MySQL + MySQLi Essentials Guide

1. MySQL Basics

```
CREATE DATABASE mydatabase;
USE mydatabase;
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100),
  age INT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
INSERT INTO users (name, email, age) VALUES ('Tahmid', 'tahmid@gmail.com', 22);
SELECT * FROM users;
UPDATE users SET age = 23 WHERE id = 1;
DELETE FROM users WHERE id = 1;
```

2. PHP Connection (db_connect.php)

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "mydatabase";

$conn = new mysqli($servername, $username, $password, $database);
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

3. Insert Example

```
<?php
include 'db_connect.php';
$name = "Tahmid Alam";
$email = "tahmid@gmail.com";
$age = 23;
$sql = "INSERT INTO users (name, email, age) VALUES ('$name', '$email', '$age')";
if ($conn->query($sql) === TRUE) {
  echo "New record created successfully";
} else {
  echo "Error: " . $conn->error;
}
$conn->close();
?>
```

4. Fetch Example

```
<?php
include 'db_connect.php';
$sql = "SELECT * FROM users";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
  while($row = $result->fetch_assoc()) {
    echo "ID: " . $row["id"] . " | Name: " . $row["name"] . " | Email: " . $row["email"] . "<br>";
  }
} else {
  echo "No results found";
}
$conn->close();
?>
```

5. Prepared Statement Example

```
<?php
include 'db_connect.php';
$stmt = $conn->prepare("INSERT INTO users (name, email, age) VALUES (?, ?, ?)");
$stmt->bind_param("ssi", $name, $email, $age);
$name = "John";
$email = "john@example.com";
$age = 28;
$stmt->execute();
echo "Data inserted safely";
$stmt->close();
$conn->close();
?>
```

6. Error Handling Example

```
<?php
include 'db_connect.php';
if ($conn->connect_errno) {
  die("Failed to connect: " . $conn->connect_error);
}
$sql = "SELECT * FROM invalid_table";
if (!$result = $conn->query($sql)) {
  echo "Query Error: " . $conn->error;
}
$conn->close();
?>
```

7. Best Practices

- Always use prepared statements with user input.
- Always close connections and statements.
- Validate and sanitize user data.
- Use LIMIT when fetching large data.
- Check for errors during database operations.

8. Conventions

```
$stmt → used for prepared statements (secure, modern)
$result → used for result sets (after query or execute)
Use $stmt whenever user input is involved.
```