## 数据库系统概论

### 一．基本概念

1. 三级模式结构

内模式

模式(全体表)

外模式

应用

1. 完整性

①实体完整性

②参照完整性

③用户定义完整性

### 二．关系代数

1.传统集合运算

2.专门运算

t[Ai] 元祖(行)t中Ai属性

象集：Zx2={Z2,Z3}并(x=x2)

1.选择

δ（条件）（表）

2.投影

π（属性）(表)

3.连接

R连接符号（下面为条件）S

4.除运算

R÷S

理解：除结果包括S中共有属性所有值的范围。

### 三．SQL

1.数据查询：SELECT

定义：CREATE DROP ALTER

操纵：INSERT UPDATE DELETE

控制；GRANT REVOKW

Datebase（房子）与schema（房间） user是房间的主人

Table是放在房间里的床，再mysql中两者相同

2.数据定义

CREATE/DROP schema/table/view/index

ALTER table/index

①模式

CREATE SCHEMA<模式名（可选）> AUTHORIZATION <>

DROP SCHEMA 模式名 <CASCADE|RESTRICT>

②基本表

1. CREATE TABLE <表名>

( 列名 类型 <约束条件>,

...........................................

表级约束性条件

）；

模式于表相联系：

①CREATE TABLE 模式名.表名 （）；

②创建模式同时创建表

③选用模式

④搜索路径

Show search\_path;

Set search\_path to “模式名”，public

b.ALTER TABLE<名>

ADD <列名><类型><约束>

ADD 表约束

DROP 列名 <CASCADE|RESTRICT>

DROP CONSTRAINT <约束性条件>

ALTER<列><类型>修改

c.删除

DROP table 名

③索引

1. 建立

CREATE [UNIQUE] [CLUSTER]

INDEX 索引名 on 表名(列名 可能次序)

ASC升序 DESC降序

b.修改

ALTER INDEX 旧索引名 RENAME TO 新名

c.删

DROP INDEX 名字

3.查询

SELECT [ALL|DISTINCT] 目标列

FROM 表名或视图名 | SELECT AS 列名（在FROM子句中建立一个临时表）

可选：WHERE+条件子句

可选：GROUP BY 列名 HAVING 条件

可选：ORDER BY ASC|DESC

单表查询：

SELECT DISTINCT Sname Name，2014-Sage BIRTH

FROM Student;

①查询列顺序决定显示的列顺序

②可以使用表达式、字符串常量、函数做查询

③后面为别名，也就是显示的列名

④DISTINCT消除重复元组

WHERE子句

①比较

同其他语言

②确定范围

（NOT）BETWEEN。。。。AND。。。。

WHERE Sage BETWEEN 20 AND 30

③集合

（NOT） IN

WHERE Sdept IN(‘CS’,’MA’,’IS’)

④字符匹配

（NOT）LIKE

%任意长度字符串 \_一个字符

WHERE Sname LIKE ‘\\_阳%’ ESCAPE ‘\’; 确定\为转义符

⑤空值

IS (NOT) NULL

⑥多重条件

AND OR NOT

ORDER BY子句

ORDER BY GRADE ASC|DESC;

聚焦函数

COUNT(\*) 统计元组的个数

COUNT (DISTINCT|ALL 列名)统计一列中值的个数

SUM (DISTINCT|ALL) 列名 统计一列值的总和

AVG (DISTINCT|ALL) 列名

MAX (DISTINCT|ALL) 列名

MIN (DISTINCT|ALL) 列名

GROUP BY子句

将查询结果按某一列或多列的值分组，值相等的为一组

HAVING短语作用于组

SELECT Sno,AVG(Grade)

FROM SC

GROUP BY Sno

HAVING AVG(Grade)>=90;

多表查询：

连接查询：

①SELECT Student.\*,SC.\*

FROM Student,SC

WHERE Student.Sno=SC.Sno;

②自身连接

SELECT FIRST.Cno,SECOND.Cpno

FROM Course FIRST,Course SECOND

WHERE FIRST.Cpno=SECOND.Cno;

③外连接

SELECT Student.Sno,Sname,Ssex,Sdept,Cno,Grade

FROM Student LEFT OUTER JOIN SC ON (Student.Sno=SC.Sno);

嵌套查询：

SELECT Sname

FROM Student

WHERE Sno IN

(SELECT Sno

FROM SC

WHERE Cno=’2’);

注意：嵌套子查询语句不能使用ORDER BY子句

EXISTS子语句 P109

用于逻辑类语句判断

集合查询

并操作：UINON 交操作：INTERSECT 差：EXCEPT

SELECT。。。。。。。（操作词）SELECT............

4.数据更新

①插入

INTO 表名 属性列（可选）

VALUES （）或子查询语句，把子查询做为输入

②修改

UPDATE 表名

SET 列名=表达式

WHERE子句（可选）

③删除

DELETE

FROM 表名

WHERE子句（可选）

5.空值

算术运算结果为空值 比较运算结果为UNKNOWN 逻辑运算见P120

6.视图

建立：

CREATE VIEW 视图名 列名。。。。

AS 子查询

WITH CHECK OPTION(可选)保证更新、删除、插入要满足子查询中条件表达式

删除：

DROP VIEW 视图名 （CASCADE）

查询和更新转化为对基本表的操作，注意表达式和函数可能出现问题

## 四．数据库安全性

1.存取控制

C2：自主存取控制

B1：强制存取控制

1. 授予

GRANT 权限。。。。

ON 对象类型 对象名。。。。。

TO 用户。。。。

（WITH GRANT OPTION)权限是否继承

Eg. GRANT INSERT

ON TABLE SC

TO U5

WITH GRANT OPTION

1. 收回权限

REVOKE 权限。。。。

ON 对象类想 对象名

FROM 用户 [CASCAD|RESTRICT];

1. 创建用户时指定权限

CREATE USER <用户名> WITH DBA|RESOURSE|CONNECT

1. 数据库角色（用于权限集合控制）

CREATE ROLE 角色名

然后给角色授权

将角色授权给用户：

GRANT 角色。。。。

TO 角色/用户

WITH ADMIN OPTION

收回权限同上

1. MAC 强制存取控制

主体：用户和用户进程

客体：系统中的被动实体

①仅当主体许可证大于等于客体才能读取

②仅当主体许可证小于等于才能写入客体

7.审计

AUDIT ALTER UPDATE

ON SC；

NOAUOIT ALTER UPDATE

ON SC;

### 数据库完整性

1.实体完整性

列级：primary key

表级：primary key(属性名)

①值唯一 ②不为空

1. 参照完整性

FOREIGN KEY（sno） REFERENCES student(Sno)

ON DELETE CASCADE

ON UPDATE CASCADE

或

ON DELETE NO ACTION

3.用户定义完整性

列级：

①NOT NULL

②UNIQUE

③CHECK子句

表级（元组约束）：

CHECK (Ssex=’女’ OR Sname NOT LIKE ‘Ms.%’)

完整性约束命名子句

CONSTRAINT 条件名 条件

可以使用DROP和ADD方便的添加和删除

第六章：关系数据理论

关系模式：R(U,D,DOM,F)

属性组U

D为属性组U来自的域

DOM为属性到域的映射

F为属性组U的一组数据依赖

码：唯一决定元组的属性

1.函数依赖

2NF:码→非码

3NF:无传递依赖

BCNF:所有函数依赖左边必须在码中，非码不能决定码

1. 多值依赖

一对多关系

4NF：只存在一组多组依赖

1. 公理系统

定义①：对任意一个关系（表），有X→Y，即称F逻辑蕴涵

闭包运算：不断从左向右推

最小依赖集：函数依赖集合等价

1. ：数据库设计