W06 – Creating, Editing and Deleting Data

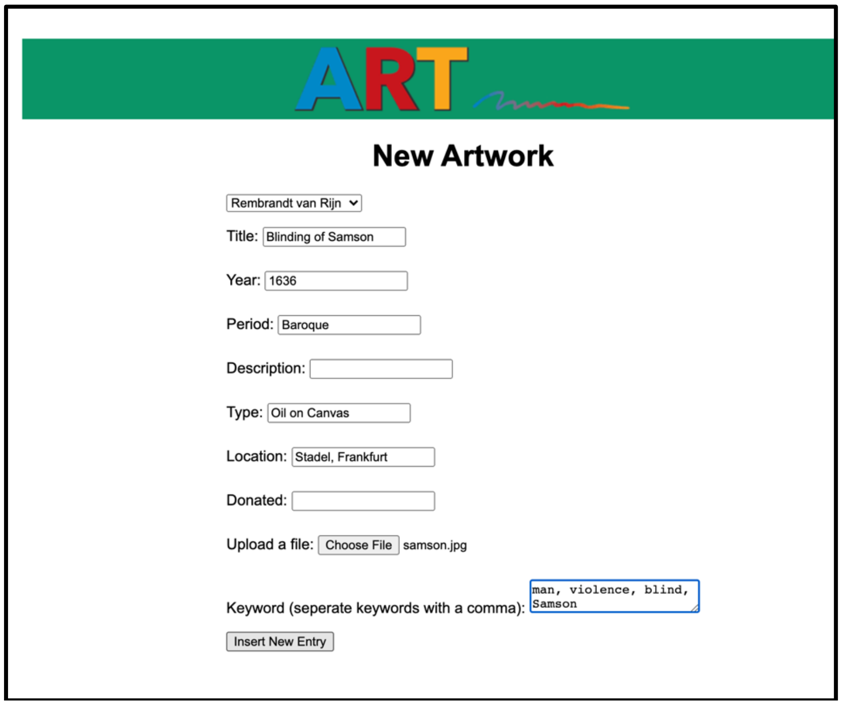
CASE STUDY - A look at how a web-based system might use a database and how insert, updates, and delete statements might be a part of the system whenever a new entity needs to be added, edited or deleted from the database.

Go to https://art.byuiwebdev.com and play around with the simple system and small database it uses. This should look familiar. It contains the same structure we were working with to design the art database last week.

Try some of the keywords in a subject search and see what happens. This system is something the public would see and maybe explore as a virtual museum.

Now let's look at the backend of the system. The backend would be something that the owner of the museum might use. This is where administration could add a new artist or painting. Or maybe they need to edit or delete an existing artist or artwork. This is not something that would be accessible to the public. A login would be required to only allow access to those who should be able to get into the system since these commands could make some big changes to the database.

What is super cool, is this admin doesn't need to know SQL additions, edits or deletions to the database. The SQL is all coded into the program. In fact, in one form that the admin fills out, 3 different entities' data is being inserted. The artwork, keyword and linking table. This all happens behind the scenes in the code and the owner never has to worry about SQL syntax or adding data to a linking table like we do.



New Artwork

Rembrandt van Rijn

Title: Blinding of Samson

Year: 1636

Period: Baroque

Description:

Type: Oil on Canvas

Location: Stadel, Frankfurt

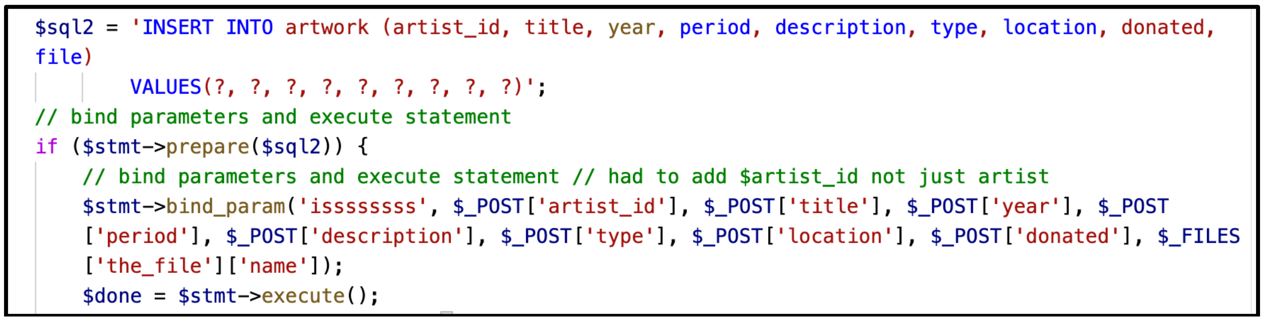
Donated:

Upload a file: Choose File samson.jpg

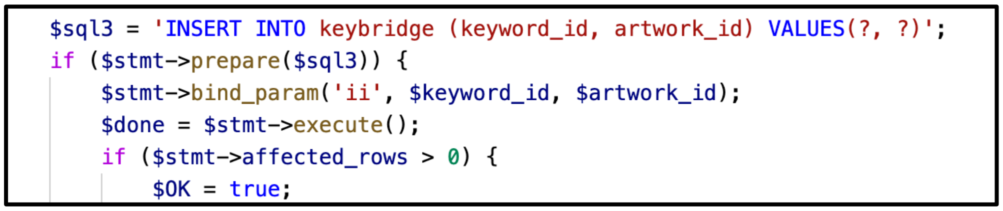
Keyword (seperate keywords with a comma): man, violence, blind, Samson

Insert New Entry

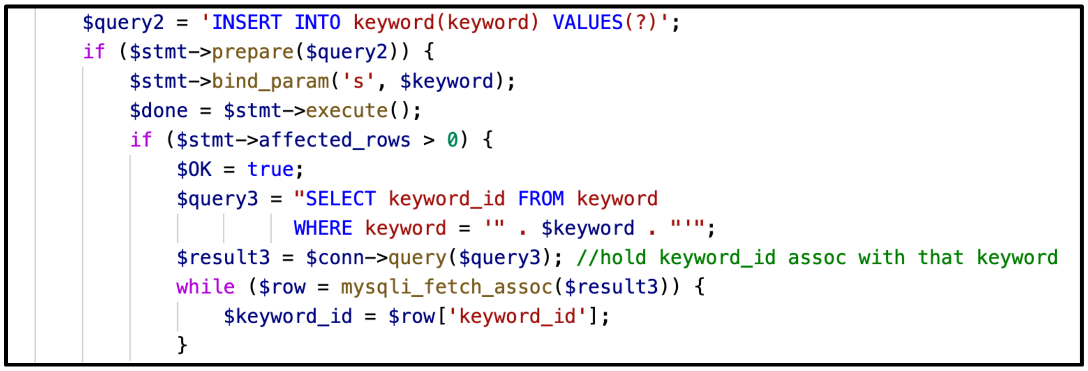
The following SQL statements look very different because they are combined with a backend language. In this case it's the language PHP. Don't worry if you don't understand this code. The point is just to show how the SQL is integrated into the PHP code here.



```
$sql2 = 'INSERT INTO artwork (artist_id, title, year, period, description, type, location, donated,
file)
        VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?)';
// bind parameters and execute statement
if ($stmt->prepare($sql2)) {
    // bind parameters and execute statement // had to add $artist_id not just artist
    $stmt->bind_param('issssssss', $_POST['artist_id'], $_POST['title'], $_POST['year'], $_POST
    ['period'], $_POST['description'], $_POST['type'], $_POST['location'], $_POST['donated'], $_FILES
    ['the_file']['name']);
    $done = $stmt->execute();
```

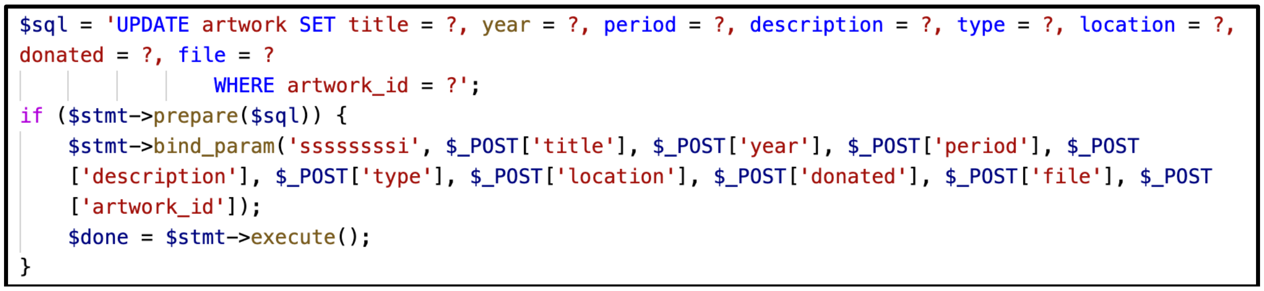


```
$sql3 = 'INSERT INTO keybridge (keyword_id, artwork_id) VALUES(?, ?)';
if ($stmt->prepare($sql3)) {
    $stmt->bind_param('ii', $keyword_id, $artwork_id);
    $done = $stmt->execute();
    if ($stmt->affected_rows > 0) {
        $OK = true;
```



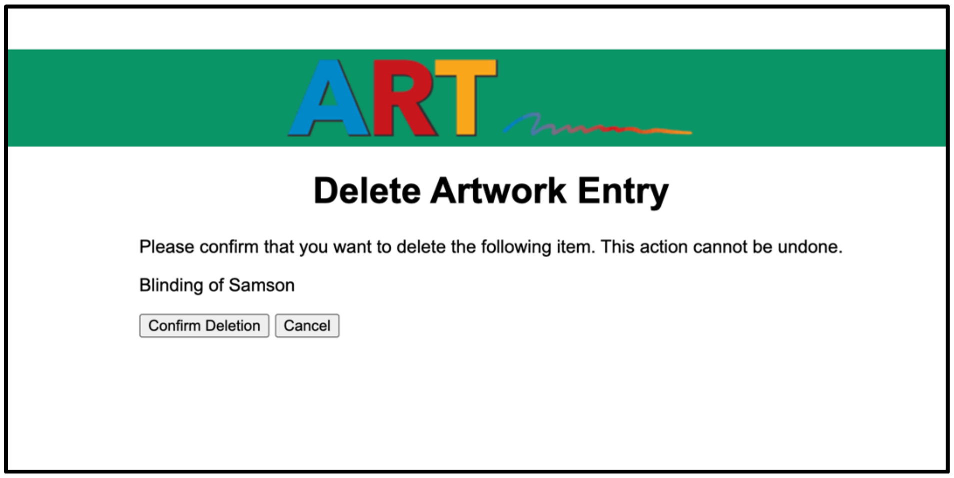
```
    $query2 = 'INSERT INTO keyword(keyword) VALUES(?)';
    if ($stmt->prepare($query2)) {
        $stmt->bind_param('s', $keyword);
        $done = $stmt->execute();
        if ($stmt->affected_rows > 0) {
            $OK = true;
            $query3 = "SELECT keyword_id FROM keyword
                    WHERE keyword = '" . $keyword . "'";
            $result3 = $conn->query($query3); //hold keyword_id assoc with that keyword
            while ($row = mysqli_fetch_assoc($result3)) {
                $keyword_id = $row['keyword_id'];
            }
```

When the admin needs to update a piece of artwork. The same type of form can be used to run an update statement.



```
$sql = 'UPDATE artwork SET title = ?, year = ?, period = ?, description = ?, type = ?, location = ?,
donated = ?, file = ?
                WHERE artwork_id = ?';
if ($stmt->prepare($sql)) {
    $stmt->bind_param('ssssssssi', $_POST['title'], $_POST['year'], $_POST['period'], $_POST
    ['description'], $_POST['type'], $_POST['location'], $_POST['donated'], $_POST['file'], $_POST
    ['artwork_id']);
    $done = $stmt->execute();
}
```

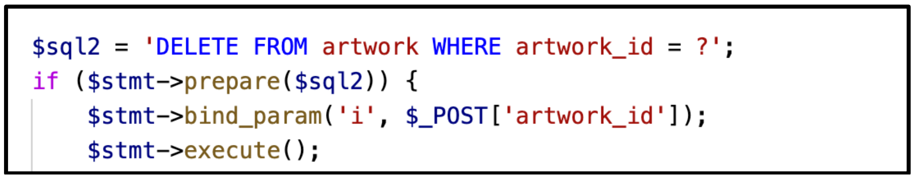
When the admin might want to delete a piece of artwork there are also safeguards in place double checking that they really want to do this.



Delete Artwork Entry

Please confirm that you want to delete the following item. This action cannot be undone.

Blinding of Samson



```
$sql2 = 'DELETE FROM artwork WHERE artwork_id = ?';
if ($stmt->prepare($sql2)) {
    $stmt->bind_param('i', $_POST['artwork_id']);
    $stmt->execute();
```

Not all end users of a database system will know SQL, but somewhere in the code before the actual database is changed, you will see SQL in action.