sec) |

**25.4.2 PHP 실습**

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

PHP 코드를 통하여 사용자 함수를 확인해 보는 메서드를 만들어 봅니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

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

**예제파일) func-100.php**

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

출력화면)

mysql connected!

쿼리성공] SHOW CREATE FUNCTION test

tables fields = 1

0=stdClass Object ( [Function] => test [sql\_mode] => NO\_ENGINE\_SUBSTITUTION [Create Function] => CREATE DEFINER=`root`@`localhost` FUNCTION `test`(a INT) RETURNS double BEGIN DECLARE s DOUBLE; SELECT a \* a INTO s; RETURN s; END [character\_set\_client] => utf8 [collation\_connection] => utf8\_general\_ci [Database Collation] => utf8\_general\_ci )

**25.5 함수 삭제**

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

작성한 사용자 함수를 삭제합니다. 삭제 또한 익숙한 DROP 명령을 통하여 삭제 할 수 있습니다.

**쿼리문법)**

|  |
| --- |
| DROP FUNCTION 함수명; |

**25.5.1 쿼리 실습**

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

사용자 함수를 삭제해 봅니다.

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

**25.5.2 PHP 실습**

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

PHP 코드를 통하여 사용자 작성 함수를 삭제하는 메서드를 만들어 봅니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

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

**예제파일) func-101.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  $name = "test";  if($db->dropFunction($name)){  echo $name." 함수 삭제성공! <br>";  if($rowss = $db->isFunction($name)){  echo "tables fields = ". count($rowss) . "<br>";    for($i=0;$i<count($rowss);$i++){  echo $i."=";  print\_r($rowss[$i]);  echo "<br>";  }  } else {  echo $name." 함수를 찾을 수 없습니다.";  }    } else {  echo "함수를 삭제 할 수 없습니다.<br>";  }    ?> |

출력화면)

mysql connected!

쿼리성공] DROP FUNCTION test

함수 삭제!

test 함수 삭제성공!

Error] SHOW CREATE FUNCTION test

test 함수를 찾을 수 없습니다.