3.2.1、基本表的定义

语句格式:

CREATE TABLE <表名>

(<列名1> <类型 1 > [列级完整性约束条件]

[, <列名2> <类型2> [列级完整性约束条件]]

…

[,<表级完整性约束条件>]);

列级完整性约束

表级完整性约束

约束的类型

- **NOT NULL**
- **UNIQUE**
- **PRIMARY KEY**
- **FOREIGN KEY;**
- **CHECK**
- **DEFAULT**

3.2.2 基本表的修改和删除

ALTER TABLE <表名>

[**ADD** <列名><类型>[完整性约束]]

[**DROP** <列名>[<完整性约束名>]]

[**ALTER COLUMN** <列名> <类型>] ;

3.2.3 索引的建立和删除

1、建立索引

CREATE
[UNIQUE][CLUSTERED|NOCLUSTERED]
INDEX <索引名> ON <表名> (<列名> [<次序>]
[, <列名> [<次序>]…])

CREATE UNIQUE INDEX U_idx_cname ON Course(Cname);

CREATE UNIQUE INDEX SCno ON SC(Sno asc,Cno desc);

- 建立表s（sno,sname,ssex)
- 建立s表的主键 sno
- 建立表sc(sno,cno,grade)
- 建立sc表的主键（sno,cno)
- 建立sc表的外键
- 建立s表的索引idx_sname

3.2.3 索引的建立和删除

2、删除索引

DROP INDEX <索引名> ON <表名>

例：

drop INDEX stusno on student ;

3.3 数据查询

基本语句格式：

SELECT <属性列表>

FROM <基本表>（或视图）

[WHERE <条件表达式>];

3.3.1　单表查询

单表查询：涉及一个表的查询。

(1) 选择表中的若干列（投影）

(2) 选择表中的若干元组（选择）

(3) 对查询分组

(4) 使用集函数

(5) 对查询结果排序

【例】在学生表中找出全体学生的学号和姓名。

```
SELECT   Sno,Sname
FROM     Student;
```

【例】在学生表中找出全体学生的所有信息。

```
SELECT   *
FROM    Student;
```

【例】在学生表中找出全体学生的姓名和出生年份。

```
SELECT   Sname,2021-Sage
FROM     Student;
```

【例】给计算字段定义别名。

```
SELECT   Sname,2021-Sage as 出生年份
    FROM     Student;
```

常用的查询条件

查询条件	谓　词
比较	=,>,<,>=,<=,<>
确定范围	BETWEEN AND， NOT BETWEEN AND
确定集合	IN ，　NOT　IN
字符匹配	LIKE，NOT　LIKE
空值	IS NULL， IS　NOT　NULL
多重条件	AND，OR

【例】查找有不及格门次的学生学号。

```
SELECT  DISTINCT  Sno
FROM  SC
WHERE  Grade< 60
```

【例】查询年龄在20-22岁（包括20岁和22岁）之间的学生的姓名、年龄和院系。

```
SELECT Sname,Sage,Sdept
FROM Student
WHERE Sage BETWEEN 20 AND 22;
```

【例】查询年龄不在20-22岁之间的学生的姓名、年龄和院系。

```
SELECT Sname,Sage,Sdept
FROM Student
WHERE Sage  NOT BETWEEN 20 AND 22;
```

【例】在表学生中找出计算机学院或机电学院的学生信息。

```
SELECT Sname,Sage,Sdept
FROM Student
WHERE Sdept IN (‘计算机’,‘机电’)
```

```
SELECT Sname,Sage,Sdept
FROM Student
WHERE Sdept=‘计算机‘ OR Sdept=’机电’;
```

【例】查询所有姓李的学生的所有情况。

```
SELECT *
FROM Student
WHERE Sname like '李*';
```

%(*)代表：任意长度的字符串

_(?)代表:任意单个字符

【例】查找DB_Design课程的课程号和学分

```
SELECT Cno, Ccredit
FROM C
WHERE Cname LIKE 'DB/_Design'  ESCAPE '/ '
```

【例】查询以“DB_”开头，倒数第三个字符为i的课程的详细信息

```
SELECT  *
FROM    Course
WHERE   Cname like ‘DB/_%i_ _’ escape ‘/’;
```

【例】查找没有成绩的学生学号及相应课程号

```
SELECT Sno, Cno
FROM SC
WHERE Grade IS NULL
```

【例】查询计算机学院年龄在20岁以下的学生姓名

```
SELECT Sname
FROM Student
WHERE Sdept='计算机'and Sage<20
```

❖对查询结果分组统计

【例】查询学生总人数。

```
SELECT COUNT(*)
FROM Student
```

聚集函数:
COUNT SUM
AVG MAX MIN

【例】查询1号课程的平均成绩。

```
SELECT   AVG(Grade)
FROM     SC
WHERE    Cno='1'
```

【例】查询各门课程的选课人数。

```
SELECT   Cno,COUNT(SNO)
FROM     SC
GROUP BY Cno
```

【例】查询选修了2门以上课程的学生学号。

```
SELECT   Sno
FROM     SC
GROUP BY Sno
HAVING COUNT(Cno)>1
```

❖对查询结果排序

【例】查询选修了2号课程的学生学号和成绩，查询结果按成绩从大到小排列。

```
SELECT   sno,grade
FROM     sc
WHERE    cno='2'
ORDER BY grade DESC
```

❖对查询结果分组排序

【例】查询学号在1-4之间至少选修了两门课程的学生学号及其选修的课程数，并对课程数降序排序。

```
SELECT  Sno,COUNT(Cno)
FROM   SC
WHERE Sno BETWEEN '1' AND '4'
GROUP BY Sno
HAVING COUNT(Cno)>1
ORDER BY COUNT(Cno) DESC
```

SELECT [ALL| DISTINCT] 属性1 AS 别名,
属性2 AS 别名….

FROM 表名1，表名2….

[**WHERE** 条件表达式]

[**GROUP BY** 属性1,属性2 …..]

[**HAVING** 条件表达式]

[**ORDER BY** 属性1**[ASC|DESC]**,属性2 ….]

3.3.2 复杂查询

1) 等值与非等值连接查询

[〈表名1〉.]〈列名〉〈比较运算符〉[〈表名2〉.]〈列名〉

2) 自身连接:一个表与其自己进行连接。

3) 外连接:

右外连接right join : = (*)

左外连接left join : (*) =

4) 复合条件连接: WHERE 子句中可以有多个连接条件。

5) 集合运算

➢等值和非等值连接查询。

查询选修了课程的学生及其选修的课程的编号。

```
SELECT   Student.*, SC.Cno
FROM   Student,  SC
WHERE   Student.Sno=SC.Sno
```

➢等值和非等值连接查询。

查询选修了数据库原理课程的学生学号

```
SELECT  Sno
FROM    Course,  SC
WHERE   Cousre.Cno=SC.Cno
AND     Cname=‘数据库原理’
```

- 自身连接。

求选修了2号学生选修的课程的学生学号

关系代数怎么写？

```
SELECT DISTINCT A.Sno
FROM   SC A, SC B
WHERE  A.Cno=B.Cno
And    B.Sno='2'
and    A.Sno<>'2'
```

A

Sno	Cno	Grade
1	1	75
1	2	50
2	1	89

B

Sno	Cno	Grade
1	1	75
1	2	50
2	1	89

A.Sno	B.Cno	C.Grade	B.Sno	B.Cno	B.Grade
1	1	75	1	1	75
1	1	75	1	2	50
1	1	75	2	1	89
1	2	50	1	1	75
1	2	50	1	2	50
1	2	50	2	1	89
2	1	89	1	1	75
2	1	89	1	2	50
2	1	89	2	1	89

- 自身连接。

求同时选修了1号和2号课程的学生学号

```
SELECT  A.Sno
FROM    SC  A,  SC  B
WHERE   A.Sno=B.Sno
And     A.Cno= “1”
and     B.Cno= “2”
```

A

Sno	Cno	Grade
1	1	75
1	2	50
2	1	89

B

Sno	Cno	Grade
1	1	75
1	2	50
2	1	89

A.Sno	B.Cno	C.Grade	B.Sno	B.Cno	B.Grade
1	1	75	1	1	75
1	1	75	1	2	50
1	1	75	2	1	89
1	2	50	1	1	75
1	2	50	1	2	50
1	2	50	2	1	89
2	1	89	1	1	75
2	1	89	1	2	50
2	1	89	2	1	89

- 自身连接。

求每一门课程的先修课课程号。

➢左外连接。

求所有学生的选课的课程号和相应成绩。

```
SELECT Student.Sno, Sname, Cno, Grade

FROM Student LEFT JOIN SC ON
Student.Sno = SC.Sno;
```

```
SELECT Student.Sno, Sname, Cno, Grade

FROM SC RIGHT JOIN Student ON
Student.Sno = SC.Sno;
```

集合查询

```
SELECT   Sno
FROM   SC
WHERE   Cno=“1”
 UNION
SELECT Sno
 FROM SC
 WHERE Cno=“2”;
```

```
SELECT   Sno
FROM   SC
WHERE   Cno="1"
 INTERSECT
SELECT Sno
 FROM SC
 WHERE Cno="2";
```

```
SELECT   Sno, Cno, Grade
FROM   SC
WHERE   Cno='1'
EXCEPT
SELECT Sno, Cno, Grade
 FROM SC
 WHERE Cno='2';
```