jinyPHP 2권 - 016. 별칭

**16. 별칭**

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

별칭이란 테이블의 컬럼명을 다른 이름으로 재정의 하는 것입니다. 별칭은 쿼리가 실행될 때 한번만 적용이 됩니다. 별칭을 사용하는 이유는 이름이 긴 컬럼 때문 입니다. 길어진 컬럼 이름은 복잡한 쿼리 문장을 작성하는 데 불편합니다. 다른 짧은 이름으로 변환하여 처리를 하게 되면 간결한 쿼리 문장을 만들어 사용할 수 있습니다.

두번째로는 쿼리의 연산결과 값의 출력 입니다. 쿼리상에서 상수나 특정 연산등을 처리할 경우에는 컬럼 명칭이 지정이 되어 있지 않습니다. 이때, 별칭을 통하여 임시로 사용을 할 수 있는 컬럼명을 생성을 할 수 있습니다.

**16.1 명칭변경**

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

별칭은 기존의 컬럼이름을 다른 이름으로 임시로 바꾸어 사용할 수 있습니다. 새로운 이름의 별칭을 정의하는데 AS 키워드를 사용을 합니다.

**쿼리문법)**

|  |
| --- |
| SELECT 컬럼명 AS 별칭, 컬럼명 AS 별칭, … FROM 테이블명; |

컬럼명 뒤에 AS 키워드와 별칭명을 넣어서 SQL 쿼리문을 작성하면 됩니다.

**16.1.1 쿼리 실습**

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

별칭을 사용하면 좀더 컬럼명을 읽기 쉽게 변경하거나, 짧은 이름으로 변경할 수도 있습니다.

기존의 회원 테이를 조회하면 다음과 같습니다.

|  |
| --- |
| mysql> select \* from members;  +----+----------+-----------+-------------+-------+---------+  | 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 |  | 4 | eric | jung | seocho-gu | seoul | korea |  +----+----------+-----------+-------------+-------+---------+  4 rows in set (0.00 sec) |

컬럼의 이름이 영문으로 출력이 됩니다. 영문 컬럼 이름을 한글 이름으로 별칭을 지정해 봅니다.

**select LastName as 이름, FirstName as 성 from members;**

|  |
| --- |
| mysql> select LastName as 이름, FirstName as 성 from members;  +--------+--------+  | 이름 | 성 |  +--------+--------+  | hojin | lee |  | jiny | NULL |  | james | kim |  | eric | jung |  +--------+--------+  4 rows in set (0.01 sec) |

컬럼의 명칭이 한글로 변경이 되었습니다.

**16.1.2 PHP 실습**

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

PHP 코드에서도 별칭을 사용하여 쿼리를 전송해 보는 실습을 해봅니다.

**예제파일) alias-01.php**

|  |
| --- |
| <?php    include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();    $tbname = "members";    // 컬럼 지정조회  echo "===== 컬럼조회 =====<br>";  if ($rowss = $db->select($tbname,"FirstName as 이름, LastName as 성")) {  echo "tables fields = ". count($rowss) . "<br>";    for ($i=0;$i<count($rowss);$i++) {  echo $i."=";  print\_r($rowss[$i]);  echo "<br>";  }  }    ?> |

출력화면)

mysql connected!

===== 컬럼조회 =====

쿼리성공] SELECT FirstName as 이름, LastName as 성 FROM members;

tables fields = 6

0=stdClass Object ( [이름] => lee [성] => hojin )

1=stdClass Object ( [이름] => hojin [성] => lee )

2=stdClass Object ( [이름] => kim [성] => james )

3=stdClass Object ( [이름] => jiny [성] => 1234 )

4=stdClass Object ( [이름] => jiny [성] => 1234 )

5=stdClass Object ( [이름] => jiny [성] => 123400 )

**16.2 결과명칭**

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

별칭은 단순히 컬럼명의 변경 뿐만 아니라 쿼리를 통하여 연산을 처리할 때의 결과값에도 많이 사용을 합니다. 결과값은 별도의 컬럼명이 없기 때문입니다.

실습을 위해서 예제 테이블과 기초 데이터를 준비합니다.

|  |
| --- |
| // 테이블 생성  CREATE TABLE `products` (  `Id` int(11) NOT NULL AUTO\_INCREMENT,  `name` varchar(255) DEFAULT NULL,  `price` int(11) DEFAULT NULL,  `vat` int(11) DEFAULT NULL,  `stock` int(11) DEFAULT NULL,  PRIMARY KEY (`Id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  // 기초데이터  INSERT INTO `products` VALUES (1,'S-827',20000,2000,5),(2,'S-532D',36000,3600,7); |

**16.2.1 연산명칭**

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

정수형 타입의 컬럼 데이터는 +, -, /, \* 등의 사칙연산을 할 수 있습니다. 또한 함수를 실행하거나 할 때도 출력필드의 명칭을 정할 수 있습니다.

|  |
| --- |
| mysql> select \* from products;  +----+--------+-------+------+-------+  | Id | name | price | vat | stock |  +----+--------+-------+------+-------+  | 1 | S-827 | 20000 | 2000 | 5 |  | 2 | S-532D | 36000 | 3600 | 7 |  +----+--------+-------+------+-------+  2 rows in set (0.00 sec) |

제품목록 테이블에는 단가와 부가세 가격만 있습니다. 합계에 대한 컬럼 항목은 없습니다. 가격과 부가세 컬럼을 기준으로 합계를 출력해 봅니다.

**select price+vat as total from products;**

|  |
| --- |
| mysql> select price+vat as total from products;  +-------+  | total |  +-------+  | 22000 |  | 39600 |  +-------+  2 rows in set (0.00 sec) |

두개의 값이 합산된 결과가 출력됩니다. 또한 출력된 값의 컬럼 명칭으로 total 로 설정하였습니다. total 컬럼 명칭은 기존 테이블에서 없는 컬럼입니다. 연산의 의해서 새롭게 별칭으로 만들어진 임시 컬럼 이름 입니다.

AS 별칭 키워드를 이용하여 새로운 가상의 컬럼명을 하나 더 만들어 출력합니다.

**16.2.2 PHP 실습**

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

PHP 코드를 통하여 별칭 실습을 해봅니다.

**예제파일) alias-02.php**

|  |
| --- |
| <?php    include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();    $tbname = "products";  // 컬럼 지정조회  echo "===== 컬럼조회 =====<br>";  if($rowss = $db->select($tbname,"price+vat as total")){  echo "tables fields = ". count($rowss) . "<br>";    for($i=0;$i<count($rowss);$i++){  echo $i."=";  print\_r($rowss[$i]);  echo "<br>";  }  }    ?> |

출력화면)

mysql connected!

===== 컬럼조회 =====

쿼리성공] SELECT price+vat as total FROM products;

tables fields = 2

0=stdClass Object ( [total] => 22000 )

1=stdClass Object ( [total] => 39600 )

**16.2.3 함수명칭**

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

사칙연산 이외에도 SQL의 내장 함수를 사용할 때도 별칭은 매우 유용합니다. 내장함수를 사용할 때 출력 명칭은 불분명 합니다. 함수를 사용하게 되면 컬럼명이 함수명으로 출력이 됩니다

함수를 사용할 때도 AS 별칭을 사용하면 새로운 출력 명칭을 선언할 수 있습니다.

**select avg(price) as avg from products;**

위의 쿼리는 제품 테이블에서 가격 컬럼의 평균을 출력하는 예제 입니다.

|  |
| --- |
| mysql> select avg(price) as avg from products;  +------------+  | avg |  +------------+  | 28000.0000 |  +------------+  1 row in set (0.00 sec) |

별칭으로 선언한 이름으로 결과를 출력합니다. 만일 함수에 별칭을 같이 설정하지 않고 함수만 실행 했을 경우에는 함수명으로 출력이 됩니다.

|  |
| --- |
| mysql> select avg(price) from products;  +------------+  | avg(price) |  +------------+  | 28000.0000 |  +------------+  1 row in set (0.00 sec) |

**16.2.4 PHP 실습**

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

PHP 코드를 통하여 별칭 실습을 해봅니다.

**SELECT CONCAT(FirstName,'\_ ',LastName) AS username FROM members;**

위의 예제는 성과 이름 필드를 합쳐서 username 별칭으로 선언합니다. AS 별칭을 사용하여 새로운 컬럼을 추가할 수 있습니다.

**예제파일) alias-03.php**

|  |
| --- |
| <?php    include "dbinfo.php";  include "mysql.class.php";    // ++ Mysqli DB 연결.  $db = new JinyMysql();    $tbname = "members";  // 컬럼 지정조회  echo "===== 컬럼조회 =====<br>";  if($rowss = $db->select($tbname,"CONCAT(FirstName,'\_ ',LastName) AS username")){  echo "tables fields = ". count($rowss) . "<br>";    for($i=0;$i<count($rowss);$i++){  echo $i."=";  print\_r($rowss[$i]);  echo "<br>";  }  }    ?> |

출력화면)

mysql connected!

===== 컬럼조회 =====

쿼리성공] SELECT CONCAT(FirstName,'\_ ',LastName) AS username FROM members;

tables fields = 6

0=stdClass Object ( [username] => lee\_ hojin )

1=stdClass Object ( [username] => hojin\_ lee )

2=stdClass Object ( [username] => kim\_ james )

3=stdClass Object ( [username] => jiny\_ 1234 )

4=stdClass Object ( [username] => jiny\_ 1234 )

5=stdClass Object ( [username] => jiny\_ 123400 )