1. 关系数据库标准语言SQL

（操作题（15\*2）

定义视图。定义索引。查询。插入。删除。（13）

删除表 删除视图 删除数据 逻辑蕴含 创建模式）

1. SQL的产生、发展以及特点

答：SQL，是一个通用的、功能极强的关系数据库语言。同时也是一种介于关系代数与关系演算之间的结构化查询语言，其功能包括数据定义，数据查询，数据操控，数据控制

SQL的特点：

综合统一

高度非过程化

面向集合的操作方式

两种使用方式，统一的语法结构

简介易学

1. 数据库创建和删除（SQL语言）

答：create database student

Drop database student

1. 模式创建和删除（SQL语言）

答：create schema test

Drop schema test cascade/restrict

1. 表定义、修改和删除（SQL语言、列级完整性约束条件、表级完整性约束条件、主键和外键）

答：--表定义

create table student

(sno char(9) primary key,

Sname varchar(20) unique,

Ssex char(2),

Sage smallint,

Sdept char(20)

);

Create table course

(cno char(4) primary key,

Cname char(40),

Cpno char(4),

Ccride smallint,

Foreign key (cpno) references course(cno)

);

Create table sc

(sno char(9),

Cno char(4),

Grade smallint,

Primary key (sno,cno),

Foreign key (sno) references student(sno),

Foreign key (cno) references course(cno)

);

--修改表

add子句：增加新列和新的完整性约束条件

drop子句：删除指定的完整性约束条件

Alter column子句：用于修改列名和数据类型

Alter table student add sentrance date;

Alter table course add unique(cname);

Alter table student alter column sage int;

命名子句？？

--删除表

Drop table student [cascade/restrict]

1. 索引的建立和删除（SQL语言、唯一性索引和聚簇索引）

答：建立索引是加快查询速度的有效手段；但需要占用一定的存储空间

索引是关系数据库管理系统的 内部实现技术，属于内模式的范畴

--建立索引

Unique唯一性索引；

对于已含重复值的属性列不能建立unique索引

对某个列建立unique索引后，插入新记录时DBMS会自动检查新记录在该列上是否取了重复值。这相当于增加了一个unique约束。

cluster聚簇索引；

建立聚簇索引后，基表中数据也需要按指定的聚簇属性值的升序或降序存放。也即聚簇索引的索引顺序与表中记录的物理顺序一致

在一个基本表上最多只能建立一个聚簇索引

聚簇索引的用途：对于某些类型的查询，可以提高查询效率

聚簇索引的适用范围：

很少对基表进行增删操作

很少对其中的变长列进行修改操作

Create unique index stusno on student(sno);

Create unique index coucno on course(cno);

Create unique index scno on sc(sno ASC, cno DESC);

--修改索引

Alter index scno rename to scsno;

--删除索引

删除索引时，系统会从数据字典中删去有关该索引的描述

Drop index stuname;

创建索引 聚集索引 索引函数

1. 数据查询语句概述

答：基本语法

子句功能：

Select——列出查询的结果

From——指明所访问的对象

Where——指定查询的条件

Group by——将查询结果按指定字段的取值分组

Having——筛选出满足指定条件的组

Order by——按指定的字段的值，以升序或降序排列查询结果

Select语句的含义：

1. 单表查询（投影查询、选择查询、order by字句、聚集函数、group by字句）

答：投影查询：

--属性名

Select sno

From student;

--属性名表达式、常量或函数

Select sname, 2017-sage as birth（列别名）

From student;

Select COUNT(sname) 学生人数（列名）

From student；

Select sname, ‘2017’（在每个学生的名字后面显示字符串2017）

From student;

选择查询：

--取消重复行

Select distinct cno, grade

From sc;

where子句常用的查询条件

|  |  |
| --- | --- |
| 查询条件 | 谓词 |
| 比较 | =，>，<，>=，<=，!=，<>，!>，!<，  NOT+上述比较运算符 |
| 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（逻辑运算） | And，or，not |

Select sname, ssex

From student

Where sdept in（‘is’，’ma’，’cs’）;

//where sname like ‘刘%’;

//where sname like ‘欧阳\_ \_’;（一个汉字是两个字符）

//where cname like ‘DB\\_%i\_ \_’ escape’\’;（转义字符 escape’\’）

Order by子句：

当排序列含空值时：

ASC：排序列为空值的元组最先显示

DESC：排序列为空值的元组最后显示

Select \*

Form student

Order by sdept ASC, sage Desc;

聚集函数：

主要聚集函数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 参数类型（列名） | 结果类型 | 描述 |
| COUNT | 任意或\* | 数值 | 计数 |
| SUM | 数值型 | 数值 | 计数总和 |
| AVG | 数值型 | 数值 | 计数平均值 |
| MAX | 数值型、字符型 | 同参数类型一样 | 求最大值 |
| MIN | 数值型、字符型 | 同参数类型一样 | 求最小值 |

Distinct短语：在计算时要取消指定列中的重复值

All语句：缺省值，不取消重复值

Where子句中不能有聚集函数，聚集函数只能放在having或select中

Select COUNT（distinct sno）

From sc;

Group by子句：

使用group by子句分组

未对查询结果分组，聚集函数将作用于整个查询结果

对查询结果分组后，聚集函数将作用于每个组

Select cno 课程号，COUNT（sno） 人数（列别名）

From sc

Group by cno;

//有group by时，select中只能出现group by中的字段或者聚集函数

Select sno

From sc

Group by sno

Having COUNT(\*)>3;

Where子句作用于基表或视图，从中选择满足条件的元组

Having短语作用于组，从中选择满足条件的组

1. 连接查询（等值连接、非等值连接、自然连接、外连接、符合条件连接）

答：同时涉及多个表的查询称为连接查询

等值连接：若连接运算符为=时，称为等值连接

Select student.\*, sc.\*

From student, sc

Where student.sno=sc.sno

非等值连接：使用其他运算符时，称为非等值连接

自然连接：在等值连接中，去掉目标列中的重复属性则为自然连接

Select student.sno,sname,ssex,sage,sdept,cno,grade

From student,sc

Where student.sno=sc.sno

自身连接：一个表与其自己进行连接，称为表的自身连接

需要给表起别名以示区别

由于所有属性名都是同名属性，因此必须使用别名前缀

（可能会考）

Select first.cno, second.cpno

From course first ,course second

Where first.cpno=second.cno;（先修课的先修课）

外链接：外链接与普通连接的区别

普通连接操作只输出满足连接条件的元组

外链接操作以指定表为连接主体，将主体表中不满足连接条件的元组一并输出

非主体表有一“万能”的虚行，该行全部由空值组成

虚表可以和主体表中所有不满足连接条件的元组进行连接

由于虚表各列全部是空值，因此与虚行连接的结果中，来自非主体表的属性值全部是空值

//外链接：

Select first.cno,second.cpno

From course first full outer join course second

On first.cpno=second.cno;

//左外链接：先写的表作为左表，主体；后写的是右表

Select student.sno,sname,ssex,sage,sdept,cno,grade

From student left outer join sc

On student.sno=sc.sno;

//右连接：

Select first.cno,second.cpno

From course first right outer join course second

On first.cpno=second.cno;

//符合条件连接

Where子句中含多个连接条件，称为符合条件连接

1. 嵌套查询（概述、相关子查询、不相关子查询、嵌套查询求解方法、引出子查询的谓词）

答：嵌套查询概述：

（子查询的限制：不能使用order by子句；有些嵌套查询可以用连接运算代替）

不相关子查询：（子查询的查询条件不依赖父查询）

Select sname

From student

Where sno in

(select sno

From sc

Where cno=’2’

);

相关子查询：（子查询的查询条件依赖父查询）

Select sno,cno

From sc x

Where Grade>=

(select AVG(grade)

From sc y

Where y.sno=x.sno

);

嵌套查询求解方法：

引出子查询的谓词：

1. 引出子查询的谓词（IN、ALL、ANY、EXISTS、NOT EXISTS）

答：带有in谓词的子查询

带有比较运算符的子查询

带有all或any谓词的子查询

Select sname,sage

From student

Where sage<any

(select sage from student

Where sdept=’IS’)

And sdept <>’IS’;

带有exists谓词的子查询

（由exists引出的子查询，其目标列表表达式通常用\*）

Select sname

Form student

Where not exists

(select \*

From sc

Where sno = student.sno and cno=’1’);

1. 集合查询

答：并操作：UNION

Select \* from student

Where sdept=’CS’

UNION

Select \* from student

Where sage<=19;

//

Select distinct \*

From student

Where sdept=’CS’ OR sage<=19;

交操作：INTERSECT

Select \* From student

Where sdept=’CS’

INTERSECT

Select \* from student

Where sage<=19;

//

Select \*

From student

Where sdept=’CS’ and sage<=19;

差操作：EXCEPT

Select \* From student

Where sdept=’CS’

EXCEPT

Select \* from student

Where sage<=19;

//

Select \*

From student

Where sdept=’CS’ and sage>19;

1. 数据的插入（必考）

答：

插入元组

Insert

Into student(sno,sname,sage,sex)

Values(01,’张’,’女’,’16’);

插入子查询结果

CREATE TABLE Deptage

(Sdept CHAR(15) ,

Avgage SMALLINT)；

Insert

Into deptage(sdept,avage)

Select sdept,avg(sage)

From student

Group by sdept;’

1. 数据的修改

答：

修改某元组的值

Update student

Set sname=’张张’

Where sno=’01’;

修改多个元组的值

Update student

Set sage=sage+1;

带子查询的修改语句

14.数据的删除（必考）

答：

删除某一个元组的值

Delete

From student

Where sno=’01’;

删除多个元组的值

Delete

From sc

带子查询的删除语句

DELETE

FROM SC

WHERE Sno in

(SELECT Sno

FROM student

WHERE student.Sdept=‘CS’)

1. 视图的定义（必考）

答：

create view student [列名]

As<子查询>

With check option

Create view is\_student

As

Select sno,sname

From student

Where sdept=’IS’

With check option;

CREATE VIEW IS\_S1(Sno，Sname，Grade)

AS

SELECT Student.Sno，Sname，Grade

FROM Student，SC

WHERE Student.Sno=SC.Sno AND

Sdept= 'IS' AND

SC.Cno= '1'；

1. 删除视图

答：drop view IS

17.查询视图

答：

1. 更新视图

答：

不允许更新视图的几种情况：

若视图是由两个以上基本表导出的，则不允许更新

若视图字段来自聚集函数，不允许更新

若视图定义中含有group by子句，则不允许更新

若视图定义中有distinct，不允许更新

一个不允许更新的视图上定义的视图，不允许更新

有嵌套查询的，并且内层查询的from子句中涉及的表也是导出该视图的基本表，不允许更新

视图字段来自表达式或者常数，只允许执行delete操作，insert和update不允许

19.视图的作用

答：

视图能够简化用户的操作

视图使用户能以多种角度看待同一数据

视图对重构数据库提供了一定程度的逻辑独立性

视图能够对机密数据提供安全保护

适当的利用视图可以更清楚的表达查询

本章考核要求：

3.1 掌握：数据定义、数据查询、数据更新：增、删、改，视图的定义、删除、查询、更新

3.2 了解：视图的作用

3.3 难点：数据查询和数据更新和视图