



DATABASE PROGRAMMING

Instructions

- Complete all tasks below.
- **TOPIC:** CONNECTING MYSQL WITH PROGRAMMING LANGUAGES

QUESTION

1. Establishing a Connection in PHP

- Write a PHP script to connect to a MySQL database named `company_db` using `mysqli` and check if the connection is successful.

2. Performing a Query in PHP

- Extend the PHP script from question 1 to execute a query that retrieves all records from the `employees` table and print the results.

3. Using MySQL with Python

- Write a Python code snippet using the `mysql-connector` library to connect to a MySQL database and fetch all rows from the `products` table.

4. Inserting Data in Python

- Extend the Python code from question 3 to perform an insertion of a new record into the `employees` table.

5. Using MySQL with Java

- Write a Java code example that establishes a connection to a MySQL database using JDBC and retrieves the names of customers from the `customers` table.

6. Handling Exceptions in Java

- Extend your Java code from question 5 to handle SQL exceptions and print an appropriate error message.

7. Connecting MySQL with Node.js

- Write a Node.js code snippet that connects to a MySQL database using the `mysql` package and queries the `orders` table.

8. Asynchronous Queries in Node.js

- Modify your Node.js example from question 7 to perform an asynchronous query to fetch all products while handling potential errors.

9. Using MySQL with C#

- Write a C# code example that connects to a MySQL database using `MySql.Data` and retrieves a list of categories from the `categories` table.

10. Parameterized Queries in C#

- Extend the C# code from question 9 to execute a parameterized query that fetches products based on a specific category ID.

11. Secure Connections

- Explain how to securely connect to a MySQL database from any programming language (e.g., using SSL) and provide an example snippet.

12. Environment Variables for Credentials

- Discuss how to use environment variables in your programming language of choice to store database credentials securely.

13. ORM Usage

- Explain what an Object-Relational Mapping (ORM) tool is and provide an example of connecting to MySQL using an ORM in any programming language (e.g., Sequelize in Node.js or SQLAlchemy in Python).

14. Connecting MySQL with Ruby

- Write a Ruby code example using the `mysql2` gem to connect to a MySQL database and query customer details from the `customers` table.

15. Closing Connections

- Write a code snippet for any programming language showing the proper way to close the database connection after performing operations.

END