

PROJECT #1: "Auction Site" v1.0

PART 3: "Creating tables for the auction site"

Objectives:

- Create all tables required for the auction project
- Populate the tables with sample data.

Instructions:

Before you can test the **Database** class, you will first need to create the tables required for the auction project. The following tables are required for the auction project:

<i>Table Name</i>	<i>The table will contain:</i>
bids	Bids made on items
categories	Different item categories
images	Images of sale items
items	The items for sale on site
payments	Payment information from PayPal
sessions	Session information
users	Users who have access to the site

1. Download the provided starter files **dbfiles_auction.zip** from blackboard and unpack the files into the project's **public/** directory.

2. Review the list of fields and their intended purpose in each table:

Table Name: bids

Field Name	Data Type	Description
id	INT UNSIGNED, AUTO_INCREMENT PRIMARY KEY	Primary key for the table. It should be auto incremented. Uniquely identifies each row in the table.
item_id	INT UNSIGNED	A foreign key which matches the bid to the item.
amount	DECIMAL(13,2)	The bid amount
user_id	INT UNSIGNED	A foreign key which matches the bid to the user who placed it.

Table Name: categories

Field Name	Data Type	Description
id	INT UNSIGNED, AUTO_INCREMENT PRIMARY KEY	Primary key for the table. It should be auto incremented. Uniquely identifies each row in the table.
cat	VARCHAR(20)	The category names for the items

Table Name: images

Field Name	Data Type	Description
id	INT UNSIGNED, AUTO_INCREMENT PRIMARY KEY	Primary key for the table. It should be auto incremented. Uniquely identifies each row in the table.
item_id	INT UNSIGNED	A foreign key which matches the image with the item.
name	VARCHAR(100)	The name of the image file.

Table Name: items

Field Name	Data Type	Description
id	INT UNSIGNED, AUTO_INCREMENT PRIMARY KEY	Primary key for the table. It should be auto incremented. Uniquely identifies each row in the table.
user_id	INT UNSIGNED	A foreign key which matches the item to the user who owns it.
cat_id	INT UNSIGNED	A foreign key which matches the item to its category
name	VARCHAR(100)	A title for the item
price	DECIMAL(13,2)	The starting price of the item
description	TEXT	The item description
date	DATETIME	The date and time the auction expires
notified	TINYINT	To indicate whether the buyer and seller has been contacted after the auction has completed.

Table Name: payments

Field Name	Data Type	Description
id	INT UNSIGNED, AUTO_INCREMENT PRIMARY KEY	Primary key for the table. It should be auto incremented. Uniquely identifies each row in the table.
txn_id	VARCHAR(20)	The transaction id provided by PayPal.
mc_gross	DECIMAL(13,2)	The total value including taxes.
payment_status	VARCHAR(25)	The PayPal status code for the transaction
item_number	VARCHAR(25)	The id of the product sold
item_name	VARCHAR(255)	The name of the product sold
payer_id	VARCHAR(50)	The id of the PayPal buyer
payer_email	VARCHAR(255)	The email of the PayPal buyer
full_name	VARCHAR(255)	The full name of the PayPal buyer
address_street	VARCHAR(255)	The address of the PayPal buyer
address_city	VARCHAR(255)	The city of the PayPal buyer

address_state	VARCHAR(255)	The state/province of the PayPal buyer
address_zip	VARCHAR(20)	The zip code/postal code of the PayPal buyer
address_country	VARCHAR(255)	The country of the PayPal buyer
payment_date	VARCHAR(255)	The transaction date for payment

Table Name: sessions

Field Name	Data Type	Description
id	CHAR(32), NOT NULL PRIMARY KEY	Primary key for the table. Uniquely identifies each row in the table.
data	MEDIUMTEXT	The serialized data for each user for each session.
last_accessed	timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()	Tracks the last time a user accessed the website

Table Name: users

Field Name	Data Type	Description
id	INT UNSIGNED, AUTO_INCREMENT PRIMARY KEY	Primary key for the table. It should be auto incremented. Uniquely identifies each row in the table.
username	VARCHAR(50)	The username used to login to the website.
password	VARCHAR(255)	The password used to login.
email	VARCHAR(100)	The email used for verification and to notify auction status.
verify	VARCHAR(20)	The verification string to confirm email validity.
active	TINYINT	To determine if the user's email address has been verified.

3. Ensure the **public/dbfiles_auction/** and all contained files are uploaded to your scweb.ca account.
4. Visit the **create_auction_tables.php** script on your scweb.ca account and press the “**Yes**” button to generate the tables in the database specified in your **config.php**.

Table Refresh Script

Would you like to refresh the Auction tables (*bids, categories, images, items, payments, sessions, users*)?

Warning: existing tables and records will be destroyed!

5. Now that you have all the tables created, you need to add some sample data to some of the tables.

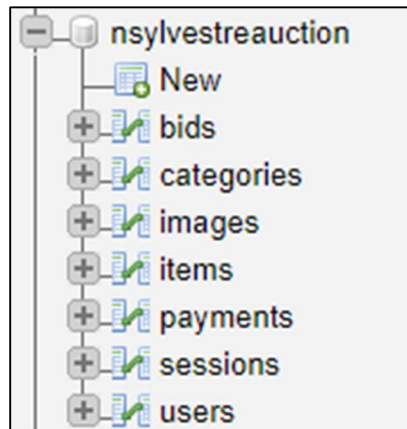
6. Using the **manage_categories.php** script, insert two records in the **categories** table:
 - Scenery
 - Wildlife
7. Using the **manage_users.php** script, insert two records in the **users** table. Pick names and passwords you will remember. These will be used as normal user accounts to create mock items and bids.
8. Using the **manage_items.php** script, insert two records in the **items** table (**See the table below**). These will act as “active” auction items.

User	Category	Name	Price	Description	Date	Image
<i>Select the first user you created</i>	Scenery	Lake Huron Sunset	50.00	Sunset on the east shore of Lake Huron	20XX-12-15 11:45 AM	sunset.jpg
<i>Select the first user you created</i>	Scenery	October Reflection	30.00	Fall reflection at the Pinery Provincial Park	20XX-12-15 11:45 AM	reflection.jpg

Note: Replace the year in the date field with the current year.

Note: Use Google Chrome if you are having issues with inserting new items into the table.

9. Verify all records were created by visiting phpmyadmin.scweb.ca and expanding the auction database. You should see that the tables have been successfully created.



Clicking on each of the tables should show all records within them. Verify all records have been successfully inserted.

10. There is one final thing you must do before you can move onto the next section - you must move the **dbfiles_auction/** and all of the scripts contained within to a directory outside of the **public/** directory. This is so no one else can delete the records from your database.

Using PHPStorm, drag and drop the **dbfiles_auction/** from the **public/** directory to the **app/** directory. This move should be mirrored on the server automatically by PHPStorm, however, verify it was also moved on the remote host.

You're now ready to move on to the next section.