net start mysql 启动MySQL服务

net stop mysql 停止MySQL服务

mysql -uroot -p -P3306 -h127.0.0.1 登陆MySQL

或mysql -uroot 直接登陆

exit || quit || \q 退出MySQL

shell>cls 清屏

shell>mysql -uroot -proot --prompt \h 修改mysql提示符 提示符由mysql> 变为localhost

localhost(或原mysql>)prompt mysql> 修改提示符 由localhost 变为mysql>

用\h变为localhost 当前服务器名称 用\D 变为当前完整的日期时间 用\d变为当前数据库 用\u变为当前用户

输入prompt \u@\h \d> 显示为root@localhost (none)>

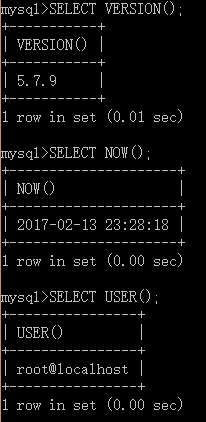
mysql>USE test; 打开系统的test数据库

常用命令：

显示当前服务器版本：SELECT VERSION();

显示当前日期时间：SELECT NOW();

显示当前用户：SELECT USER();



MySQL语句的规范

关键字与函数名称全部大写

数据库名称、表名称、字段名称全部小写

SQL语句必须以分号结尾

(大写小写是为了迅速而明显地区分出关键字、函数、数据库、表、字段等)

创建数据库

CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db\_name

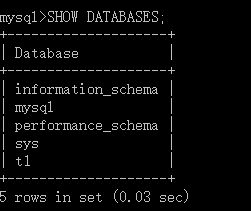
[DEFAULT] CHARACTER SET [=] charset\_name （本句表示创建数据库时所用的编码方式） 花括号表必选项 |表二者（或更多）选其一 中括号表可选项 如CREATE DATABASE t1;

IMG_257

查看当前服务器下的数据表列表

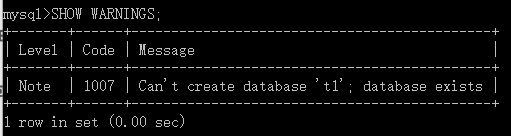
SHOW {DATABASES | SCHEMAS}

[LIKE 'pattern' | WHERE expr]



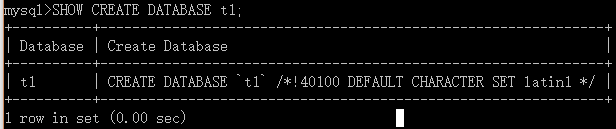
查看警告信息

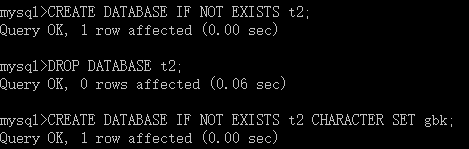
SHOW WARNINGS;



查看数据库创建时所使用的指令

SHOW CREATE DATABASE t1; （t1为自己起的数据库名称）

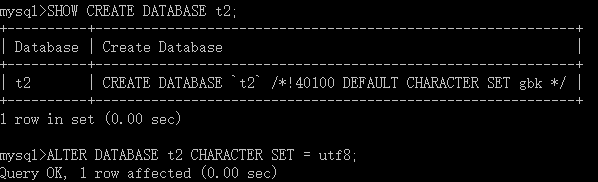




修改数据库（编码方式）

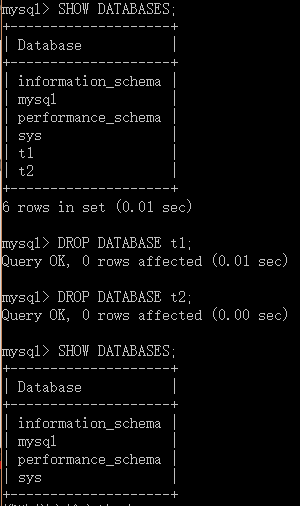
ALTER {DATABASE | SCHEMA} [db\_name]

[DEFAULT] CHARACTER SET [=] charset\_name



删除数据库

DROP {DATABASE | SCHEMA} [IF NOT EXISTS] db\_name(即database\_name)





MySQL默认端口号：3306

MySQL中的超级用户：root

数据类型：是指列、存储过程参数、表达式和局部变量的数据特征，它决定了数据的存储格式，代表了不同的信息类型。

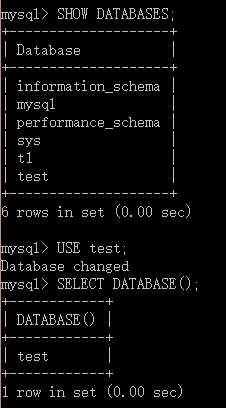
MySQL数据类型：<http://www.runoob.com/mysql/mysql-data-types.html>

数据表（或称表）是数据库最重要的组成部分之一，是其他对象的基础。

关系型数据库：二维表格，行为记录，列为字段。

用USE打开数据库，USE+数据库名称；

SELECT DATABASE();显示当前打开的数据库



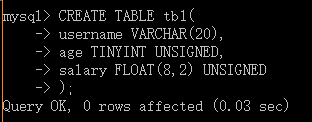
创建数据表

CREATE TABLE [IF NOT EXISTS] table\_name(

column\_name data\_type,

...

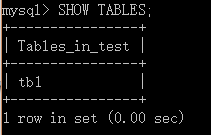
)

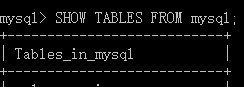


查看数据表列表

SHOW TABLES [FROM db\_name]

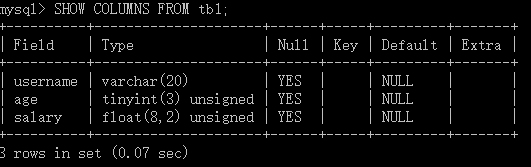
[LIKE 'pattern' | WHERE expr]





查看数据表结构

SHOW COLUMNS FROM tbl\_name

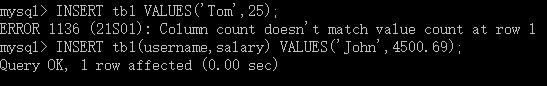


插入记录

INSERT [INTO] tbl\_name [(col\_name,...)] VALUES(val,...)

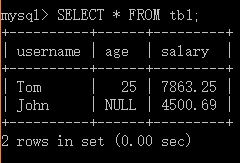
当省略列名或字段名，则所有的字段都要赋值

IMG_270



记录查找

SELECT expr,... FROM tb1\_name (expr 表示表达式)

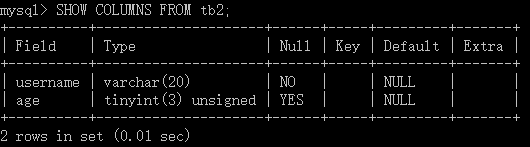


空值与非空

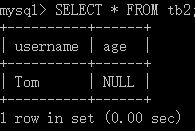
NULL 字段值可以为空

NOT NULL 字段值禁止为空





IMG_275



IMG_277

AUTO\_INCREMENT

自动编号，且必须与主键组合使用

默认情况下，起始值为1，每次的增量为1

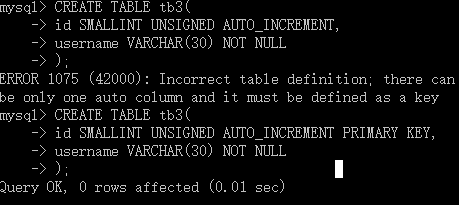
PRIMARY KEY

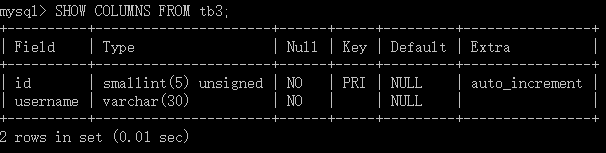
主键约束（主键约束的字段自动禁止为空）

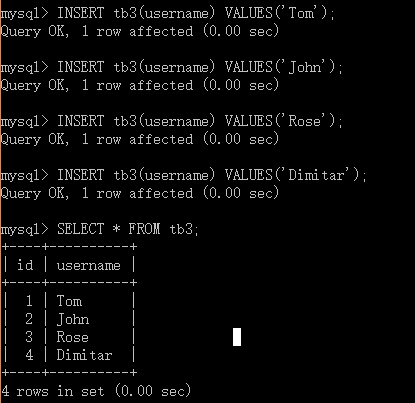
每张数据表只能存在一个主键

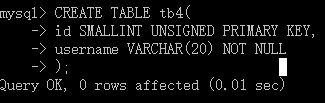
主键保证记录的唯一性

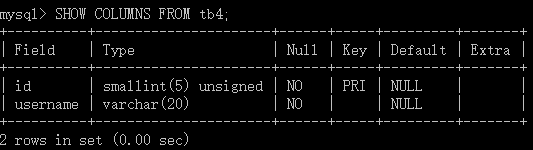
主键自动为NOT NULL









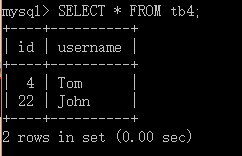


有主键PRIMARY KEY不一定要有AUTO\_INCREMENT

但有AUTO\_INCREMENT一定要有PRIMARY KEY

IMG_283

IMG_284



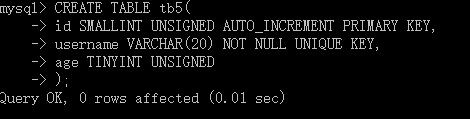
IMG_286

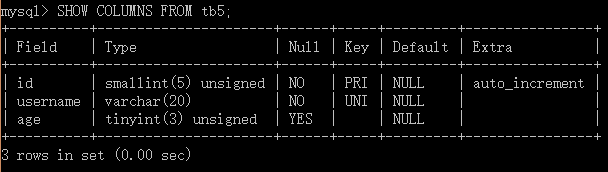
UNIQUE KEY 唯一约束

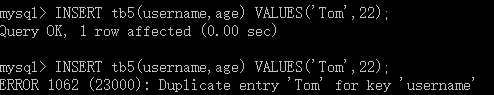
唯一约束可以保证记录的唯一性

唯一约束的字段可以为空值（NULL）

每张数据表可以存在多个唯一约束

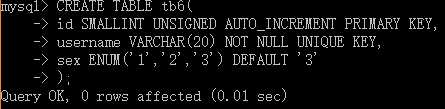


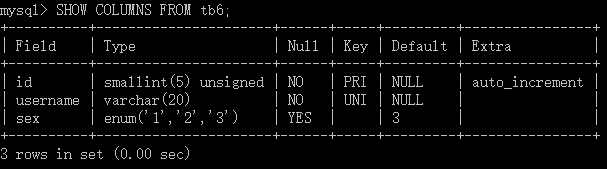




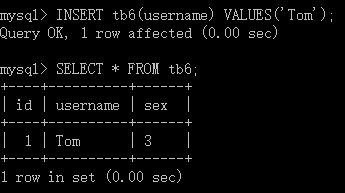
DEFAULT 默认值

当插入记录时，如果没有明确为字段赋值，则自动赋予默认值





ENUM--枚举



数据类型：字符型；整型；浮点型；日期时间型

数据表操作：插入记录；查找记录

记录操作：创建数据表；约束的使用

约束类型：主键约束、唯一约束、默认约束、非空约束、外键约束

约束保证数据的完整性和一致性

约束分为表级约束和列级约束

FOREIGN KEY

保持数据一致性，完整性。实现一对一或一对多关系

外键约束的要求：父表和字表必须使用相同的存储引擎，而且禁止使用临时表。数据表的存储引擎只能为InnoDB

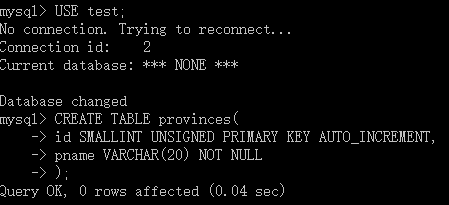
外键列和参照列必须具有相似的数据类型，其中数字的长度或是否有符号位必须相同；而字符的长度则可以不同

外键列和参照列必须创建索引。如果外键列不存在索引的话，MySQL将自动创建索引

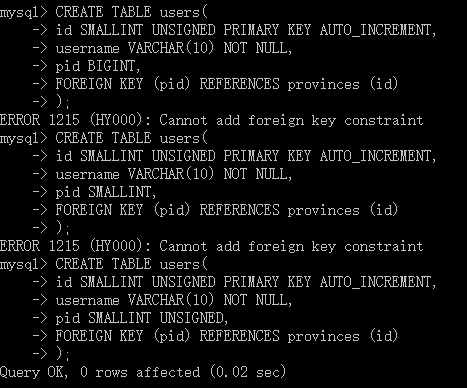
编辑数据表的默认存储引擎

MySQL配置文件

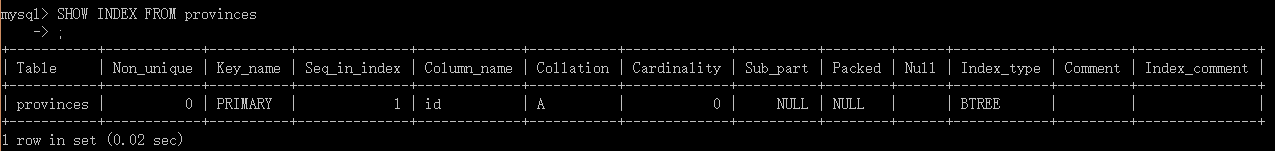
default-storage-engine=INNODB

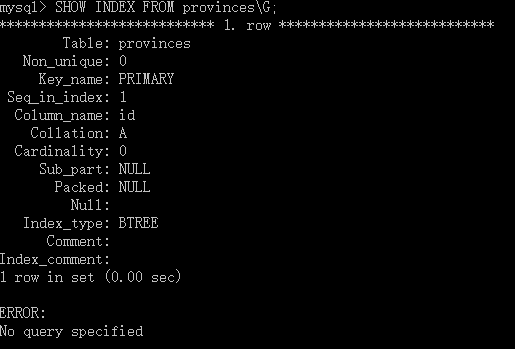


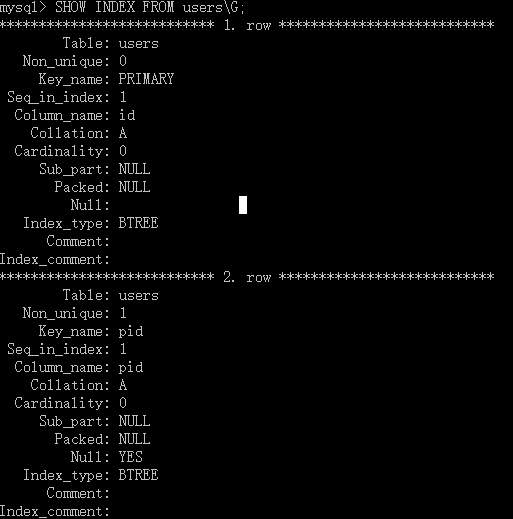


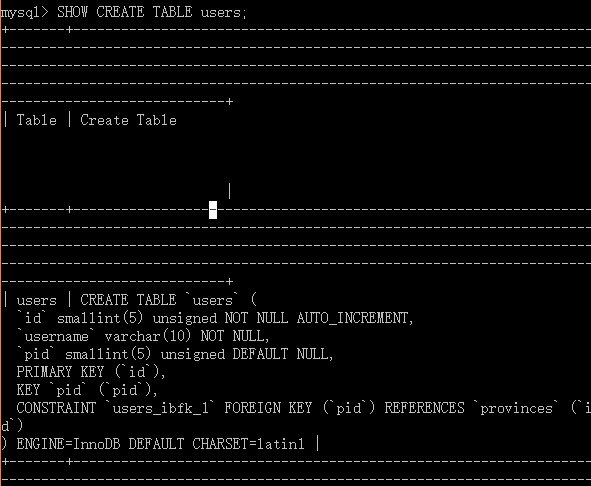


其中users为字表，provinces为父表。pid为外键列，id为参照列。











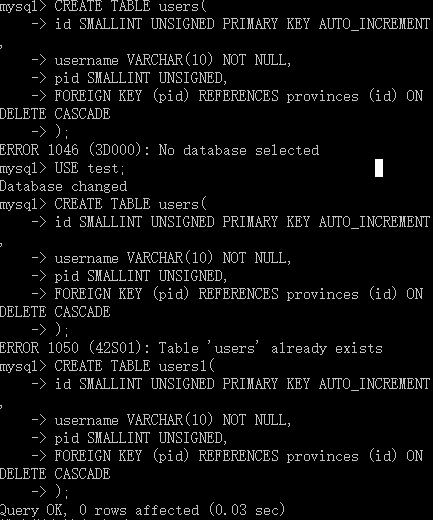
外键约束的参照操作

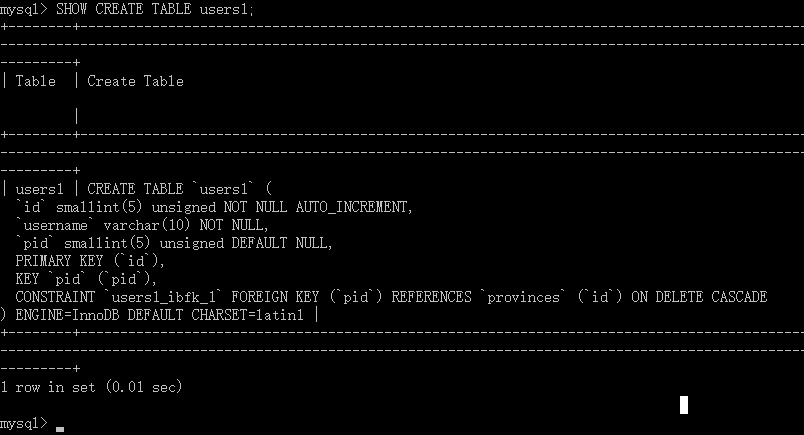
CASCADE：从父表删除或更新且自动删除或更新字表中匹配的行

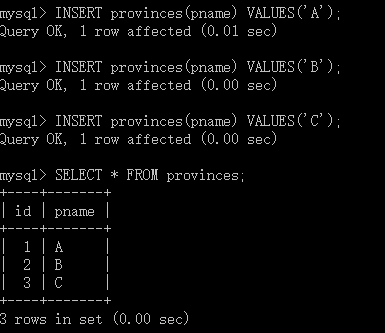
SET NULL ：从父表删除或更新行，并设置字表中的外键列为NULL。如果使用该选项，必须保证字表列没有指定NOT NULL

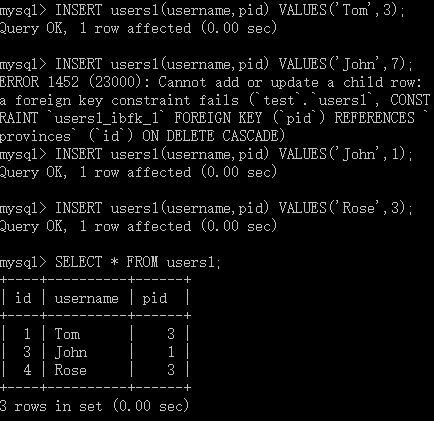
RESTRICT:拒绝对父表的删除或更新操作

NO ACTION: 标准SQL的关键字，在MySQL中与RESTRICT相同

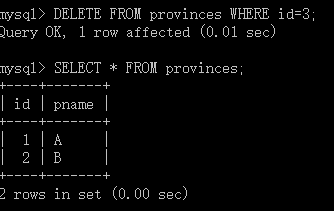


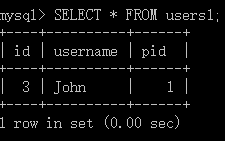






删除的命令：DELETE FROM provinces WHERE id=3;





从父表删除或更新记录，则自动删除或更新字表中匹配的行。

表级约束与列级约束

对一个数据列建立的约束，称为列级约束；

对多个数据列建立的约束，称为表级约束；

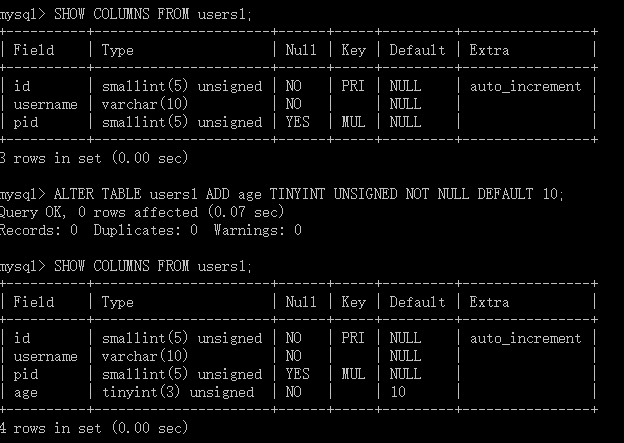
列级约束既可以在列定义时声明，也可以在列定义后声明。

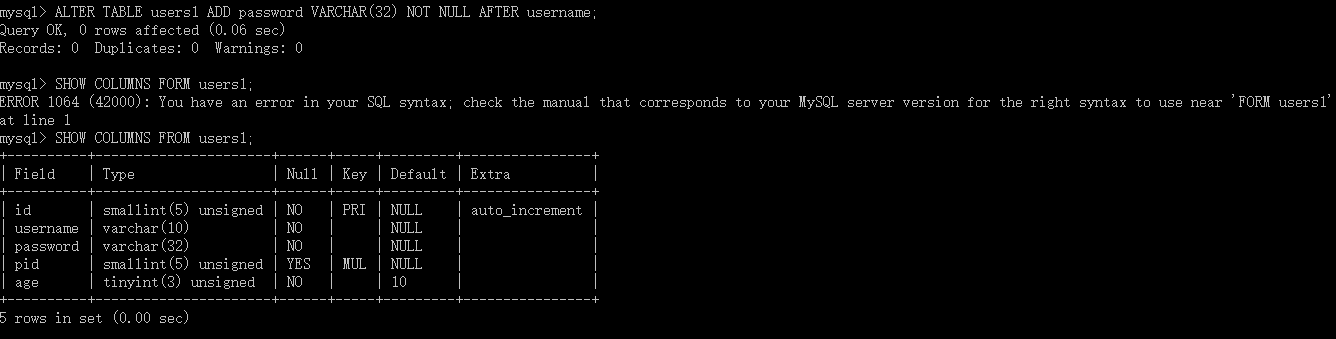
表级约束只能在列定义后声明。

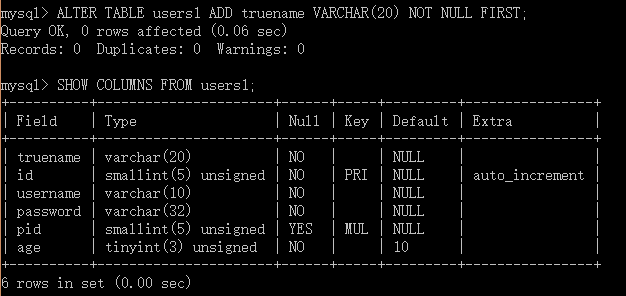
修改数据表

添加单列

ALTER TABLE tbl\_name ADD [COLUMN] col\_name column\_definition [FIRST | AFTER col\_name]







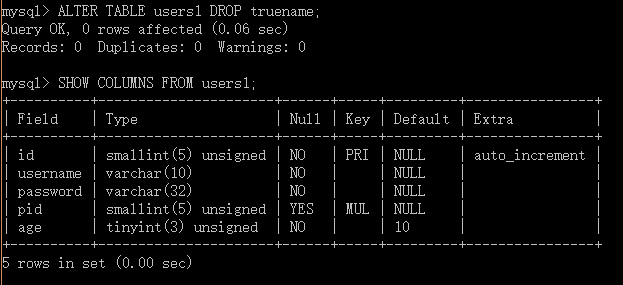
修改数据表

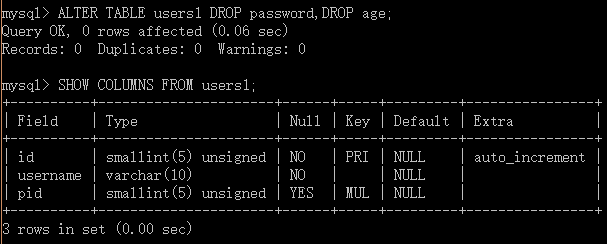
添加多列

ALTER TABLE tbl\_name ADD [COLUMN] (col\_name column\_definition,...)

删除列

ALTER TABLE tbl\_name DROP [COLUMN] col\_name

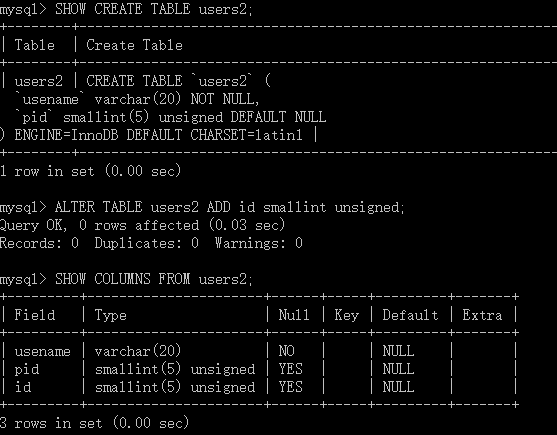


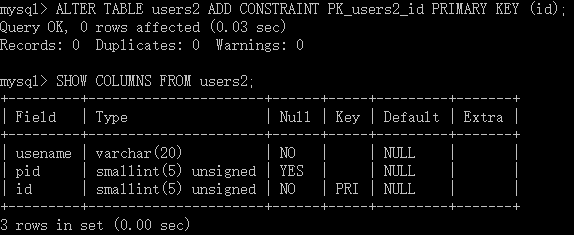


修改数据表

添加主键约束

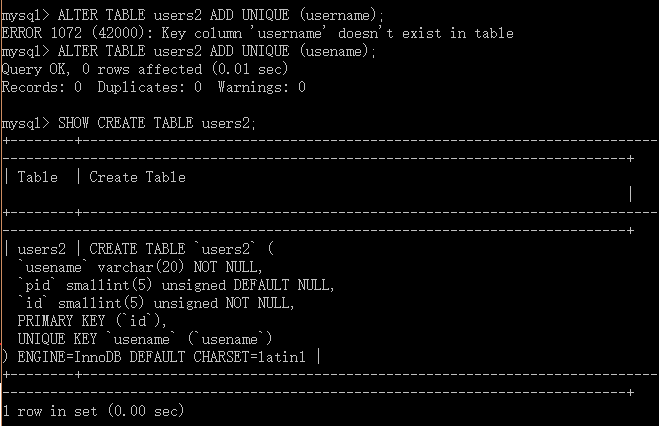
ALTER TABLE tbl\_name ADD [CONSTRAINT [symbol]] PRIMARY KEY [index\_type] (index\_col\_name,...)





添加唯一约束

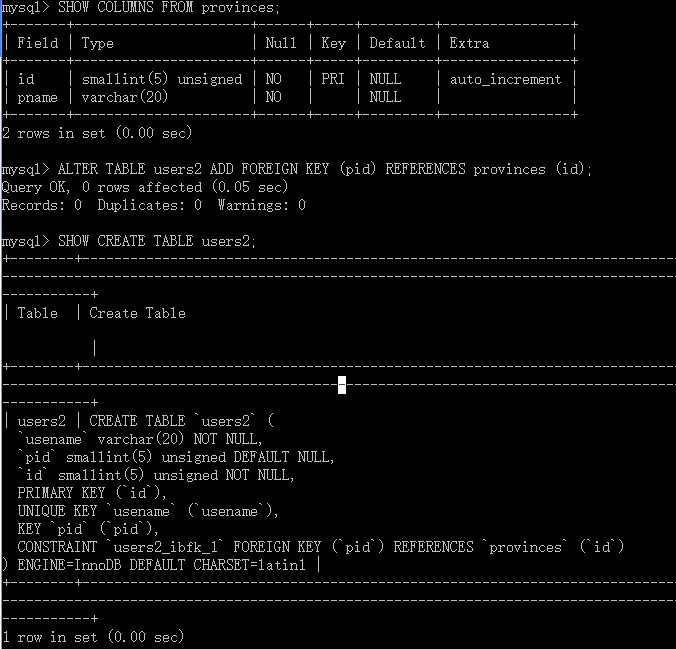
ALTER TABLE tbl\_name ADD [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY] [index\_name] [index\_type] (index\_col\_name,...)



修改数据表

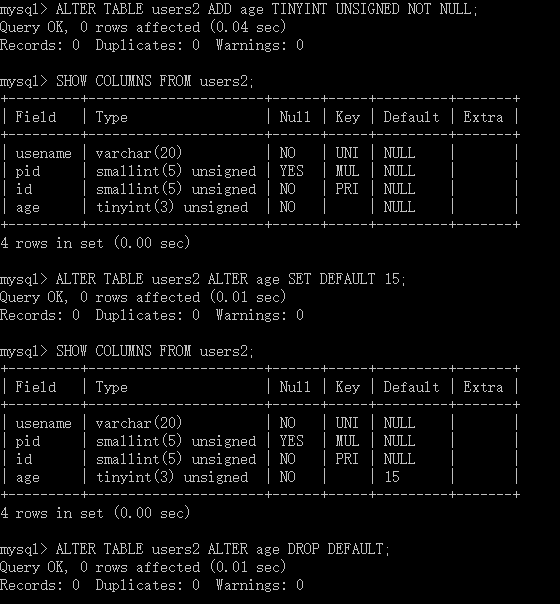
添加外键约束

ALTER TABLE tbl\_name ADD [CONSTRAINT [symbol]] FOREIGN KEY [index\_name] (index\_col\_name,...) reference\_definition



添加/删除默认约束

ALTER TABLE tbl\_name ALTER [COLUMNS] col\_name {SET DEFAULT literal | DROP DEFAULT}

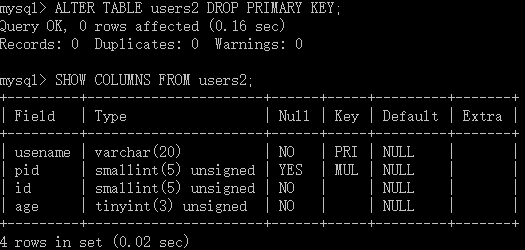


修改数据表

删除主键约束

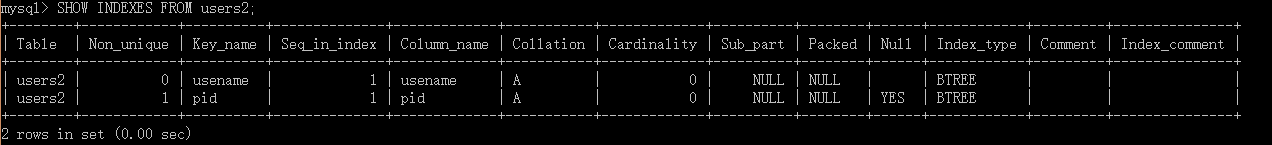
ALTER TABLE tbl\_name DROP PRIMARY KEY

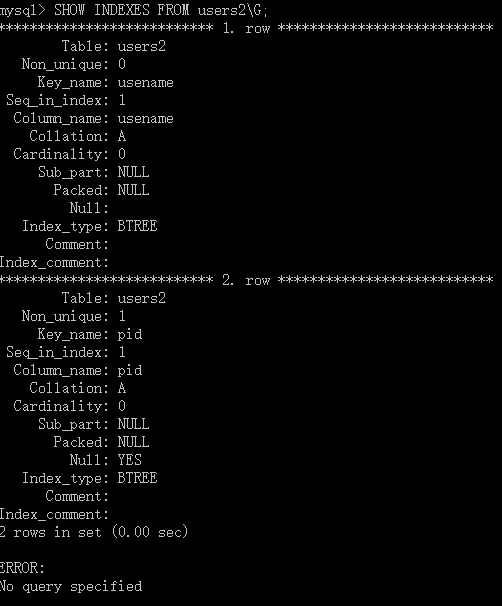
(任何一张数据表有且只有一个主键)

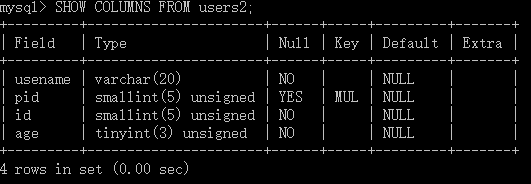


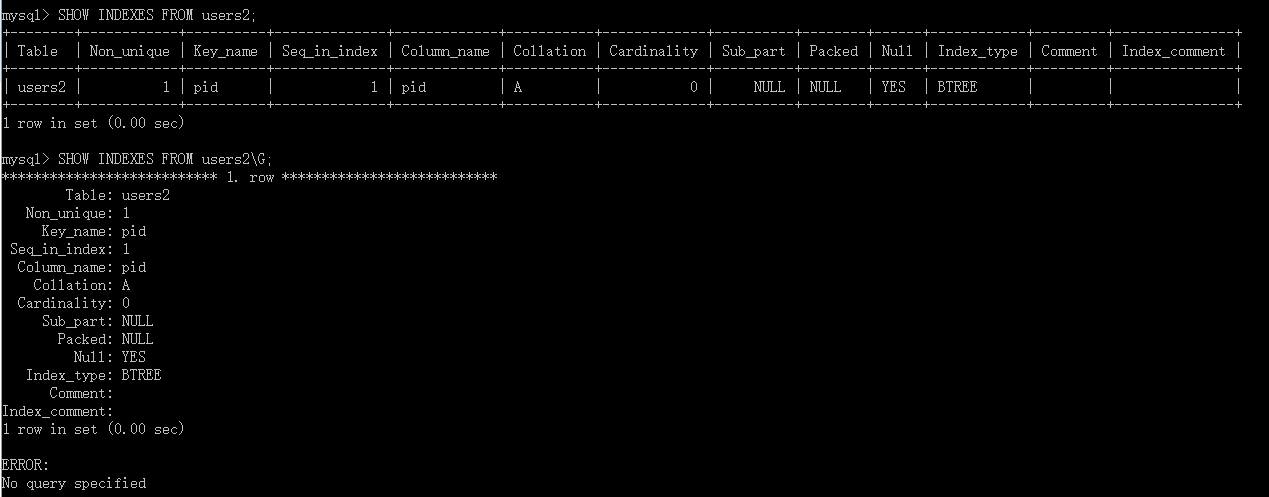
删除唯一约束

ALTER TABLE tbl\_name DROP {INDEX|KEY} index\_name



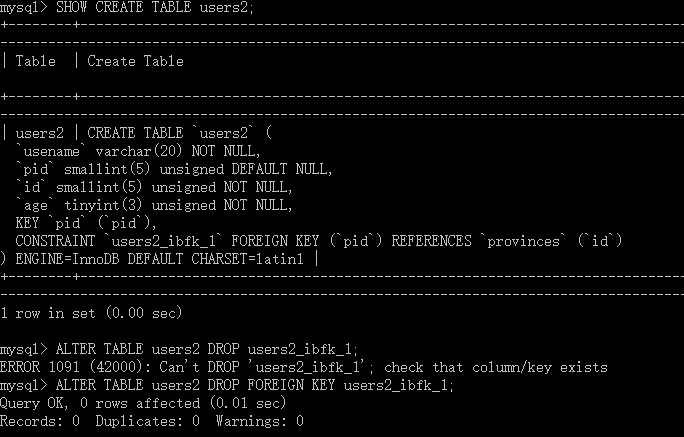


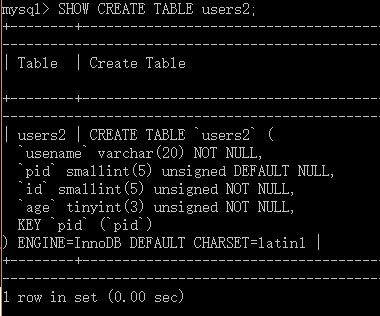


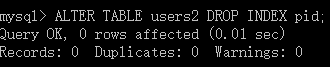


删除外键约束

ALTER TABLE tbl\_name DROP FOREIGN KEY fk\_symbol



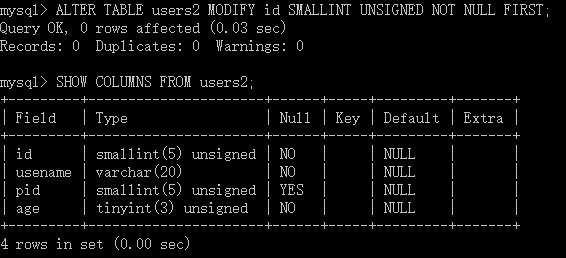


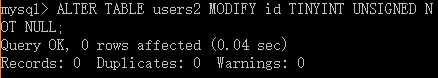


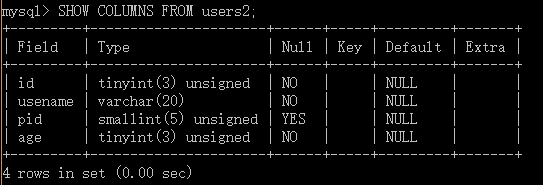
修改数据表

修改列定义

ALTER TABLE tbl\_name MODIFY [COLUMN] col\_name column\_definition [FIRST | AFTER col\_name]



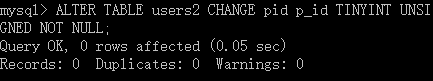


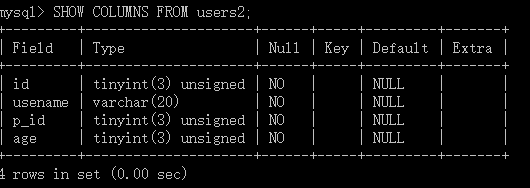


由大类型改为小类型，有可能会造成数据的丢失

修改列名称

ALTER TABLE tbl\_name CHANGES [COLUMNS] old\_col\_name new\_col\_name column\_definition [FIRST|AFTER col\_name]

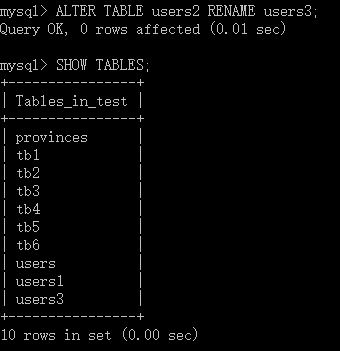


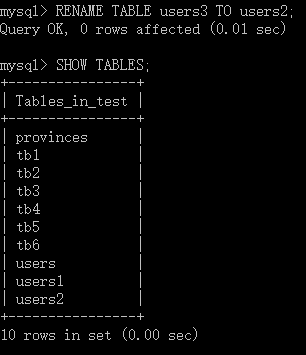


数据表更名

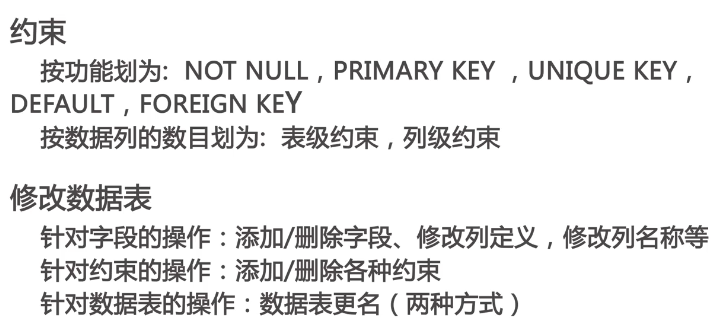
方法一：ALTER TABLE tbl\_name RENAME [TO|AS] new\_tbl\_name

方法二：RENAME TABLE tbl\_name TO new\_tbl\_name [,tbl\_name2 TO new\_tbl\_name2]...





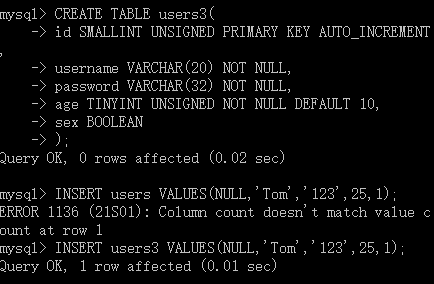
尽量不去随意更改数据列的名字以及数据表的名字



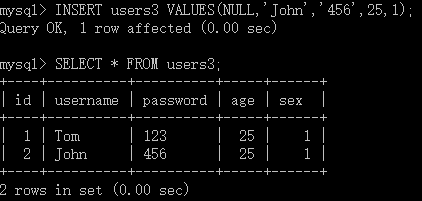
插入记录(INSERT)

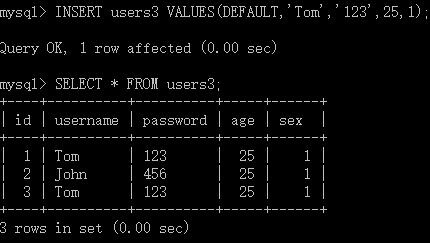
INSERT [INTO] tbl\_name [(col\_name,...)] {VALUES | VALUE} ({expr | DEFAULT},...),(...),...

(可一次性插入多条记录)

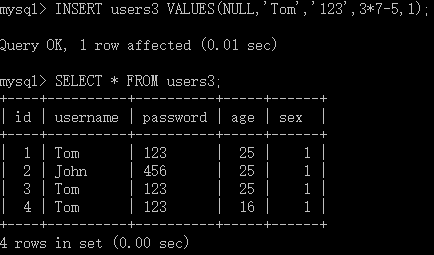


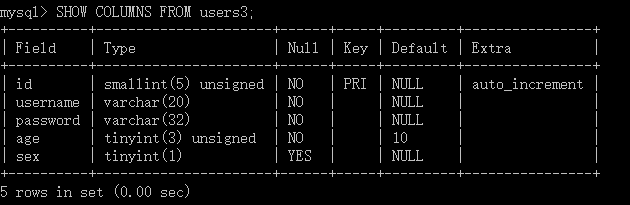


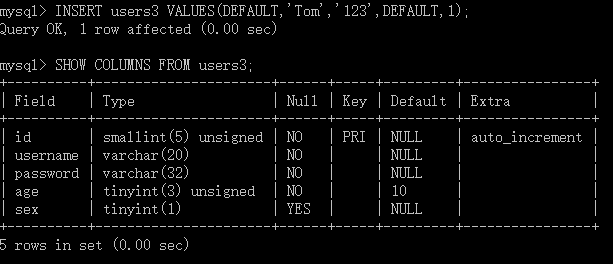


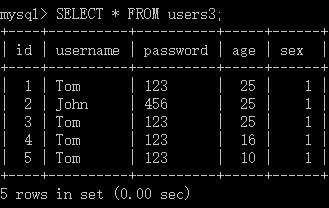


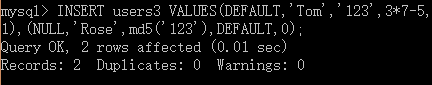
IMG_337

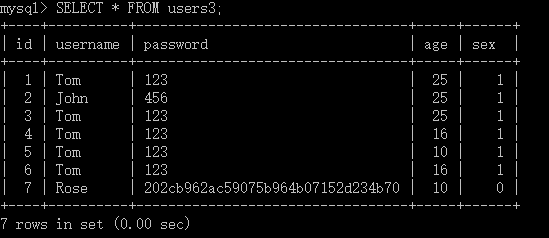








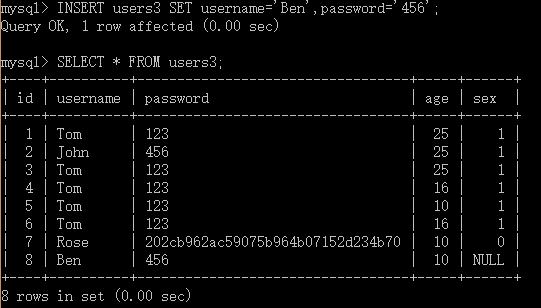




插入记录

INSERT [INTO] tbl\_name SET col\_name={expr | DEFAULT},..

(与第一种方式的区别在于，此方法可以使用子查询（SubQuery）)

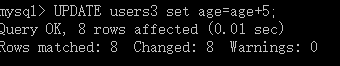


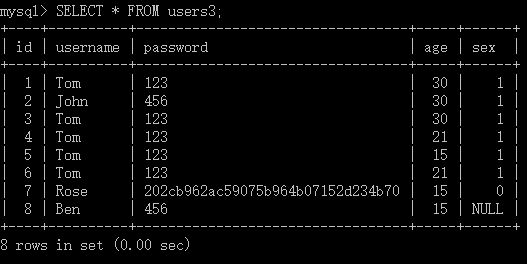
插入记录

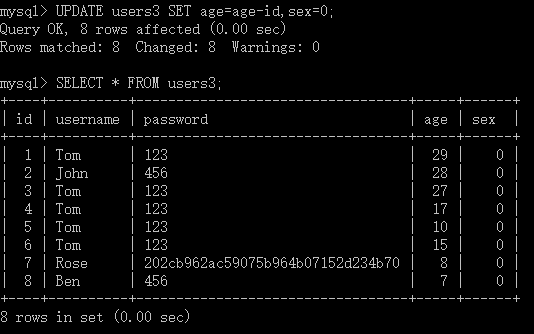
INSERT [INTO] tbl\_name [(col\_name,...)] SELECT... 此方法可以将查询结果插入到指定数据表

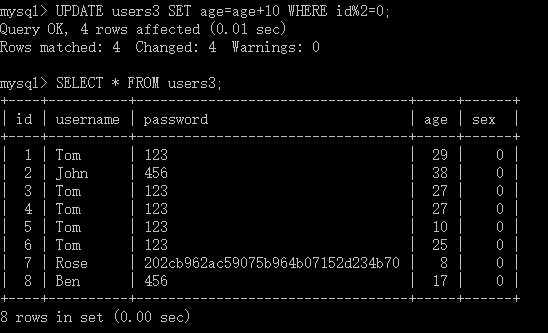
更新记录（单表更新）

UPDATE [LOW\_PRIORITY] [IGNORE] table\_reference SET col\_name1={expr1|DEFAULT}[,col\_name2={expr2|DEFAULT}]...[WHERE where\_condition]



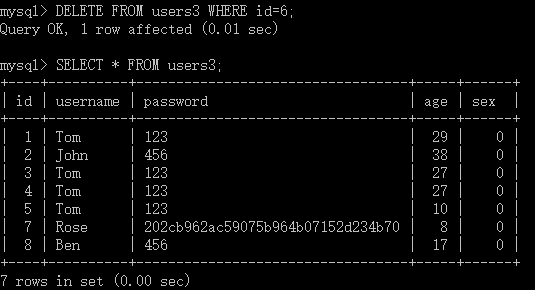






删除记录（单表删除）

DELETE FROM tbl\_name [WHERE where\_condition]





查找记录

SELECT select\_expr [,select\_expr ...]

[

FROM table\_references

[WHERE where\_condition]

[GROUP BY {col\_name | position} [ASC | DESC],...]

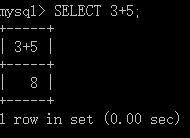
[HAVING where\_condition]

[ORDER BY {col\_name | expr |position} [ASC |DESC],...]

[LIMIT {[offset,] row\_count | row\_count OFFSET offset}]

]





查询表达式

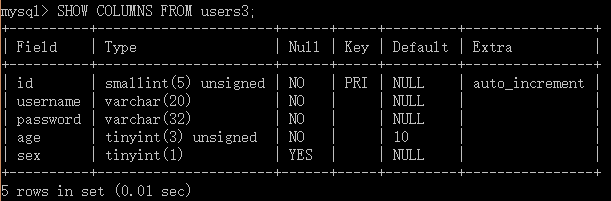
每一个表达式表示想要的一列，必须有至少一个

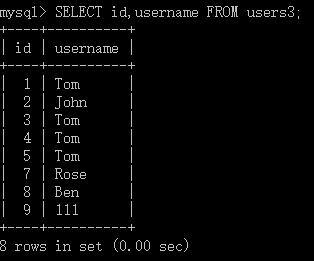
多个列之间以英文逗号分隔

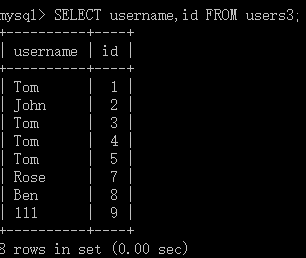
星号（\*）表示所有列。tbl\_name.\*可以表示命名表的所有列

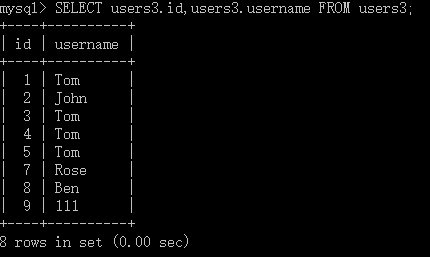
查询表达式可以使用[AS] alias\_name 为其赋值别名

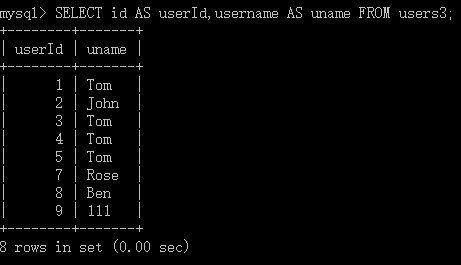
别名可用于GROUP BY, ORDER BY 或 HAVING 子句。

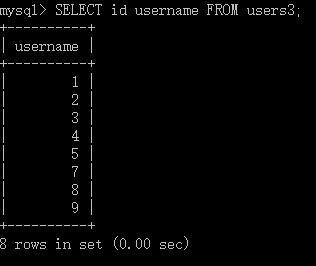


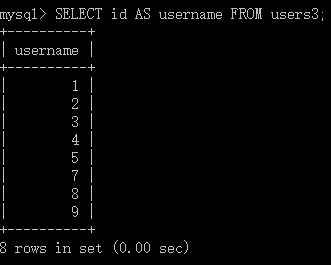












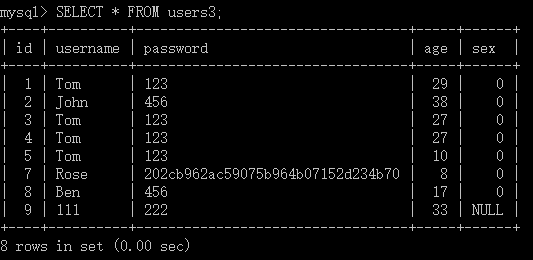
WHERE 条件表达式

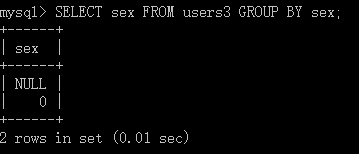
对记录进行过滤，如果没有指定WHERE 子句，则显示所有记录。

在WHERE表达式中，可以使用MySQL支持的函数或运算符。

查询结果分组

[GROUP BY {col\_name | position} [ASC | DESC],...] 其中ASC | DESC 分别表示升序和降序



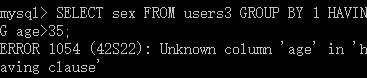


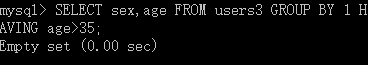


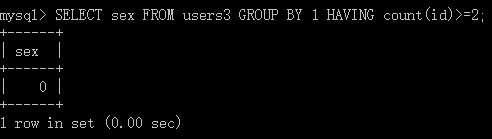
1 是指SELECT 语句中第一个出现的字段

分组条件（HAVING）

[HAVING where\_condition]

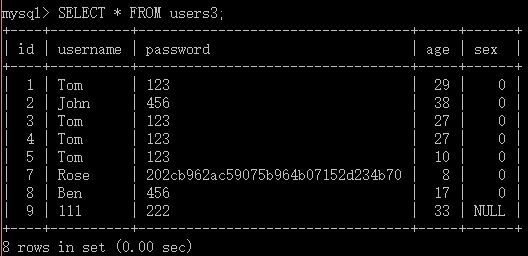


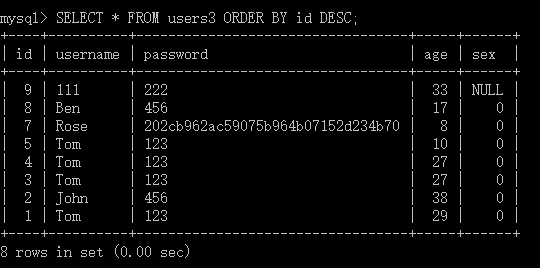


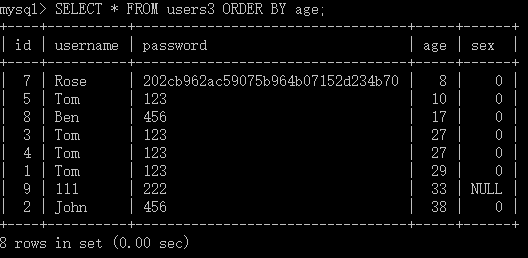


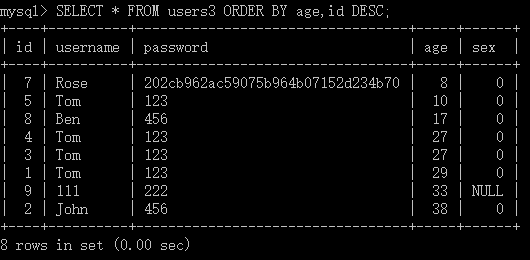
对查询结果进行排序

[ORDER BY {col\_name | expr | position} [ASC | DESC],...]



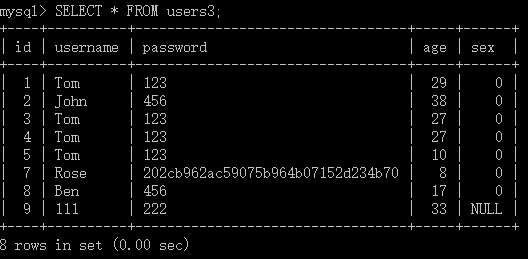


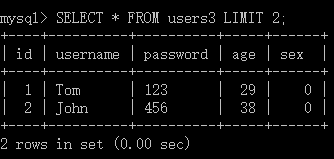


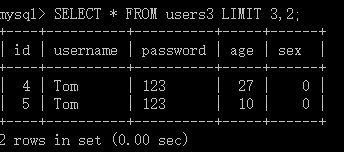


限制查询结果返回的数量

[LIMIT {[offset,] row\_count | row\_count OFFSET offset}]







SELECT语句从0开始编号，因此3，2实际为从第四个开始选择两个（包括第四个）

