## 一、填空题：共10分

1、 数据库的本质特征：永久存储、有组织、可共享

数据库的目标特征：较小的冗余、较高的独立性、较强的扩展性

1. DBMS的主要功能：DDL、DML、DCL

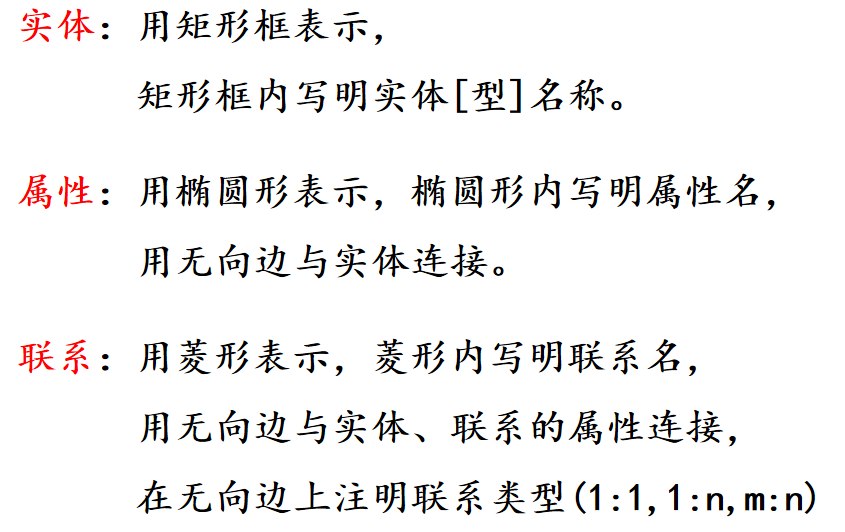
传统的数据模型：层次模型、网状模型、关系模型

1. 关系代数传统的集合运算：并、差、交、积
2. 关系代数专门的关系运算：选择、投影、连接、除.

## 单选题：共10分

1. 网状模型的数据结构：图结构

层次模型的数据结构：树结构

1. E-R图的属性、联系、实体



3、

4、2NF在1NF的基础上，消除了非主属性对码的部分函数依赖。

3NF在2NF的基础上，消除了非主属性对码的传递函数依赖。

3NF在1NF的基础上，消除了非主属性对码的部分函数依赖和传递函数依赖。

1. 二维表，一行对应的关系：元组

一行一列的值的对应关系：分量

元组(Tuple)：对应表中的一行

分量(Component)：元组中的一个属性值

1. 选择若干列：投影（投影运算是从列的角度对关系进行运算。）

选择若干行：选择（选择运算是从行的角度对关系进行运算。）

7、关系R(A,B,C)，关系S(C,D,E,F)，则R×S的属性个数是：7

关系R(A,B,C)，关系S(C,D,E,F)，则R /S的属性个数是：2

8、设W=R ⋈S，W、R、S的属性个数分别为w、r、s，那么三者之间满足：w< (r+s)

[ 自然连接是关系R和S在所有公共属性（common attribute）上的等接（Equijoin）。但在得到的结果中公共属性只保留一次，其余删除。]

设关系R(A,B,C)有10个元组，关系S(D,E,F,G)有20个元组，

则R ×S的元组的个数是：200（10 × 20=200）

1. 物理结构设计的成果：内模式

概念结构设计的成果：E-R图

1. NULL表示不确定

数据库管理系统：DBMS

## 简答题：4小题，共20分

1、关系的实体完整性规则

实体完整性规则：

若属性A是基本关系R的主属性，

则属性A不能取空值。

关系的参照完整性规则

参照完整性规则：

若属性(组)F是基本关系R的外码，它与基本关系S的主码K 相对应，则对于R中每个元组在F上的值必须为：

①或者取空值；

②或者等于S中某个元组的主码值。

2、数据库的逻辑独立性

当数据库的模式改变时，只需修改外模式到模式的映像，使外模式保持不变；由于应用程序只与外模式相关，从而应用程序不变。这种数据的逻辑结构和应用程序之间的独立性称为数据的逻辑独立性。

数据库的物理独立性

当数据库的内模式改变时，只需要修改内模式到模式的映像，使模式不变，外模式也不变；由于应用程序只与外模式相关， 从而应用程序不变。这种数据的物理结构与应用程序之间的独立性称为数据的物理独立性。

1. 并发可能导致的数据不一致的情形。

（1）丢失修改：事务T1和T2同时读入同一数据并修改，T2修改的的结果覆盖了T1修改的结果，丢失了T1的修改。这种情形称为丢失修改(Lost Update)

（2）读"脏"数据：事务T1修改数据后，将其写回数据库；事务T2读T1写回的数据。由于某种原因T1撤消，T1前面修改的数据恢复原值；T2读到的数据与数据库中的数据不一致,是"脏"数据。这种情形称为读"脏"数据(Dirty Read)

（3）不可重复读：事务T2第一次读取数据a后；事务T1修改a；若T2再一次读a，同一事务中两次读a的结果不同。这种情形称为不可重复读(Non-Repeatable Read)

（丢失修改：

两个事务T1和T2读入同一数据并修改，T2提交的结果破坏了T1提交的结果，导致T1的修改被丢失

不可重复读：

是指事务T1读取数据后，事务T2执行更新操作，使T1无法再现前一次读取结果。

读“脏”数据

是指事务T1修改某一数据并将其写回磁盘，事务T2读取同一数据后，T1由于某种原因被撤销，这时被T1修改过的数据恢复原值，T2读到的数据就与数据库中的数据不一致，则T2读到的数据就为“脏”数据。）

数据库系统可能产生的故障。

（1）事务故障：某一事务在运行过程中发生故障，导致该事务无法继续运行。

（2）系统故障：数据库系统在运行过程中，突然发生硬件损坏、操作系统崩溃、DBMS错误等，导致整个数据库系统停止运行。

（3）介质故障：存储介质损坏而产生的故障。

（事务内部故障（事务故障）

有的是可以通过事务程序本身发现的，有的是非预期的，不能由事务程序处理。更多的故障是非预期的。

系统故障

是指造成系统停止运转的任何事件，使得系统要重新启动

介质故障

指外存故障。这类故障将破坏数据库或部分数据库，并影响正在存取这部分数据的所有事务。）

4、死锁的预防。

①一次封锁法：一次封锁法要求每个事务必须一次将所有要使用的数据对象全部加锁，否则就不能继续执行。

②顺序封锁法：顺序封锁法对所有数据对象规定一个封锁顺序，所有事务都按这个顺序实行封锁。

死锁的检测（诊断）。

①超时法：如果一个事务等待的时间超过了规定时限，就诊断为发生了死锁。

②等待图法：事务等待图是一个有向图G=(T,U)T为结点集合，每个结点表示正运行的事务。U为有向边集合，每条边表示事务等待的情况。

## 四、应用题：10小题，共30分

关系代数表达和SQL语句

学生：student(sno,sname,sage) sno学号，sname姓名，sage年龄

课程：course(cno,cname) cno课号，cname课名

学习：study(sno,cno,grade) sno学号，cno课号，grade成绩

1、2用关系代数表达式，其他用sql语句

1. 查询所有课程的课号、课名

Πcno.name(course)

1. 查询年龄小于20的学生学号、姓名、年龄

δsage<20(student)

1. 查询所有姓'张'的学生学号、姓名

Select sno, sname

From student

Where sname like ‘张%’

1. 查询每门课程的平均成绩

SELECT cno avg(grade) from study Group By cno

1. 查询年龄大于所有学生平均年龄的学号、姓名

Select sno,sname from student where sage>(SELECT AVG(sage) from student)

1. 查询学生成绩单，成绩单包括学号、姓名、课名、成绩。

Selcet student.sno,sname,cname,grade from student ,study ,course

where student.sno=study.sno and course.cno=study.cno

1. 查询所有学生的学号、姓名、课号、成绩(包括没有学习任何课程的学生)。

Select student. sno,sname,cno,grade from student left join study on (student.sno=study.sno);

1. 增加student表学生记录('S1',王敏',21)。

Insert into student values(‘s1’, ‘王敏 ’ ,21)

1. 删除study表中学号为'S1'的学生学习课号为'C1'课程的记录

Delete from study where sno=’s1’ and cno=’c1’

1. 修改course表中课号为'C1'的课名为'数据结构'。

Update course set cname=’数据结构’ where cno=’c1’

eg：

学生：student(sno,sname,sage) sno学号，sname姓名，sage年龄

课程：course(cno,cname) cno课号，cname课名

学习：study(sno,cno,grade) sno学号，cno课号，grade成绩

要求：1-2小题用关系代数表达式实现，3-10小题用SQL实现。

1. 查询所有课程的课号、课名。

∏cno,cname(course);

1. 查询年龄小于20的学生学号、姓名、年龄。

δsage<20(student); ∏sno,sname,sage(δsage<20(student)) ;

1. 查询所有姓'张'的学生学号、姓名。

Select sno,sname (p95) %是姓刘的学生

From student -是：张某，后面只有一个字

Where sname like ‘张%’ ; not like ‘张%’：不姓张；-雪：名字中第二个字为“雪”

1. 查询每门课程的平均成绩。

Select cno,avg(grade)

From study

Group by cno;

1. 查询年龄大于所有学生平均年龄的学号、姓名。

Select sno,sname

From student

Where sage>(select avg(sage)

From student

);

1. 查询学生成绩单，成绩单包括学号、姓名、课名、成绩。

Select student.sno,sname,cname,grade

From student,course,study

Where student.sno=study.sno and study.cno=course.cno;

1. 查询所有学生的学号、姓名、课号、成绩(包括没有学习任何课程的学生)。

Select student.sno,sname,cno,grade

From student left outer join

No student.sno=study.sno;

右外连接

查询所有系部的学号、姓名、系部名

SELECT dname,sno,sname

FROM student RIGHT OUTER JOIN dept

ON(student.dno=dept.dno);

等价写法：

SELECT dname,sno,sname

FROM dept LEFT OUTER JOIN student

ON(dept.dno=student.dno);

1. 增加student表学生记录('S1',王敏',21)。

Insert into student Values ('S1',王敏',21);

向学习study(sno,cno,grade)表中插入一行值(‘S1’,‘C2’)。 （只包含部分数据）

Insert into study(sno,cno) Values('S1','C2');

9、 删除study表中学号为'S1'的学生学习课号为'C1'课程的记录

Delete from study

Where sno=’s1’and cno=’c1’;

例1:删除employee表 CASCADE是级联删除，删除表的同时，删除表所附带的视图、

DROP TABLE employee CASCADE; 索引、触发器、过程和函数等。

例2:删除student表 RESTRICT是限制删除，删除的表不能有附带的视图、索引、

DROP TABLE student; 过程和函数、触发器、参照表等

1. 修改course表中课号为'C1'的课名为'数据结构'。

Update course set cname=’数据结构’

Where cno=’c1’;

11.查姓名中有‘清’字的学生

SELECT \*

FROM Student

WHERE Sname LIKE ‘%清%’;

12.查询成绩不及格的学生学号

Select distinct sno

From study

WhereE grade<60;

13.查询年龄不在20～23之间学生姓名和年龄

Select sname,sage

from student

where sage not between 20 and 23

14.查询信息系('IS')和计算机系('CS')的学生的姓名和性别

Select sname,ssex

From student

Where sdept in ('IS','CS');

或者Select sname,ssex

from student

Where (sdept='IS') or (sdept='CS');

15.查询课名以'DB\_C'打头的课号和课名 ESCAPE，取消通配，使之转义。

Select cno,cname FROM course

Where cname like 'DB\\_C%'escape '\';

16.查询成绩为空的学号和课号

Select sno,cno

From study

Where grade is nullL;

17.统计性别不为空的学生人数

Select count(\*) cn

From student

Where ssex is not null;

18.查询学生情况，按系部的降序排序，如果系部相同，按学号的升序排序

Select \*

From student

Order by sdept DESC,sno;

19.查询学生情况，按姓名的升序排序，如果姓名相同，按年龄的降序排序

Select \* from student

Order by sname,sage desc;

20.查询学习了3门和3门以上课程的学生学号

Select sno

From study

Group by sno

Having count(\*)>=3;

21.查询学习人数在30个以上的每一门课程的平均成绩

Select cno,AVG(grade) 平均成绩

From study

Group by cno

Having count(\*)>=30;

22.查询及格人数在30个以上的课号

Select cno

From study

WhereE grade>=60

Group by cno

Having count(\*)>=30;

23.删除计算机系('CS')所有学习成绩

Delete

from study

Where sno in

(Select sno

From student

Where sdept='CS');

在插入行时，不能违反完整性约束条件：

①PRIMARY KEY 列值不能重复，也不能取空值 (实体完整性)。

②FOREIGN KEY 列值要么取被引用表的主码值， 要么为NULL(参照完整性)。

③NOT NULL 列的值不能为空。

④UNIQUE 列的值不能重复。

⑤CHECK 使函数值为真。

24.查询学习了'C1'号课程和'C2'号课程的学号

SELECT sno FROM study WHERE cno='C1'

INTERSECT

SELECT sno FROM study WHERE cno='C2'; 注：一般相同列的多个条件不能用 AND

25.查计算机系('CS')且年龄不等于20学生的

学号、姓名

SELECT sno,sname FROM student WHERE sdept='CS'

EXCEPT

SELECT sno,sname FROM student WHERE sage=20;

等价于：

SELECT sno,sname

FROM student

WHERE (sedpt='CS') AND (sage<>20);

## 计算题，4小题，共20分

**1、设有关系模式，求Fmin，写出计算步骤。**

**R(U,F),U={A,B,C,D},F={A→C,B→AC,BD→A,C→A,D→AC}**

**2、设有关系模式，求出R的所有码，写出计算步骤。**

**R(U,F),U={A,B,C,D},Fmin={D→B,B→D,AC→D})**

**3、判断上述第2题中的关系模式R满足的最高范式，说明判断理由。**

**4、判断下面的模式分解ρ是否为无损连接的分解，写出判断过程。**

**R(U,F),U={A,B,C,D},Fmin={A→B,C→D,D→B}**

**ρ={R1(AD),R2(BC),R3(BD)}**

**1、设有关系模式，求Fmin，写出计算步骤。**

**R(U,F),U={A,B,C,D},F={AB→CD,C→A,A→D,D→B}**

**第一步：将F中函数依赖的右边分解单属性，得到**

**F={AB→C,AB→D,C→A,A→D,D→B}**

**第二步：去掉多余依赖**

**AB→C G={ AB→D,C→A,A→D,D→B} ABG+={ABD} 保留AB→C**

**AB→D G={AB→C, C→A,A→D,D→B} ABG+={ABCD} 去掉AB→D**

**C→A G={AB→C, A→D,D→B} CG+={C} 保留C→A**

**A→D G={AB→C, C→A, D→B} AG+={A} 保留A→D**

**D→B G={AB→C, C→A, A→D } DG+={D} 保留D→B**

**得到：F={AB→C,C→A,A→D,D→B}**

**第三步：去掉左部多余属性**

**A→C:AF+={ADBC}用A→C代替AB→C**

**Fmin={A→C,C→A,A→D,D→B}**

2.求码、判断最高范式

第1步：求Fmin

第2步：将属性集U中的属性划分为3集合：

UL：UL=左部属性集-右部属性集

UR：UR=右部属性集-左部属性集

UB：UB=U-UL-UR

第3步：若UL ，则求(UL)F＋

若(UL)F＋=U，则UL为R的唯一码，算法终止。

若(UL)F＋  U，则转向第4步；

若UL=，则转向第5步；

第4步：若UB，UL依次与UB中属性组合后求组合属性的闭包。根据码的定义求出所有的码，算法终止。

第5步：若UB，求UB中子集的属性闭包，根据码的定义求出所有的码，算法终止。

说明： 上述的F就是Fmin

最高范式的判断

从低往高判断法

1）根据求出的码，确定主属性和非主属性

2）判断Fmin函数依赖中，是否存在非主属性对码的部分函数依赖

存在：则该关系模式属于1NF

不存在，则进入下一步

3）判断Fmin函数依赖中，是否存在非主属性对码的传递函数依赖

存在：则该关系模式属于2NF

不存在，则进入下一步

4）判断Fmin函数依赖中每一个函数依赖的决定因素（右边）是否都包含有码

不是：则该关系模式属于3NF

是：则该关系模式属于BCNF

**2、设有关系模式，求出R的所有码，写出计算步骤。**

**R(U,F),U={A,B,C,D},Fmin={A→B,A→D,B→C}**

**判断上述关系模式R满足的最高范式，说明判断理由。**

**第一步：求**

**UL={AB}-{BCD}={A}**

**UR={BCD}-{AB}={CD}**

**UB={ABCD}-{A}-{CD}={B}**

**由于UL不为空，而(A)F+={ABCD}=U**

**故A为唯一的码。**

**判断最高范式**

**①因为A为唯一码，且是单属性码，所以不存在非主属性对码的部分函数依赖，故R∈2NF。**

**②对于非主属性C有：A→B,B→C，存在对码A的传递函数依赖，故R∉3NF。**

**③所以R的最高范式为2NF**

3.判断是否为保持函数依赖的分解

(1)构造初始表

构造一个k行n列的表

每行对应Ri(Ui)(i=1,2,…,k)的属性集Ui

每列对应一个属性Aj(j=1,2,…,n)

如果Aj在Ui中，在第i行第j列填上aj

否则填上bij

(2)修改表

取F（即Fmin）中的每一个函数依赖X→Y

找出表中在X分量上相等的行，

修改这些行在Y分量上的值，

使这些行在Y分量上的值也相等。

修改分为两种情况：

①如果这些行的Y分量中有一个是aj

全部修改成aj

②如果这些行的Y分量中没有aj

全部修改成行号i较小的bij

(3)判断

修改完成后的表格中若有一行全是a

即有一行为： a1,a2,…,an

则相对于F是无损连接分解

否则相对于F是有损连接分解

**3、判断下面的模式分解ρ是否为无损连接的分解，写出判断过程。**

**R(U,F),U={A,B,C,D},Fmin={A→B,B→C,C→D}**

**ρ={R1(AB),R2(BC),R3(CD)}**

4.判断是否为无损连接分解

①令G=∪Fi，F=F-G，Result＝True

②对于F中的函数依赖X→Y，计算XG＋

并令F=F-{X→Y}

③若Y ! XG＋，则令Result＝False，转向④

否则，若F≠Φ，转向②；

若F=Φ， 转向④

④若Result＝True， 则是保持依赖的分解；

若Result＝False，则不是保持依赖的分解。

关键是计算分解后的函数依赖并集G=∪Fi

Fi=Ui(F)={ X→Y | X→YF＋ ∧ XYUi }

注：函数依赖Fi是针对F+，而不是F

## 六、设计题：共10分

画E-R图

将E-R图转换为模式

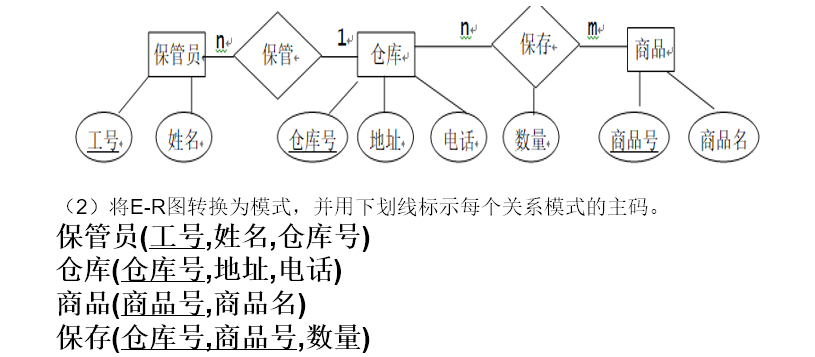
1. 设有仓库、保管员、商品三个实体集。仓库有仓库号、地址、电话三个属性，保管员有工号、姓名两个属性，商品有商品号、商品名两个属性。

一个仓库可由多名保管员，一个保管员只能"保管"一个仓库；

一个仓库可"保存"多种商品，一种商品可在多个仓库中"保存"，商品"保存"在某仓库中有数量属性。

（1）根据语义，画出E-R图，标明实体及其属性、联系及其类型和属性，并用下划线标示每个实体的主码。

（2）将E-R图转换为模式，并用下划线标示每个关系模式的主码。



2. 设有班级、学生、课程三个实体集。班级有班号、班名两个属性，学生有学号、姓名、性别三个属性，课程有课号、课名两个属性。

一个班级可"拥有"多名学生，一个学生只能在一个班级；

一个学生可"学习"多门课程，一门课程可被多名学生"学习"，学生"学习"课程有成绩属性。

（1）根据语义，画出E-R图，标明实体及其属性、联系及其类型和属性，并用下划线标示每个实体的主码。

（2）将E-R图转换为模式，并用下划线标示每个关系模式的主码。

