6. Database Interaction with PHP

Interacting with databases is a fundamental aspect of web development with PHP. Most web applications rely on databases to store and retrieve data. Therefore, understanding basic SQL and how to connect to and manipulate databases using PHP is crucial for junior developers.

SQL Basics for PHP Developers

SQL (Structured Query Language) is the standard language for managing and manipulating data in relational database management systems. As a PHP developer, you should be familiar with the following fundamental SQL commands 37:

- SELECT: Used to retrieve data from one or more tables based on specified criteria.
- INSERT INTO: Used to add new rows of data into a table.
- UPDATE: Used to modify existing data within a table.
- DELETE FROM: Used to remove rows from a table.
- CREATE TABLE: Used to define the structure of a new table, including column names and data types.
- ALTER TABLE: Used to modify the structure of an existing table (e.g., adding or removing columns).
- DROP TABLE: Used to completely remove a table from the database.

Beyond these basic commands, it's helpful to understand core database concepts such as tables, columns, rows, primary keys (uniquely identify each record in a table), and foreign keys (establish relationships between tables).37

Connecting to MySQL using PHP

PHP provides several extensions for interacting with MySQL databases. Two of the most commonly used are mysqli (MySQL Improved Extension) and PDO (PHP Data Objects).7

- mysqli: This extension is specific to MySQL and offers good performance along with access to MySQL-specific features.7 To establish a connection using mysqli, you typically use the mysqli_connect() function, providing the database hostname, username, password, and database name.34
- PDO (PHP Data Objects): PDO is a more general-purpose database abstraction layer that provides a consistent way to access various database systems, including MySQL, PostgreSQL, SQLite, and others.4 PDO is often recommended for new projects due to its flexibility and enhanced security features, particularly prepared statements for preventing SQL injection.7 To connect using PDO, you create a new PDO object, providing a Data Source Name (DSN) that specifies the database type and connection details, along with the username and password.34

Performing Common Database Operations

Once a connection to the database is established, you can perform various operations:

- Querying Data (SELECT): You can execute SELECT queries using methods like $conn->query() (for mysqli) or $pdo->query() or by preparing a statement with $pdo->prepare() and then executing it with $statement->execute() (for PDO).35 The results can then be fetched using methods such as $result->fetch_assoc() (for mysqli) or $statement->fetch(PDO::FETCH_ASSOC) (for PDO) to retrieve each row as an associative array.34
- Inserting Data (INSERT INTO): Inserting new data involves constructing an INSERT SQL query and executing it using similar methods as querying data.35
- Updating Data (UPDATE): Modifying existing records in the database requires constructing an UPDATE SQL query specifying which table, columns, and rows to update, and then executing the query.35
- Deleting Data (DELETE FROM): Removing records from a table is done by constructing a DELETE SQL query with appropriate conditions and executing it.35

Preventing SQL Injection

SQL injection is a critical security vulnerability that can occur when user-supplied data is directly embedded into SQL queries. To prevent this, it is essential to use prepared statements with parameterized queries, especially when working with PDO.7 Prepared statements allow you to define the structure of your SQL query first, with placeholders for the user-provided values. These values are then bound to the placeholders separately, ensuring that they are treated as data rather than executable SQL code.

Common Interview Questions:

- How do you connect to a MySQL database in PHP?
    - **Solution:** In PHP, you can connect to a MySQL database using either the mysqli extension or PDO.
        - With mysqli, you use the mysqli_connect() function, providing the hostname, username, password, and database name:

          ```
          $conn = mysqli_connect($hostname, $username, $password, $database);
          if (!$conn) {
             die("Connection failed: " . mysqli_connect_error());
          }
          ```

        - With PDO, you create a new PDO object with a Data Source Name (DSN):

          ```
          $dsn = "mysql:host=$hostname;dbname=$database";
          try {
             $pdo = new PDO($dsn, $username, $password);
             $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
          // Error handling
          } catch (PDOException $e) {
             die("Connection failed: " . $e->getMessage());
          ```

~~~
}
``` 5
~~~

● What is PDO in PHP?
* Solution: PDO (PHP Data Objects) is a database abstraction layer that provides a consistent interface for accessing various database systems (MySQL, PostgreSQL, SQLite, etc.). It offers flexibility and enhanced security features, including prepared statements to help prevent SQL injection. 4

● What's the difference between using mysql_ functions and PDO?
* Solution:
* The mysql_ functions are an older, deprecated extension that is specific to MySQL. They are no longer recommended due to security vulnerabilities and lack of support for modern MySQL features.
* PDO is a more modern, general-purpose extension that supports multiple database systems. It offers improved security with prepared statements and a more object-oriented approach. 70

● How can one prevent SQL injections in PHP?
* Solution: SQL injection can be prevented by using prepared statements with parameterized queries. Instead of directly embedding user-supplied data into SQL queries, you use placeholders and bind the values separately. This ensures that the data is treated as data, not as executable code. 7

● What are the steps to create a new database using MySQL and PHP?
* Solution:

1. Connect to the MySQL server: Use mysqli_connect() or PDO to establish a connection. For this operation, you might not specify a database initially.
2. Create the database: Execute a CREATE DATABASE SQL query using $conn->query() (for mysqli) or $pdo->exec() (for PDO).
3. Select the database (if needed): If you need to perform further operations on the new database within the same script, select it using mysqli_select_db() (for mysqli) or by including the database name in the DSN when creating the PDO connection.

Here's an example using PDO:

~~~
```php
try {
$pdo = new PDO("mysql:host=$hostname", $username, $password); // No database in DSN
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
      $databaseName = "my_new_database";
      $pdo->exec("CREATE DATABASE IF NOT EXISTS $databaseName");
      $pdo->exec("USE $databaseName"); // Select the database
      echo "Database '$databaseName' created and selected successfully.";

   } catch (PDOException $e) {
      die("Error creating database: " . $e->getMessage());
   }
~~~

``` 4

● What are the different ways of handling the result set of MySQL in PHP?
* Solution: When querying data from a MySQL database in PHP, you can handle the result set in several ways:
* mysqli:
* mysqli_fetch_assoc(): Fetches a result row as an associative array.
* mysqli_fetch_row(): Fetches a result row as a numeric array.
* mysqli_fetch_array(): Fetches a result row as both an associative and numeric array.
* mysqli_fetch_object(): Fetches a result row as an object.
* PDO:
* $statement->fetch(PDO::FETCH_ASSOC): Fetches the next row as an associative array.
* $statement->fetch(PDO::FETCH_NUM): Fetches the next row as a numerically indexed array.
* $statement->fetch(PDO::FETCH_BOTH): Fetches the next row as both an associative and numerically indexed array (the default).
* $statement->fetch(PDO::FETCH_OBJ): Fetches the next row as an object.
* $statement->fetchAll(): Returns an array containing all of the result set rows.

The choice of which method to use depends on how you want to access the data (by column name or index) and whether you need an object or an array.
```