

# INLS 672

## Web Development 2

# PHP Data Objects (PDO)



Joan Boone  
[jpboone@email.unc.edu](mailto:jpboone@email.unc.edu)

# Web Application Design

- As applications incorporate more features, they become more complex and may include
  - more functions
  - more detailed user interfaces
  - more database content
- Similarly, as you develop more code, it can become more difficult to manage, maintain, test, and debug
- Two techniques that can improve the design and maintainability of your application
  - [Model-View-Controller \(MVC\) pattern](#) to structure your web application
  - Data-access abstraction via [PHP Data Objects \(PDO\)](#)

# PHP Data Objects (PDO)

- PDO extension is a consistent interface for accessing different databases in PHP.
- PDO extension provides a data-access abstraction layer that enables portability: You can use the same functions to execute queries and fetch data with different databases



- Database-specific drivers are needed to use the abstraction layer
  - PDO Drivers are available for SQLite, MySQL, Oracle, IBM, MS SQL Server, PostgreSQL, others
  - SQLite driver is enabled by default

## References:

- PHP Manual: [PHP Data Objects](#)
- [Why you Should be using PHP's PDO for Database Access](#)

# PHP Data Objects (PDO)



PDO is an object-oriented interface with the following classes:

- **PDO** class represents a connection between PHP and a database server
  - **query, prepare, setAttribute**
- **PDOStatement** class represents a prepared statement, and after the statement is executed, an associated result set
  - **execute, fetch, fetchAll, columnCount**
- **PDOException** class represents an error raised by PDO
- **PDO Drivers** implement the PDO interface

# PDO: Database Connections

Connect to a database by

- Creating a PDO object (an instance of PDO base class)
- `$db = new PDO('sqlite:books.db');`

Close a connection by

- Setting the object to NULL – this destroys all references to the database
- `$db = null`
- If not closed explicitly, PHP will automatically close the connections when your script ends

PHP Manual: [PDO Connections](#)

# PDO: Query Execution

## Prepared vs. dynamic SQL statements

- Prepared statements can be more efficient because they are compiled and optimized once, so subsequent executions should run faster. Prepared statements prevent “SQL injection”

## Prepare and execute

```
$query = "SELECT * FROM BOOK";  
$statement = $db->prepare($query);  
$statement->execute();
```

After a statement is executed, a result set is associated with `$statement`, and is available for fetching the results.

## Dynamic SQL statements

```
$statement = $db->query($query);
```

# PDO: Fetching Query Results

- Prepare and execute the query

```
$query = "SELECT * FROM BOOK WHERE Category='Poetry'";  
$statement = $db->prepare($query);  
$statement->execute();
```

- The result set associated with `$statement` is an array containing the data from the BOOK table
- Fetch the first row from the result set


```
$result = $statement->fetch(PDO::FETCH_ASSOC)
```

- The row (`$result`) is represented as an array where each element is indexed by the column name from the database table.
- Use the fetch style, `PDO::FETCH_ASSOC`, to obtain an array with column name indexing

# PDO: Fetching Query Results

Fetch the first row from the result set:

```
$result = $statement->fetch(PDO::FETCH_ASSOC)
```



|   | ISBN           | Title                | Author         | Publisher           | Format    | Category | Description   | Rating |
|---|----------------|----------------------|----------------|---------------------|-----------|----------|---|--------|
| 1 | 978-1555975845 | Life on Mars         | Tracy K. Smith | Graywolf Press      | Paperback | Poetry   | With allusions to David Bowie and interplanetary... | 3      |
| 2 | 978-1556593109 | The Shadow of Sirius | W.S. Merwin    | Copper Canyon Press | Paperback | Poetry   | "A collection of luminous, often tender poems th... | 5      |
| 3 | 978-1556592010 | Delights & Shadows   | Ted Kooser     | Vintage             | eBook     | Poetry   | As Poet Laureate of the United States, Ted Koose... | 4      |
| 4 | 978-1933527147 | Without Saying       | Richard Howard | Turtle Point Press  | Audio     | Poetry   | In Richard Howard's new collection, voices of my... | 5      |
| 5 | 978-1556592461 | Angle of Yaw         | Ben Lerner     | Copper Canyon Press | Audio     | Poetry   | In his bold second book, Ben Lerner molds philo...  | 3      |

Display contents of `$result`  
with `print_r($result);`

```
Array ( [ISBN] => 978-1555975845
       [Title] => Life on Mars
       [Author] => Tracy K. Smith
       [Publisher] => Graywolf Press
       [Format] => Paperback
       [Category] => Poetry
       [Description] => With allusions to ...
       [Rating] => 3 )
```



# PDO: Fetching All Query Results

```
<?php
$db = new PDO('sqlite:Books.db');
$query = "SELECT * FROM BOOK";
$stmt = $db->prepare($query);
$stmt->execute();
```

```
echo "<strong>All Books in BOOKS database</strong><br>";
while($result = $stmt->fetch(PDO::FETCH_ASSOC)) {
    $title = $result['Title'];
    $author = $result['Author'];
    $category = $result['Category'];
    $publisher = $result['Publisher'];
    echo "<br>" . $title . " by " . $author . ", " . $category .
        ", " . $publisher . "<br>";
}
$stmt->closeCursor();
?>
```

## All Books in BOOKS database

The Swerve: How the World Became Modern by Stephen Greenblatt, History, W. W. N  
George F. Kennan: An American Life by John Lewis Gaddis, Biography, Penguin Press H  
Malcolm X: A Life of Reinvention by Manning Marable, Biography, Penguin Books  
Life on Mars by Tracy K. Smith, Poetry, Graywolf Press  
Tinkers by Paul Harding, Novel, Bellevue Literary Press  
Washington: A Life by Ron Chernow, Biography, Bellevue Literary Press  
The Road by Cormac McCarthy, Literature, Vintage Books  
March by Geraldine Brooks, Literature, Penguin Books

# PDO: Handling Errors

Problem: what if the database connection fails?

Fatal error: Uncaught PDOException: SQLSTATE[HY000]:.... in *filename.php* on line *x*

Solution: use a try/catch construct to intercept the error, display a message, and possibly exit the script.

```
try {  
    // your potentially offending code goes here  
}  
  
catch (PDOException $e) {  
    // display error message, and exit if non-recoverable  
}
```

PHP Manual: [PDO: Errors and error handling](#)

# PDO: Handling Errors

- Ensure that `PDOException` are being raised (thrown)!
- There are 3 different error modes:
  - `PDO::ERRMODE_SILENT` (default)
    - Sets error code, but no display of messages or exceptions
  - `PDO::ERRMODE_WARNING`
    - Sets error code, and displays warning message
  - `PDO::ERRMODE_EXCEPTION` (*recommended!*)
    - Sets error code, throws a `PDOException` with additional error info
    - Recommended during development and testing

## How to set the error mode

```
$db = new PDO('sqlite:Books.db');  
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

# Handling Database Connection Errors

- Create the database connection in **try** clause
- Display an appropriate error message in **catch** clause
- PDOException object has useful methods to obtain error information

```
try {  
    $db = new PDO('sqlite:Books.db');  
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
}  
catch (PDOException $e) {  
    $errorMessage = $e->getMessage(); // Descriptive message for error  
    echo "<p>SQL error: $errorMessage </p>";  
    $db = null;    // "closes" a connection  
    exit();  
}
```

# With and without exception handling

Example: wrong database name: `books.dbb`

With `try...catch` exception handler:

```
SQL error: SQLSTATE[HY000]:
```

```
General error: 1 no such table: BOOK
```

Without `try...catch` exception handler:

```
Fatal error: Uncaught PDOException: ... PDO->prepare('SELECT
* FROM B...') #1 {main} thrown in
C:\xampp\htdocs\2021_INLS672\SQL-PDO\view_all_books.php on
line 9
```

If your application does not catch the exception, the default action by the server is to terminate the script, often with a Fatal Error.

# Practice: Add PDO Exception Handlers

- Modify `view_all_books.php` to query a subset of books, e.g., all History, Biography, or Poetry books.
- Add an exception handler when connecting to database
- Add an exception handler when executing a query
- Include descriptive messages for each exception
- Test the handlers, e.g., specify an invalid database name and an invalid SQL query