jinyPHP 2권 - 023. 트리거

**23. 트리거**

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

트리거(Trigger)의 사전적 뜻은 “**방아쇠**를 당기다” 입니다. 즉, 하나의 SQL 동작을 함과 동시에 또 다른 동작을 처리하기 트랩을 걸어 놓은 것과 비슷합니다. 트리거 기능은 MYSQL 5.0 버전 이상에서는 새롭게 등장한 기능입니다.

트리거는 가장 많이 사용하는 INSERT, UPDATE, DELETE 등의 명령과 함께 사용을 하면 유용합니다. 특히, DELETE 와 같은 명령은 데이터를 삭제합니다. 만일 삭제 명령을 처리할 때 트리거를 설정하여 자동 백업 기능 트랩을 만들 수도 있습니다.

트리거는 하나의 쿼리 명령에 반응하는 또다른 쿼리 명령을 자동으로 연동을 만들어 놓은 것입니다. 과도한 트리거의 설정은 데이터 쿼리 처리시에 동작 속도가 느려 질수 있는 단점도 있습니다.

**23.1 트리거 생성**

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

트리거의 선언은 CREATE TRIGGER 명령을 통하여 생성합니다. 트리거는 쿼리 명령에 대해서 트랩을 설정하는 것과 유사합니다. 트리거를 명령 실행 전에 설정을 할 수도 있고, 명령이 실행 후에 동작을 하도록 선언을 할 수 있습니다.

**쿼리문법)**

|  |
| --- |
| CREATE TRIGGER 트리거명 BEFORE/AFTER 명령키워드  ON 테이블명 FOR EACH ROW  BEGIN  처리할 내용  END |

**BEFORE/AFTER**

BEFORE/AFTER는 뒤에 있는 쿼리명령 전에 작업을 할것인지, 이후에 작업을 할것인지를 선택하는 키워드 입니다. BEFORE 를 입력하면 명령전에 작업을 합니다. AFTER 를 입력하면 명령 후에 작업을 합니다.

**OLD. NEW.**

트리거는 명령키워드로 인하여 변경되는 데이터를 접근할 수 있습니다. 명령전의 데이터값은 OLD.컬럼 으로 , 명령후의 데이터값은 NEW.컬럼 으로 확인을 할 수 있습니다.

**23.1.1 쿼리 실습**

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

트리거에 대한 쿼리 실습을 합니다. 회원 테이블에서 회원을 삭제할 경우, 삭제된 회원을 다른 테이블에 백업 저장하는 예제 입니다.

CREATE TRIGGER backup\_mem BEFORE DELETE

ON members FOR EACH ROW

BEGIN

INSERT INTO members1 (FirstName,LastName) VALUES (OLD.FirstName, OLD.LastName);

END

위의 백업 트리거는 회원 정보의 백업 입니다. 데이터가 삭제가 된 이후에 백업을 할 수는 없기 때문에 BEFORE로 지정하여 삭제 명령전에 데이터를 백업하고, 실제 데이터를 테이블에서 삭제를 합니다.

|  |
| --- |
| mysql> DELIMITER //  mysql> CREATE TRIGGER backup\_mem BEFORE DELETE  -> ON members FOR EACH ROW  -> BEGIN  -> INSERT INTO members1 (FirstName,LastName) VALUES (OLD.FirstName, OLD.LastName);  -> END  -> //  Query OK, 0 rows affected (0.02 sec)  mysql> DELIMITER ; |

members 테이블에 대해서 DELETE 동작에 대한 트리거를 설정합니다. 트리거 동작은 DELETE 동작이 실행 되기 전(BEFORE)에 동작이 됩니다. 트리거의 이름은 backup\_mem 입니다

members 테이블의 데이터를 삭제 명령을 입력하게 되면, 삭제가 되기 전에 백업 테이블 members1 에 회원 정보를 복사한 후에 삭제를 합니다. 먼저 members 와 members1의 테이블의 내용을 확인해 봅니다.

|  |
| --- |
| mysql> select \* from 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 |  | 4 | eric | jung | seocho-gu | busan | korea | info | eric@jinyphp.com |  +----+----------+-----------+-------------+-------+---------+-----------+-------------------+  4 rows in set (0.01 sec)  mysql> select \* from members1;  +----+----------+-----------+-------------+-------+---------+  | 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 |  +----+----------+-----------+-------------+-------+---------+  3 rows in set (0.00 sec) |

members 테이블에는 4개의 데이터가 있고, 백업 테이블인 members1 에는 3개의 데이터가 있습니다. 트리거 설정을 통하여 데이터를 삭제해 봅니다.

|  |
| --- |
| mysql> delete from members where Id=4;  Query OK, 1 row affected (0.00 sec) |

삭제 명령 실행후의 테이블의 내용을 다시 한번 확인해 봅니다.

|  |
| --- |
| mysql> select \* from 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 (0.00 sec)  mysql> select \* from members1;  +----+----------+-----------+-------------+-------+---------+  | 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 |  | 0 | eric | jung | NULL | NULL | NULL |  +----+----------+-----------+-------------+-------+---------+  4 rows in set (0.00 sec) |

members 테이블에서 회원아이디 4를 삭제합니다. 정상적으로 삭제가 되어 members 테이블의 회원 목록은 3명으로 줄어 들었습니다. 하지만, members 테이블은 DELETE 에 대한 트리거가 설정되어 있기 때문에 실제적인 DELETE 작업을 하기전에 지정한 백업 쿼리가 먼저 실행이 됩니다.

members 테이블에서 삭제되는 회원을 membets1에 복사를 처리 한 후에, 요청한 삭제 쿼리를 실행하게 됩니다.

**23.1.2 PHP 실습**

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

PHP 코드를 통하여 트리거 실습을 진행해 봅니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

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

**예제파일) tri-01.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  $queryTrigger = "  CREATE TRIGGER backup\_mem BEFORE DELETE  ON members FOR EACH ROW  BEGIN  INSERT INTO members1 (FirstName,LastName) VALUES (OLD.FirstName, OLD.LastName);  END";  if ($db->createTrigger($queryTrigger)){  echo "Trigger 성공!";  } else {  echo "트리거를 생성이 되지 않았습니다.";  }    ?> |

출력화면)

mysql connected!

DELIMITER //

CREATE TRIGGER backup\_mem BEFORE DELETE ON members FOR EACH ROW BEGIN INSERT INTO members1 (FirstName,LastName) VALUES (OLD.FirstName, OLD.LastName); END

트리거 성공] CREATE TRIGGER backup\_mem BEFORE DELETE ON members FOR EACH ROW BEGIN INSERT INTO members1 (FirstName,LastName) VALUES (OLD.FirstName, OLD.LastName); END

DELIMITER ;

Trigger 성공!

**23.2 트리거 확인**

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

작성한 트리거는 데이터베이스 시스템에 저장이 됩니다. SHOW TRIGGERS; 명령은 트리거 설정에 대한 정보들을 출력합니다.

**쿼리문법)**

|  |
| --- |
| SHOW TRIGGERS; |

**23.2.1 쿼리 실습**

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

트리거의 내영을 콘솔명령을 통하여 확인해 봅니다. 앞에서 실습한 트리거의 내용들이 출력이 되는 것을 확인 할 수 있습니다.

|  |
| --- |
| mysql> SHOW TRIGGERS;  +------------+--------+---------+-------------------------------------------------------------------------------------------+--------+---------+------------------------+----------------+----------------------+----------------------+--------------------+  | Trigger | Event | Table | Statement | Timing | Created | sql\_mode | Definer | character\_set\_client | collation\_connection | Database Collation |  +------------+--------+---------+-------------------------------------------------------------------------------------------+--------+---------+------------------------+----------------+----------------------+----------------------+--------------------+  | backup\_mem | DELETE | members | BEGIN  INSERT INTO members1 (FirstName,LastName) VALUES (OLD.FirstName, OLD.LastName);  END | BEFORE | NULL | NO\_ENGINE\_SUBSTITUTION | root@localhost | utf8 | utf8\_general\_ci | utf8\_general\_ci |  +------------+--------+---------+-------------------------------------------------------------------------------------------+--------+---------+------------------------+----------------+----------------------+----------------------+--------------------+  1 row in set (0.03 sec) |

**23.2.2 PHP 실습**

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

PHP 코드를 통하여 트리거의 내용을 확인 할 수 있습니다. 관련 처리 메서드를 만들어 봅니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| 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);  }    } |

**예제파일) tri-02.php**

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

출력화면)

mysql connected!

쿼리성공] SHOW TRIGGERS

tables fields = 1

0=stdClass Object ( [Trigger] => backup\_mem [Event] => DELETE [Table] => members [Statement] => BEGIN INSERT INTO members1 (FirstName,LastName) VALUES (OLD.FirstName, OLD.LastName); END [Timing] => BEFORE [Created] => [sql\_mode] => NO\_ENGINE\_SUBSTITUTION [Definer] => root@localhost [character\_set\_client] => utf8 [collation\_connection] => utf8\_general\_ci [Database Collation] => utf8\_general\_ci )

**23.3 트리거 삭제**

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

작성한 트리거를 삭제할 수 있습니다. 트리거 삭제 또한 기존에 익숙한 DROP 명령을 통하여 삭제할 수 있습니다.

**쿼리문법)**

|  |
| --- |
| DROP TRIGGER 트리거명; |

**23.3.1 쿼리 실습**

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

기존에 실습한 트리거를 삭제해 보도록 합니다.

**drop trigger backup\_mem;**

|  |
| --- |
| mysql> drop trigger backup\_mem;  Query OK, 0 rows affected (0.00 sec) |

**23.3.2 PHP 실습**

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

PHP 코드를 통하여 트리거 삭제 메서드를 만들어 봅니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

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

**예제파일) tri-03.php**

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

출력화면)

mysql connected!

쿼리성공] DROP TRIGGER backup\_mem

트리거 삭제!

backup\_mem 트리거 삭제성공!

쿼리성공] SHOW TRIGGERS

backup\_mem 트리거를 찾을 수 없습니다.