# 第一章．初涉MySQL

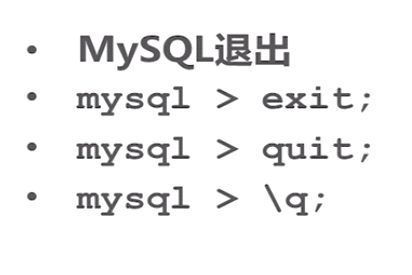


## MySQL登录：

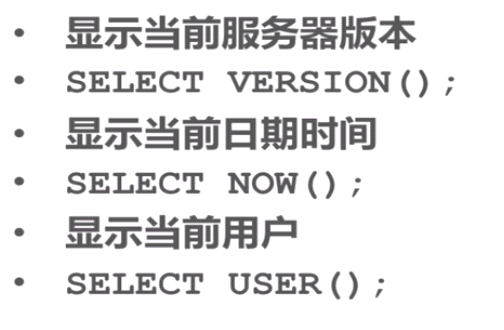
C:\Users\ai>mysql -uroot -p -P3306 -h127.0.0.1

Enter password: \*\*\*\*\*\*\*\*\*

MySQL退出：



## MySQL常用命令：

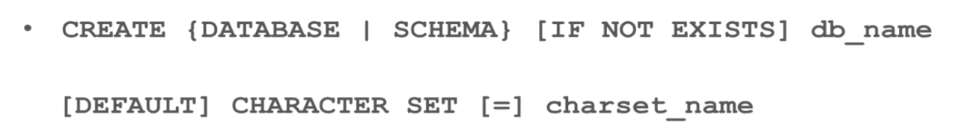


关键字与函数名称全部大写

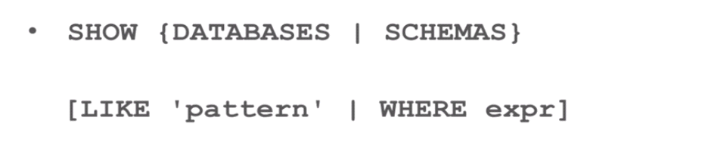
数据库名称、表名称、字段名称全部小写

SQL语句必须以分号结束“;”

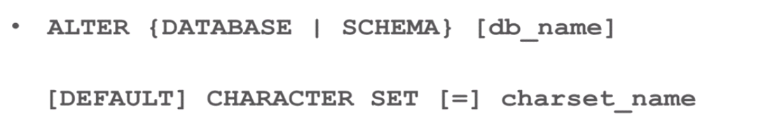
## 创建数据库：



## 查看当前服务器下的所有数据库：



## 修改数据库：



## 删除数据库：





# 第二章．数据类型与操作数据库

## 数据类型

### 整型



数据优化时选择最合理最合适的数据类型存储。

### 浮点型



### 日期时间型



### 字符型



## 数据表的操作

数据表（表）时数据库最重要的组成成员之一，是其他对象的基础。

### 登录到数据库：

C:\Users\ai>mysql -u root -p

Enter password: \*\*\*\*\*\*\*\*\*

### 创建数据库：

CREATE DATABASE test;

### 查看已存在的数据库：

SHOW DATABASES;

### 打开数据库：

USE test;

### 查看当前已打卡的数据库：

SELECT DATABASE();

### 创建表：

mysql> CREATE TABLE tb1(

-> username VARCHAR(20),

-> age TINYINT UNSIGNED,

-> salary FLOAT(8,2) UNSIGNED

-> );

Query OK, 0 rows affected (0.43 sec)

### 查看（指定）数据库下的所有表：

查看当前已打开数据库下的表：mysql> SHOW TABLES;

查看指定数据库下的表：mysql> SHOW TABLES FROM mysql;

### 查看数据表结构：

mysql> SHOW COLUMNS FROM tb1;

mysql> desc tb\_samples;

…………..

3 rows in set (0.08 sec)

查看表创建命令

### 向数据表中插入记录

省略列名时默认是（必须）给所有列都插入值

mysql> INSERT tb1 VALUES('Tom',25,7894.25);

Query OK, 1 row affected (0.07 sec)

指定要插入到哪些列中

mysql> INSERT tb1 (username,salary) VALUES('John',4500.23);

Query OK, 1 row affected (0.11 sec)

### 简单记录查找（验证插入命令）

mysql> SELECT \* FROM tb1;

### 空值与非空

NULL：字段值可以为空

NOT NULL：字段值不能为空

### 主键

PRIMARY KEY

主键约束

每张数据表只能存在一个主键

主键保存记录的唯一性

主键自动为NOT NULL

### 自动编号

AUTO\_INCREMENT

自动编号必须为**数值型**，若为浮点数小数后应为0

自动编号应与主键组合使用（但主键不一定有此属性）

默认情况下起始值为1，每次增量为1

mysql> CREATE TABLE tb3(

-> id SMALLINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

-> username VARCHAR(30) NOT NULL

-> );

Query OK, 0 rows affected (0.22 sec)

插入姓名时主键id自动从1开始递增。当然主键也允许赋值但不能出现相同的值。

mysql> SELECT \* FROM tb3;

+----+--------------+

| id | username |

+----+--------------+

| 1 | oen |

| 2 | TWO |

| 3 | three |

+----+--------------+

3 rows in set (0.00 sec)

### 唯一约束

UNIQUE KEY

唯一约束

唯一约束可以保证记录的唯一性（类似主键）

唯一字段的值可以为空（NULL）--多个记录值可以同时为空

每张数据表可以存在多个唯一约束

mysql> CREATE TABLE tb4(

-> id SMALLINT AUTO\_INCREMENT PRIMARY KEY,

-> username VARCHAR(20) NOT NULL UNIQUE KEY,

-> age TINYINT UNSIGNED

-> );

Query OK, 0 rows affected (0.25 sec)

### 默认约束

DEFAULT

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。

mysql> CREATE TABLE tb5(

-> id SMALLINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

-> sex ENUM('1','2','3') DEFAULT '3'

-> );

Query OK, 0 rows affected (0.25 sec)

# 第三章．约束及修改数据表

## 约束

约束保证数据的完整性和一致性

约束分为表级约束（一个字段以上）和列级约束（仅一个字段）

约束类型包括：

非空约束：NOT NULL

主键约束：PRIMARY KEY

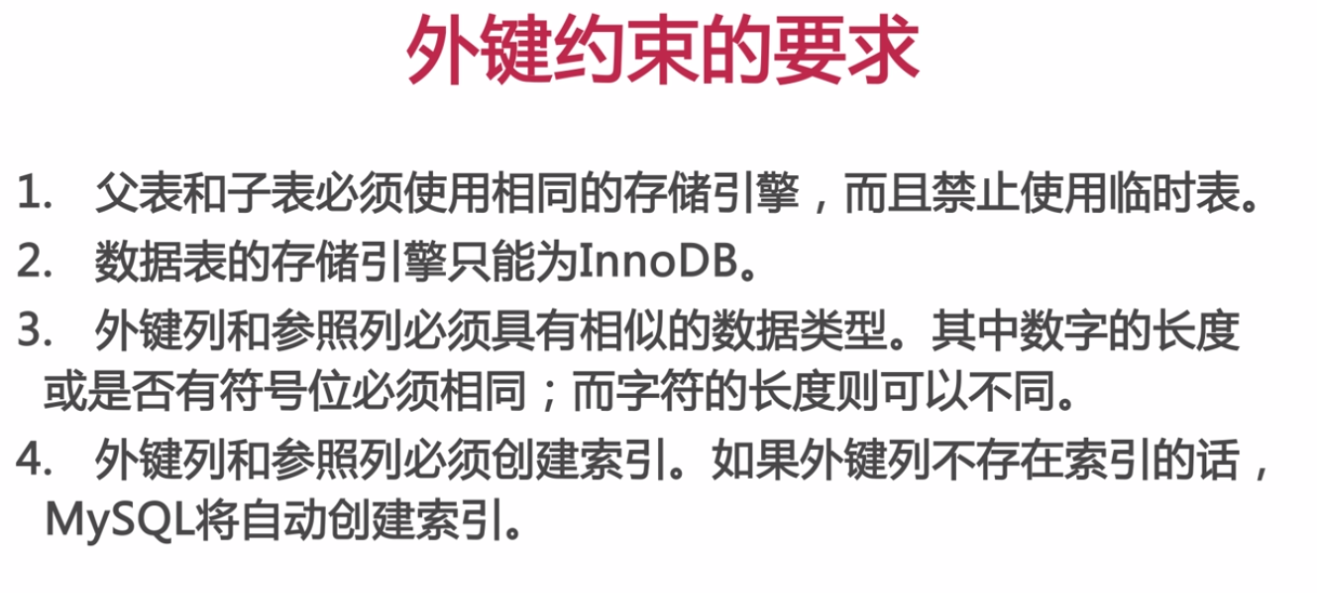
唯一约束：UNIQUE KEY

默认约束：DEFAULT

外键约束：FOREIGN KEY

### 外键约束

实现一对一或一对多关系



编辑数据表的默认存储引擎

配置文件：My.ini

编辑：default-storage-engine=INNODB

创建

mysql> CREATE TABLE provinces(

-> id SMALLINT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

-> pname VARCHAR(20) NOT NULL

-> );

Query OK, 0 rows affected (2.21 sec)

mysql> CREATE TABLE users(

-> username VARCHAR(10) NOT NULL,

-> pid SMALLINT UNSIGNED,

-> FOREIGN KEY (pid)REFERENCES provinces (id)

-> );

Query OK, 0 rows affected (0.34 sec)

pid对应的表users为子表，子表参照的表provinces为父表。

pid和id类型为整形所以必须相同

查看索引

mysql> SHOW INDEXES FROM users\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Table: users

Non\_unique: 1

Key\_name: pid

Seq\_in\_index: 1

Column\_name: pid

Collation: A

Cardinality: 0

Sub\_part: NULL

Packed: NULL

Null: YES

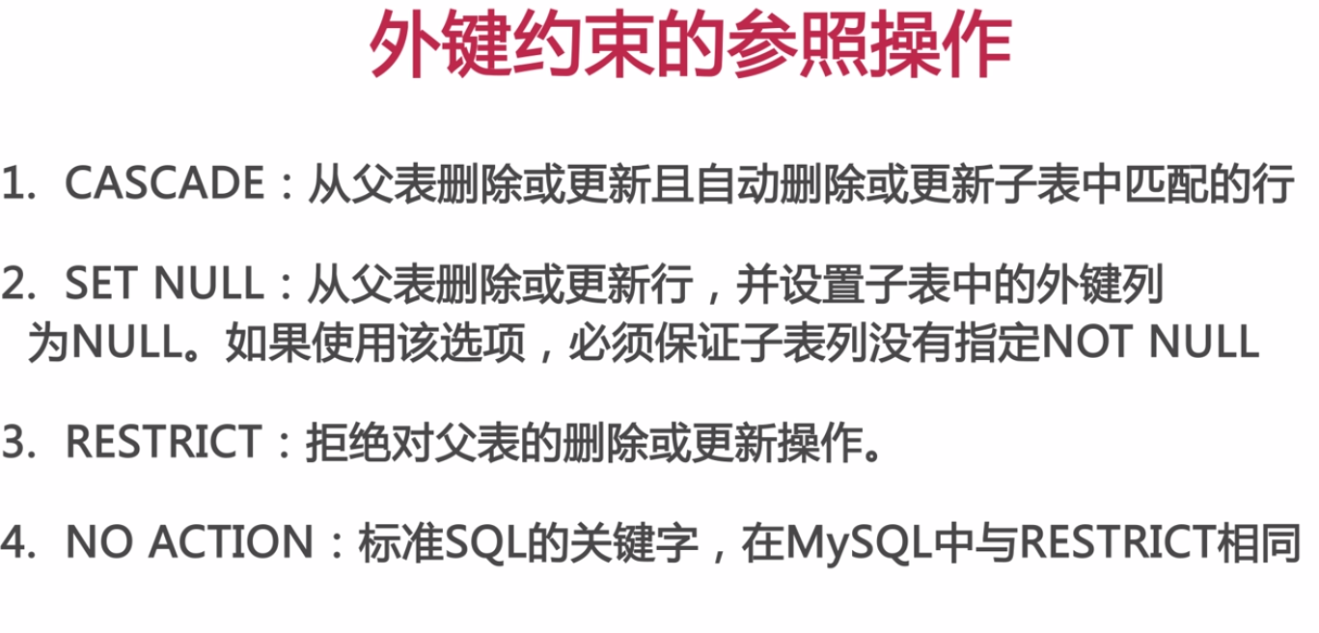
Index\_type: BTREE

Comment:

Index\_comment:

1 row in set (0.00 sec)

### 外键约束的参照条件



mysql> CREATE TABLE users1(

-> id SMALLINT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

-> username VARCHAR(20) NOT NULL,

-> pid SMALLINT UNSIGNED,

-> FOREIGN KEY (pid) REFERENCES provinces (id) **ON DELETE CASCADE**

-> );

Query OK, 0 rows affected (0.29 sec)

查看创建命令

mysql> SHOW CREATE TABLE users1;

在两张表中插入信息验证外键约束的查找条件CASCADE

1. 在两张表中插入信息（必须先在父表中插入）
2. 在父表中删除值，同时子表中对应的值也会被删除

### 表级约束和列级约束

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——ALTER…{ ADD | DROP}

### 添加列

1. 添加一列：

ALTER TABLE tb\_name **ADD** [CLOUMN] col\_name column\_definition [FIRST|AFTER col\_name]

mysql> ALTER TABLE users1 **ADD** age TINYINT UNSIGNED NOT NULL DEFAULT 10 **AFTER** username;

若不加FIRST或AFTER…默认添加到尾部

添加多列：

1. ALTER TABLE tb\_name ADD [COLUMN] （col\_name column\_definition,………）

**不能指定位置关系**，只能添加到数据表末尾。

### 删除列

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

mysql> ALTER TABLE users1 DROP username;

删除多列只需在DROP后添加多个列名用逗号分隔

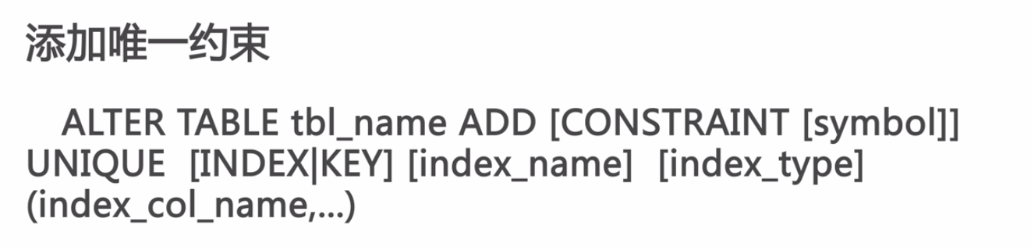
### 添加主键约束

ALTER TABLE tb\_name ADD [CONSTRAINT [symbol]] PRIMARY EKY [index\_type] (index\_col\_name,…)

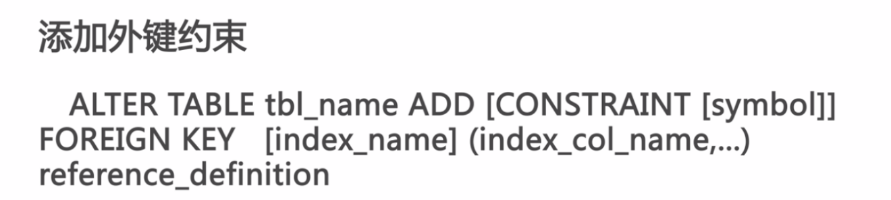
mysql> ALTER TABLE users2 ADD CONSTRAINT PK\_users2\_id PRIMARY KEY (id);

PK\_users2\_id为主键的名字

### 添加唯一约束



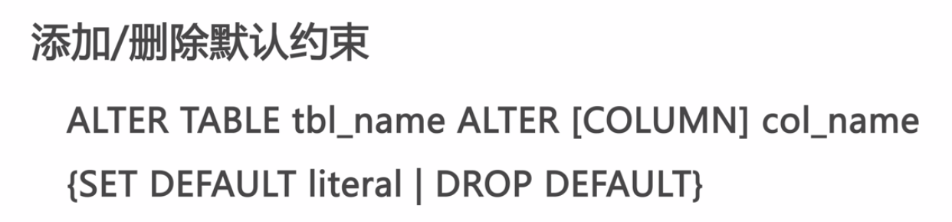
### 添加外键约束



### [添加联合唯一约束](http://blog.csdn.net/yumushui/article/details/38960619)

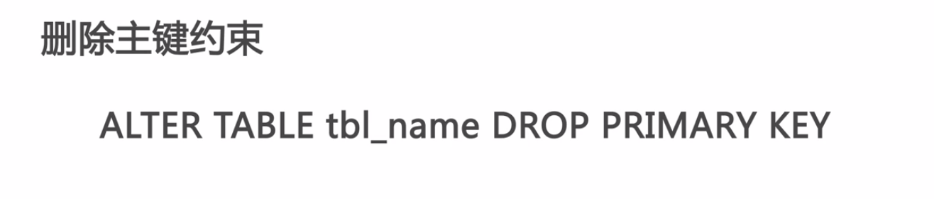
mysql> ALTER TABLE jw\_resource ADD **UNIQUE KEY(**resource\_name, resource\_type**)**;

## 添加/删除默认约束——ALTER…{ ALTER | DROP}



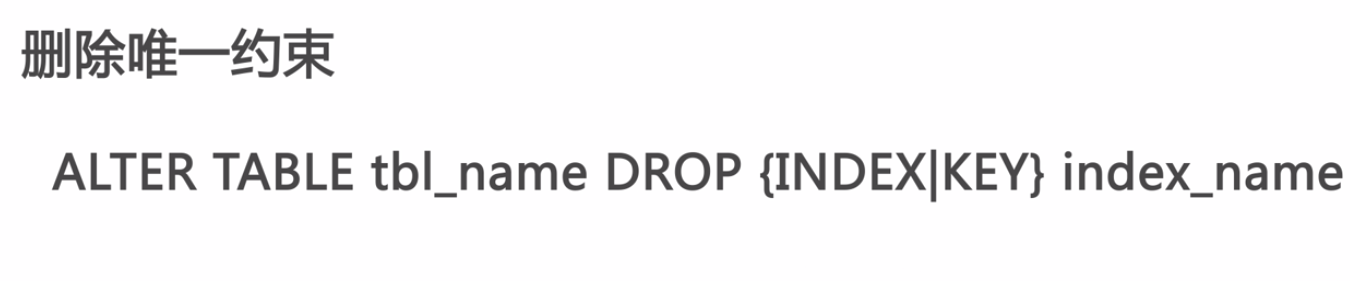
大括号为单选

### 删除主键约束



### 删除唯一约束

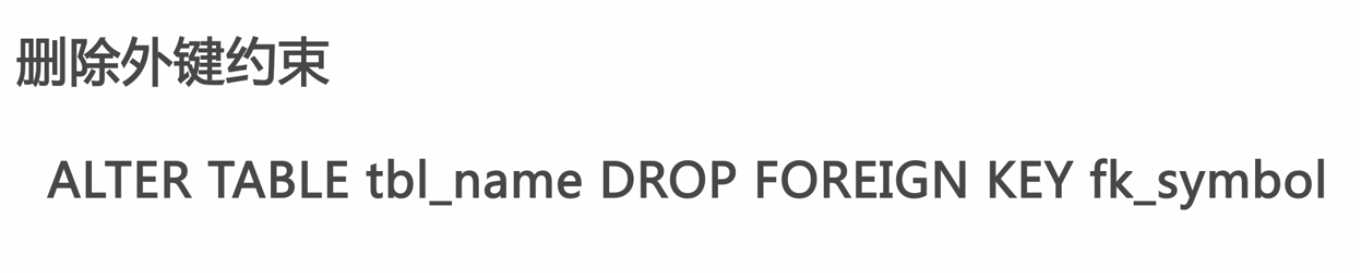
通过SHOW INDEX FROM tb\_name;查看索引（Key\_name对应项）。



删除的是字段上的约束，而并非字段。

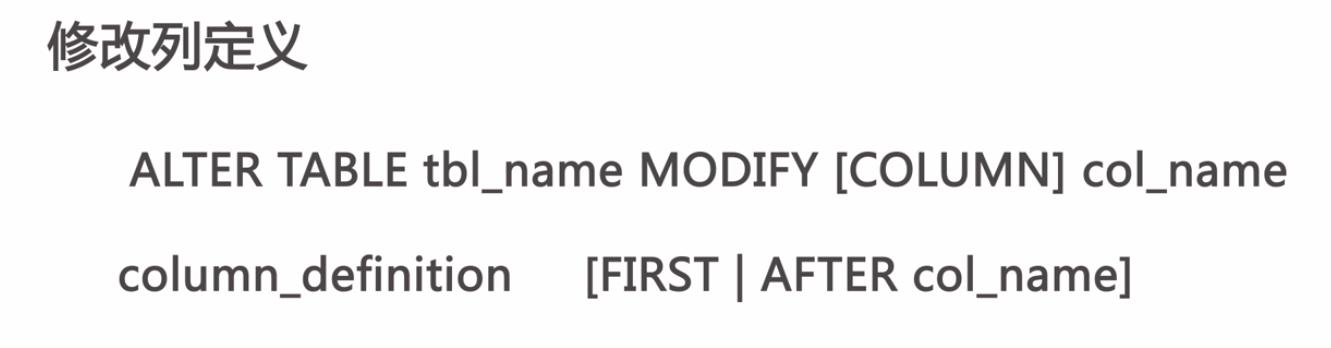
### 删除外键约束

通过SHOW CREATE TABLE tab\_name;查看外键约束。



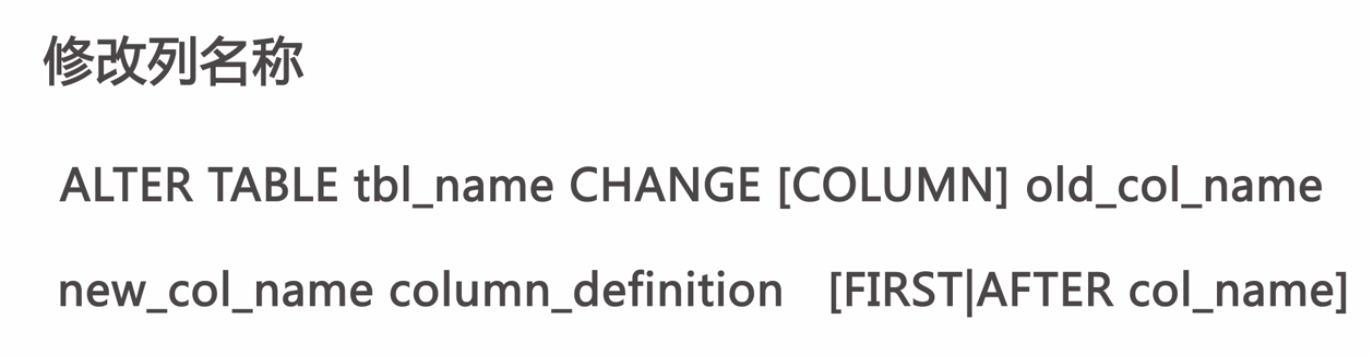
## 修改列定义和更名数据表

### 修改列定义——ALTER…MODIFY

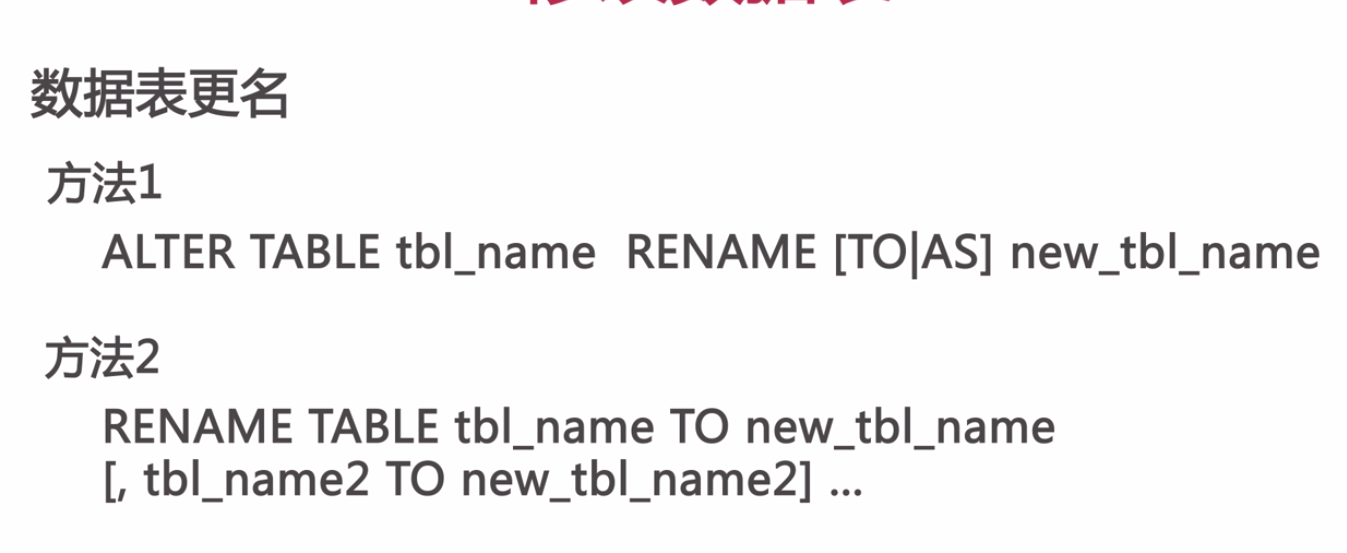


ALTER TABLE tb5 MODIFY sex ENUM('1','2','3','4','5') DEFAULT 5 FIRST;

### 修改列名及定义——ALTER…CHANGE



### 数据表重命名——ALTER…RENAME



## 查看表的字段信息：desc 表名;

## 查看表的创建语句：show create table 表名;

## 添加主键约束：alter table 表名 add constraint 主键 （形如：PK\_表名） primary key 表名(主键字段);

## 添加外键约束：alter table 从表 add constraint 外键（形如：FK\_从表\_主表） foreign key 从表(外键字段) references 主表(主键字段);

## (alter table 主表名 add foreign key (字段 ) references 从表名(字段) on delete cascade)

## 添加唯一约束:ALTER table 表名 add unique key 约束名 (字段);

## 删除主键约束：alter table 表名 drop primary key;

## 删除外键约束：alter table 表名 drop foreign key 外键（区分大小写）;

## 修改表名：alter table t\_book rename to bbb;

## 添加列：alter table 表名 add column 列名 varchar(30);

## 删除列：alter table 表名 drop column 列名;

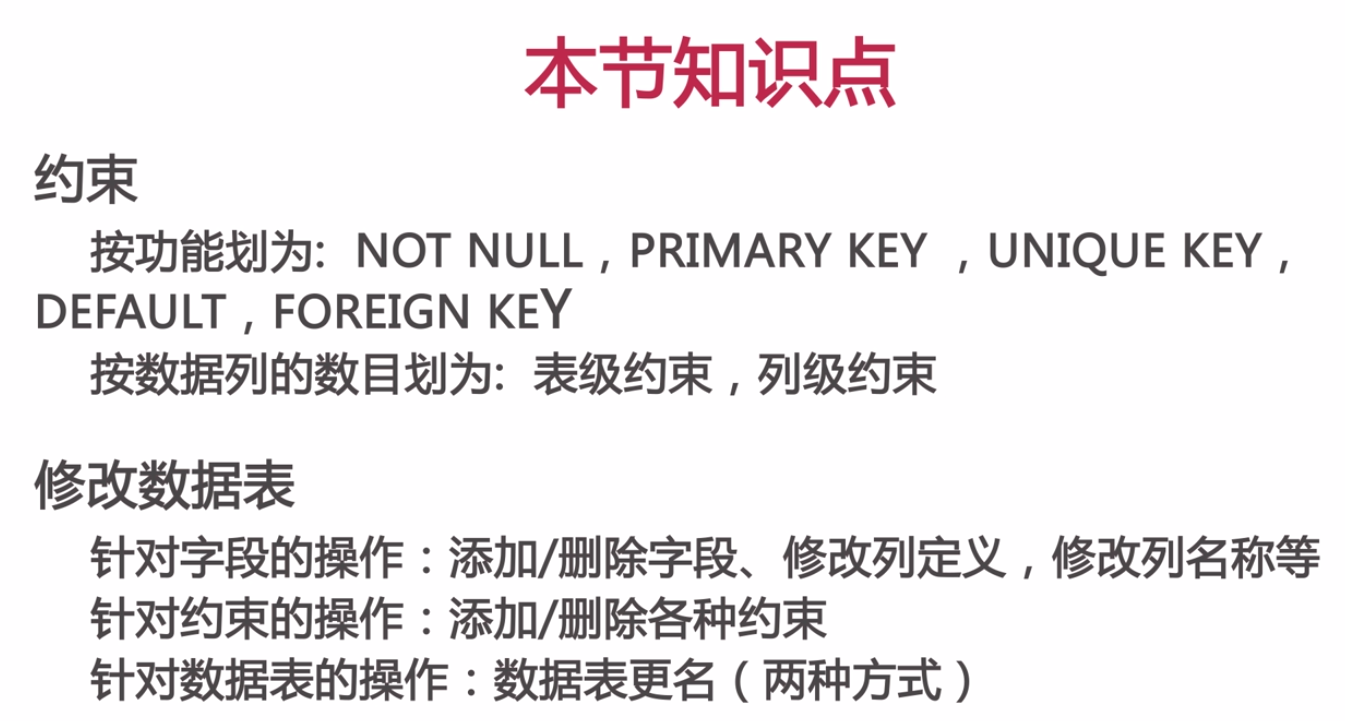
## 修改列名MySQL： alter table bbb change nnnnn hh int;

## 修改列名SQLServer：exec sp\_rename't\_student.name','nn', 'column' ;

## 修改列名Oracle：alter table bbb rename column nnnnn to hh int;

## 修改列属性：alter table t\_book modify name varchar(22);

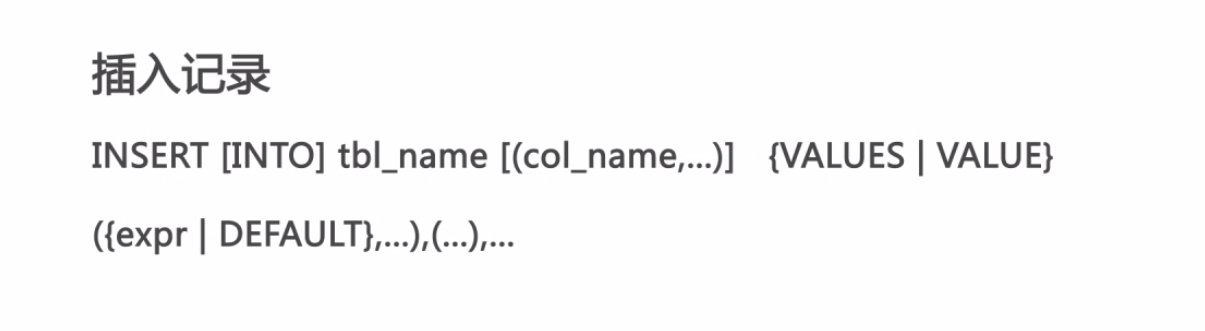
## 总结





# 第四章．操作数据表中的记录

## 记录的插入——INSERT



mysql> CREATE TABLE five01(

-> id SMALLINT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

-> username VARCHAR(20) NOT NULL,

-> password VARCHAR(32) NOT NULL,

-> age TINYINT UNSIGNED NOT NULL DEFAULT 13,

-> sex BOOLEAN

-> );

Query OK, 0 rows affected (0.29 sec)

### 方法一：单值，多值插入

一次插入一个值

mysql> INSERT fivee01 VALUES(***NULL***,'Tom','123',23,1);

mysql> INSERT fivee01 VALUES(***DEFAULT***,'Tom','123',23,1);

DEFAULT和NULL对应自增的主键id，不为其赋值而使之按默认的自增可赋予这两个值。

一次插入多个值

mysql> INSERT five01 VALUES(***NULL***,'Tom','123',23,1),(***DEFAULT,***'Join',***md5(123)***,***DEFAULT***,0);

md5()：为PHP函数，将传入参数转换为哈希值。

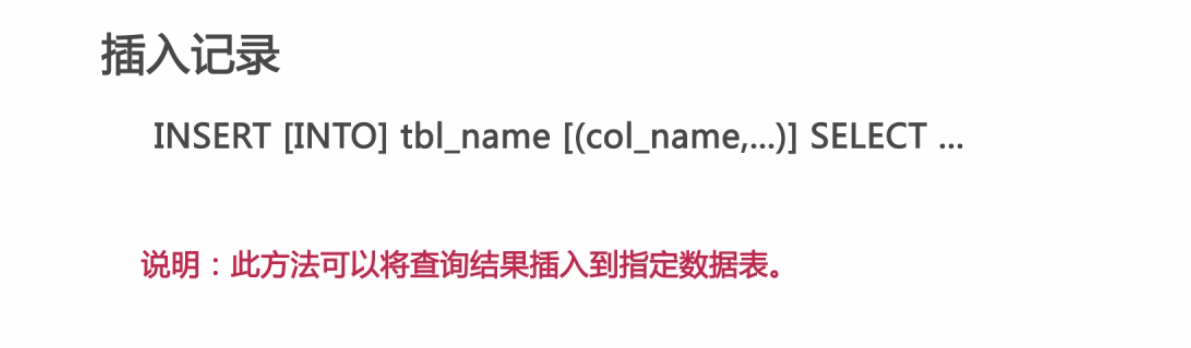
### 方法二：set可使用子查询



一次只能插入一个记录

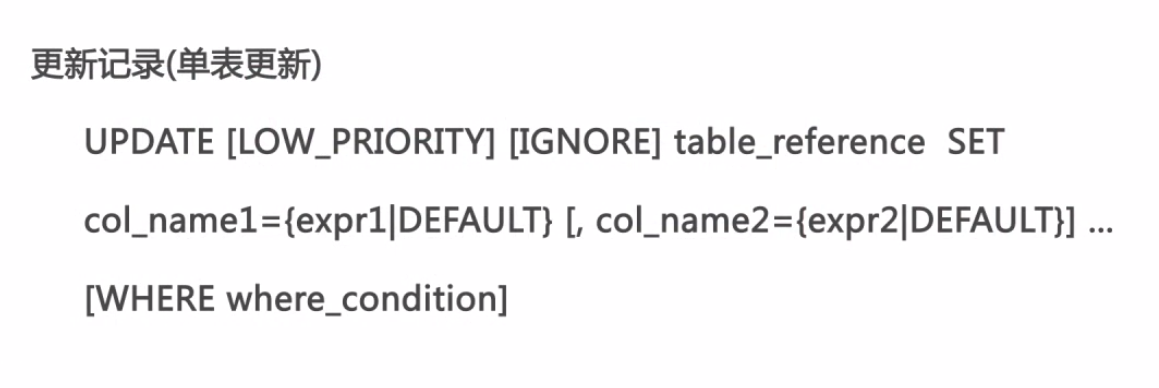
mysql> INSERT five01 **SET** username='Bean',password='4567';

### 方法三：插入查询结果



mysql> INSERT tb4\_a(username,age) SELECT username,age FROM tb4 WHERE age>=25;

## 更新记录——UPDATE



### **只更新一列：**

mysql> UPDATE five01 SET age=age+3;

此命令将five01表中所有记录的age值更改为age+3；

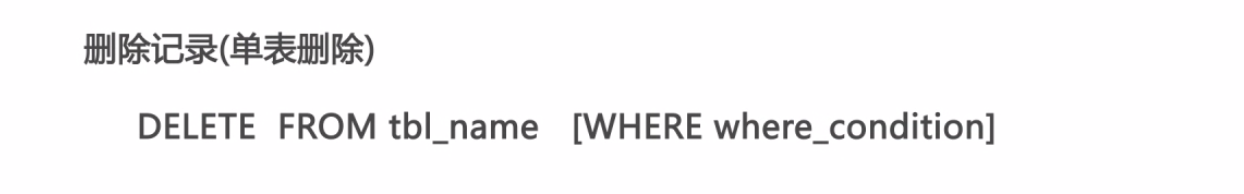
### **更新多列：**

mysql> UPDATE five01 SET age=age+2,sex=0;

### **更新符合指定条件的列**

mysql> UPDATE five01 SET age=age+10 WHERE id%2=0;

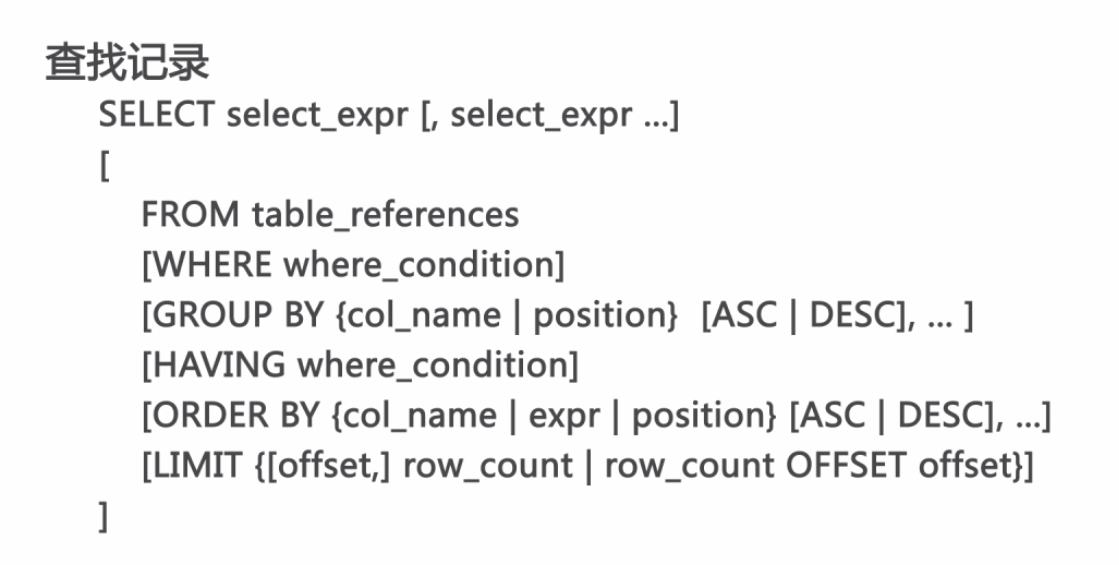
## 删除记录——DELETE



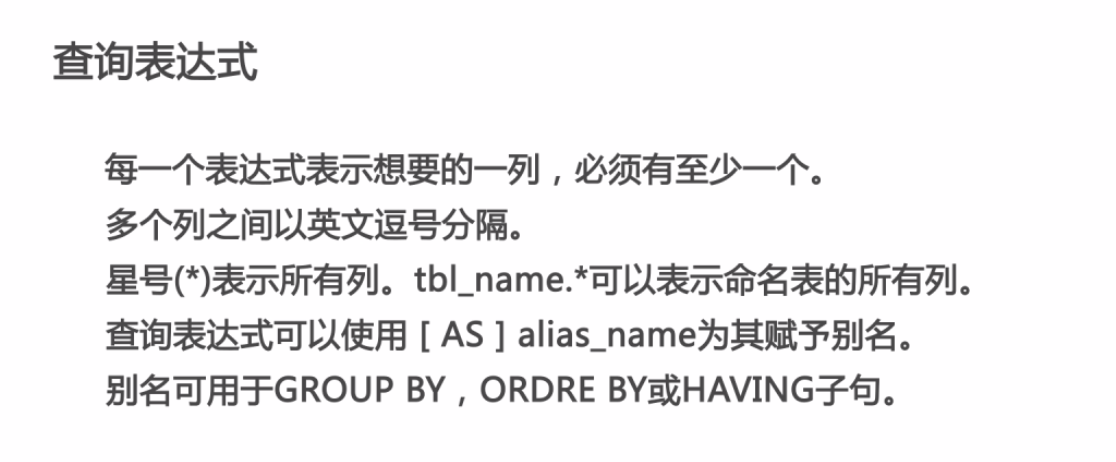
mysql> DELETE FROM five01 WHERE id=2;

## 清空表 truncate table

## 查找记录——SELECT



### 查询表达式



mysql> SELECT id AS uid,username AS uname FROM five01;

mysql> SELECT 78-34+34\*122;

+-----------------------+

| 78-34+34\*122 |

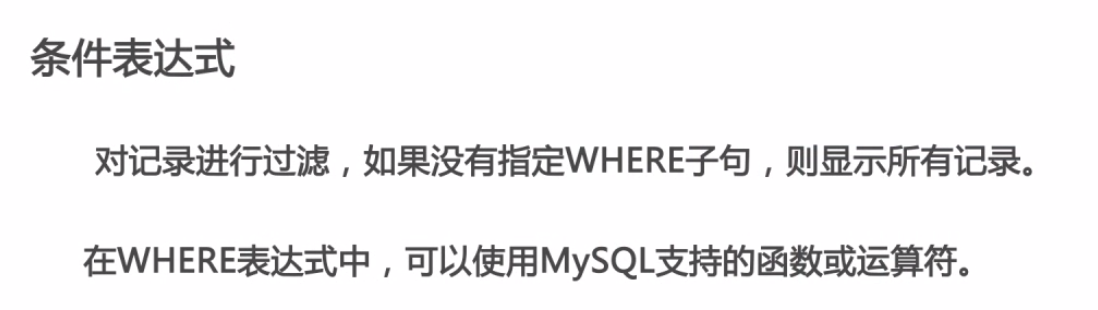
+-----------------------+

| 4192 |

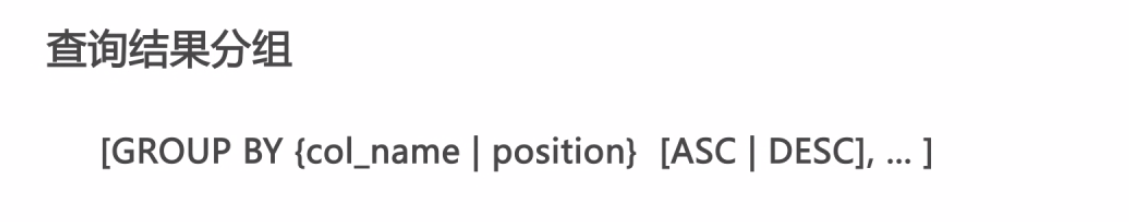
+------------------------+

1 row in set (0.04 sec)

### 查询条件

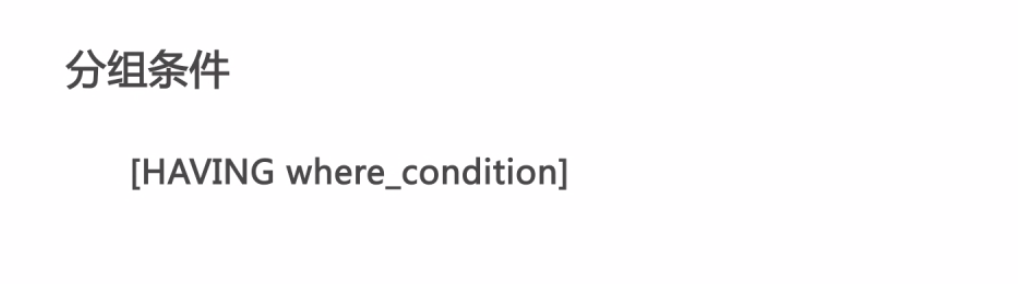


### 查询结果分组——GROUP BY

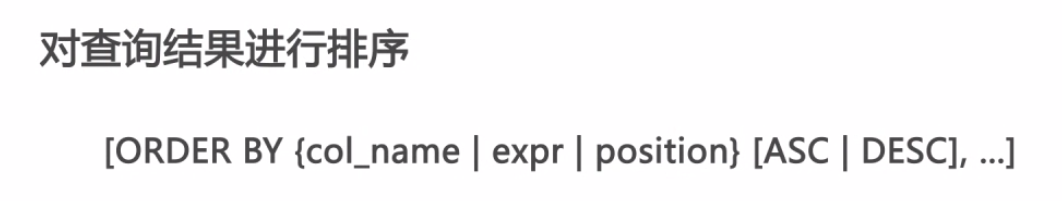


mysql> SELECT sex FROM five01 GROUP BY sex;

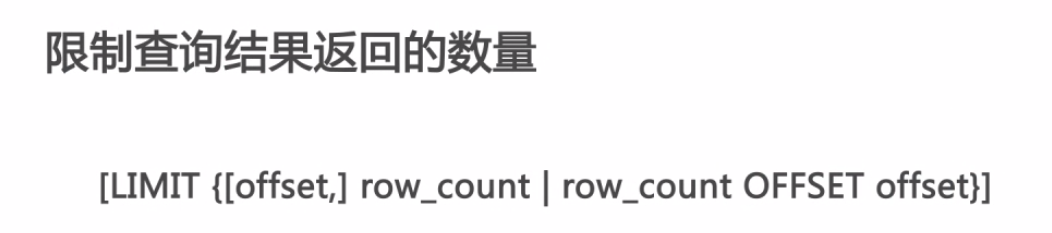
#### **分组条件——HAVING**



#### **对查询结果排序——ORDER BY**

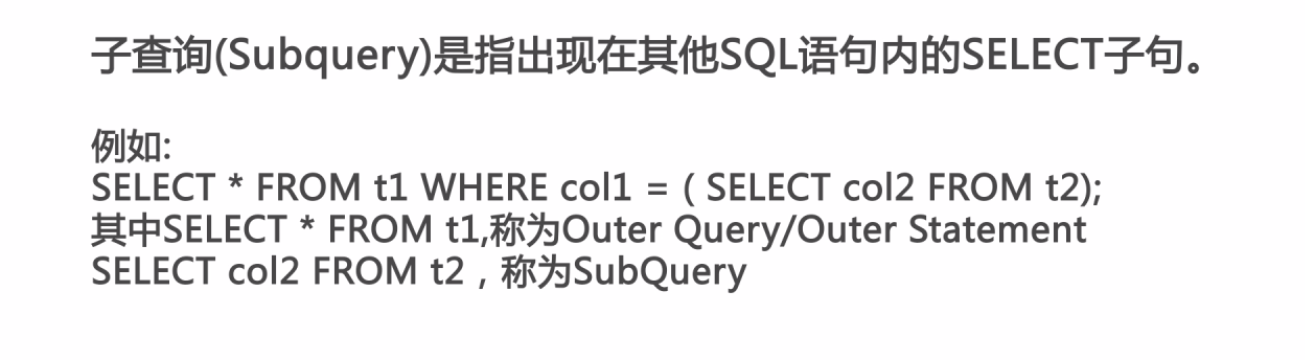


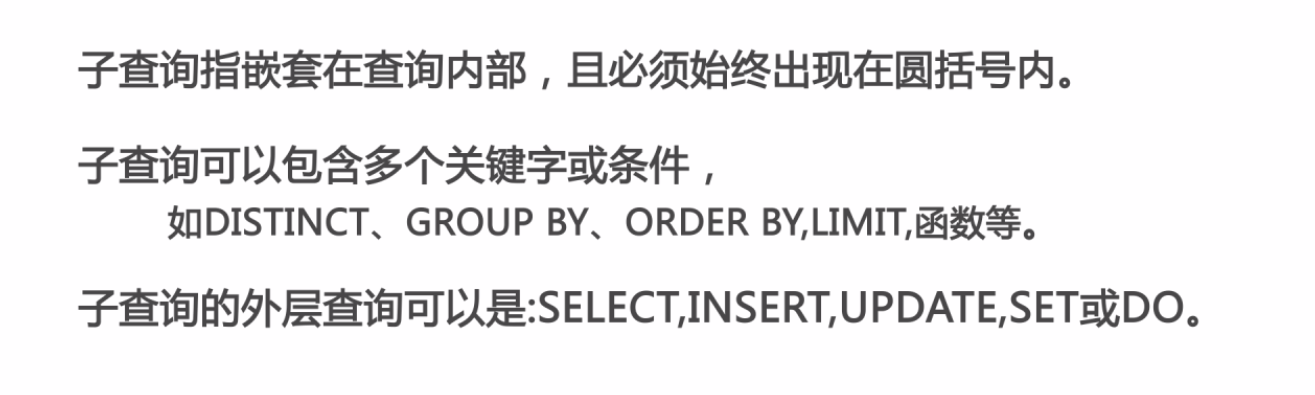
#### **限制查询结果返回数量——LIMIT**



# 第五章 子查询与连接

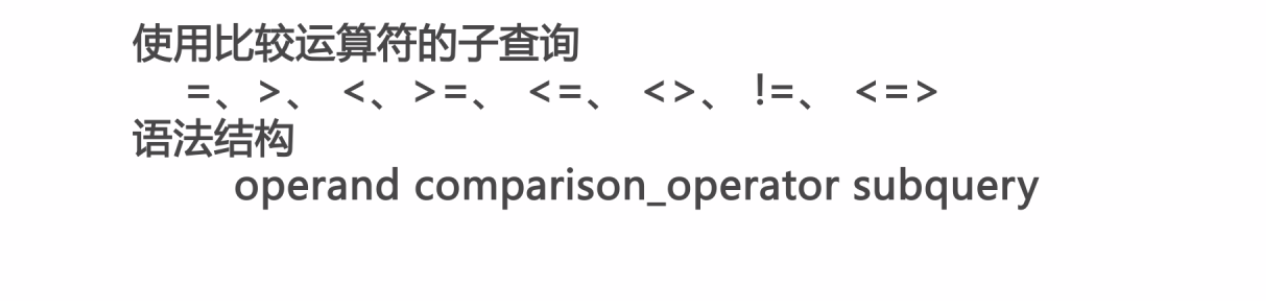
简介





子查询可以返回标量、一行、一列或子查询

## 使用比较运算符的子查询



mysql> SELECT ROUND(AVG(goods\_price),2) FROM tdb\_goods;

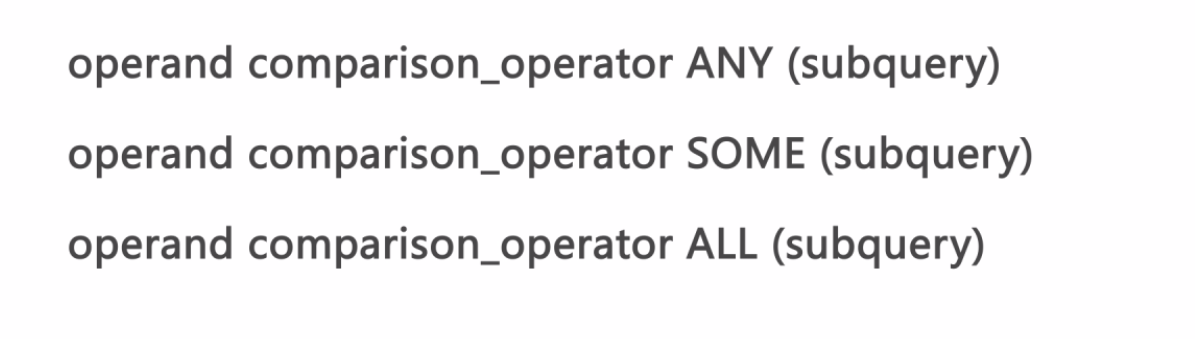
ROUND(r,2)对r进行四舍五入并保留小数点后两位

AVG(a)对a字段的所有值求平均值

mysql> SELECT goods\_id,goods\_name,goods\_price FROM tdb\_goods WHERE goods\_price>=(SELE

CT ROUND(AVG(goods\_price),2) FROM tdb\_goods);

当子查询返回多个结果时选用ANY/SOME/ALL来选出其中唯一一个。

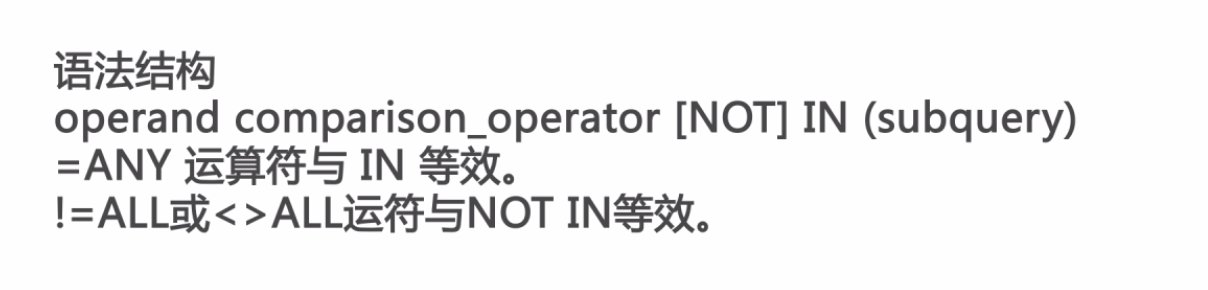




mysql> SELECT goods\_id,goods\_name,goods\_price FROM tdb\_goods WHERE goods\_price>=***ALL*** (SELECT goods\_p

rice FROM tdb\_goods WHERE goods\_cate='超级本');

## 使用[NOT]IN的子查询



**一.MYSQL的命令行模式的设置：  
桌面->我的电脑->属性->环境变量->新建->  
PATH=“；path/mysql/bin;”其中path为MYSQL的安装路径。  
二.简单的介绍一下命令行进入MYSQL的方法：  
1.C:/>mysql -h hostname -u username -p   
按ENTER键，等待然后输入密码。这里hostname为服务器的名称，如localhost，username为MYSQL的用户名，如root。  
进入命令行后可以直接操作MYSQL了。  
2.简单介绍一下MYSQL命令：  
mysql->CREATE DATABASE dbname;//创建数据库  
mysql->CREATE TABLE tablename;//创建表  
mysql->SHOW DATABASES;//显示数据库信息，有那些可用的数据库。  
 mysql->USE dbname;//选择数据库  
mysql->SHOW TABLES;//显示表信息，有那些可用的表  
mysql->DESCRIBE tablename;//显示创建的表的信息**

## 从数据库导出数据库文件

1. 将数据库mydb导出到e:/mysql/mydb.sql文件中：  
   打开开始->运行->输入cmd 进入命令行模式  
   c:/>mysqldump -h localhost -u root -p mydb >e:/mysql/mydb.sql  
   然后输入密码，等待一会导出就成功了，可以到目标文件中检查是否成功。  
   2.将数据库mydb中的mytable导出到e:/mysql/mytable.sql文件中：  
   c:/>mysqldump -h localhost -u root -p mydb mytable>e:/mysql/mytable.sql   
   3.将数据库mydb的结构导出到e:/mysql/mydb\_stru.sql文件中：  
   c:/>mysqldump -h localhost -u root -p mydb --add-drop-table >e:/mysql/mydb\_stru.sql

## 从外部文件导入数据到数据库中：

从e:/mysql/mydb2.sql中将文件中的SQL语句导入数据库中：  
1.从命令行进入mysql，然后用命令CREATE DATABASE mydb2;创建数据库mydb2。  
2.退出mysql 可以输入命令exit；或者quit；  
3.在CMD中输入下列命令：  
c:/>mysql -h localhost -u root -p mydb2 < e:/mysql/mydb2.sql  
然后输入密码，就OK了。  
五.下面谈一下关于导入文件大小限制问题的解决：  
默认情况下：mysql 对导入文件大小有限制的，最大为2M，所以当文件很大时候，直接无法导入，下面就这个问题的解决列举如下：  
1.在php.ini中修改相关参数：  
影响mysql导入文件大小的参数有三个：  
memory\_limit=128M,upload\_max\_filesize=2M,post\_max\_size=8M  
修改upload\_max\_filesize=200 M 这里修改满足你需要的大小，  
可以同时修改其他两项memory\_limit=250M post\_max\_size=200M  
这样就可以导入200M以下的.sql文件了。