

All mysql database commands that we use in php

In PHP, you can interact with MySQL databases using a variety of database commands, often facilitated through the MySQLi or PDO (PHP Data Objects) extension. Below is a list of commonly used MySQL database commands and operations in PHP:

1. **Connecting to a MySQL Database:**
 - **mysqli_connect()** or **new PDO()**: Establish a connection to the MySQL database.
2. **Selecting a Database:**
 - **mysqli_select_db()**: Select a specific database to work with.
3. **Executing SQL Queries:**
 - **mysqli_query()** or PDO's prepared statements: Execute SQL queries.
4. **Fetching Data:**
 - **mysqli_fetch_array()**, **mysqli_fetch_assoc()**, **mysqli_fetch_row()**, or PDO fetch methods: Retrieve data from query results.
5. **Inserting Data:**
 - **INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);**: Insert data into a table.
6. **Updating Data:**
 - **UPDATE table_name SET column1 = value1, column2 = value2 WHERE condition;**: Update existing data in a table.
7. **Deleting Data:**
 - **DELETE FROM table_name WHERE condition;**: Delete data from a table.
8. **Prepared Statements (to prevent SQL injection):**
 - **mysqli_prepare()** and **mysqli_stmt_bind_param()** for MySQLi, or PDO prepared statements: Prepare and execute SQL queries with placeholders for data.
9. **Fetching Multiple Rows:**
 - Use loops (e.g., **while** or **foreach**) in combination with fetch functions to retrieve multiple rows from a query.
10. **Transaction Management:**
 - **mysqli_begin_transaction()**, **mysqli_commit()**, **mysqli_rollback()**, or PDO transaction methods: Manage database transactions.
11. **Error Handling:**
 - **mysqli_error()** or PDO error handling: Handle database errors.
12. **Closing the Database Connection:**

- **mysqli_close()** or PDO's connection close method: Close the database connection when you're done.

13. Fetching the Number of Rows:

- **mysqli_num_rows()**: Get the number of rows in a result set.

14. Getting the Last Inserted ID:

- **mysqli_insert_id()**: Retrieve the auto-generated ID from the last **INSERT** query.

15. Escaping Data:

- **mysqli_real_escape_string()** or prepared statements with placeholders: Safely escape data to prevent SQL injection.

16. Fetching Field Metadata:

- **mysqli_fetch_field()**: Retrieve information about fields in a result set.

17. Executing Stored Procedures:

- Use **mysqli_query()** or prepared statements to execute MySQL stored procedures.

18. Executing Custom Functions:

- Call MySQL custom functions using SQL queries.

Here's a basic example of how you might use some of these commands in PHP to connect to a MySQL database, execute a query, and fetch data:

```
<?php

$hostname = "localhost";
$username = "your_username";
$password = "your_password";
$database = "your_database";

// Create a connection
$conn = mysqli_connect($hostname, $username, $password, $database);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Execute a query
$sql = "SELECT * FROM your_table";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Fetch and display data
```

```
while ($row = mysqli_fetch_assoc($result)) {  
    echo "ID: " . $row["id"] . " - Name: " . $row["name"] . "<br>";  
}  
} else {  
    echo "No results found.";  
}  
// Close the connection  
mysqli_close($conn);  
?>
```