

1. 创建表

```
create table 表(  
    列 类型 [约束],  
    列 类型 [约束]  
);
```

2. float(M,D). double(M,D) M表示显示长度,D表示指定小数个数

3. char与varchar区别

字符串

固定长度 char(n) eg: char(20), 最大能存放20个字符. 'aaa', 还是占20个字符的空间

可变长度 varchar(n) eg: varchar(20), 最大能存放20个字符. 'aaa', 占3个字符的空间

一般使用varchar(n) 节省空间; 如果长度(eg:身份证)是固定的话 可以使用char(n) 性能高一点

4. 约束种类:

not null: 非空 ; eg: username varchar(40) not null username这个列不能有null值

unique:唯一约束, 后面的数据不能和前面重复; eg: cardNo char(18) unique; cardNo列里面不可以有重复数据

primary key; 主键约束(非空+唯一); 一般用在表的id列上面. 一张表基本上都有id列的, id列作为唯一标识的

auto_increment: ==自动增长, 必须是设置了primary key之后, 才可以使用

auto_increment==

id int primary key auto_increment; id不需要我们自己维护了, 插入数据的时候直接插入null, 自动的增长进行填充进去, 避免重复了.

修改表

增加一列; alter table 表 add 字段 类型 约束;

修改列的类型约束; alter table 表 modify 字段 类型 约束 ;

修改列的名称, 类型, 约束; alter table 表 change 旧列 新列 类型 约束;

删除一列; alter table 表名 drop 列名;

修改表名 ; rename table 旧表名 to 新表名;

对表的增删改查

添加: insert into 表名(列, 列) values(值, 值)

删除: delete from 表 [where 条件]

修改: update 表 set 列 =值, 列 =值 [where 条件]

查询:select [*] [列名 ,列名] [列名 as 别名 ...] [distinct 字段] from 表名
[where 条件]

去重查询 :SELECT DISTINCT 字段名 FROM 表名

别名查询 select 列名 as 别名 ,列名 from 表

条件查询

比较运算符	> < <= >= = <>	大于、小于、大于(小于)等于、不等于
	between .. and ..	显示在某一区间的值
	in(set)	显示在in列表中的值,例: in(100,200)
	like '张pattern'	模糊查询
	Is null	判断是否为空
逻辑运算符	and	多个条件同时成立
	or	多个条件任一成立
	not	不成立,例: where not(salary>100);

单列排序

SELECT 字段名 FROM 表名 [WHERE 条件] ORDER BY 字段名 [ASC|DESC]; //ASC: 升序,默认值; DESC: 降序

组合排序

SELECT 字段名 FROM 表名 WHERE 字段=值 ORDER BY 字段名1 [ASC|DESC], 字段名2 [ASC|DESC];

聚合函数

SELECT 聚合函数(列名) FROM 表名 [where 条件];

聚合函数	作用
max(列名)	求这一列的最大值
min(列名)	求这一列的最小值
avg(列名)	求这一列的平均值
count(列名)	统计这一列有多少条记录
sum(列名)	对这一列求总和

分组查询

SELECT 字段1, 字段2... FROM 表名 [where 条件] GROUP BY 列 [HAVING 条件];

where和having的区别【面试】

子名	作用
where 子句	1) 对查询结果进行分组前, 将不符合where条件的行去掉, 即在分组之前过滤数据, 即先过滤再分组。2) where后面不可以使用聚合函数
having字句	1) having 子句的作用是筛选满足条件的组, 即在分组之后过滤数据, 即先分组再过滤。2) having后面可以使用聚合函数

分页查询

select ... from limit a ,b

LIMIT a,b;
a起始行数，从0开始计数，如果省略，默认就是0; $a=(\text{当前页码}-1)*b$;
b: 返回的行数