mysql 数据库

## 数据库的发展史

1.数据库

Oracle商用数据库，DB2

MS Sql Server

MySql 免费数据库

sqlite 手机用嵌入式数据库

2.mysql

============================================

\*开源免费

\*mysql 卖给sun公司，

一年后，Oracle收购了sun

\*mysql 免费社区版

闭源的收费版

\*mysql 未来有完全闭源的风险

\*mysql 创始人，有开发了一个mysql的分支mariadb

\*开源社区已经放弃支持 mysql,转为支持mariadb

3.mysql 安装

==================================================

\*安装 mysql 社区版 community

\*安装 marriadb

\*linux yum安装 marriadb

## mysql 使用

==================================================

### 登录 mysql 服务器

mysql -hlocalhosst -P3306 -uroot -p

mysql -uroot -p

password:root

### 查看数据库列表

show databases;

show schemas;

\*)databases 数据库

\*)schema 数据库中一组对象的集合，即数据库

### 进入数据库

use test;

use mysql;

### 查看数据库表

show tables;

### 查询表中的数据

select \* from user;

### 创建数据库

create database Jayus charset utf8；

(定义字符编码）

show databases;

### 查看编码

asc ii

0-127

iso-8859-1

latin-1

160-255

双字节编码

GBK

unicode 统一码 100万+字符

双字节常用字符 65535

utf8

英文单字节

中文三字节

### 数据库保存目录

linux:

ls /var/lib/mysql

windos:

mysql安装目录\data

### 查看数据库的配置文件

linux:

cat /var/lib/mysql/dbl/db.opt

windows:

把文件拖拽到记事本

配置两项内容：

\*）数据库默认字符编码

编码

asc ii

0-127

iso-8859-1

latin-1

160-255

双字节编码

GBK

unicode 统一码 100万+字符

双字节常用字符 65535

utf8

英文单字节

中文三字节

\*）字符的排序方式

collation 排列、校对、字符排序

show collation;

show collation like 'gbk%'；

show collation like 'utf8\\_%';

(\\_转义字符)

一般只能用一下两种默认方式

utf8\_general\_ci （ci不区分大小写）

gbk\_chinese

## 修改数据库

### 修改数据库编码

alter database db1 charset gbk;

### 修改数据库排序(很少用)

alter database db1 collate gbk\_bin;

## 创建表

column/field 列、字段

use db1;

-- 用户表 （-- 表示注释）

create table users(

name varchar(16),

age int

) engine=innodb charset=gbk;

(对表参数设置)

show tables;

desc user;

## 存储引擎 engine

\*innodb

\*)支持数据库高级功能

\*）事务

\*）外键

\*)myisam

\*）数据存储的基本功能

\*）效率非常高

### 在表中插入数据

insert into user values('aa',23);

insert into users(age,name) values(23，'Jml')；

### 数据库目录中的表文件

user.frm --表结构文件

user.myd --数据存储文件

user.myi --索引人文件

linux:

ls -al /var/lib/mysql/db1

windows:

找到 data/db1 目录

### 数据库分类

==================================================

\*关系型数据库

\*）用“表”保存数据

相关数据存入一张表

\*非关系型数据库

\*）对象数据库

\*）键值数据库

## 修改表

### 修改表名

rename table user to tb\_user;

## 修改表属性（引擎，编码）

alter table tb\_user

engine=myisam charset=utf8;

show create table tb\_user;

### 添加字段

first 加载第一个

after 指定加载某个字段之后

alter table tb\_user add id int first;

alter table tb\_user add (

gender char (1) ,

tel char(11)

);

select \*from tb\_user;

### 修改字段属性

alter table tb\_user

change gender sex varchar(10);

select \*from tb\_user;

desc tb\_user;

### 修改字段名

alter table tb\_user

modify tel varchar(20);

select \*from tb\_user;

### 修改字段顺序

alter table tb\_user

modify sex varchar(20) after tel;

desc tb\_user;

### 删除字段

alter table tb\_user

drop sex;

desc tb\_user;

### 创建修改类型

create database

alter database

create table

alter table

rename table

### 删除表(不可恢复）

drop table tb\_use;

### 删除库(不可恢复)

drop database db1;

# 数据类型

## 终端的使用

1.右键桌面， 终端

2.mysql -uroot -p

password:直接回车

3.show databases; 显示所有数据库

4.create database test2 charset utf8;

创建数据库，指定默认字符编码

5.use test2;

进入数据库

6.退出 mysql

exit 或 \q

==================================================

创建数据表

1.数据类型

2.约束

3.表的增删改查

创建表

删除表

修改表结构

查看表

建表语句

==============================================

create table user(

id int,

username varchar(32),

password varchar(32)

) engine=innodb charset=utf8;

show tables;

## 数据类型

===========================================

### 字符串

-----------------------------------

char(固定长度) 定长字符串,最多 255 个字节

varchar(最大长度) 变长字符串,超过出错,65535 个字节,一般不超过 255

text 65535 字节

blob 大数据对象，以字节方式存储数据

### 数字

-----------------------------------

不带符号, 添加 unsigned,

zerofill 显示时，位数不足，可以补0

tinyint 1字节

SMALLINT 2字节

int 4字节

BIGINT 8字节

int(6)

不影响保存的数据范围

只影响查询显示的位数

float 4字节

double 8字节

可能保存不精确的值

DECIMAL/mumeric

保存精确浮点数

需要指定总长度和小数位位数

decimal(10,2) 八.二

### 日期

-----------------------------------

date 年月日

time 时分秒

datetime 年月日时分秒

到 9999 年

timestamp 同 datetime

到 2038 年

第一个 timestamp 字段，

在行数据更新时，自动修改为系统当前时间

也可以使用 bigint 整数表示时间

java 中时间是用长整型毫秒值表示

毫秒值： 1970-1-1 0点开始

### 表字段总字节量限制为 65535

====================================================

\* varchar 需要额外两个字节来表示长度

\* 一个字段如果允许 null 值，需要额外 1 个字节

\* text 类型 65535 个字节，

但仅占用表的 10 个字节额度

## 约束 5种

====================================================

\* 所有的约束保存在 information\_schema 库中

table\_constraints 表中

\* 限制字段的取值

\* 主键、非空、唯一、外键、检查(mysql不支持)

## 主键

\*)对一行数据的唯一标识

\*)不重复

\*)不能为空 null

\*)尽量使用业务无关数据作为主键

\*)连续自增整数 auto\_increment

\*)随机字符串

\*)主键自动添加索引

\*)添加主键

### 建表时添加

create table tb1(

id bigint primary key auto\_increment,

...

)，

### 修改表添加

alter table tb1 add primary key(id);

### 删除主键

alter table tb1 drop primary key;

\*)auto\_increment

### 查看下一个自增值:

show create table tb1\G;

\*)一直增加不会退

\*)如果在表中手动插入一个更大的主键值，

会从这个值继续递增

\*)只有主键才能自增

\*)如果删除自增主键约束，

先取消自增，

再取消主键约束

\*) 双主键(不推荐)

两个字段和并作为主键

create table tb1(

id ..,

name ...,

...,

primary key(id, name)

);

## 非空

不能取 null 值

### 添加非空约束

\*)创建表时

create table tb1(

id int primary key,

name varchar(20) not null

);

### 查看表中的非空约束

desc tb1;

show create table tb1\G

### 修改字段,添加/删除非空约束

alter table tb1 modify name varchar(32) not null;

alter table tb1 modify name varchar(32) null;

## 唯一约束

限制字段的值不能重复，但允许多个 null 值

### 添加唯一约束

\*) 创建表时

create table tb1(

...,

name varchar(20) unique,

...

);

create table tb1(

...

name ...,

gender ...,

...,

unique key(name, gender)

)；

### 修改字段，添加唯一约束

alter table student

modify name varchar(20) unique;

alter table student

add unique key(name, gender);

### 查看唯一约束

show create table tb1\G

### 取消唯一约束

alter table tb1 drop index 约束名

## 外键约束

外键一般引用另一个主键中的值

\*)mysql的 innodb 引擎才支持外键,效率低

myisam 引擎不支持外键,效率高

学生表 联系方式表

id name .. id stu\_id tel .. ..

1 张三 1 1 43634634

2 李四 2 3 47542345

3 王五 3 2 678434

4 2 845345354

5 null 45568753

6 1 44523234

### 添加外键约束

\*)创建表时添加

create table tb2(

...,

stu\_id ...,

...,

foreign key(stu\_id) references student(id)

);

### 修改字段添加外键

alter table tb2

add foreign key(stu\_id) references student(id);

### 取消外键约束

alter table tb2 drop foreign key 约束名;

### 检查约束(mysql 不支持)

create table tb1(

...

gender

age

...,

check(gender='M' or gender='F'),

check(age>=16 and age<=60)

);

默认值

===========================================

create table tb\_item(

...

status tinyint default 0,

...

);

### 练习

====================================================

1. 测试数据类型，以及表的 65535 字节量限制

drop table tb1; -- 删除表

create table tb1(

id int(6), -- 4字节

title varchar(21842) -- 21842\*3+2+1

)engine=innodb charset=utf8;

2.测试 text 类型

drop table tb2; -- 删除表

create table tb2(

id int(6), -- 4字节

title varchar(21839), -- 21842\*3+2+1

des text -- 占10个字节

)engine=innodb charset=utf8;

3. 测试主键

create table student(

id bigint primary key auto\_increment,

name varchar(20)

);

-- 插入一行数据

insert into student(name) values('张三');

insert into student(name) values('李四');

-- 查询

select \* from student;

4. 手动向自增主键插入更大的值

insert into student(id,name) values(10000, '王五');

select \* from student;

show create table student\G;

5. 删除主键约束

-- 取消自增

alter table student modify id bigint;

-- 删除主键约束

alter table student

drop primary key;

6. 学生姓名非空

use test2;

drop table if exists student;

create table student(

id bigint unsigned primary key auto\_increment,

name varchar(20) not null,

gender char(1)

);

desc student;

show create table student\G;

insert into student(gender) values('M');-- 失败

insert into student(name, gender)

values('张三', 'M');

show warnings;

select \* from student;

7. 修改学生表添加学号字段，不能重复

alter table student

add stu\_num int unsigned unique

after id; -- 字段顺序添加在 id 后面

desc student;

show create table student\G; -- 查看唯一约束名

-- 在约束表中查看

use information\_schema;

select \* from table\_constraints

where table\_schema='test2'; -- 过滤test2库中的约束

use test2;

-- 插入重复学号

insert into student(name, stu\_num)

values('张三', 2017001);

insert into student(name, stu\_num)

values('张三', 2017002);

-- 删除唯一约束,约束名字叫stu\_num

alter table student drop index stu\_num;

-- 再添加唯一约束

-- 如果已经有重复数据，不能添加唯一约束

alter table student add unique key(stu\_num);

\*) 添加联系方式表,用 stu\_id 引用学生的主键id值

create table contacts(

id bigint unsigned primary key auto\_increment,

stu\_id bigint unsigned,

tel varchar(20),

foreign key(stu\_id) references student(id)

);

desc contacts;

show create table contacts\G;

select \* from student;

insert into contacts(stu\_id,tel)

values(1, '345364564');

insert into contacts(stu\_id,tel)

values(1, '956454');

insert into contacts(stu\_id,tel)

values(2, '7455');

insert into contacts(stu\_id,tel)

values(5, '4565363');

insert into contacts(stu\_id,tel)

values(5, '2467457');

insert into contacts(stu\_id,tel)

values(9999, '2467457');

-- 取消外键约束

alter table contacts

drop foreign key contacts\_ibfk\_1;

\*) 创建 分类表和商品表

-- 删除数据库 jtds

drop database if exists jtds;

-- 重新创建数据库 jtds

create database jtds charset utf8;

-- 进入 jtds 库

use jtds;

-- 新建分类表 tb\_item\_cat

create table tb\_item\_cat(

id bigint primary key auto\_increment comment '自增主键',

parent\_id bigint comment '父分类',

name varchar(50) comment '分类名称',

status tinyint default 1 comment '状态码',

sort\_order int comment '排序顺序号',

is\_parent tinyint default 1 comment '是否父分类',

created datetime comment '创建时间',

updated datetime comment '修改时间',

foreign key(parent\_id) references tb\_item\_cat(id)

)engine=innodb charset=utf8;

show create table tb\_item\_cat;

create table tb\_item(

id bigint primary key auto\_increment comment '主键',

cid bigint not null comment '分类id',

brand varchar(50) comment '品牌',

model varchar(50) comment '型号',

title varchar(100) comment '名称',

sell\_point varchar(500) comment '卖点',

price bigint comment '价格\*100',

num int comment '库存数量',

barcode varchar(30) comment '条码',

image varchar(500) comment '图片路径',

status tinyint default 2 comment '状态码',

created datetime comment '创建时间',

updated datetime comment '修改时间',

foreign key(cid) references tb\_item\_cat(id)

)engine=innodb charset=utf8;

作业

================================================

4.2.2 商品描述表(tb\_item\_desc)

item\_id 外键引用商品主键

timestamp 修改成 datetime

4.2.1 商品规格参数表(tb\_item\_param)

Item\_cat\_id 分类id引用分类表主键

timestamp 修改成 datetime

4.2.1 用户表(tb\_user)

timestamp 修改成 datetime