

CUBRID PHP 함수 cubrid_fetch_all 추가하기

.소개: 본 문서는 cubrid_fetch_all 함수를 추가하는 방법에 대하여 기술한다.

2009-6-1 고객지원팀, 남재우 **CUBRID**

적용 대상: CUBRID2008

목 차	
적용 환경.....	1
cubrid_fetch_all 함수.....	2
제약 사항.....	2
header file 수정.....	2
source file 수정.....	2
함수 목록에 추가.....	2
함수 추가.....	2
사용방법.....	5

적용 환경

본 문서는 아래와 같은 환경에서 적용 되었다.

큐브리드 2008 이상

Apache2.x, PHP5.x

cubrid_fetch_all 함수

일부 다른 데이터베이스에서는 *_fetch_all 형태의 함수를 제공한다. 이 함수는 데이터베이스에 select 질의후 결과를 배열에 한번에 다 가져오는 함수이다. 현재 CUBRID에서는 이 기능을 제공하지 않아 기존 소스를 조금 수정하여 해당 함수를 구현하도록 하였다.

CUBRID는 특성상 commit/rollback을 통한 트랜잭션 종료시 cursor 닫히게 되어 select 질의후 fetch가 진행중이었다면, commit/rollback 후 더 이상 진행할 수 없게 된다. 따라서 이런 경우 소스를 수정하거나 하여야 하는데 fetch_all 함수를 이용하여 결과를 모두 배열로 가져와 처리를 한다면 구지 소스를 수정하지 않아도 된다.

제약 사항

일부 데이터베이스에서 제공하는 fetch_all 함수는 옵션을 이용하여 행 기준 또는 열 기준으로 배열이 만들어 지는데 반해, CUBRID에서는 행 기준으로만 되도록 하였다. 즉 row[1][2]는 1번째 레코드의 2번째 필드를 의미하게 된다.

또한 결과 레코드 개수는 별도의 함수(cubrid_num_rows)를 이용하여 얻어야 한다. 예, \$num_rows = cubrid_num_rows(\$req);

header file 수정

CUBRID PHP 모듈 소스 디렉토리(예, cubrid-php-scm-latest) 아래 src 디렉토리의 php-cubrid.h 를 수정하여 cubrid_fetch_all 함수 정보를 추가한다. 기존 cubrid_fetch 를 찾아 그 아래에 추가한다.

```
i
PHP_FUNCTION(cubrid_fetch);

PHP_FUNCTION(cubrid_fetch_all); // cubrid_fetch_all 함수를 추가한다.
i
```

source file 수정

CUBRID PHP 모듈 소스 디렉토리(예, cubrid-php-scm-latest) 아래 src 디렉토리의 php-cubrid.c 를 수정한다.

함수 목록에 추가

cubrid_fetch_all 함수에 대한 함수 목록을 추가한다. 기존 cubrid_fetch 를 찾아 그 아래에 추가한다.

```
i
PHP_FE(cubrid_fetch, NULL)

PHP_FE(cubrid_fetch_all, NULL)
i
```

함수 추가

1. cubrid_fetch 함수를 찾아(PHP_FUNCTION(cubrid_fetch) 를 찾으면 된다) 그 내용을 그대로 복사한 후, 함수명을 cubrid_fetch_all 로 변경한다.

```
i
PHP_FUNCTION(cubrid_fetch_all)
i
```

- 함수 내부에 아래 내용들을 추가한다. 추가한 부분은 빨간색 및 설명을 표시하였다.

```
i
    T_CCI_ERROR error;

    int res;

// 추가 시작

    int i = 0;

    pzval temp_element;

// 추가 끝

    init_error();

i
```

- 함수 내부에 아래 내용들을 삭제한다. 그다음에는 삭제된 위치에 코드를 추가한다.

```
i

    res = cci_cursor(request->handle, 0, CCI_CURSOR_CURRENT, &error);

    if (res == CCI_ER_NO_MORE_DATA) {

        RETURN_FALSE;

    }

    res = cci_fetch(request->handle, &error);

    if (res < 0) {

        handle_error(res, &error);

        RETURN_FALSE;

    }

    if ((res = fetch_a_row(return_value, request->handle, real_type))

        != SUCCESS){

        handle_error(res, NULL);

        RETURN_FALSE;

    }

}
```

```

if (real_type & CUBRID_OBJECT) {

    if (return_value->type==IS_ARRAY) {

        convert_to_object(return_value);

    }

}

res = cci_cursor(request->handle, 1, CCI_CURSOR_CURRENT, &error);

if (res < 0 && res != CCI_ER_NO_MORE_DATA) {

    handle_error(res, &error);

    RETURN_FALSE;

}

```

4. 위에서 삭제한 부분에 아래 코드를 추가한다. 간단히 설명하면 결과를 저장할 변수를 만들고, 질의 결과를 가져와 loop 를 돌면서 해당 변수에 추가한다.

```

array_init(return_value);

while (1) {

    res = cci_fetch(request->handle, &error);

    if (res < 0) {

        handle_error(res, &error);

        RETURN_FALSE;

    }

    ALLOC_ZVAL(temp_element);

    if ((res = fetch_a_row(temp_element, request->handle, real_type))

        != SUCCESS){

        handle_error(res, NULL);

        FREE_ZVAL(temp_element);

        RETURN_FALSE;

    }

    INIT_PZVAL(temp_element);

```

```

        if (real_type & CUBRID_OBJECT) {
            if (return_value->type==IS_ARRAY) {
                convert_to_object(return_value);
            }
        }

        zend_hash_index_update(return_value->value.ht, i++,
            (void *)&temp_element, sizeof(pzval), NULL);

        res = cci_cursor(request->handle, 1, CCI_CURSOR_CURRENT, &error);

        if (res == CCI_ER_NO_MORE_DATA) {
            break;
        }
    }
}

```

사용방법

빌드후 cubrid_fetch_all 함수는 아래와 같이 사용할 수 있다.

```

i
$req = cubrid_execute($con, $sql);
$row_cnt = cubrid_num_rows($req);
$rows = cubrid_fetch_all($req);
cubrid_close_request($req);

i
$some_value = $rows[$row_index][$column_index];
$some_value2 = $rows[$row_index][$column_name];

i

```