测试用例：

Create table s (sno int ,sname char(10));

Create table c(cno int,cname char(10));

Create table sc(sno int,cno int,score float);

Insert into s values(1,’lcj’);

Insert into s values(2,’kzx’);

Insert into s values(3,’hlb’);

Insert into s values(4,’ll’);

Insert into s values(5,’cz’);

Insert into c values(1,’math’);

Insert into c values(2,’db’);

Insert into c values(3,’ds’);

Insert into c values(4,’english’);

Insert into sc values(1,1,98.0);

Insert into sc values(1,2,86.0);

Insert into sc values(1,3,97.0);

Insert into sc values(1,3,97.0);

Insert into sc values(2,2,99.0);

Insert into sc values(2,3,89.0);

Insert into sc values(3,3,98.0);

Insert into sc values(3,4,86.5);

Insert into sc values(5,2,93.5);

Insert into sc values(4,4,100.0);

Select s.sno ,s.sname from s;

Select c.cno,c.cname from c;

Select sc.sno,sc.cno,sc.score from sc;

Update s set s.sname = ‘ldl’ where s.sname =’ll’;

delete from sc where sc.sno = 5;

//选取成绩为100分的学生及科目

select s.sname, c.cname from s,c,sc where s.sno = sc.sno and c.cno = sc.cno and sc.score = 100.0;

//选取成绩大于90的学生及科目

select s.sname, c.cname from s,c,sc where s.sno = sc.sno and c.cno = sc.cno and sc.score > 90.0;

//选取课程号为4大于90分的学生学号及姓名

select s.sno,s.sname from s,c,sc where s.sno = sc.sno and c.cno = sc.cno and c.cno = 4 and sc.score >90.0;

数据库思路：

1. 创建数据库：

1.创建一个系统表，一个系统列文件

1. 删除数据库：

1.把整个文件夹删除就行

1. 打开数据库：
2. 创建表：
3. 打开系统表，系统列
4. 将数据表的相关信息分别填入系统表和系统列文件中
5. 将数据表中所有的属性长度加起来，作为记录长度，创建数据表文件
6. 删除表：
7. 打开系统表，系统列文件
8. 删除系统表和系统列中与要删除表对应的项
9. 删除数据表文件
10. 创建索引：
11. 删除索引：
12. 插入：
13. 打开系统表和系统列文件，读取要插入的数据表的相关信息
14. 根据属性的个数控制循环，循环插入到一片内存区，调用InsertRec函数
15. 删除：
16. 打开系统表系统列文件，读取相关信息。
17. 逐一读取记录 逐一比较条件，首先是获取左右两边的值，比较，看是否成立
18. 操作
19. 更新：

与删除类似

1. 查询：

RC Select(int nSelAttrs, RelAttr \*\*selAttrs, int nRelations, char \*\* relations,

int nConditions, Condition \*conditions, char \*\*RES)

1. 首先还是获取系统表和系统列中的信息。
2. 获取selAttrs对应的笛卡尔积的位置，及其对应的属性信息，在最后向res中写时要用。
3. 将nRelations个关系连接起来，一条笛卡尔记录由这nRealtions个关系的各一条记录组成
4. 读一条记录判断一条记录分为左属性右值，右属性左值，左右均属性三种情况。
5. 向res中写
6. Res的结构是按照头20字节 列数，20字节查询结果数，各属性列名，数据。