jinyPHP 2권 - 013. 데이터 제한

**13. 데이터 제한**

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

지속적으로 데이터베이스 시스템을 운영하다가 보면 저장되는 데이터의 양은 매우 방대해 집니다. 데이터를 검색하기 위해서 사용하는 SELECT쿼리 명령과 WHERE 조건은 일치하는 데이터 모두를 출력합니다. 조건에 맞는 데이터의 개수의 상관이 없이 출력되는 것입니다.

만일 조건에 일치하는 데이터의 개수가 1,000개 10,000개라고 하면 출력되는 데이터의 양은 상당할 것입니다. 그리고 이렇게 출력되는 데이터를 확인하기도 힘들 거십니다.

우리는 인터넷 게시판 글 목록, 쇼핑몰의 상품목록 등을 검색할 때 하단에 페이지 번호를 보신적이 있습니다. 많은 상품과 글을 하나의 목록으로 출력을 한다면 우리가 쉽게 찾아보고 읽기가 어려울 것입니다. 이렇게 데이터가 많은 때에는 일정한 양을 나누어서 보여주는 것이 중요합니다.

SQL에서는 출력되는 데이터에 대해서 조건이 아닌, 출력되는 데이터의 개수를 제한할 수 있는 LIMIT 명령을 제공합니다.

**13.1 출력 수**

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

데이터베이스 시스템에 검색을 위하여 명령 쿼리를 실행을 합니다. 만일 실행되는 쿼리에 일치하는 조건의 데이터가 많을 경우 처리 시간은 오래 걸립니다. 또한 많은 출력 결과물을 처리하여 보기 좋게 화면에 출력하는 것 또한 오래 걸릴 것입니다.

빠른 처리와 가독성을 위해서 데이터를 나누어서 표시합니다. 이때 사용할 수 있는 키워드가 LIMIT 입니다. LIMIT 키워드는 출력되는 데이터의 개수를 제한합니다.

**쿼리문법)**

|  |
| --- |
| SELECT 컬럼명,컬럼명, … FROM 테이블명 LIMIT 표시할수; |

**13.1.1 쿼리실습**

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

쿼리의 맞는 데이터 검색을 처리하는데 LIMIT에서 지정한 숫자 만큼의 데이터만 가지고 옵니다.

**select code from orders limit 5;**

주문 목록에서 데이터를 가지고 오는 데 제일 위에서부터 5개만 가지고 옵니다. 이를 위해서 쿼리 뒤에 limit 5 명령을 추가하였습니다.

|  |
| --- |
| mysql> **select code from orders limit 5;**  +-------+  | code |  +-------+  | O\_001 |  | O\_002 |  | O\_003 |  | O\_004 |  | O\_005 |  +-------+  5 rows in set (0.00 sec) |

데이터를 5개만 출력합니다.

**13.1.2 PHP 실습**

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

PHP 코드를 이용하여 데이터 제한을 실습합니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // 입력된 쿼리의 출력제한 limit를 설정합니다.  public function limit($query, $a, $b)  {  $queryString = $query . " limit $a ";  If ($b) {  $queryString .= ",".$b;  }  return $queryString;  } |

**예제파일) limit-01.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  // 기본 select 쿼리  $query = "SELECT Id, FirstName, LastName FROM members";  // 기본쿼리에 limit 제한을 설정합니다.  $queryString = $db->limit($query,2);    if ($rowss = $db->selectRowss($queryString)) {  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] SELECT Id, FirstName, LastName FROM members limit 2

tables fields = 2

0=stdClass Object ( [Id] => 1 [FirstName] => lee [LastName] => hojin )

1=stdClass Object ( [Id] => 2 [FirstName] => [LastName] => jiny )

**13.2 부분출력**

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

쿼리 다음에 **LIMIT 숫자** 만 작성하는 것은 상단 데이터의 일부를 선택하는 것입니다. 데시보드(dashboard)와 같이 데이터의 상위 몇 개만 출력할 때는 이와 같은 간단한 출력 제한은 편리합니다.

하지만 제일 상단이 아닌 중간의 일부분만을 데이터로 읽어오고자 할 경우에는 LIMIT 인자를 더 주어야 합니다.

**쿼리문법)**

|  |
| --- |
| SELECT 컬럼명,컬럼명, … FROM 테이블명 LIMIT 시작, 표시할수; |

LIMIT 다음에 콤마(,)를 통하여 시작값과 출력할 개수를 지정하면 데이터의 일부 영역의 내용만 제한하여 출력을 할 수 있습니다.

**13.2.1 콘솔 실습**

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

시작값은 데이터를 출력할 offset 과 같은 성격을 가집니다. 데이터의 출력 위치를 이동후에, 지정한 데이터의 수 많큼 읽어서 화면에 출력을 합니다.

**select code from orders limit 2, 3;**

주문 목록에서 위치를 2로 이동한 다음에 연속된 3개의 데이터를 출력합니다.

|  |
| --- |
| mysql> **select code from orders limit 2, 3;**  +-------+  | code |  +-------+  | O\_003 |  | O\_004 |  | O\_005 |  +-------+  3 rows in set (0.00 sec) |

따라서 2번째 데이터부터 연속된 3개의 데이터만 출력을 합니다.

**13.2.2 PHP 실습**

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

PHP 코드를 이용하여 데이터의 일정 부분의 영역을 출력합니다.

**예제파일) limit-02.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  // 기본 select 쿼리  $query = "SELECT Id, FirstName, LastName FROM members";  // 기본쿼리에 limit 제한을 설정합니다.  $queryString = $db->limit($query,2,3);    if ($rowss = $db->selectRowss($queryString)) {  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] SELECT Id, FirstName, LastName FROM members limit 2 ,3

tables fields = 3

0=stdClass Object ( [Id] => 3 [FirstName] => kim [LastName] => james )

1=stdClass Object ( [Id] => 5 [FirstName] => jiny [LastName] => 1234 )

2=stdClass Object ( [Id] => 6 [FirstName] => jiny [LastName] => 1234 )

**13.3 OFFSET**

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

데이터의 출력 개수를 제어하는 OFFSET 키워드가 있습니다. OFFSET 은 LIMIT를 이용하여 부분 출력 하는 동작은 비슷합니다.

**쿼리문법)**

|  |
| --- |
| SELECT 컬럼명,컬럼명, … FROM 테이블명 LIMIT 레코드수 OFFSET 이동레코드수; |

데이터를 출력할 때 OFFSET 수 만큼 이동한 다음에 지정한 레코드 수 만큼을 출력합니다.

**13.3.1 콘솔 실습**

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

LIMIT 에서 처음에 이동하는 시작점을 OFFSET 부분의 위치로 이동을 헌것과 비슷합니다.

**select code from orders limit 3 offset 3;**

주문 목록에서 옵셋위치를 3으로 이동합니다. 이동 시점에서 연속된 데이터 3개를 읽어서 출력을 합니다.

|  |
| --- |
| mysql> select code from orders limit 3 offset 3;  +-------+  | code |  +-------+  | O\_004 |  | O\_005 |  | O\_006 |  +-------+  3 rows in set (0.00 sec) |

데이터를 3만큼 시작 위치를 이동한 상태에서 3개의 데이터만 출력합니다. 즉 LIMIT 3, 3 과 비슷합니다.

**13.3.2 PHP 실습**

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

PHP 코드를 통하여 OFFSET 처리에 대한 실습을 합니다.

**PHP 예제)**

mysql.class.php 파일에 메서드 예제를 추가합니다.

|  |
| --- |
| // 지정한 위치많큼 이동후, 갯수를 출력합니다.  public function offset($query, $a, $b){  $queryString = $query . " limit $b offset $a";  return $queryString;  } |

**예제파일) offset-01.php**

|  |
| --- |
| <?php  include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();  // 기본 select 쿼리  $query = "SELECT Id, FirstName, LastName FROM members";  // 기본쿼리에 2만큼 이동후 3개를 출력함.  $queryString = $db->offset($query,2,3);    if($rowss = $db->selectRowss($queryString)){  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] SELECT Id, FirstName, LastName FROM members limit 3 offset 2

tables fields = 3

0=stdClass Object ( [Id] => 3 [FirstName] => kim [LastName] => james )

1=stdClass Object ( [Id] => 5 [FirstName] => jiny [LastName] => 1234 )

2=stdClass Object ( [Id] => 6 [FirstName] => jiny [LastName] => 1234 )