数据库：

历史:

1. excle 二维表模型；
2. access 基础数据库（可视化）
3. SQlServer 标准化数据库（sql）
4. MySQL 可视化/非可视化
5. Oracle 实现大数据操作吗、

Mysql 数据库

01 驱动com.mysql.jdbc.Driver

02 数据库连接 connection

03 数据库的基本操作

MySQL

登陆：

Mysql -u root -p

Password:

02 查看已有数据库

Show databases;

03 创建数据库

Create database//无s；

Database\_mame;

选择操作数据库

Use datebase\_name;

查看已有数据表

Show tables；

Create table table\_name

(

Name varchar(20),

Sex varchar(30),

Age int )

);

查看结构

desc table name

show full fields from tb\_name

数据操作

插入数据：

全局插入：

Insert into table name

Values(“jine”,”joy”,18);

局部插入：

Insert into table -name

(name,age)value(“lili”,12);

多条记录插入：

Insert into tablename values

(),(),(),(),();

查询数据：

Select \*from table\_name

举例：

（只单独查询name）

Select name from table\_name;

删除数据

Delete from table\_name where

Field\_name=value;

//where表示参考性删除

修改数据

Update table\_name set

Filed\_name=value where

field \_name=value;//参考位置

查看字符集

Show character set;

查看数据库编码

Show variables like “%char%”;

查看数据表编码

Show create table table\_name;

数据库修改字符集

alter database database\_name default character set ‘utf8’;

表修改字符集

Alter database database\_name default character set ‘utf8’;

修改属性

alter table table\_name

modify number varchar(18)

primary key not null;

添加某一选项

alter table table\_name add number varchar(18) primary key not null first;

删除数据库用

Drop database database\_name;

删除表中的某一个属性

Alter table table\_name drop value\_name;

联合查询

select \*from

userinfo join student\_information

on userinfo.number=student\_information.number””;

用户管理

User 可连接数据库服务器的用户

Db 设定用户可访问哪些数据库

Host 设定用户可以从哪些主机连接数据库，不同主机可设定不同的权限

Tables\_priv 设定用户可以访问哪些表

Columnts\_priv 设定用户可访问哪些字段

难点数据库的链接

代码

#Include<stdio.h>

#include<stdlib.h>

#include<winsock2.h>

#include<c:/program File/MySQL Server 8.0/include/mysql.h>

#define HOST “localhost”

#define USERNAME “root”;

#define PASSWORD “123456”

#define DATABASE “test”

Int main()

{

MYSQL my\_connection;

Int ser;

Mysql\_init(&my\_connection);

If(mysql\_real\_connect(&my\_connction,HOST,USERNAME,PASSWORD,DATABASE,0,NULL,CLIENT\_FOUND\_ROWS)){

Printf(“连接成功”)；

Mysql\_query(&my\_connection,”set name utf8”);

}

}

第二种链接代码

#include<stdio.h>

#include<stdlib.h>

#include<winsock2.h> //必须加这个文件

#include<mysql包含文件路径>

Int main()

{

MYSQL mysql; //一个数据库结构体

MYSQL\_RES\* res;//一个结果集结构体

MYSQL\_ROW row; //一个char\*\*二位数组

Mysql\_init(&mysql);//初始化数据库

Mysql\_options(&mysql,MYSQL\_SET\_CHARSET\_NAME,”gbk”); //字符编码设置

Mysql\_real\_connect（&mysql,”localhost”,”root”,”toor”,”student\_information”,）3306,NULL,CLINET\_FOUND\_ROW）; //数据库的链接

Mysql\_query(&mysql,”select\*from birthday\_date”); //数据查询

//获取数据集

Res = mysql\_store\_result(&mysql);

}

Vs2013链接需要将：

项目属性

Vc/c++目录改变

包含文件添加include文件

库文件添加lib文件

链接器输入里边的附加依赖项添加libmysql.lib文件名

记住编译环境要在64位系统下

编译后会出现debug文件将libmysql.dll文件添加进去；

数据库高级查询方法

Select sum(tmony) as money,store from sales\_infor group by store order by money desc

解释：查询表sales\_infor中的 store 的 tmoney 的和并按照store 分组然后根据tmony从大到小的排序不写desc就可以实现从小到大的排序

Select money as smoney from (select date(ctime) as time ,tmoney as money from sales\_infor group by time) temp order by mony desc;

解释先对ctime时间进行date（）函数的处理然后通过date（ctime）分组然后进行根据取出来的值money进行排序处理

数据库各项键的建立

主键：primary key

创建时

Create table tb\_name(

Name varchar(12) not null primary key);

追加添加primary key

Alter table table\_name modify name var char(12) not null primary key;

删除主键

Alter table table\_name drop primary key;

复合主键

有多个字段创建主键称为复合主键

自增：

自增键只能是整型必须是一个索引

必须是整型数字

通常与主键连用

Crate table table\_name(

Id int primary key auto\_increment

);

添加自增

Alter table table\_name modify id int auto\_increment;

唯一键

创建表时

Create table table\_name(

Name varchar(12) not null unique key;

Name\_2 varchar(12) not null,

Unique key (Name\_2)

);

Alter 修改

Alter table table\_name modify column\_name varchar(5) unique key;

Alter table table\_name add constraint num\_uk unique key (name);

删除唯一键

alter table my\_unique drop index num\_uk;

触发器

关键词：trigger

分类：块级/行级

Mysql 备注触发器：pl/sql

Delimiter 符号

Create trigger 名称

After /before

Insert /update/delete

On 表名

For each row

Begin

Sql 语句；

End；

符号

注意：

本表触发必须添加时间延迟（包）

Insert new.name

Update new.name,old.name

Delete old.name

视图

view

create view test\_view (学号,姓名,月,日,星期,签到时间,签到类型,实际时间)

as

select num\_number as "学号",

student\_name as "姓名",

month as "月份" ,days as "号",week as "星期",

statu as "签到时间",

type as "签到状态" , syste\_dat as "记录时间" from student\_infor,

stutime,

stutype ,main\_table where main\_table.student\_id=student\_infor.stunum\_id and

main\_table.statime\_id=stutime.time\_id and

main\_table.type\_id=stutype.type\_id order by student\_infor.stunum\_id;

通过视图的DML操作（）

select

insert

可以通过视图插入数据，但是只能基于一个基础表进行插入不能跨表更新数据

修改视图

create or replace view view\_name as

创建索引

create index index\_name on table\_name

(col\_name[,....n])

create index test1 on student\_infor (stunum\_id,num\_number,student\_name);