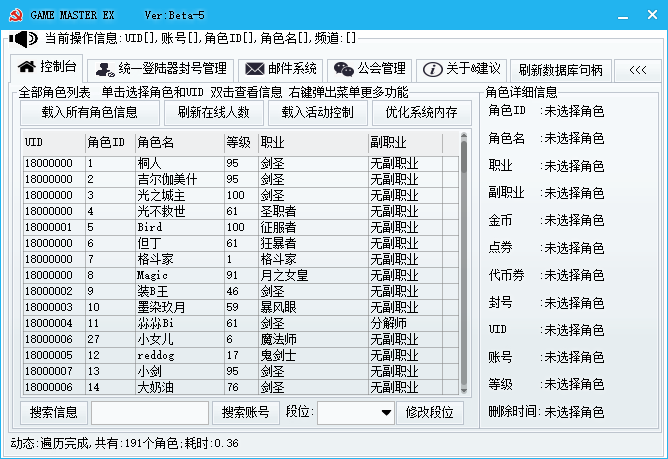
# GM工具的各种操作在数据库中对应的位置

## 以及一些简单的易语言应用实例



下面先说一些位置

最后我会解析易语言操作的代码以及SQL的一些语法

如有疑问可联系本人QQ1048893645 交流群327052207

转载请注明出处

贴吧 台服DNF吧：<https://tieba.baidu.com/f?kw=%E5%8F%B0%E6%9C%8Ddnf&fr=ala0&tpl=5>

论坛 能源图书馆：<http://lukebook.cc>

1. 角色信息

Taiwan\_cain数据库中的charac\_info表

简单解析:（一些没用的东西我就不说了，诸君自行摸索即可）

m\_id=账号ID，accounts里面的UID

Charac\_no=角色ID

Charac\_name=角色名

Job=职业类型；例如 鬼剑士 格斗家 神枪手

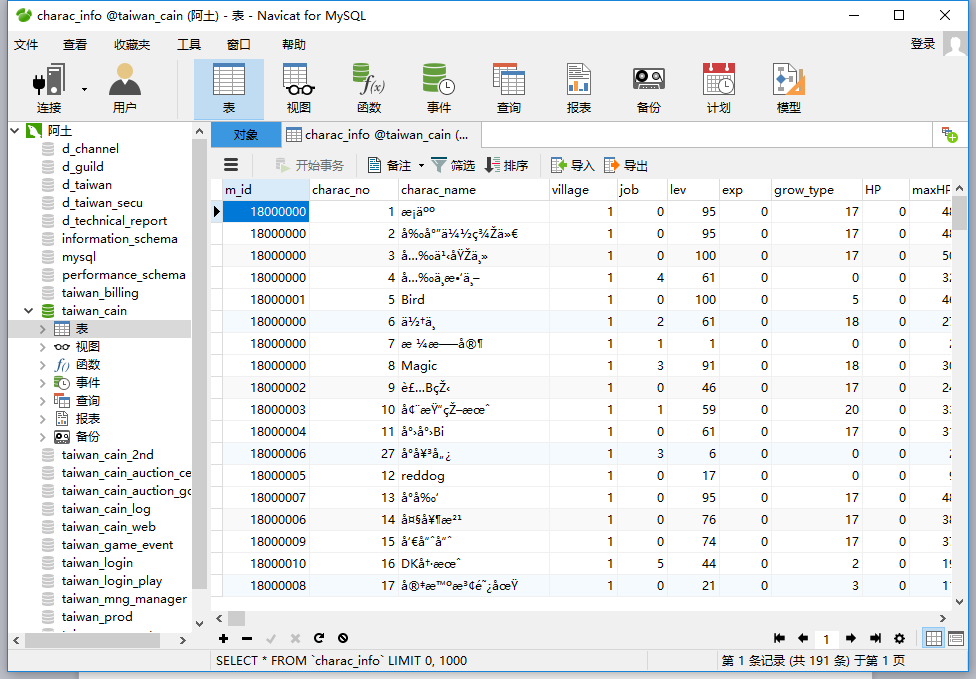
Lev=角色等级

Grow\_type=职业分支；例如 鬼剑士的鬼泣 剑魂 狱血魔神 大暗黑天等

Delete\_time和delete\_flag分别是角色删除的时间和删除的状态 flag的数值0未删除1删除

Guild\_id=所在工会ID

Expert\_job=副职业



1. 账号列表

D\_taiwan数据库中的accounts表

简单解析：

UID=账号的ID

AccountName=账号

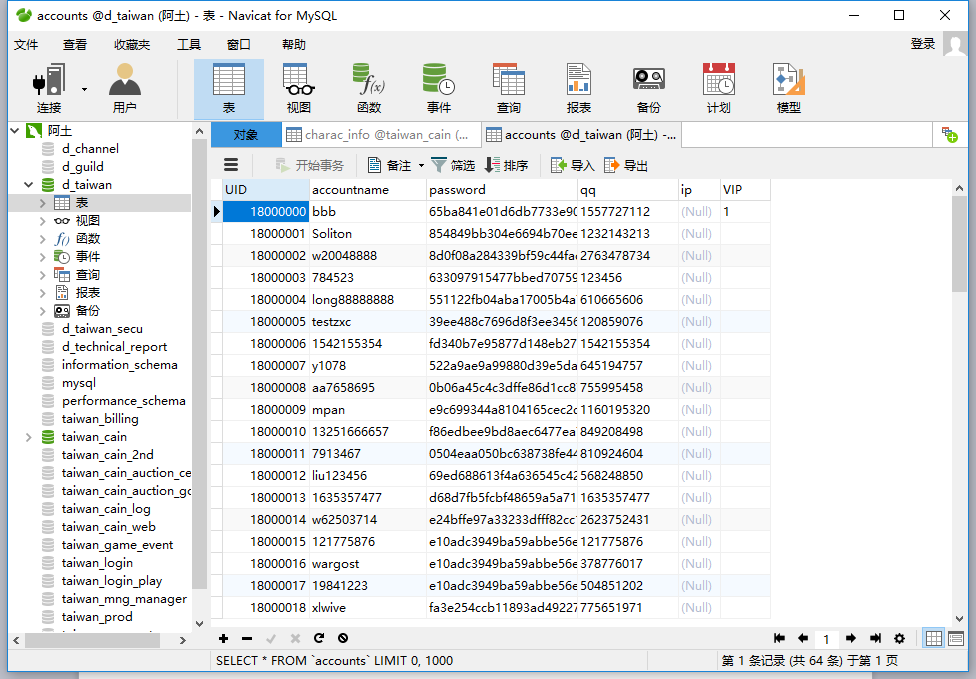
Password=密码

后面三个都是各种一键端自己创建的，可以用于二级密码或登录IP等验证

登陆器是通过账号和加密后的密码在数据库中寻找对应的UID

然后通过几个参数来加密UID获得一个参数 这个参数大小一般为300字节以上

直接传递给dnf.exe



1. 在线账号的表

Taiwan\_login数据库中的login\_account\_3表

简单解析：

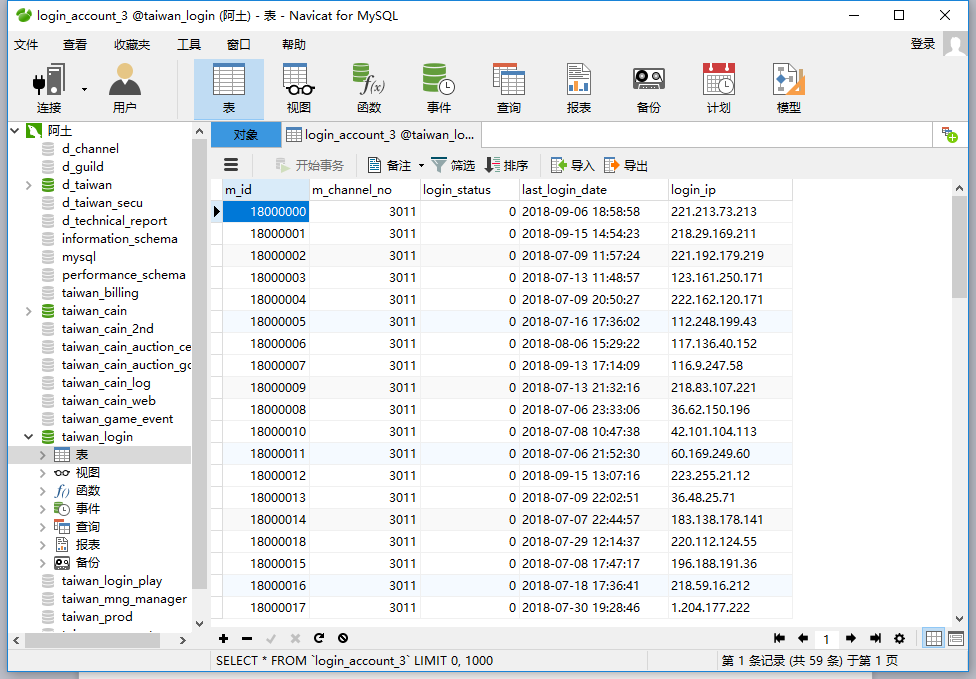
m\_id=账号ID 也叫UID UserID

M\_channel\_no=所在频道 这里是3011 用数值减去3000即可 56频道就是3056

Login\_status=在线状态；0不在线1在线

Last\_login\_date=登录的时间

Login\_ip=登录时候的IP地址



1. 活动信息

d\_taiwan数据库中的dnf\_event\_info表

默认编码为Latin1 在操作的时候需要先转为utf-8编码再读取出来转为GB2312编码

这个表是记录的所有活动信息

Event\_id=和名字一样 活动ID

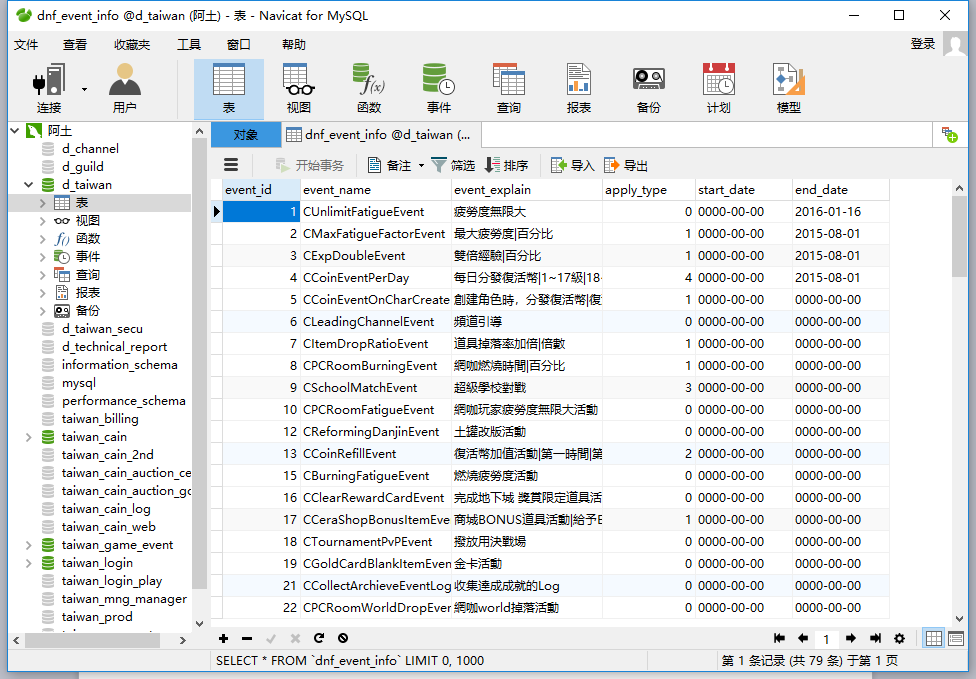
Event\_name=英文的活动名称

Event\_explain=中文的活动名称

Apply\_type=应该是活动类型

后面的start\_date和end\_date虽说应该是时间 但是并没有什么用

可能在一些残缺的活动中需要这两个参数



1. 活动控制

d\_taiwan数据库中的dnf\_event\_log表

这里是打开活动的一些参数

简单解析为log\_id应该是记录活动的序号

Event\_type 是打卡活动的ID

Parameter1和parameter2是活动的两个参数

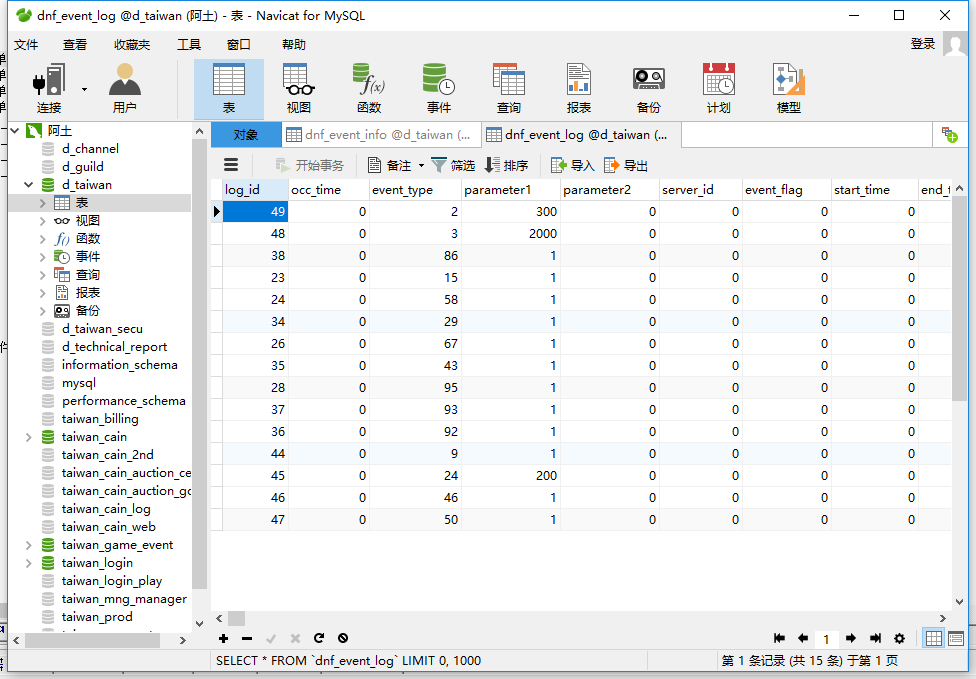
Serverid 应该是服务器的ID 但是因为现在私服都是只有一个希洛克的库

所以为0应该是所有服务器

当然这个不在我们写工具的考虑范围之内

下面还有两个star\_time和end\_time 应该就是和名字一样，开始时间和结束时间了

更多控制参数欢迎一起交流



1. 工会管理

有两个表，一个是工会的信息，一个是工会成员的信息

6.1工会信息

D\_guild数据库中的guild\_info

Guild\_id=工会ID

Guild\_name=工会名

Master\_id=会长的UID

Master\_no=会长的角色ID

Master\_name=会长的角色名

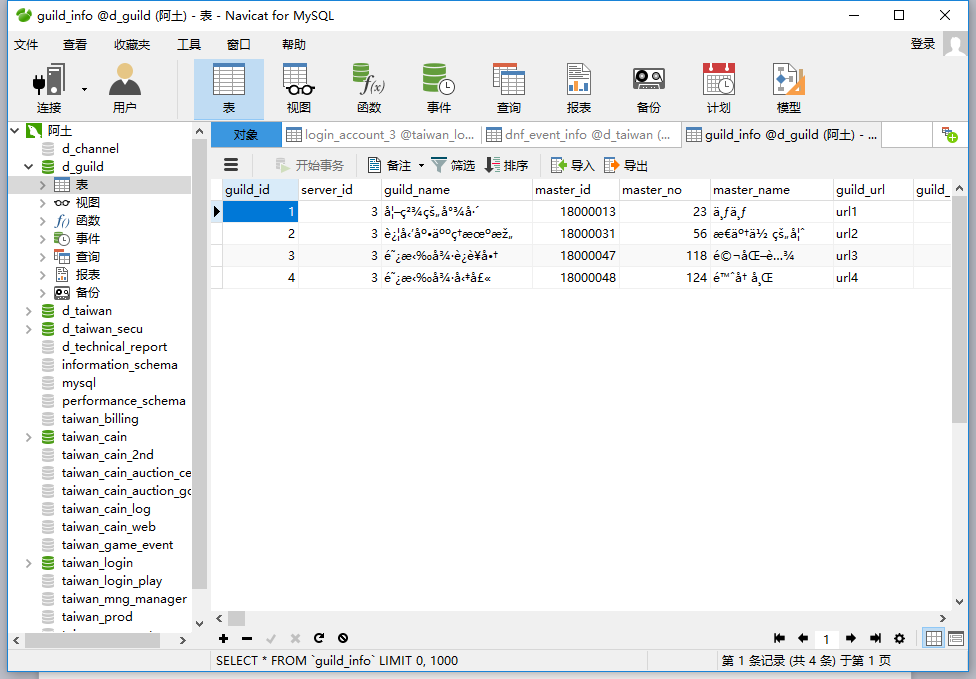
Crate\_time=创建时间

Lev=工会等级

Member\_count=工会人数

Guild\_exp=工会经验

Guild\_fund=工会基金



6.2工会成员信息

D\_guild数据库中的guild\_member表

Guild\_id=所在工会ID

M\_id=该角色的账号ID UID

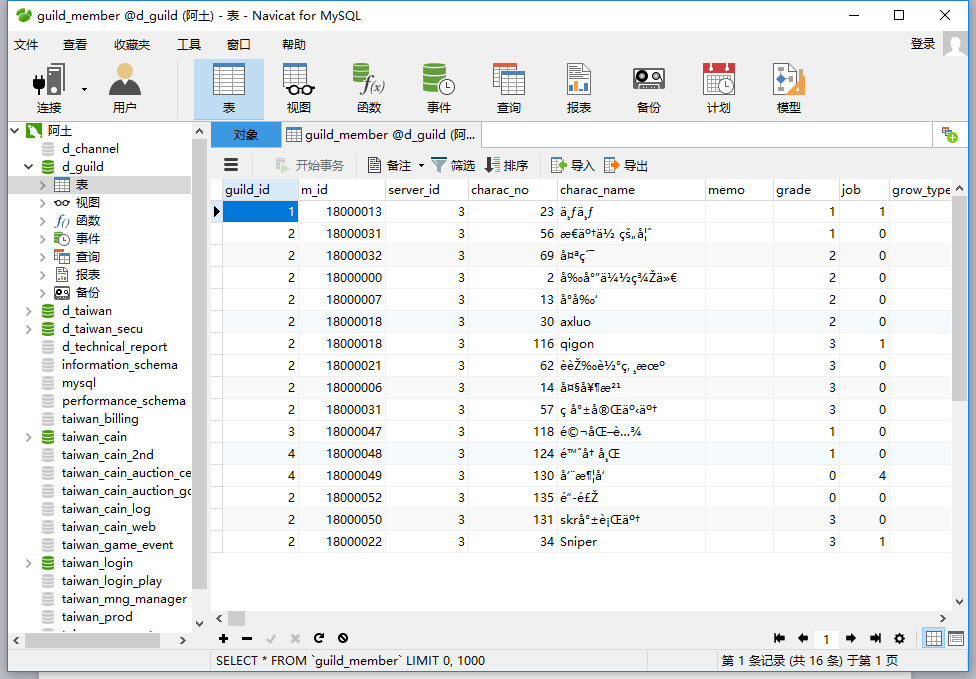
Charac\_no=角色ID

Charac \_name=角色名

Memo=留言板

Grade=角色在工会中的职位

job和grow\_type还有lev分别是职业、职业分支、等级 同上面的角色信息一样

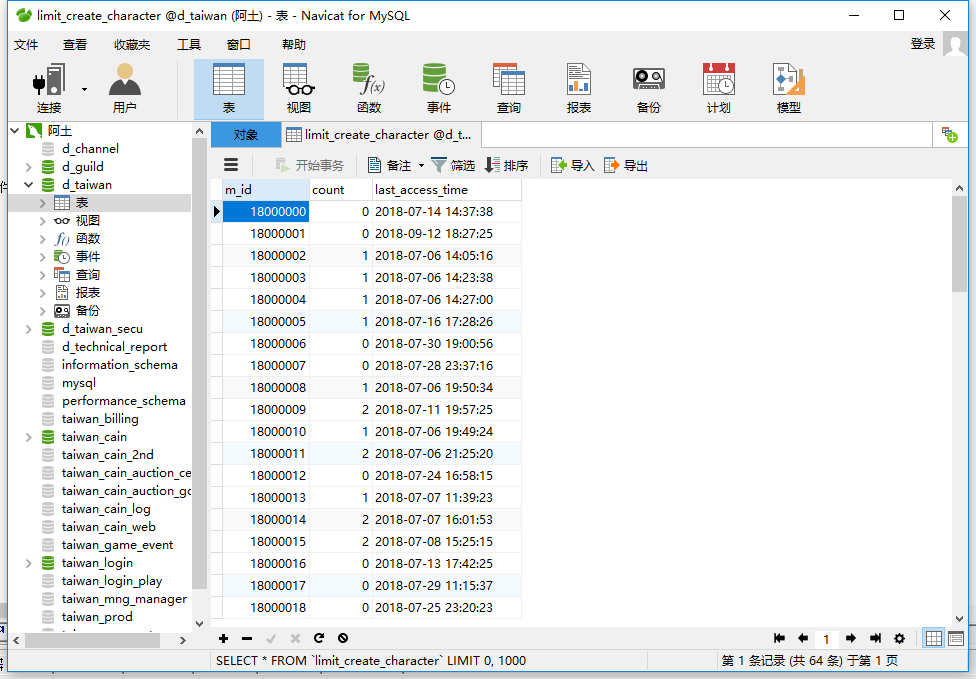


1. 重置创建角色每天2个的限制

d\_taiwan数据库中的limit\_create\_character

m\_id=账号ID UID

Count=创建数量（GM操作中把这个字段写0即可）



1. 开启账号所有副本难度（格蓝迪以下，一些活动图和自制副本除外）

Taiwan\_cain数据库中的member\_dungeon表

m\_id=账号ID UID

Dungeon=副本数据 我的理解为 x|x 为一组 或者说 一个副本

例如 1|0 就是副本ID1中拥有普通级权限

1|1=冒险级

1|2=王者

1|3=地狱级

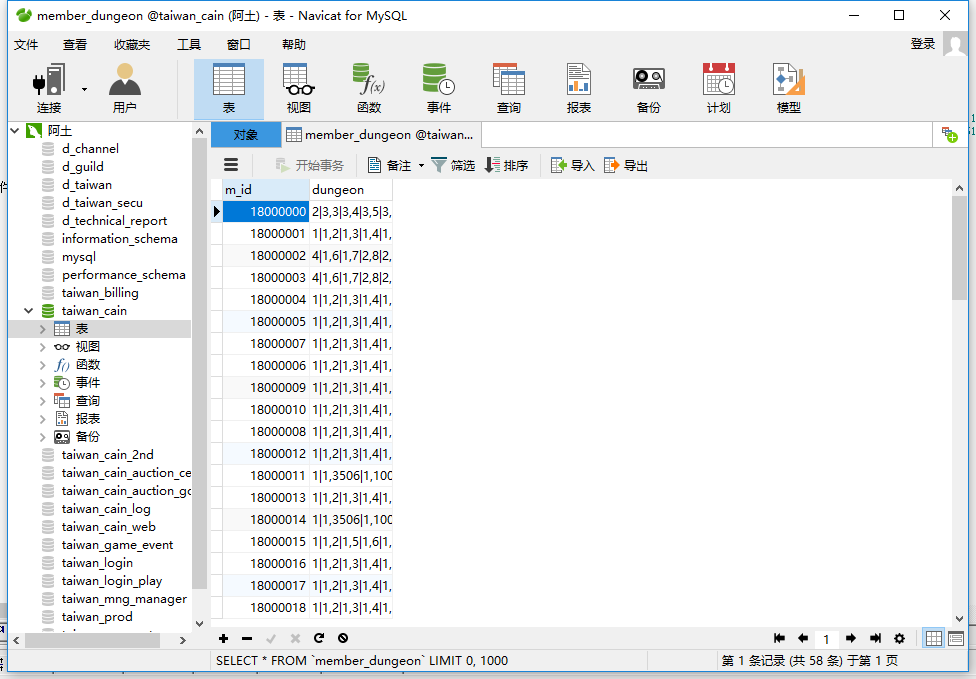
和pvf内的难度数据应该是相关联的

然后用英文半角逗号（,）分割每组数据

我在这里提供一段格蓝迪以下普通副本的难度数据

诸位看得懂的自行写入即可

2|3,3|3,4|3,5|3,6|3,7|3,8|3,9|3,11|3,12|3,13|3,14|3,15|3,17|3,21|3,22|3,23|3,24|3,25|3,26|3,27|3,31|3,32|3,33|3,34|3,35|3,36|3,37|3,40|3,42|3,43|3,44|3,45|3,50|3,51|3,52|3,53|3,60|3,61|3,65|2,66|1,67|2,70|3,71|3,72|3,73|3,74|3,75|3,76|3,77|3,80|3,81|3,82|3,83|3,84|3,85|3,86|3,87|3,88|3,89|3,90|3,91|2,92|3,93|3,100|3,101|3,102|3,103|3,104|3,110|3,111|3,112|3,140|3,141|3,502|3,511|3,521|3,1000|3,1500|3,1501|3,1502|3,1504|1,1506|3,3506|3,10000|3



1. 封号系统（进入频道提示已被停权）

D\_taiwan数据库中member\_punish\_info表

m\_id=账号ID UID

Punish\_type=封号类型或状态？这个我没有深入研究 总之写1封了就完了

Punish\_value同上

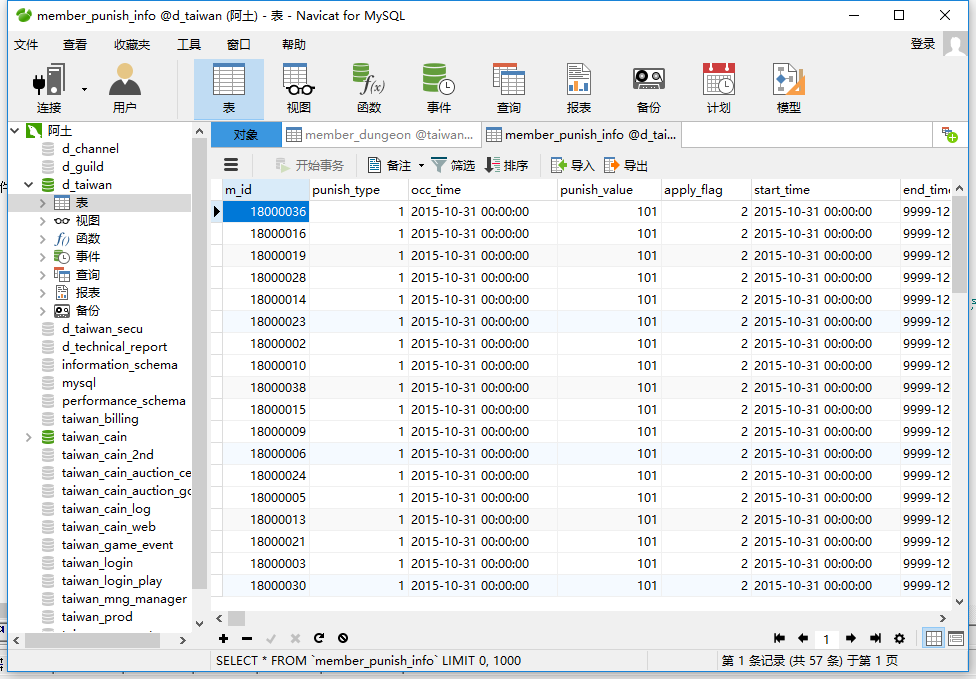
Apply\_flag=可能是操作的管理人员ID

Start\_time=封停时间

End\_time=解封时间

时间并没有深入的去研究 总之今天封掉 然后9999解封就是永久封号没错了

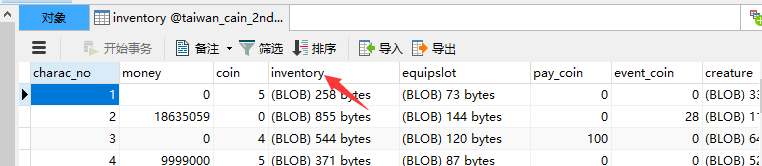
解封的话直接用m\_id作为参数删除这一条数据就是了



1. 清空数据；背包 宠物 装扮 邮件 任务

10.1背包数据

taiwan\_cain\_2nd数据库inventory表



inventory这个字段写空就行了

\* “update taiwan\_cain\_2nd.inventory set inventory = '' where charac\_no = '角色ID'”

下面的我直接提供操作语句

10.2宠物数据

\* “delete from taiwan\_cain\_2nd.creature\_items where charac\_no=’角色ID’”

10.3装扮数据

\* “delete from taiwan\_cain\_2nd.user\_items where charac\_no='角色ID’”

10.4邮件数据

\* “delete from taiwan\_cain\_2nd.postal where receive\_charac\_no=’角色ID’”

\* “delete from taiwan\_cain\_2nd.letter where charac\_no=’角色ID’”

10.5任务数据

我这里提供一段黑岩以前某版本客户端的数据

其余客户端的版本请根据表的位置自行提取

下面我也会说任务和一些字节集数据的提取方法

“Update taiwan\_cain.new\_charac\_quest set clear\_quest=quest\_notify=0xE1100000000000000000000000000000000000000000000000000000000000000000000000000000 where charac\_no='角色ID”

”

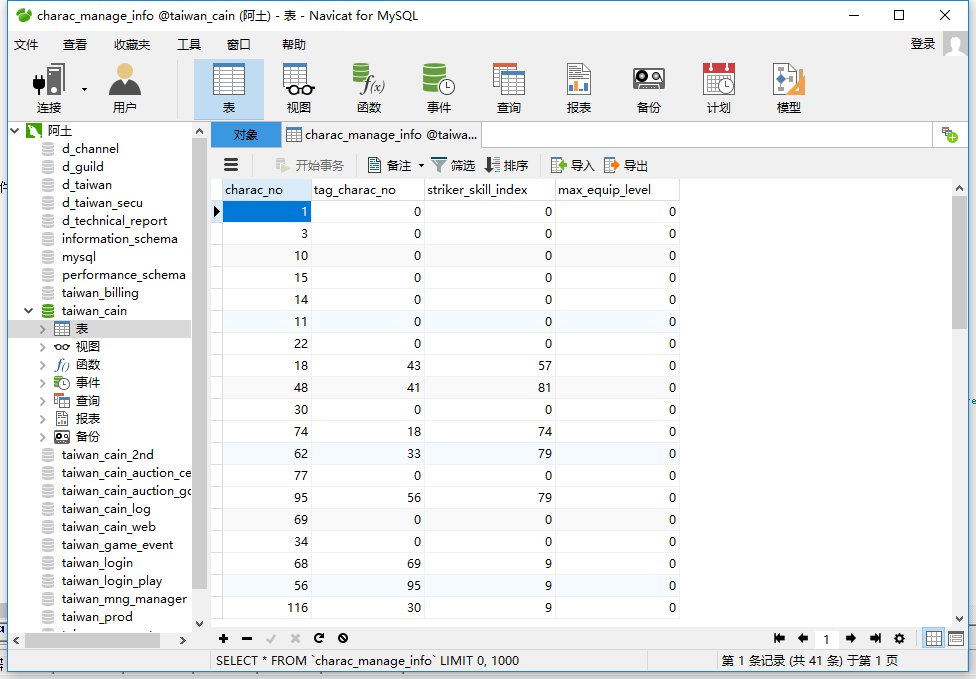
“update taiwan\_cain.charac\_titlebook set specific\_section=0xA82F0000789CEDD3370A02611445E179D889B1356741B1300710145130A0829B70FF2B1057207AC0E1F1DFAF98EE1433EF4E14C5C60ACFF7F3C7B848E21289CB244E90B802E2F85895BC738DC475123748DC74792AA72C494ED522719BC41D8DC4094B913BA7499C217196C43912E7B5EDAF58977CED1E89FB719DCA061AC9FFD8908C6444E23189271A89133625779E91784EE2058997245E69DB4ED85AA70A816DC8FFBC25F14E0B0B81EDC9480E243E6A6121B01319C999C4172D2C04762523B96924F281DDC9C21E5A98888888888888888878F202ADDA1D05,general\_section=0xAE100000789CEDD24B0E82401084E12EC35DF58C82BA16E516826F451045E3EB0EFE09643253FB6FD1D5655A0DCD4CF65794135C10BC267843F016601FA31D697B4FF081E023C12782CF0497045F9CDCB62A72734DF095E086E01BC17727FFDC5FD492B61F043FFB7A955E6124DD456F32920FC15F826D04B0081E101C01EC68342685C50427044F089E123CF36F24289A93B6538217042F09CEC2483ACB0FFC373322,despair=0x981C0000789CEDC13101000000C2A0F54F6D0A3FA000000000000000BE061C980001,event=0x4C0E0000789CEDC13101000000C2A0F54F6D067FA0000000E0330E4C0001 where charac\_no='角色ID'”

1. 角色的装备穿戴等级上限

Taiwan\_cain数据库的charac\_manage\_info表

最后一项

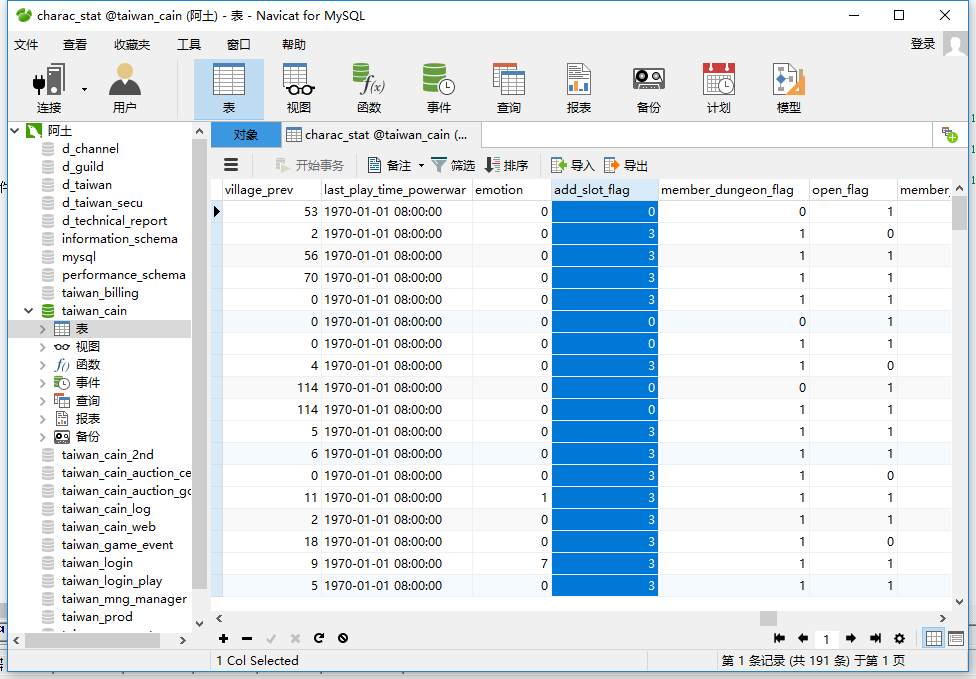
Max\_equip\_level=装备穿戴等级上限



1. 不动任务单开启角色左右槽

Taiwan\_cain数据库中的charac\_stat表

Add\_slot\_flag=3即可



1. 副职业一键满级

这个功能需要先判断一下角色是否有副职业，没有副职业强改数据我没有试过后果

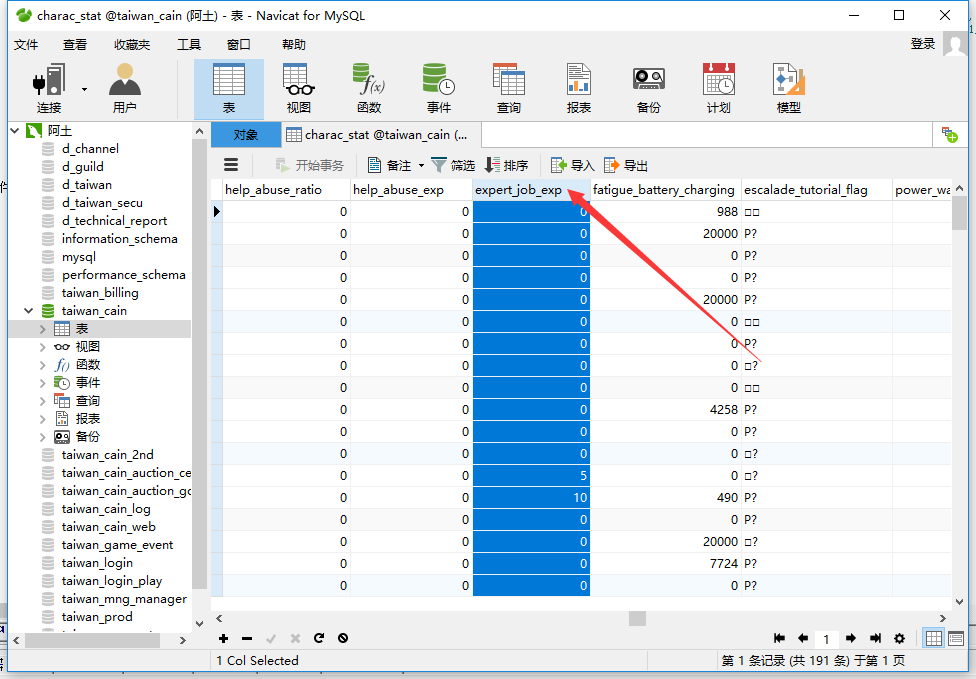
而副职业的信息在角色信息的那个表里面，本文档第一节就是

和上面那个功能【左右槽】在同一个表里面

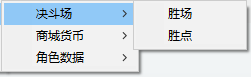
只是字段不一样

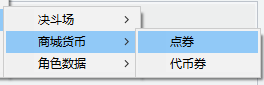
副职业经验值的字段是

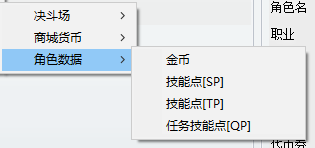
Expert\_job\_exp=经验值 11级写2054即可



1. 可以充值或修改的一些角色数据







这几个东西，我直接在下一页把易语言的代码贴出来好了，诸位自行去转换

充值系统源代码

.版本 2

.支持库 mysql

.子程序 账号\_充值系统, 文本型

.参数 执行参数, 文本型, , 角色charac\_no或账号UID

.参数 数值, 文本型, , 需要充值或修改的数量

.参数 充值类型, 整数型, , 0点券 1代币券 2金币 3SP 4TP 5QP 6决斗胜场 7决斗胜点

.参数 执行方式, 整数型, 可空, 默认为0修改 压入1为充值

.局部变量 记录集, 整数型

.局部变量 取出数据, 文本型

.局部变量 计算后的数据, 文本型

.如果真 (是否为空 (执行方式))

执行方式 ＝ 0

.如果真结束

.判断开始 (执行方式 ＝ 0)

.如果真 (充值类型 ＝ 0) ' 点券

.如果真 (取反 (执行SQL语句 (SQL句柄, “update taiwan\_billing.cash\_cera set cera='” ＋ 数值 ＋ “' where account='” ＋ 执行参数 ＋ “'”)))

返回 (“ExecuteFail!ErrorCode:0”)

.如果真结束

返回 (“ExecuteSuccess!”)

.如果真结束

.如果真 (充值类型 ＝ 1) ' 代币券

.如果真 (取反 (执行SQL语句 (SQL句柄, “update taiwan\_billing.cash\_cera\_point set cera\_point='” ＋ 数值 ＋ “' where account='” ＋ 执行参数 ＋ “'”)))

返回 (“ExecuteFail!ErrorCode:0”)

.如果真结束

返回 (“ExecuteSuccess!”)

.如果真结束

.如果真 (充值类型 ＝ 2) ' 金币

.如果真 (取反 (执行SQL语句 (SQL句柄, “update taiwan\_cain\_2nd.inventory set money='” ＋ 数值 ＋ “' where charac\_no='” ＋ 执行参数 ＋ “'”)))

返回 (“ExecuteFail!ErrorCode:0”)

.如果真结束

返回 (“ExecuteSuccess!”)

.如果真结束

.如果真 (充值类型 ＝ 3) ' SP

.如果真 (取反 (执行SQL语句 (SQL句柄, “update taiwan\_cain\_2nd.skill set remain\_sp='” ＋ 数值 ＋ “' where charac\_no='” ＋ 执行参数 ＋ “'”)))

返回 (“ExecuteFail!ErrorCode:0”)

.如果真结束

返回 (“ExecuteSuccess!”)

.如果真结束

.如果真 (充值类型 ＝ 4) ' TP

.如果真 (取反 (执行SQL语句 (SQL句柄, “update taiwan\_cain\_2nd.skill set remain\_sfp\_2nd='” ＋ 数值 ＋ “',remain\_sfp\_1st='” ＋ 数值 ＋ “' where charac\_no='” ＋ 执行参数 ＋ “'”)))

返回 (“ExecuteFail!ErrorCode:0”)

.如果真结束

返回 (“ExecuteSuccess!”)

.如果真结束

.如果真 (充值类型 ＝ 5) ' QP

.如果真 (取反 (执行SQL语句 (SQL句柄, “update taiwan\_cain.charac\_quest\_shop set qp='” ＋ 数值 ＋ “' where charac\_no='” ＋ 执行参数 ＋ “'”)))

返回 (“ExecuteFail!ErrorCode:0”)

.如果真结束

返回 (“ExecuteSuccess!”)

.如果真结束

.如果真 (充值类型 ＝ 6) ' 胜场

.如果真 (取反 (执行SQL语句 (SQL句柄, “update taiwan\_cain.pvp\_result set win='” ＋ 数值 ＋ “' where charac\_no='” ＋ 执行参数 ＋ “'”)))

返回 (“ExecuteFail!ErrorCode:0”)

.如果真结束

返回 (“ExecuteSuccess!”)

.如果真结束

.如果真 (充值类型 ＝ 7) ' 胜点

.如果真 (取反 (执行SQL语句 (SQL句柄, “update taiwan\_cain.pvp\_result set pvp\_point='” ＋ 数值 ＋ “',win\_point='” ＋ 数值 ＋ “' where charac\_no='” ＋ 执行参数 ＋ “'”)))

返回 (“ExecuteFail!ErrorCode:0”)

.如果真结束

返回 (“ExecuteSuccess!”)

1. 最后说一下小白们最关心的邮件

邮件表的位置在数据库taiwan\_cain\_2nd中

又分为四个表；letter、letter\_del、post、post\_del

但是一般我们写工具来说，只需要操作两个表即可 那就是letter和post

先说letter

15.1

位置：taiwan\_cain\_2nd.letter

字段解析：

Letter\_id=这一封邮件的编号，类型为自动递增的一个整数

我们写发送邮件的时候 这个参数可以省略

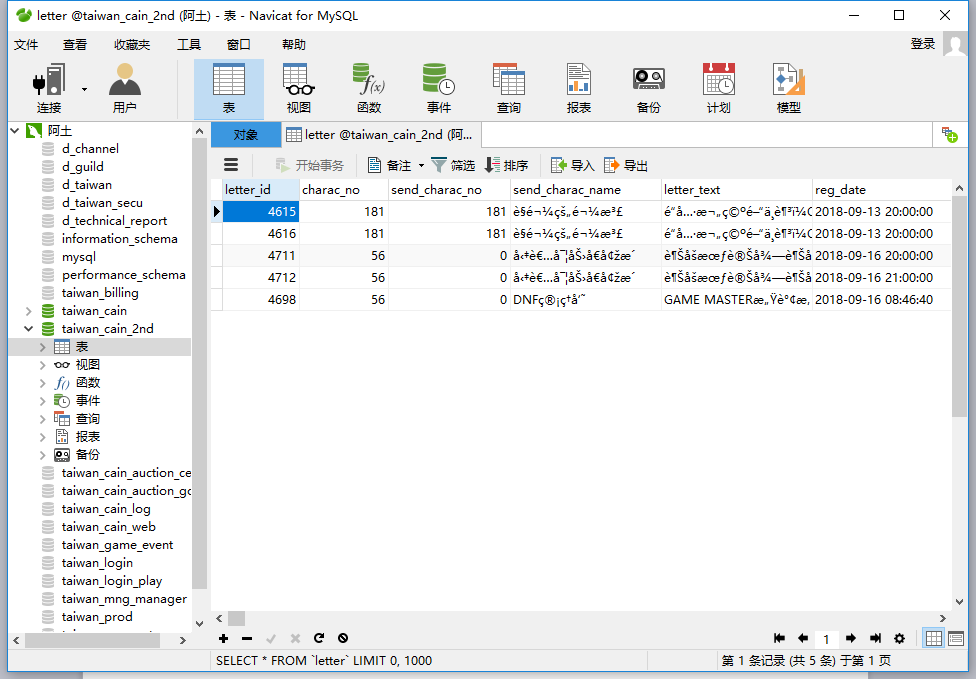
Charac\_no=收件人的角色ID

Send\_charac\_no发件人的角色ID 这个如果用工具写的话就随意

send\_charac\_name=发件人的名字

Letter\_text=邮件的文本内容

reg\_date=发送邮件的时间



15.2 post表

位置：taiwan\_cain\_2nd.post

字段解析：

Postal\_id=邮件的ID 和letter\_id一样自动递增

Occ\_time=发送时间

Send\_charac\_no=发件人ID

Send\_charac\_name=发件人名字

发件人的ID和名字最好和letter的同步，没有同步大概也就是不会显示，具体后果怎么样我忘记了

Receive\_charac\_no=收件人角色ID

Item\_id=物品代码

Add\_info=物品数量，装备的品级（最下级 下级 中级 上级 最上级）

Endurance=装备的耐久度 物品的话不知道有什么用写0即可 装扮要写1

Upgrade=装备的强化数值

amplify\_option=装备附加的红字属性 （力量 智力 体力 精神 ）

amplify\_value=装备附加的红字数值 最大65535

（发送+31红字这个再写65535就是传说中的红字六万八）

Gold=邮件附带的金币数量

receive\_time=邮件发送时间

delete\_flag=这个是删除状态 发送写0即可 角色领取了这个物品后就会自动变成1

seal\_flag=装备是否封装 0不封1封装 物品不知道什么用（SS切勿封装）

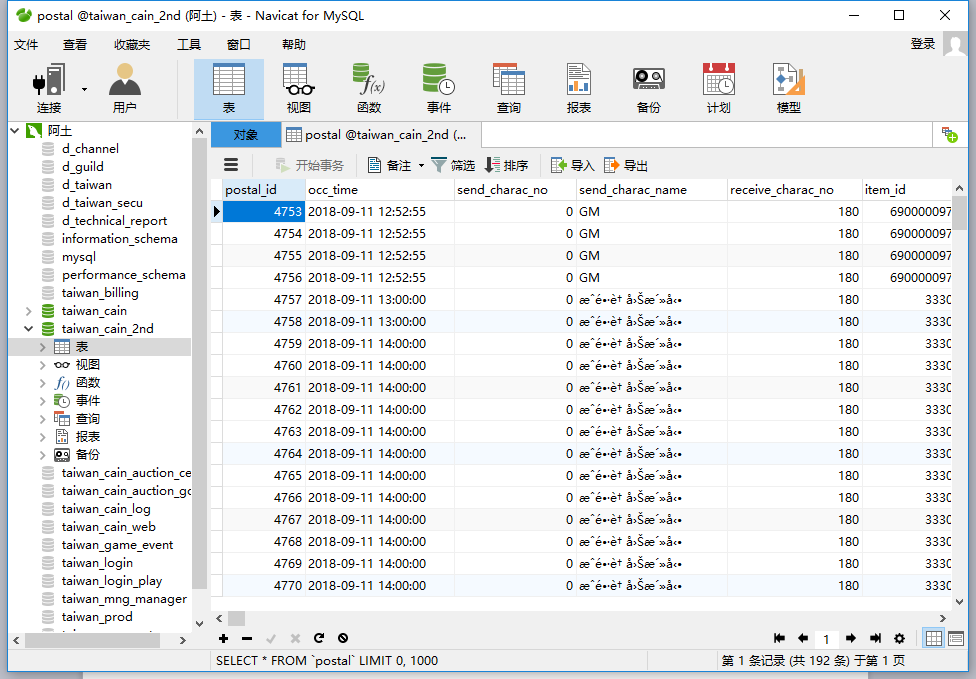
发送宠物蛋的话也要写1封装一下

creature\_flag=这个可能是邮件类型吧，反正要发送宠物的话 这个要写1 其余写0即可

letter\_id=这个就是对应的邮件编号了，可以通过select语句来获得letter\_id的最大值

seperate\_upgrade=装备的锻造等级

unlimit\_flag=装扮写1 宠物或物品或装备写0



这是在邮件的数据，单单这样的话是发送不了宠物装扮宠物蛋的

宠物装扮和宠物蛋还有另外两个表

先说宠物和宠物蛋吧

宠物和宠物蛋都在一个表里面 只是有一个字段的内容不一样 所以类型不一样

位置：taiwan\_cain\_2nd.creature\_items

字段解析：

ui\_id=类似于那个邮件ID和letterID 不过写法一样 自动递增的不用手动干预

Charac\_no=角色ID

it\_id=道具代码（宠物或宠物蛋的代码）

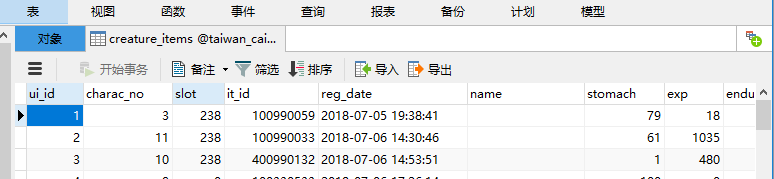
reg\_date=获得时间

Name=宠物名字（用了宠物改名卡后的那个名字）

Exprire\_date=到期时间

Delete\_date=删除时间

Stomach=用处不明 写100就好

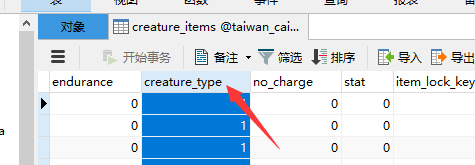


关键点来了，区分宠物和宠物蛋

由creature\_type这个字段控制

发送宠物这个就写1

宠物蛋就写0



这里是装扮

装扮 同样要写另一个表

位置：taiwan\_cain\_2nd.user\_items

字段解析；

charac\_no=角色ID

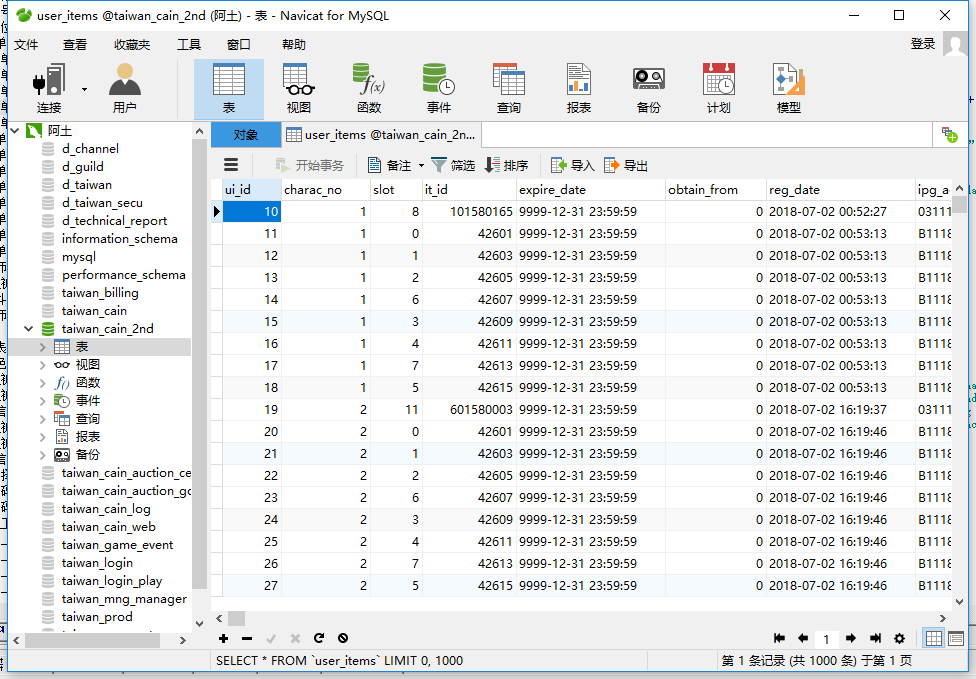
it\_id=道具代码（装扮的代码）

expire\_date=到期时间

reg\_date=获得时间

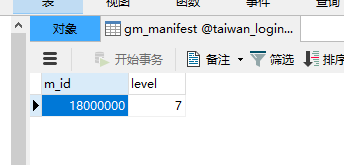
obtain\_from=用处不明 写1即可

Stat=用处不明 写2



番外；GM账号

位置：taiwan\_login.gm\_manifest



第一个字段为账号UID

第二个字段就是GM等级

等级应该还有其他地方关联，不过没什么人研究这个鬼东西

就目前来说 这个东西的作用只是能发那个[阿拉德守护者]的喇叭

以及普通玩家在大街上看到此账号的角色不能查看信息

这里说一些常用的SQL语句在易语言中的使用方法

增删改查

新增 insert into

删除 delete

修改 update

查询 select

条件 where

和 AND

且 OR

新增的两种语法使用方式

“insert into 数据库名.表名 set 字段1=’字段1的内容’,字段2=’字段2的内容’”

“insert into 数据库名.表名 (字段1,字段2) value (字段1的内容,字段2的内容)”

删除的语法使用方式

“delete from 数据库名.表名 ”

//这是直接清空表了 删除指定项目的话需要配合条件(where)语句

条件删除语法使用方式

“delete from 数据库名.表名 where 字段1=’判断的内容’”

//条件语法解析

Where 字段1=’判断的内容’

如果操作的表的指定字段（字段1）的内容=判断的内容 则会执行删除命令

易语言应用实例：

执行SQL语句 (sql句柄, “delete from d\_taiwan.accounts where uid=’18000001’”)

执行结果为

如果uid=18000001 表中的字段[uid]等于18000001的项目都会删除 而不会删除其他的

修改的两种语法使用方式

“update 数据库名.表名 (字段1,字段2) value (字段1的新内容,字段2的新内容)”

“update 数据库名.表名 set 字段1=’字段1的新内容’,字段2=’字段2的新内容’”

和新增的语法使用方式差不多 当然 修改也是需要条件语句（where）的

使用方法参考上面的【条件语法解析】

查询语法的使用方式以及易语言应用实例

“select \* from 数据库名.表名”

这一句将会取出这个表全部的数据

如果想单独取出某个字段的要怎么做呢？

答案在select和from中间的那个\*符号

这个\*号在这里代表了所有

如果只想取出某个或多个字段 只需要把\*号改为字段的名称即可

例如

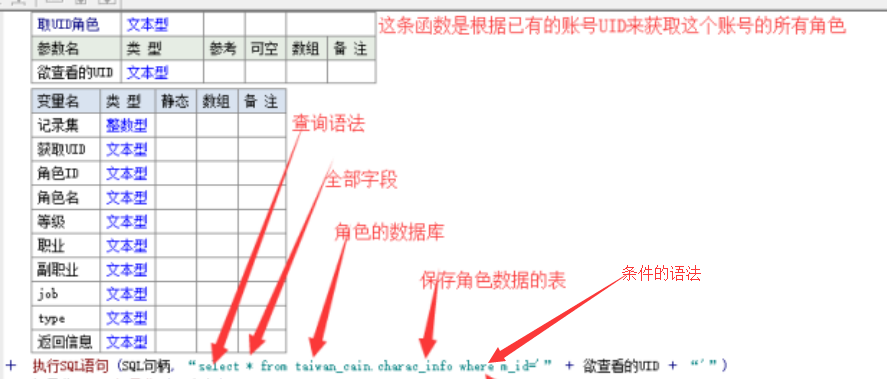
“select 字段1,字段2,字段3 from 数据库名.表名”

这样取出来的数据就会只有字段1和字段2还有字段3的全部数据

如果想根据已知的条件来得到指定的数据呢？

去参考上面的条件语法

下面是我的GM工具使用查询的实例



那么查询到后要怎么取出数据呢

使用函数

取记录集()返回的是一个整数型的记录集句柄 需要的参数是执行SQL语句的那个SQL句柄

取出数据就要操作这个记录集的句柄

具体的使用方法为函数

读字段值()这个函数没有返回值 但是有三个参数

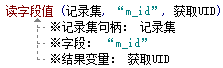
第一个就是要操作的记录集的句柄

第二个是在数据库中的字段（select 字段 from）名

第三个就是存放取出结果的一个变量

这个变量是一个通用型的数据类型 可以为文本型 整数型 字节集 字节 小数 双精度小数

不过首先给他定义一个变量名



既然知道了取出数据的函数

那么怎么确定取出的是哪一条呢？

使用对记录集操作的行函数

到首行(记录集句柄)

到尾行(记录集句柄)

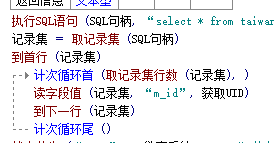
到上一行(记录集句柄)

到下一行(记录集句柄)

首先给一个循环函数（鬼知道你取出来的数据有多少条啊）

计次循环的函数有两个参数 1是次数 2是记录次数的一个变量

下面是一个简单的循环取每一行记录集的应用实例



到首行() //把当前操作的行定义到记录集的第一行

读字段值 (记录集, “m\_id”, 获取UID)

读取记录集中“m\_id”这个字段的数据保存到【获取UID】这个变量中

读取完之后我们可以做一些其他的变量赋值操作

因为读字段值的第三个参数是直接赋值给那个变量的 在下一次覆盖之前

我们可以先把这个变量的数据保存到另一个静态的变量中



保存完毕后再执行下一步操作



这个函数的作用就像名字一样 把操作的记录集行定义到下一行

这个函数执行完毕后就相当于这一个循环函数执行完毕了 将会进行下一轮的循环

简单的操作教程至此结束

感谢闰土大佬（养老服群号713021425）和豆腐（豆腐百科560713584）提供测试环境。

如有疑问或进阶学习可联系本人或加入交流群

QQ1048893645 交流群327052207

转载请注明出处