一.单表查询

1.单表查询的总语法

```
select 列名1,列名2,...
from 表名
[where 条件]                    //边查询边筛选
[group by 列名1,列名2...]      //将查询结果进行分组
[having 条件]                   //将查询分组后的结果进一步筛选
[order by 列名1 asc/desc,列名2 asc/desc]   //排序,默认升序asc
[limit 数字1,数字2]              //限制输出的记录条数或分页
```

2.查询表中所有记录

语法:select 列名1,列名2,...from 表名;(推荐)

select * from 表名;(属于模糊查询,效率比上面的低)

eg:#查询表中所有的记录

```
select * from t_student;
select sid,sname,sage,sex,address from t_student;
```

3.根据条件查询:语法:select 列名1,列名2,...from 表名 where 条件;

eg:#边查询边筛选:查询年龄小于20岁的所有的学生信息

select sid,sname,sage,sex,address from t_student where sage<20;

4.查询完后对结果进行排序:select 列名1,列名2,...from 表名 order by 列名1...;

eg:#查询完后对结果进行一个列的排序:查询学生信息,按学号来排/按姓名排序

```
select sid,sname,sage,sex,address from t_student order by sname;

#查询完后对结果进行多个列的排序:查询学生信息,先姓名排序,再按年龄排序
select sid,sname,sage,sex,address from t_student order by sname asc,sage desc;
```

5.聚合函数:sum()求总和,,max()最大值min()最小值,avg()求平均值,count()求总

记录 数.

注意:一般情况下,普通不能与聚合函数列一起使用(语法上可以,但逻辑上有问题), 结果集无法匹配,除非普通列作为分组条件就可以和聚合函数列一起使用

聚合函数不能用在where,其他关键字后面可以用.

eg:#聚合函数查询

```
select count(sid),sum(sage),max(sage),min(sage),avg(sage) from t_student;
#一般情况下,普通不能与聚合函数列一起使用(语法上可以,但逻辑上有问题),结果集无法匹配
select sname,count(sid) from t_student;
```

6.模糊查询:

6.1:通配符:一种特殊符号,每种符号有特定含义

*: 代表表中所有列

_: 代表一个任意字符

%: 代表任意长度的任意字符

6.2:like查询: select 列名1,列名2... from 表名 where 列名 like 通配符;

注意:like一般要与通配符配合使用;能用Like查询的列是字符串列.

eg:#like模糊查询:查询所有姓王的学生

```
select sid,sname,sage,sex,address from t_student
    where sname like '王%';
```

6.3:between...and...:select 列名1,列名2... from 表名 where 列名 between 小值 and 大值;

注意:between...and...适用于数据列;

between...and...小范围的值写在前面,大范围的值写在后面.反之,语法 不报错,但是无意义.

eg:#BETWEEN and: 查询年龄在18到20岁的所有的学生信息

```
select sid,sname,sage,sex,address from t_student
where sage>=18 and sage<=20;

select sid,sname,sage,sex,address from t_student
where sage BETWEEN 18 and 20;
```

6.4:in(值1,值2...): select 列名1,列名2... from 表名
where 列名 in(值1,值2...);

eg:#in:查询年龄在18到20岁的所有的学生信息

```
select sid,sname,sage,sex,address from t_student where sage in(18,19,20);

select sid,sname,sage,sex,address from t_student
where sage=18 or sage=19 or sage=20;

#in:查询地址是千锋1或者是千锋2
select sid,sname,sage,sex,address from t_student where address in('千锋1','千锋2');

    select sid,sname,sage,sex,address from t_student
    where address='千锋1' or address='千锋2'
```

6.5:is [not] null:select 列名1,列名2... from 表名
where 列名 is null;

eg:#is null/is not NULL

```
    select sid,sname,sage,sex,address from t_student
    where sage = null;#查询不到数据

    select sid,sname,sage,sex,address from t_student
    where sage is null;
    select sid,sname,sage,sex,address from t_student
    where sage is not nul
```

7.分组查询和筛选:

7.1:分组查询:select 列1,列2...from 表名 group by 列名1,列名2;

eg:#分组查询:分别查询出男生和女生的总人数

```
select sex,count(sid) from t_student GROUP BY sex
```

7.2.分组查询后再筛选:select 列1,列2...from 表名 group by 列名1,列名2

having 条件;

注意:有having的地方一定有group by,

但是有group by的地方不一定有having.

#分组查询后筛选:按性别分组查询,查询出总人数大于6的性别

select sex,count(sid) from t_student GROUP BY sex having count(sid)>6

8.(MySql)limit的使用:

8.1:限制输出记录条数:select 列名1,列名2,...from 表名 limit 记录数;

eg:#limit:限制输出记录条数;查询出学生表中前三条记录

select sid,sname,sage,sex,address from t_student limit 3;

8.2:分页:select 列名1,列名2,...from 表名 limit 每页起始记录数,每页显示的记录条数;

eg:#limit:对数据进行分页: 每页显示5条记录数,查询出第三页

select sid,sname,sage,sex,address from t_student limit 10,5;

分页总语法:select 列名1,列名2,...from 表名

limit (当前页码-1)*每页显示的记录条数,每页显示的记录条数;

9.(MySql,Oracle)distinct:去重

eg:#DISTINCT:如果有多条记录结果相同去重

select DISTINCT sex from t_student;

二.在SQL语句中,运算符(算术运算符,赋值运算符,比较运算符,逻辑运算符)可以直接用.

1.在MySql中不等于: <> 或 !=

等于: =

2.逻辑运算符: and(并且),or(或者)

三.数据的完整性:数据库最重要的就是数据的完整性.

1.完整性:数据的可靠性和准确性.

2.数据完整性:实体完整性,域完整性,引用完整性,自定义完整性.

3.实体完整性:主键约束,唯一约束,标识列.

注意:一般情况下,每张表都要有一个主键.

4.域完整性:数据类型,默认约束,非空,检查约束(check).

注意:MySql5.5之后不支持检查约束

5.引用完整性:外键约束.

外键约束的优点:保护数据的安全性

缺点:效率低

综合外键约束的优缺点:一般不建立外键约束.

6.自定义完整性:规则、存储过程、触发器

回顾

1.dml(插入,修改,删除)

2.dql(基本查询,where,order by,聚合函数,模糊查询.group by,having,limit,distinct)

3.数据的完整性.