**一、库操作**

(create/ drop/ show/ alter )

1. 创建数据库：

* create database db\_name，会在mysql的数据目录下，创建一个目录，目录名是数据库名

1. 创建表

* create table student(
* name varchar(20),
* age int
* );

1. 查看当前存在的数据库

* Show database;
* 查看数据的创建语句：Show create database db\_name；

1. 数据库删除

* Drop database db\_name

1. 修改数据库信息

* Alter database db\_name [修改指令]
* Alter database db\_name character set gbk

问题：如何将数据库内容导出，新建一个数据库，将内容导入，并删除旧的数据库

**二、表操作:（alter /add/drop/modify/change）**

1、创建表：(create table database.table)

\* 确定表的所属数据库，表必须属于某个数据库

\* create table ‘1234’.’java\_class’

\*

2、列定义：

\* 列名 + \*列的数据类型+列的属性（约束）

\* 字符串类型：varchar();

\* 日期类型：date

3、表明前缀（为了区分相同逻辑表名的不同应用，给逻辑表名，增加前缀，行程真实表名）

/\*学生管理\*/

\* create table info\_student(name varchar(20) DEFAULT NULL,stu\_no varchar(20) DEFAULT NULL);

/\*在线考试\*/

\* create table exam\_student(name varchar(20),stu\_no varchar(20),score int);

4、查看

\* Show tables;

\* Show create table tbl\_name;

1. 查看表结构describe

* describe exam\_student; == desc exam\_student

1. 删除表

* drop table if exists db\_name
* drop table db\_name;

1. 修改表名
   1. Rename table old\_tbl\_name to new\_tbl\_name;
   2. Rename table old\_tbl\_name to new\_tbl\_name ,old\_student\_name to new\_sutdent\_name;
2. 跨数据库重命名
   1. rename table exam\_user to ‘1234’.user
   2. 使用rename 将就数据库中的表，rename到新的数据库中，删除就的数据库
3. 修改列定义
   1. alter table tbl\_name [add|drop|change|modify](子命令)
   2. 增加一个新列
      1. alter table exam\_student add height int;
   3. 删除一个列
      1. alter table exam\_student drop height;
4. 修改一个列的定义(modify column )
   * 1. alter table exam\_student modify stu\_no varchar(40) ;//从int改为了字符串
   1. 重命名一个列的定义（change old new ）
      1. alter table exam\_table change score fenshu int(将score改为fenshu)

**三、数据操作**

**(create read,update delete)**

**创建数据（插入数据）、获得数据（查询数据）、修改数据、删除数据**

1. 创建数据
   1. Insert into 表名（字段列表）values(值列表)
      1. Insert into exam\_student <name,age,score>values<’xiaowang’,’12’,’100’>;
      2. 如果要为所有的字段赋值那么字段列表可以省略。
      3. Insert into 表名 <‘小明’,’12’,’12’>
2. 查询数据
   1. select 字段列表 from 表名 查询条件（字段列表可以用\*代替，代表所有的字段，查询条件可以省略，标识所有的记录都获得，相当于where 1）
      1. select name,stu\_no from exam\_student;(where 默认为1，代表永远成立，相当于默认)
      2. select \*from exam\_student
      3. select name,stu\_no from exam\_student where 1;
      4. select \*from exam\_studen where fenshu>=72
3. 删除数据
   1. Delete from 表名 条件
   2. 删除需要在逻辑上严格给条件，否则，容易造成数据误操作，导致损失，语法上可以没有where。删除所有数据可以用where 1
   3. delete from exam\_student where 1
4. 修改数据
   1. update exam\_student set fenshu=100 where fenshu>=72;

### 1　基础查询

1. 查询所有列
   1. SELECT \*FROM STU
2. 查询指定列
   1. SELECT sid ,sname ,age FROM stu;
3. 条件查询
   1. 查询时给出WHERE子句，WHERE子句中可以使用如下运算符和关键字：
      1. =、！=、<>、<、<=、>、
      2. BETWEEN…AND
      3. IN(set)
      4. IS NULL;IS NOT NULL
      5. AND
      6. OR
      7. NOT in(set)
   2. 查询性别为女，并且年龄为50的记录
      1. SELECT \*FROM stu WHERE gender=’female’AND age <50;
   3. 查询学号为S\_1001,或者姓名为lisi的记录
      1. SELECT \*FROM stu WHERE stu\_no=’S\_1001’OR name=’lisi’;
   4. 查询学号为S\_1001,S\_1002,S\_1003的记录
      1. SELECT \*FROM stu WHERE sid IN(‘S\_1001’,’S\_1002’,’S\_1003’)
   5. 查询学号不为S\_1001,S\_1002,S\_1003的记录
      1. SELECT \*FROM stu WHERE sid NOT IN(‘S\_1001’,’S\_1002’,’S\_1003’);
   6. 查询年龄为null的记录
      1. SELECT \*FROM stu WHERE age IS NULL;
   7. 查询年龄在20-40之间的学生记录
      1. SELECT \*FROM stu WHERE age>=20 AND age <= 40
      2. SELECT \*FROM stu WHERE age BETWEEN 20 AND 40;
   8. 查询性别非男的学生记录
      1. SELECT \*FROM stu WHERE gender != ‘male’
      2. SELECT \*FROM stu WHERE gender <>’male’;
      3. SELECT \*FROM stu WHERE NOT gender=’male’;
   9. 查询姓名不为null的学生记录
      1. SELECT \*FROM stu WHERE name IS NOT NULL
      2. SELECT \*FROM stu WHERE NOT name IS NULL
   10. 模糊查询（当想查询姓名中包含a字母的学生时就需要模糊查询，关键字LIKE）
       1. 通配符：
          1. **\_ ：代表任意一个字符**
          2. **%：代表任意0~n个字符**
       2. 查询姓名由5个字母构成的学生记录
          1. SELECT \*FROM stu WHERE name LIKE’\_\_\_\_\_’;
       3. 查询姓名由5个字母构成，并且第5个字母为i的学生记录
          1. SELECT \*FROM stu WHERE name LIKE ‘\_\_\_\_i’;
       4. 查询姓名以’z’开头的学生记录
          1. SELECT \*FROM stu WHERE name LIKE ‘z%’
       5. 查询姓名中第2个字母为‘i‘的学生记录
          1. SELECT \*FROM stu WHERE name LIKE ‘\_i%’;
       6. 查询姓名中包含‘a‘字母的学生记录
          1. SELECT \*FROM stu WHERE name LIKE ‘%a%’;
4. **排序（order by 列名 asc(默认) desc）**

* 查询所有学生记录，按年龄升序排序
  + SELECT \*FROM stu ORDER BY age ASC
* 查询所有雇员，按月薪降序排序，如果月薪相同，按照编号升序
  + SELECT \*FROM emp ORDER BY sal desc,num ASC

1. **聚合函数（sum/avg/max/min/count）:用来做纵向运算的函数**
   1. COUNT():统计指定列不为NULL的函数
   2. MAX():指定列的最大值
   3. MIN()指定列的最小值
   4. SUM()：指定列的数值和，不是数值类型电话，结果为0
   5. AVG()：指定列的平均值，不是数值类型电话，结果为0
2. 分组查询 GROUP BY
3. LIMIT
   1. select \* from emp limit 0,4;第一页（0,1,2,3） 1-0
   2. SELECT \*FROM emp LIMIT 0,5;
4. ****
   1. 约束外键：
      1. CREATE TABLE t\_bookType{id int primary key auto\_increment,bookTypeName varchar(20),bookTypeDesc varchar(200)};
      2. CREATE TABLE t\_book(id int primary key auto\_increment,bookName varchar(20),author varchar(10),price decimal(6,2),bookTypeId int , constraint ‘fk’foreign key (‘bookTypeId’)references ‘t\_bookType’(‘id’))
      3. 外键相当于本表中的某个字段关联另一张表的主键