

```

<?php
/*
 * userapp.php                                Colin MacCreery
 *
 * main program which handles the trans and log files.
 * Creates a DBHandler object that deals with all SQL
 * commands. Has four methods for retrieving, inserting
 * deleting and updating data.
 *
 *****/

/* MAIN *****/
// CREATE DATABASE HANDLER OBJECT
include 'dbhandler.php';
$db = new DBHandler('world.db');
// OPEN TRANSACTION FILE
$file = file('WorldTrans.txt');
// CREATE LOG FILE (OVERWRITE)
$log = fopen('log.txt', 'w');

// PROCESS EACH TRANSACTION
foreach($file as $line) {
    // WRITE OUT TRANSACTION TO LOG
    fwrite($log, $line);
    // CALL APPROPRIATE METHOD FOR TRANSACTION
    switch($line{0}) {
        case 'S':
            retrieveData($line, $db, $log);
            break;
        case 'I':
            insertData($line, $db, $log);
            break;
        case 'D':
            deleteData($line, $db, $log);
            break;
        case 'U':
            updateData($line, $db, $log);
            break;
    }
    // WRITE OUT BLANK LINE
    fwrite($log, "\n");
}
// CLOSE DATABASE & LOG FILE
$db->close();
fclose($log);

/* FUNCTIONS *****/
function retrieveData($line, $db, $log) {
    // GET RESULT FROM DB OBJECT
    $result = $db->select($line);
    // WRITE OUT HEADER
    for($i = 0; $i < $result->numColumns(); $i++) {
        if($i == $result->numColumns() - 1) {
            fwrite($log, $result->columnName($i));
        } else fwrite($log, $result->columnName($i) . "|");
    }
    fwrite($log, "\n");
    // WRITE OUT ENTRIES
    while($row = $result->fetchArray()) {
        for($i = 0; $i < $result->numColumns(); $i++) {
            if($i == $result->numColumns() - 1) {
                fwrite($log, $row[$result->columnName($i)]);
            } else fwrite($log, $row[$result->columnName($i)] . "|");
        }
        fwrite($log, "\n");
    }
}
}

```

```

function insertData($line, $db, $log) {
    // SPLIT RELEVANT DATA FROM LINE
    $line = explode("|", substr($line, 2));
    // IF LINE IS ALL-COLUMN INSERT ...
    if(sizeof($line) == 2) {
        // BUILD SQL COMMAND
        $insert = "INSERT INTO " . $line[0] . " VALUES ("
            . rtrim($line[1]) . ")";
        fwrite($log, $insert . "\n");
        // EXECUTE IT USING DB OBJECT
        $db->exec($insert);
    }
    // ... OTHERWISE IT IS A SOME-COLUMNS INSERT
    } else {
        // BUILD SQL COMMAND
        $insert = "INSERT INTO " . $line[0] . "(" . $line[1]
            . ") VALUES (" . rtrim($line[2]) . ")";
        fwrite($log, $insert . "\n");
        // EXECUTE IT USING DB OBJECT
        $db->exec($insert);
    }
    // WRITE OUT CONFIRMATION
    fwrite($log, "OK, INSERT WORKED\n");
}

function deleteData($line, $db, $log) {
    // SPLIT RELEVANT DATA FROM LINE
    $line = explode("|", substr($line, 2));
    // BUILD SQL COMMAND
    $delete = "DELETE FROM " . $line[0] . " WHERE " . $line[1]
        . " = " . rtrim($line[2]);
    fwrite($log, $delete . "\n");
    // EXECUTE IT USING DB OBJECT
    $db->exec($delete);
    // WRITE OUT CONFIRMATION
    fwrite($log, "OK, DELETE WORKED\n");
}

function updateData($update, $db, $log) {
    // ALREADY AN SQL COMMAND, EXECUTE IT USING DB OBJECT
    $db->exec($update);
    // WRITE OUT CONFIRMATION
    fwrite($log, "OK, UPDATE WORKED\n");
}
?>

```