Task C: SQL Files, CSV Files, & PHP Files

**create.sql:**

*-- Create a new database named 'bookstore'*

CREATE DATABASE bookstore;

*-- Select the 'bookstore' database for use*

USE bookstore;

*-- Create the 'User' table*

CREATE TABLE User (

user\_id int NOT NULL,

user\_password varchar(20),

name varchar(30),

email varchar(50),

CONSTRAINT UserPK PRIMARY KEY(user\_id)

);

*-- Create the 'Customer' table*

CREATE TABLE Customer (

user\_id int NOT NULL,

owned\_books varchar(255) NULL,

address varchar(100),

CONSTRAINT CustomerPK PRIMARY KEY (user\_id),

FOREIGN KEY (user\_id) REFERENCES User(user\_id) ON DELETE CASCADE

);

*-- Create the 'Employee' table*

CREATE TABLE Employee (

user\_id int NOT NULL,

salary int,

CONSTRAINT EmployeePK PRIMARY KEY (user\_id),

FOREIGN KEY (user\_id) REFERENCES User(user\_id) ON DELETE CASCADE

);

*-- Create the 'Book' table*

CREATE TABLE Book (

book\_id int NOT NULL,

price int,

age\_rating int,

genre varchar(20),

quantity\_available int,

title varchar(100),

CONSTRAINT BookPK PRIMARY KEY(book\_id)

);

*-- Create the 'Orders' table*

CREATE TABLE Orders (

order\_id int NOT NULL,

customer\_id int,

book\_id int,

quantity int,

order\_date DATE,

CONSTRAINT OrderPK PRIMARY KEY (order\_id),

FOREIGN KEY (customer\_id) REFERENCES Customer(user\_id) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY (book\_id) REFERENCES Book(book\_id) ON DELETE SET NULL ON UPDATE CASCADE

);

*-- Create the 'Supplier' table*

CREATE TABLE Supplier (

supplier\_id int NOT NULL,

book\_id int,

price double,

quantity int,

CONSTRAINT SupplierPK PRIMARY KEY (supplier\_id),

FOREIGN KEY (book\_id) REFERENCES Book(book\_id) ON DELETE SET NULL ON UPDATE CASCADE

);

*-- Create the 'Author' table*

CREATE TABLE Author(

book\_id int NOT NULL,

author\_id int NOT NULL,

CONSTRAINT AuthorPK PRIMARY KEY (author\_id),

FOREIGN KEY (book\_id) REFERENCES Book(book\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

*-- Create the 'Owned\_By' relation table*

CREATE TABLE Owned\_By(

book\_id int NOT NULL,

user\_id int NOT NULL,

CONSTRAINT OwnedByPK PRIMARY KEY (book\_id, user\_id),

FOREIGN KEY (book\_id) REFERENCES Book(book\_id) ON DELETE RESTRICT ON UPDATE CASCADE,

FOREIGN KEY (user\_id) REFERENCES User(user\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

**load.sql:**

USE bookstore;

LOAD DATA LOCAL INFILE 'User.csv'

INTO TABLE User

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE 'Customer.csv'

INTO TABLE Customer

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE 'Employee.csv'

INTO TABLE Employee

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE 'Book.csv'

INTO TABLE Book

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE 'Orders.csv'

INTO TABLE Orders

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE 'Supplier.csv'

INTO TABLE Supplier

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE 'Author.csv'

INTO TABLE Author

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE 'Owned\_By.csv'

INTO TABLE Owned\_By

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

**User.csv:**

user\_id,user\_password,name,email

1,pass123,Alice Johnson,alice.johnson@example.com

2,pass456,Bob Smith,bob.smith@example.com

3,pass123,John Doe,john.doe@example.com

4,pass456,Jane Doe,jane.doe@example.com

5,pass150,Michael Smith,michael.smith@example.com

6,pass155,Emma Johnson,emma.johnson@example.com

7,pass160,Olivia Williams,olivia.williams@example.com

8,pass165,Noah Brown,noah.brown@example.com

9,pass170,Ava Jones,ava.jones@example.com

10,pass175,Isabella Garcia,isabella.garcia@example.com

11,pass180,Sophia Miller,sophia.miller@example.com

12,pass185,Liam Davis,liam.davis@example.com

13,pass190,Mason Rodriguez,mason.rodriguez@example.com

14,pass195,Logan Martinez,logan.martinez@example.com

15,pass200,Lucas Hernandez,lucas.hernandez@example.com

16,pass205,Jackson Lopez,jackson.lopez@example.com

17,pass210,Aiden Gonzalez,aiden.gonzalez@example.com

18,pass215,Ethan Wilson,ethan.wilson@example.com

19,pass220,James Anderson,james.anderson@example.com

20,pass225,Charlotte Thomas,charlotte.thomas@example.com

21,pass230,Amelia Taylor,amelia.taylor@example.com

22,pass235,Mia Moore,mia.moore@example.com

23,pass240,Harper Jackson,harper.jackson@example.com

24,pass245,Evelyn Martin,evelyn.martin@example.com

**Supplier.csv:**

supplier\_id,book\_id,price,quantity

1,1,18,100

2,2,22,150

3,3,17,125

4,4,19,150

5,5,21,175

6,6,23,200

7,7,25,225

8,8,27,250

9,9,29,275

10,10,31,300

11,11,33,325

12,12,35,350

**Owned\_By.csv:**

book\_id,user\_id

1,1

2,2

19,26

10,30

16,23

13,21

12,15

14,6

15,11

17,30

22,25

20,15

18,9

21,24

11,12

24,13

23,20

27,32

**Orders.csv:**

order\_id,customer\_id,book\_id,quantity,order\_date

1,1,1,2,2024-01-10

2,2,2,1,2024-02-15

3,10,15,5,2024-02-18

4,14,10,5,2024-03-04

5,12,7,3,2024-03-13

6,11,1,3,2024-03-31

7,17,5,2,2024-04-15

8,19,20,4,2024-05-13

9,18,10,4,2024-06-08

10,20,19,4,2024-06-16

**Employee.csv:**

user\_id,salary

3,60000

4,50000

5,45000

6,70000

7,55000

8,52000

9,48000

11,64000

**Customer.csv:**

user\_id,owned\_books,address

1,"1,2",123 Maple St, Anytown

2,"3,4",456 Oak Rd, Anycity

10,"5,6",789 Pine Ln, Anyville

11,"7",1012 Birch Blvd, Somewhere

12,"8,9,10",345 Cedar Ave, Anyplace

13,"11,12,13",678 Spruce St, Nowhere

14,"14,15",901 Willow Way, Everywhere

15,"16",234 Elm St, ThisTown

16,"17,18,19",567 Aspen Ct, ThatTown

17,"20",890 Redwood Rd, TheOtherTown

18,"1,21",123 Maple St, NewTown

19,"2,22,23",456 Oak Rd, OldCity

20, “4,13”,324 Windingham Rd, NewCity

21, “25”,635 Stardust Ln, OldTown

**Book.csv:** book\_id,price,age\_rating,genre,quantity\_available,title

1,20,12,Fantasy,10,Magic Quest

2,25,16,Sci-Fi,5,Star Voyage

6,15,12,Sci-Fi,2,Cloak of Darkness

18,10,15,Sci-Fi,6,Defenders of Peace

19,17,20,Sci-Fi,3,Blue Shadow Virus

3,18,10,Mystery,15,The Secret of the Old Clock

4,22,18,Horror,8,Nightmare Alley

5,30,14,Adventure,12,Jungle Escape

10,17,12,Fantasy,20,The Enchanted Forest

7,24,16,Sci-Fi,7,Galactic Odyssey

8,20,10,Historical,10,The Forgotten Empire

9,27,18,Thriller,5,The Last Witness

13,15,14,Romance,9,Love in the Time of Cholera

11,19,12,Fantasy,17,The Crystal Cave

12,23,16,Mystery,11,Murder on the Orient Express

20,21,14,Drama,4,Echoes of the Past

21,26,18,Non-Fiction,8,The Innovators

22,28,12,Children,15,The Adventures of Pixie

23,16,10,Comedy,10,Laughter in the Air

24,23,16,Action,6,The Final Stand

25,19,14,Fantasy,9,Guardians of the Realm

26,22,18,Mystery,5,Shadows in the Mist

27,25,12,Sci-Fi,12,Voyage Beyond

28,17,10,Biography,7,Life of Discovery

29,30,20,Thriller,3,Edge of Tomorrow

30,12,22,Action,16,Shadow of Malevolence

**Author.csv:**

book\_id,author\_id

1,101

2,102

3,201

4,105

5,183

6,192

7,106

8,231

9,091

10,211

**PHP Files for Functionality:**

**addBook.php:**<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

//$bookId = $\_POST['book\_id'];

//$price = $\_POST['price'];

//$ageRating = $\_POST['age\_rating'];

//$genre = $\_POST['genre'];

//$quantityAvailable = $\_POST['quantity\_available'];

//$title = $\_POST['title'];

// Input validation

$bookId = filter\_input(INPUT\_POST, 'book\_id', FILTER\_VALIDATE\_INT);

$price = filter\_input(INPUT\_POST, 'price', FILTER\_VALIDATE\_FLOAT);

$ageRating = filter\_input(INPUT\_POST, 'age\_rating', FILTER\_VALIDATE\_INT);

$genre = filter\_input(INPUT\_POST, 'genre', FILTER\_SANITIZE\_STRING);

$quantityAvailable = filter\_input(INPUT\_POST, 'quantity\_available', FILTER\_VALIDATE\_INT);

//$title = filter\_input(INPUT\_POST, 'title', FILTER\_SANITIZE\_STRING);

if ($bookId === false || $price === false || $ageRating === false || $quantityAvailable === false) {

die('Invalid input');

}

// Database connection

$conn = new mysqli('localhost', 'root', '', 'bookstore');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "INSERT INTO Book (book\_id, price, age\_rating, genre, quantity\_available, title) VALUES (?, ?, ?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("idssis", $bookId, $price, $ageRating, $genre, $quantityAvailable, $title);

$stmt->execute();

if ($stmt->affected\_rows > 0) {

echo "New book added successfully.";

} else {

echo "Error: " . $stmt->error;

}

$stmt->close();

$conn->close();

}

?>

**addCustomer.php:**<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

//$userId = $\_POST['user\_id'];

//$password = $\_POST['user\_password'];

//$name = $\_POST['name'];

//$email = $\_POST['email'];

//$address = $\_POST['address'];

//$ownedBooks = $\_POST['owned\_books'];

// Database connection

$conn = new mysqli('localhost', 'root', '', 'bookstore');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// First INSERT INTO User

$sqlUser = "INSERT INTO User (user\_id, user\_password, name, email) VALUES (?, ?, ?, ?)";

$stmtUser = $conn->prepare($sqlUser);

$stmtUser->bind\_param("isss", $userId, $password, $name, $email);

$stmtUser->execute();

if ($stmtUser->affected\_rows > 0) {

echo "New user added successfully. ";

} else {

echo "Error: " . $stmtUser->error;

}

$stmtUser->close();

// Then INSERT INTO Employee

$sqlEmployee = "INSERT INTO Customer (user\_id, address, owned\_books) VALUES (?, ?, ?)";

$stmtEmployee = $conn->prepare($sqlEmployee);

$stmtEmployee->bind\_param("iss", $userId, $address, $ownedBooks);

$stmtEmployee->execute();

if ($stmtEmployee->affected\_rows > 0) {

echo "Customer record created successfully.";

} else {

echo "Error: " . $stmtEmployee->error;

}

$stmtEmployee->close();

$conn->close();

}

?>

**addEmployee.php:**<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$userId = $\_POST['user\_id'];

$password = $\_POST['user\_password'];

$name = $\_POST['name'];

$email = $\_POST['email'];

$salary = $\_POST['salary'];

// Database connection

$conn = new mysqli('localhost', 'root', '', 'bookstore');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// First INSERT INTO User

$sqlUser = "INSERT INTO User (user\_id, user\_password, name, email) VALUES (?, ?, ?, ?)";

$stmtUser = $conn->prepare($sqlUser);

$stmtUser->bind\_param("isss", $userId, $password, $name, $email);

$stmtUser->execute();

if ($stmtUser->affected\_rows > 0) {

echo "New user added successfully. ";

} else {

echo "Error: " . $stmtUser->error;

}

$stmtUser->close();

// Then INSERT INTO Employee

$sqlEmployee = "INSERT INTO Employee (user\_id, salary) VALUES (?, ?)";

$stmtEmployee = $conn->prepare($sqlEmployee);

$stmtEmployee->bind\_param("ii", $userId, $salary);

$stmtEmployee->execute();

if ($stmtEmployee->affected\_rows > 0) {

echo "Employee record created successfully.";

} else {

echo "Error: " . $stmtEmployee->error;

}

$stmtEmployee->close();

$conn->close();

}

?>

**addSupplier.php:**

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Collect input values

$supplierId = $\_POST['supplier\_id'];

$bookId = $\_POST['book\_id'];

$price = $\_POST['price'];

$quantity = $\_POST['quantity'];

// ... add more inputs as needed

// Database connection

$conn = new mysqli('localhost', 'root', '', 'bookstore');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// SQL query

$sql = "INSERT INTO Supplier (supplier\_id, book\_id, price, quantity) VALUES (?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("iidi", $supplierId, $bookId, $price, $quantity);

$stmt->execute();

if ($stmt->affected\_rows > 0) {

echo "Supplier added successfully.";

} else {

echo "Error: " . $stmt->error;

}

$stmt->close();

$conn->close();

}

?>

**addUser.php:**

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$userId = $\_POST['user\_id'];

$password = $\_POST['user\_password'];

$name = $\_POST['name'];

$email = $\_POST['email'];

// Database connection

$conn = new mysqli('localhost', 'root', '', 'bookstore');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "INSERT INTO User (user\_id, user\_password, name, email) VALUES (?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("isss", $userId, $password, $name, $email);

$stmt->execute();

if ($stmt->affected\_rows > 0) {

echo "New user added successfully.";

} else {

echo "Error: " . $stmt->error;

}

$stmt->close();

$conn->close();

}

?>

**createOrder.php:**<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Collect input values

$orderId = $\_POST['order\_id'];

$customerId = $\_POST['customer\_id'];

$bookId = $\_POST['book\_id'];

$quantity = $\_POST['quantity'];

$orderDate = $\_POST['order\_date'];

//$orderDate = date('Y-m-d', strtotime($\_POST['order\_date']));

// Convert the order date from string to a DateTime object and check if it's a valid date

$orderDateObject = DateTime::createFromFormat('Y-m-d', $orderDate);

$currentDateObject = new DateTime(); // Today's date

// Ensure the order date is a valid future date

if ($orderDateObject === false || $orderDateObject <= $currentDateObject) {

die("Invalid order date. The date must be after today.");

}

// Database connection

$conn = new mysqli('localhost', 'root', '', 'bookstore');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// SQL query

$sql = "INSERT INTO Orders (order\_id, customer\_id, book\_id, quantity, order\_date) VALUES (?, ?, ?, ?, ?)"; // Modify as per table

$stmt = $conn->prepare($sql);

$stmt->bind\_param("iiiis", $orderId, $customerId, $bookId, $quantity, $orderDate);

$stmt->execute();

if ($stmt->affected\_rows > 0) {

echo "Order created successfully.";

} else {

echo "Error: " . $stmt->error;

}

$stmt->close();

$conn->close();

}

?>

**updateBook.php:**

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$bookId = $\_POST['book\_id'];

$price = $\_POST['price'];

$ageRating = $\_POST['age\_rating'];

$genre = $\_POST['genre'];

$quantityAvailable = $\_POST['quantity\_available'];

$title = $\_POST['title'];

// Database connection

$conn = new mysqli('localhost', 'root', '', 'bookstore');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Update statement instead of insert

$sql = "UPDATE Book SET price = ?, age\_rating = ?, genre = ?, quantity\_available = ?, title = ? WHERE book\_id = ?";

$stmt = $conn->prepare($sql);

// Note the order of variables and their types in bind\_param has changed

$stmt->bind\_param("dssisi", $price, $ageRating, $genre, $quantityAvailable, $title, $bookId);

$stmt->execute();

// Check if the update was successful

if ($stmt->affected\_rows > 0) {

echo "Book updated successfully.";

} else {

echo "Error: " . $stmt->error;

}

$stmt->close();

$conn->close();

}

?>

**login.php:**

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$email = $\_POST['email'];

$password = $\_POST['user\_password'];

// Database connection

$conn = new mysqli('localhost', 'root', '', 'bookstore');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

//to prevent from mysqli injection

$email = stripcslashes($email);

$password = stripcslashes($password);

$email = mysqli\_real\_escape\_string($conn, $email);

$password = mysqli\_real\_escape\_string($conn, $password);

$sql = "select \*from User = '$email' and password = '$user\_password’;

$result = mysqli\_query($conn, $sql);

$row = mysqli\_fetch\_array($result, MYSQLI\_ASSOC);

$count = mysqli\_num\_rows($result);

if($count == 1){

echo "<h1><center> Login successful </center></h1>";

}

else{

echo "<h1> Login failed. Invalid username or password.</h1>";

}

$conn->close();

}

?>