#include <mysql/mysql.h>

初始化连接 → 建立连接 → 执行 SQL → 处理结果 → 释放资源

**MYSQL\* mysql\_init(MYSQL\* mysql)**

‌功能‌：初始化 MySQL 连接对象 ‌12

‌参数‌：传入 nullptr 创建新对象

‌返回值‌：成功返回 MYSQL\* 指针，失败返回 nullptr

**MYSQL\* mysql\_real\_connect(…)**

功能：建立与mysql服务器的连接

返回值：成功返回MYSQL\*指针，和mysql\_init的返回值一样，失败返回nullptr

**int mysql\_query(MYSQL \*mysql, const char \*query)‌**

‌功能‌：执行 SQL 查询语句 ‌

‌返回值‌：成功返回 0，失败返回非 0 ‌

**MYSQL\_RES \*mysql\_store\_result(MYSQL \*mysql)‌**

‌功能‌：获取 SELECT 查询结果集 ‌

‌返回值‌：成功返回 MYSQL\_RES\* 指针，是一个指向结果集的指针，失败返回 nullptr

‌配套函数‌：

MYSQL\_ROW mysql\_fetch\_row(MYSQL\_RES \*result)：逐行读取数据 ‌

int mysql\_num\_fields(MYSQL\_RES \*result)：获取列数 ‌

void mysql\_free\_result(MYSQL\_RES \*result)：释放结果集 ‌

与 mysql\_use\_result 的区别‌，二者使用方法一样。

mysql\_store\_result：一次性加载所有数据到内存，适合结果集较小的情况。

mysql\_use\_result：逐行从服务器读取数据，适合结果集较大且需流式处理的情况。

**mysql\_fetch\_row 函数‌**

**‌功能‌**

从结果集中逐行提取数据，每次调用返回下一行的数据。

**‌函数原型‌**

**MYSQL\_ROW mysql\_fetch\_row(MYSQL\_RES \*result);**

**‌返回值‌**

返回 MYSQL\_ROW 类型（即 char\*\*），表示当前行的列数据。

每列数据以字符串形式返回（即使数据库中存储的是数值）。

示例：row 表示第一列的值（如 "1" 对应 id 字段）。

返回 nullptr，表示结果集已遍历完毕。

**‌配套函数‌**

**mysql\_num\_fields(result)**‌

获取结果集的列数，用于遍历所有字段。

‌**获取错误信息**‌：使用 mysql\_error(conn) 输出错误详情

解决编译时找不到动态库的问题

[root@laluo ~]# g++ main.cpp -o app -lmysqlclient -std=c++11会报错，因为

/usr/lib64/mysql //动态库所在路径

/usr/lib //编译器搜索的路径

所以才会找不到动态库

创建软链接到 /usr/lib 如下：

[root@laluo mysql]# ln -s /usr/lib64/mysql/libmysqlclient.so /usr/lib/libmysqlclient.so



