安装mysql 参考网上图解

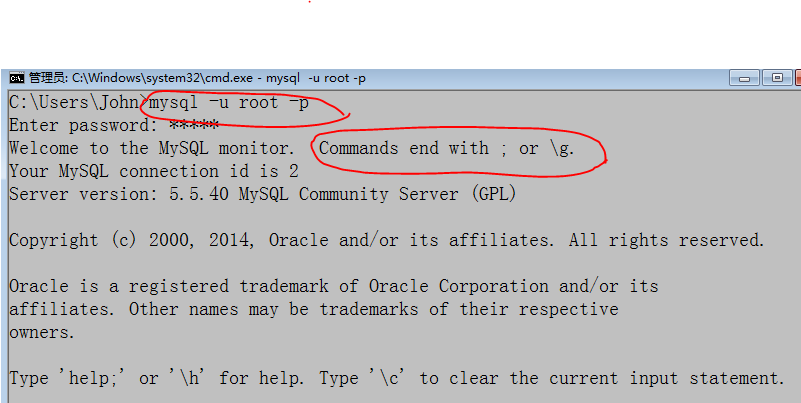
在cmd中进入mysql使用 mysql –u roo –p 加密码 命令

可以为： mysql –uroot –pmysql

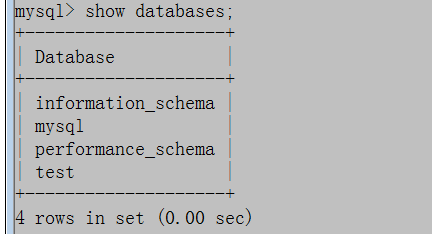
Mysql –u root –p 回车 密码

在mysql中， 每一条命令的结束都是使用“分号”

退出mysql 使用 “quit”命令



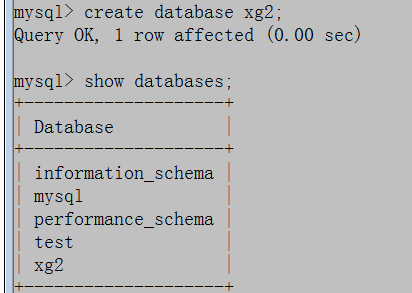
# 库

1. 查看所有的数据库
   1. show databases;
   2. 

2, 创建数据库

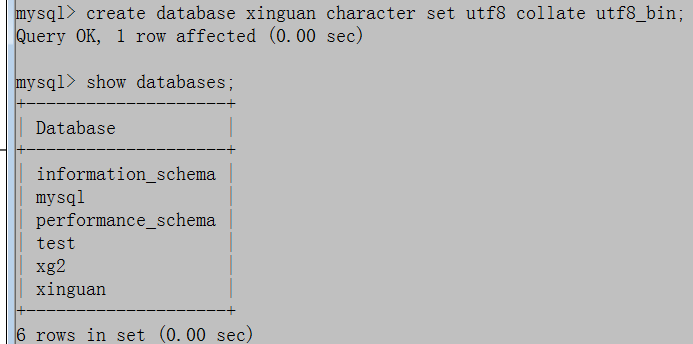
create database 数据库名称;

create database xg2;



|  |
| --- |
| CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] *db\_name*      [*create\_specification* [, *create\_specification*] ...]    *create\_specification*:      [DEFAULT] CHARACTER SET *charset\_name*    | [DEFAULT] COLLATE *collation\_name* |

create database xinguan character set utf8 collate utf8\_bin;



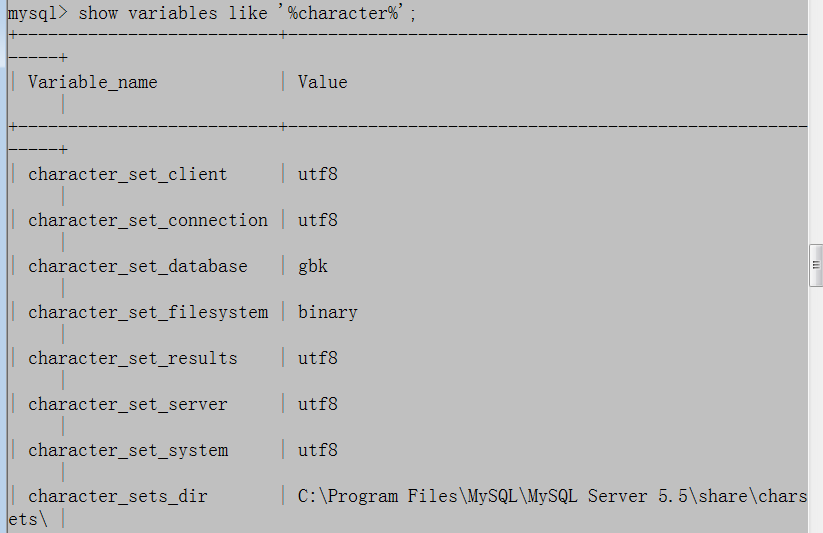
3, 修改数据库的字符集

alter database xinguan character set gbk;



查看数据库中的字符集

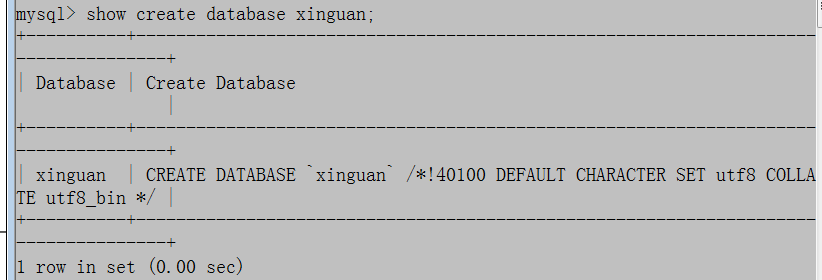
show variables like ‘character\_set\_database’;



4, 修改数据库的名称

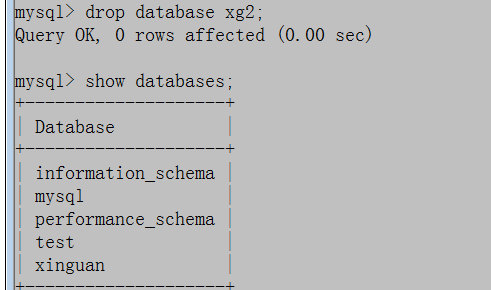
5, 查看数据库的建库语句

show create database 数据库名称;

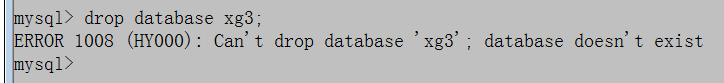


6, 删除数据库

drop database xg2;

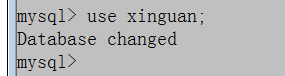


注意点:



7, 使用数据库

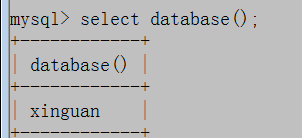
use 数据库名称;



# 表

首先要进入到数据库

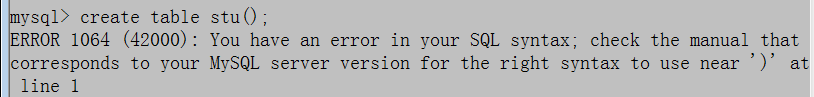
查看当前正在哪个数据库实例中.



1, 创建表

|  |
| --- |
| CREATE [TEMPORARY] TABLE [IF NOT EXISTS] *tbl\_name*      [(*create\_definition*,...)]      [*table\_options*] [*select\_statement*]  或：  CREATE [TEMPORARY] TABLE [IF NOT EXISTS] *tbl\_name*      [(] LIKE *old\_tbl\_name* [)];    *create\_definition*:  *column\_definition*    | [CONSTRAINT [*symbol*]] PRIMARY KEY [*index\_type*] (*index\_col\_name*,...)    | KEY [*index\_name*] [*index\_type*] (*index\_col\_name*,...)    | INDEX [*index\_name*] [*index\_type*] (*index\_col\_name*,...)    | [CONSTRAINT [*symbol*]] UNIQUE [INDEX]          [*index\_name*] [*index\_type*] (*index\_col\_name*,...)    | [FULLTEXT|SPATIAL] [INDEX] [*index\_name*] (*index\_col\_name*,...)    | [CONSTRAINT [*symbol*]] FOREIGN KEY          [*index\_name*] (*index\_col\_name*,...) [*reference\_definition*]    | CHECK (*expr*) |

create table stu();



create table stu(

id int primary key auto\_increment,

sname varchar(32),

gender bit default 0,

age int

);

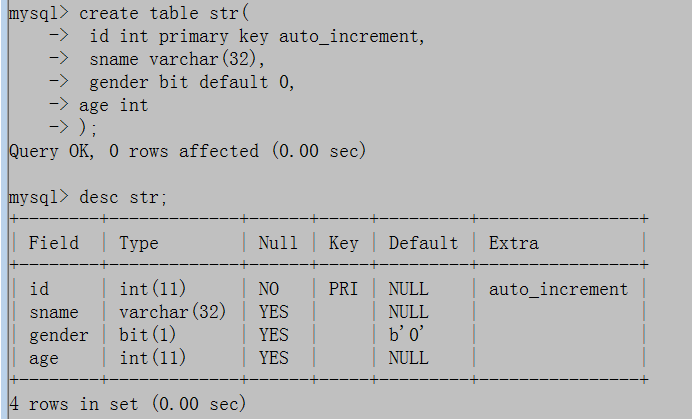
primary key 主键, 唯一, 并且非空

unique 唯一键

not null 非空键

auto\_increment : 自动增长

desc 表名称 查看表结构



拓展:

Java中的数据类型和 Mysql中数据类型的对应

在Java中数据类型可以分为: 基本数据类型和引用数据类型

基本数据类型包括:

byte short int long

float double

char

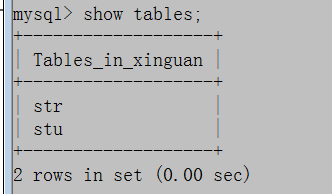
boolean

拓展: char类型和varchar类型和varchar2的区别

2, 查看表

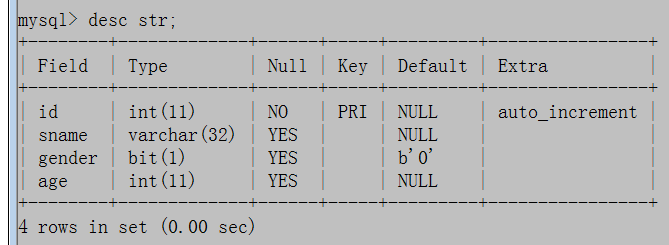
a, 查看一个数据库实例中所有的表

show tables;



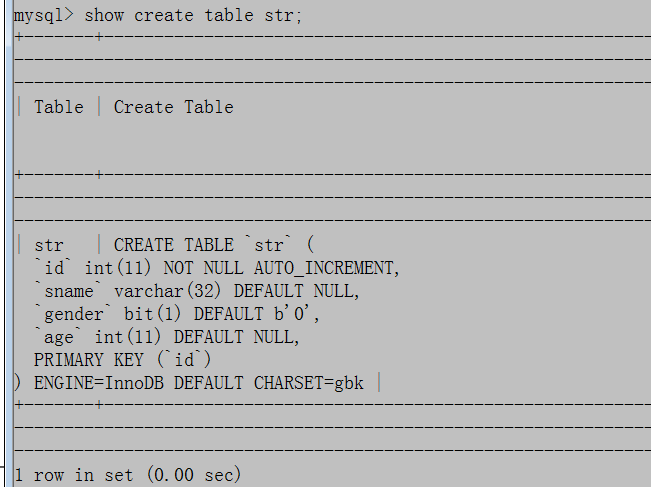
b, 查看表结构

desc 表名称;



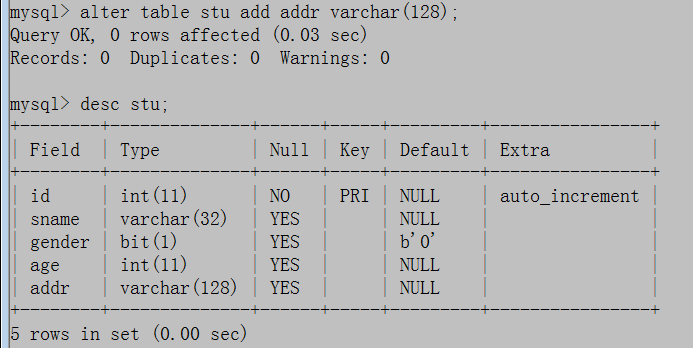
c, 查看建表语句

show create table 表名称;



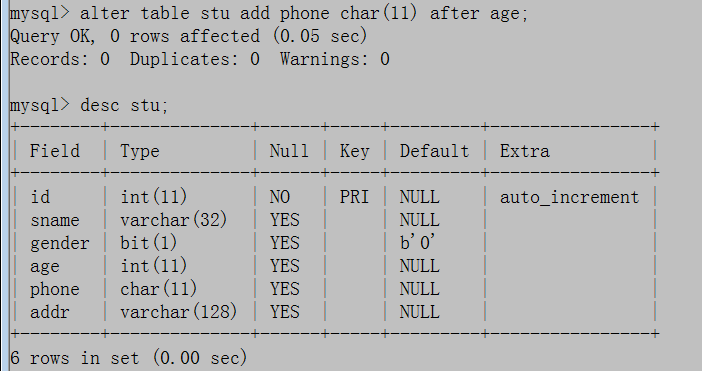
3, 修改表

a, 增加一个字段



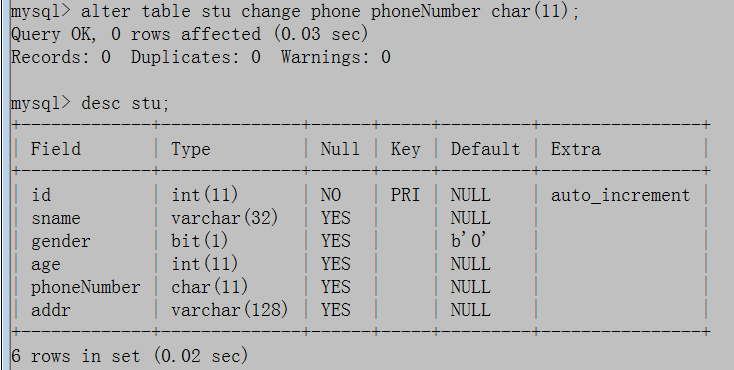
b, ;在指定的位置增加一个字段

alter table stu add phone char(11) after age;



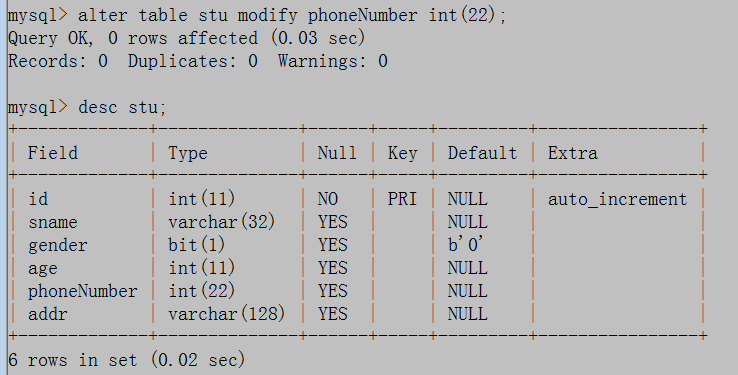
c, 修改字段的名称

alter table stu change phone phoneNumber char(11);



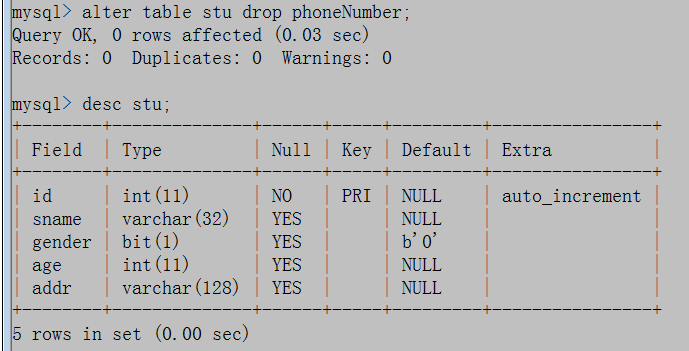
d, 修改字段的类型

alter table stu modify phoneNumber int(22);



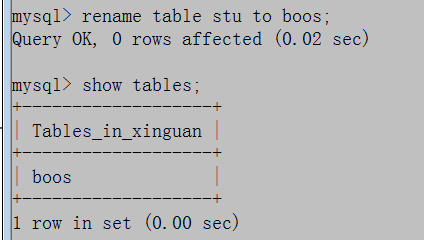
e, 删除某一个列

alter table stu drop phoneNumber;

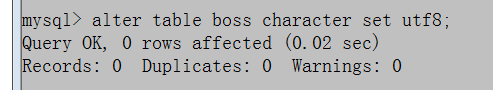


f, 修改表的名称

rename table stu to boos;



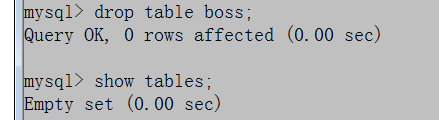
g, 修改表的字符集



拓展: alter, change, modify 区别???

h, 删除表

drop table 表名称;



拓展: drop, delete, truncate的区别???

# 数据

CRUD 是什么???

C ------------ create

R -------------- retrieve

U -------------- update

D ------------- delete

1, insert

|  |
| --- |
| INSERT [LOW\_PRIORITY | DELAYED | HIGH\_PRIORITY] [IGNORE]      [INTO] *tbl\_name* [(*col\_name*,...)]      VALUES ({*expr* | DEFAULT},...),(...),...      [ ON DUPLICATE KEY UPDATE *col\_name*=*expr*, ... ]  或：  INSERT [LOW\_PRIORITY | DELAYED | HIGH\_PRIORITY] [IGNORE]      [INTO] *tbl\_name*      SET *col\_name*={*expr* | DEFAULT}, ...      [ ON DUPLICATE KEY UPDATE *col\_name*=*expr*, ... ]  或：  INSERT [LOW\_PRIORITY | HIGH\_PRIORITY] [IGNORE]      [INTO] *tbl\_name* [(*col\_name*,...)]      SELECT ...      [ ON DUPLICATE KEY UPDATE *col\_name*=*expr*, ... ] |

create table stu(

id int primary key auto\_increment,

sname varchar(32) not null,

gender char(2),

age int

);

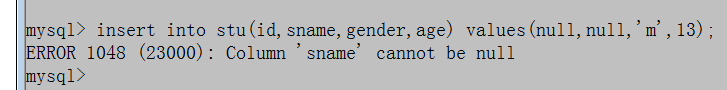
insert into stu(id,sname,gender,age) values(null,'siri','f',5);

insert into stu(id,sname,gender,age) values(null,'lily','f',12), (null,'lucy','f',12);

insert into stu(id,sname,gender,age) values(5,'tom','m',5);

insert into stu(id,sname,gender,age) values(null,'jack','m',13);

insert into stu(id,sname,gender,age) values(null,null,'m',13);



insert into stu(sname)values(‘rose’);

拓展: 如果写的数据超出了数据类型所表示的范围, 会怎么样???

拓展: 我们现在需要做大量的数据来做测试, 那么该怎么样在最短的时间内, 存入大量数据到表中.???

“蠕虫复制”:

insert into stu (sname, gender, age) select sname, gender, age from stu;

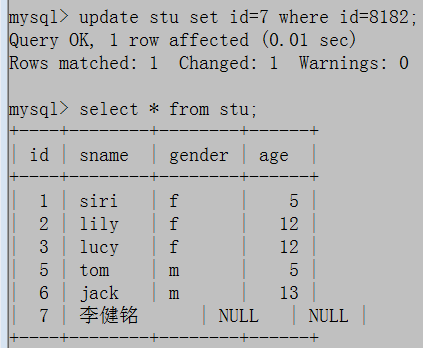
insert into stu(sname) values(‘李健铭’)

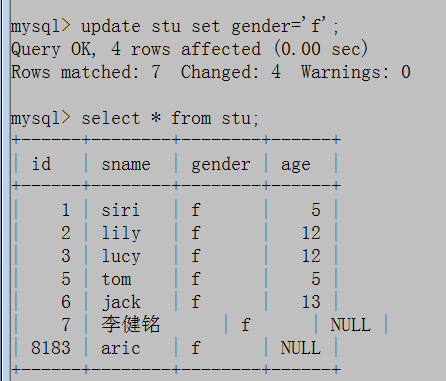
2, update

|  |
| --- |
| Single-table语法：  UPDATE [LOW\_PRIORITY] [IGNORE] *tbl\_name*      SET *col\_name1*=*expr1* [, *col\_name2*=*expr2* ...]      [WHERE *where\_definition*]      [ORDER BY ...]      [LIMIT *row\_count*]  Multiple-table语法：  UPDATE [LOW\_PRIORITY] [IGNORE] *table\_references*      SET *col\_name1*=*expr1* [, *col\_name2*=*expr2* ...]      [WHERE *where\_definition*] |

where条件一定要设置, 如果没有设置, 则会改变所有的值.

使用where就是对更新数据加以条件限定.





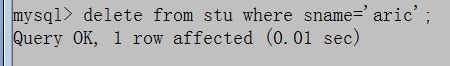
update语句千万不要和from连用

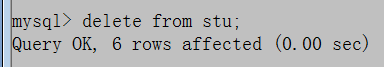
3, delete

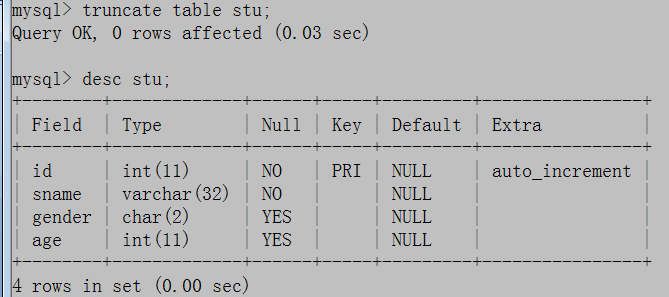
|  |
| --- |
| 单表语法：  DELETE [LOW\_PRIORITY] [QUICK] [IGNORE] FROM *tbl\_name*      [WHERE *where\_definition*]      [ORDER BY ...]      [LIMIT *row\_count*]  多表语法：  DELETE [LOW\_PRIORITY] [QUICK] [IGNORE]  *tbl\_name*[.\*] [, *tbl\_name*[.\*] ...]      FROM *table\_references*      [WHERE *where\_definition*]  或：  DELETE [LOW\_PRIORITY] [QUICK] [IGNORE]      FROM *tbl\_name*[.\*] [, *tbl\_name*[.\*] ...]      USING *table\_references*      [WHERE *where\_definition*] |

一定要注意: 在删除操作的时候, 一定要加上限定条件. 否则的话, 一张表中的数据就被全部删除了.

在几乎所有的大公司中是不允许有”删除”操作的.







4 select

建表:

create table exam(

id int primary key auto\_increment, -- sql中的注释

name varchar(32) not null,

chinese int,

math int,

english int

);

insert into exam values(null, 'lily', 95, 96, 97);

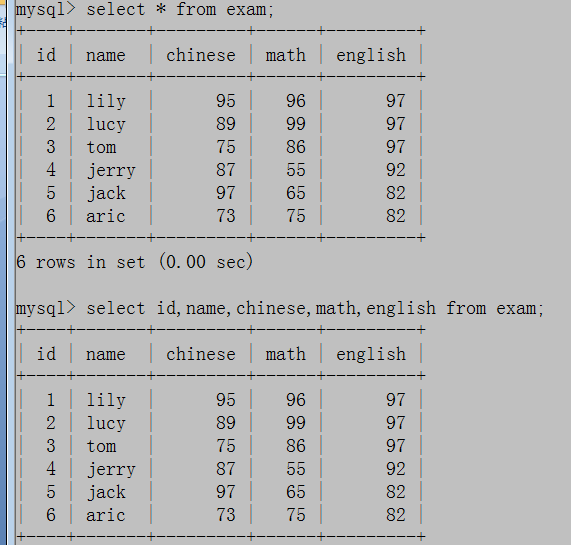
insert into exam values(null, 'lucy', 89, 99, 97);

insert into exam values(null, 'tom', 75, 86, 97);

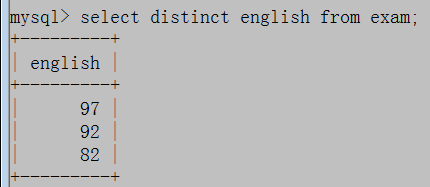
insert into exam values(null, 'jerry', 87, 55, 92);

insert into exam values(null, 'jack', 97, 65, 82);

insert into exam values(null, 'aric', 73, 75, 82);

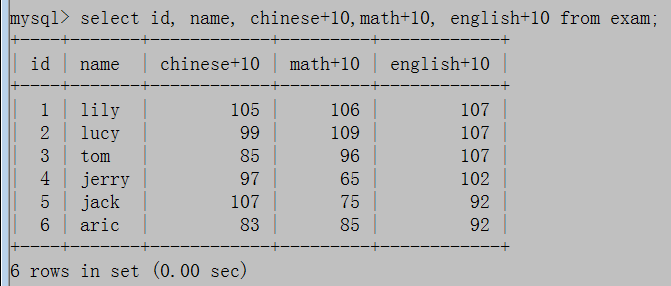


查询的时候, 去除重复



对每一个学生的各科成绩都加10分

select id, name, chinese+10,math+10, english+10 from exam;



查询lily的总成绩

select id, name, chinese+math+english

from exam

where name=’lily’;

使用as是给字段或者表, 等取一个别名(就是取一个外号) 可以省略不写

select id, name, chinese+math+english 总成绩

from exam

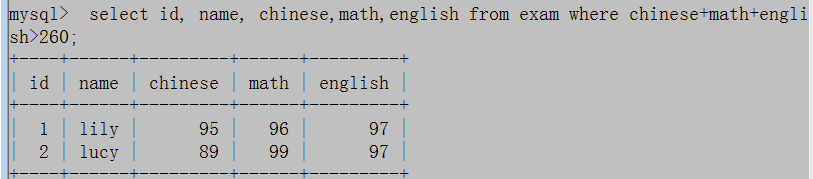
where name=’lily’;

where条件查询

在where的后面添加查询的详细条件, 使得我们查询到的结果更加的细致

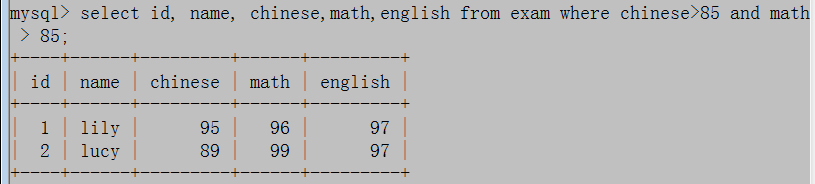
查询总成绩大于260的所有学生信息

select id, name, chinese,math,english from exam where chinese+math+english>260;



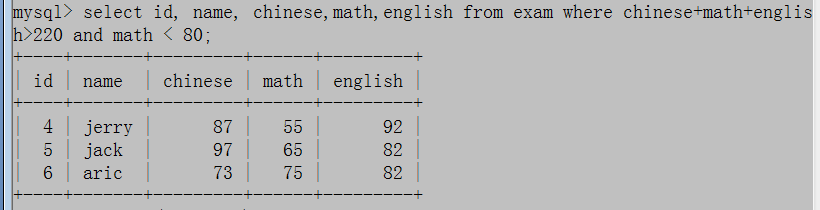
查询语文成绩和数学成绩都大于85的所有学生信息

select id, name, chinese,math,english from exam where chinese>85 and math > 85;



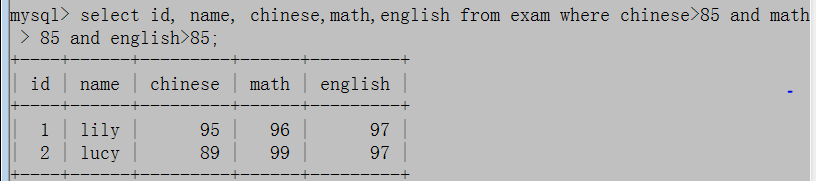
查询总成绩大于220,但是数学成绩小于80的所有学生信息

select id, name, chinese,math,english from exam where chinese+math+english>220 and math < 80;



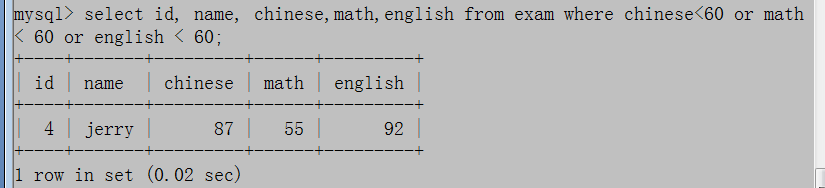
查询, 三科成绩都大于85的学生信息

select id, name, chinese,math,english from exam where chinese>85 and math > 85 and english>85;



查询出所有有挂科同学的信息

select id, name, chinese,math,english from exam where chinese<60 or math < 60 or english < 60;



查询所有语文成绩在85 至 95之间的学生信息

select id, name, chinese from exam where chinese >= 85 and chinese <= 95;

select id, name, chinese from exam where chinese between 85 and 95;

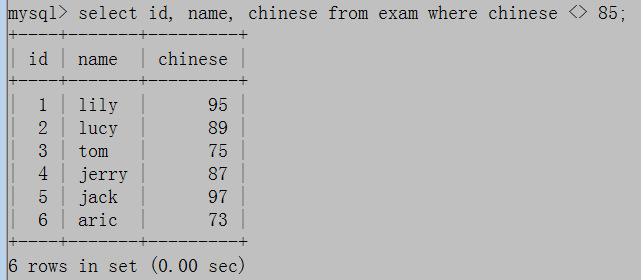
查询所有语文成绩不在85到95之间的学生信息

select id, name, chinese from exam where chinese < 85 or chinese > 95;

select id, name, chinese from exam where not chinese between 85 and 95;

查询语文成绩不是85的学生的信息

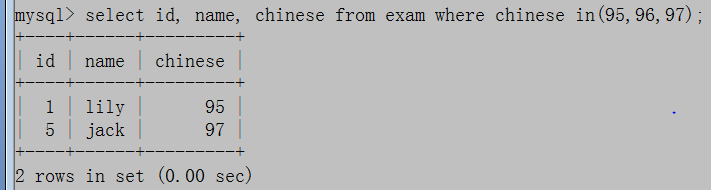
select id, name, chinese from exam where chinese <> 85;



查询所有语文成绩是95, 96, 97 的学生的信息

select id, name, chinese from exam where chinese = 95 or chinese = 96 or chinese = 97;

select id, name, chinese from exam where chinese in(95,96,97);



拓展: != 和 <> 的区别???

数据库的备份和恢复

备份:

在cmd中输入命令

mysqldump –u root –p 数据库名称>路径和文件名称

mysqldump –u root –p xinguan>c:/xinguan.sql

恢复数据的前提: 必须要有对应的数据库实例.

恢复:

mysql -u root –p 数据库名称<文件路径和文件名

mysql –u root –p xinguan<c:/xinguan.sql

另一种恢复数据的方式

在mysql中直接使用source命令

source 文件路径和名称

CREATE TABLE employees (

id int(11) NOT NULL AUTO\_INCREMENT,

name char(32) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT NULL,

gender bit(1) DEFAULT b'1',

age int(11) DEFAULT NULL,

birthday date DEFAULT NULL,

entry\_date date DEFAULT NULL,

position varchar(32) DEFAULT NULL,

salary double DEFAULT NULL,

resume text CHARACTER SET utf8 COLLATE utf8\_bin,

bonus double DEFAULT NULL,

PRIMARY KEY (id)

) ENGINE=InnoDB AUTO\_INCREMENT=31 DEFAULT CHARSET=utf8;

INSERT INTO employees VALUES (1,'赵财旺','',23,'1993-04-01','2014-04-01','经理',7000,'重庆理工大学',100),(2,'张样','',22,'1993-02-17','2014-04-02','员工',5000,'重庆工商大学',500),(3,'孙一菲','\0',23,'1993-05-18','2014-02-17','前台',4000,'重庆科技学院',600),(4,'李阳光','',21,'1993-02-22','2014-03-22','经理',7000,'重庆交通大学',900),(5,'钱坤','',24,'1993-01-21','2014-04-09','经理',7000,'重庆大学',1200),(6,'赵有钱','',30,'1989-10-10','2014-04-09','老板',15000,'清华大学毕业',NULL),(7,'钱二货','',46,'1976-07-07','2015-03-30','员工',8000,'北京大学毕业',1300),(8,'孙三傻','',26,'1990-03-05','2015-04-20','经理',15000,'重庆理工大学毕业',1000),(9,'李四楞','\0',20,'1996-04-27','2015-04-25','经理',15000,'重庆交通大学毕业',1200),(10,'周五王','',21,'1995-03-10','2016-01-20','员工',15000,'西南交通大学毕业',500),(11,'赵亮光','',22,'1994-02-21','2012-06-15','经理',5000,'有前途的人',2000),(12,'李逵','',23,'1995-04-15','2012-07-20','员工',3000,'虽然我人黑, 但是我牙白啊',1000),(13,'孙姬妃','\0',24,'1992-06-09','2012-08-21','经理',4500,'客官上来玩啊',2000),(14,'王维','\0',20,'1996-10-28','2016-02-12','员工',2500,'我要飞的更高',500),(15,'李狗蛋','',30,'1986-03-06','2006-07-10','员工',8000,'我是老实人',5000),(16,'赵钱多','',30,'1986-10-10','2015-03-20','经理',12000,'清华大学',1000),(17,'孙李','',30,'1986-10-10','2015-03-20','员工',9000,'重庆工商',100),(18,'周吴','',30,'1986-10-10','2015-03-20','前台',9000,'重庆理工',300),(19,'郑王','',30,'1986-10-10','2015-03-20','员工',10000,'家里蹲',500),(20,'冯陈楚','\0',30,'1986-10-10','2015-03-20','员工',4000,'波大',1000),(21,'周云天','',21,'1995-02-01','2015-04-13','员工',5000,'重庆大学',150),(22,'吴三桂','',24,'1992-06-01','2015-02-01','主管',3000,'天津大学',200),(23,'郑先','',20,'1996-01-30','2015-07-23','员工',4500,'西南大学',300),(24,'王娅','',24,'1992-06-02','2015-09-16','前台',6000,'广州大学',200),(25,'李云','',22,'1994-12-01','2015-01-11','员工',5000,'海南大学',300),(26,'张三','',24,'1992-05-12','2013-05-12','经理',7000,'重庆理工大学',800),(27,'李四','',23,'1993-03-18','2013-08-08','经理',6500,'重庆工商大学',600),(28,'王五','',23,'1993-05-15','2014-02-12','员工',4500,'重庆科技学院',500),(29,'赵六','',21,'1995-02-22','2014-03-03','员工',4000,'重庆交通大学',400),(30,'孙七','',22,'1994-06-06','2014-04-04','员工',3500,'重庆大学',300);

模糊查询

使用like关键字

% 匹配所有

\_ 匹配单个字符

查询所有姓赵的员工

select \* from employees where name like ‘%赵%’;

查询员工姓名中带有”三”的员工信息

select \* from employees where name like ‘%三%’;

查询所有姓周的并且姓名是两个字的员工信息

select \* from employees where name like '周\_';

查询员工信息, 已知其姓名的第二个字是”三”;

select \* from employees where name like ‘\_三%’;

对查询结果进行排序

order by

asc 升序排序 (默认)

desc 降序排序

对所有员工的年龄进行降序排序

select name, age from employees order by age desc;

对所有员工的年龄进行降序排序, 并且去除年龄的重复

select distinct age from employees order by age desc;

拓展: 如果一个数和null进行了运算, 结果会怎么样???

按照员工的年薪进行降序排序

select name,(salary+bonus)\*12 年薪 from employees order by 年薪 desc; (这种方式不行)

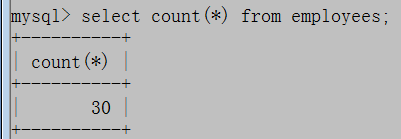
重点注意: ifnull()函数的使用

select name,(ifnull(salary,0)+ifnull(bonus,0))\*12 年薪 from employees order by 年薪 desc;

一些常用的函数

count统计函数 当统计的列中含有null值, 则不被统计在内

一般使用select count(\*) 来统计表中有多上条数据



统计一下在公司中有多少姓赵的员工

select count(\*) from employees where name like ‘赵%’;

SUM函数 用来求和

统计公司一个月需要发放多少薪资

select sum(ifnull(salary,0)+ifnull(bonus,0)) 人力成本 from employees;

AVG函数 求平均数

查询公司中一个月的员工平均工资

select avg(ifnull(salary,0)+ ifnull(bonus,0)) from employees;

MAX函数 求最大值

MIN函数 求最小值

GROUP BY 分组查询

对于公司员工按照职位分组查询, 并且求出每一类岗位工作人员的人数和总工资.

select position 职位, count(position) 人数, sum(ifnull(salary,0)+ifnull(bonus,0)) 总工资 from employees group by 职位;

对公司员工按照职位进行分类, 并且求每一类岗位工作人员的人数和总工资, 并且总工资降序显示.

select position 职位, count(position) 人数, sum(ifnull(salary,0)+ifnull(bonus,0)) 总工资 from employees group by 职位 order by 总工资 desc;

对公司员工职位进行分类查询, 并且求出总工资大于20000的每一类工作岗位的员工总工资. 并且降序显示.

select position 职位, count(position) 人数, sum(ifnull(salary,0)+ifnull(bonus,0)) 总工资 from employees group by 职位 having 总工资>20000 order by 总工资 desc;

查询每一种职位上年龄在25以上的员工个数..

select position, count(position) from employees where age > 25 group by position;

select position, count(position) from employees group by position having age > 25;

对公司员工按照职位分类, 并且要求员工年龄都在25岁以上, 并且要求总工资大于20000的每一类工作岗位的员工工资, 并且按照工资的降序排序.

select position 职位, sum(ifnull(salary,0)+ifnull(bonus,0)) 总工资 from employees where age > 25 group by 职位 having 总工资>20000 order by 总工资 desc;

分页查询 使用 limit关键字

使用的方式 limit 第一个参数, 第二个参数;

select \* from employees limit 0, 3;

select \* from employees limit 1, 3;

select \* from employees limit 2, 3;

select \* from employees limit 3, 3;

select \* from employees limit 3, 4;

select \* from employees limit 3, 5;

select \* from employees limit 3, 6;