

复旦大学计算机科学技术学院

2022-2023 第一学期《数据库引论》期中考试试卷

A 卷 共 4 页

课程代码: COMP130010.01

考试形式: ☐ 开卷 ☒ 闭卷

2022 年 11 月

(本试卷答卷时间为 120 分钟, 答案必须写在试卷上, 做在草稿纸上无效)

专业_____学号_____姓名_____成绩_____

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

一、单项选择题 (本大题共 8 小题, 每小题 2 分, 共 16 分)

- 下列关于 DB 的**数据模型**的叙述中, 正确的是_____ **(D)**。
 - 概念模型从用户使用观点出发对数据建模, 它是**设计人员**对应用项目数据库的描述。
 - 逻辑模型中的层次模型采用有向图结构表达实体类型以及实体间的联系。
 - 内部模型又称为物理模型, 它独立于软件但依赖于硬件。
 - 表达用户使用观点的 DB 全局逻辑结构的模型称为“外部模型”。
- 下列关于关系模型和关系代数语言的说法中, 错误的是_____ **(C)**。
 - 关系是一个属性数目相同元组的集合, 因此不考虑元组间的顺序。
 - 关系模型的完整性规则要求元组在组成的主键属性上不能有空值。
 - R 和 S 的半连接操作定义为 R 和 S 的自然连接在关系 S 的属性集上的投影。
 - 笛卡尔积是一个关系代数的基本操作, 不能由其它基本操作推出。
- 有如下表结构: S (S#, SNAME, AGE, SEX), SC (S#, C#, GRADE), C (C#, CNAME, Teacher-name) 关于这个表的 SQL 语句, 下列说法中正确的是: _____ **(A)** 。
 - 对基本表 SC 中的(S#, C#)建立索引, 可用语句: CREATE INDEX SC_INDEX ON SC(S#, C#);
 - 查询所有课程中平均成绩大于 80 分的课程号, 可用语句: SELECT DISTINCT C# FROM SC GROUP BY C# WHERE AVG (GRADE) > 80;
 - 在表 S 中将 S#的长度修改为 6, 可用语句: ALTER TABLE S UPDATE S# CHAR(6);
 - 将 C 表删除的同时也将所有引用的视图和完整性约束删除, 可用语句: DROP TABLE C RESTRICT;

(装订线内不要答题)

- 下列关于数据库语言 SQL 的描述中，下列说法中错误的是 (B)。
 - 在基本表的视图中，对于视图的查询操作与基本表一样，但对于视图中元组的更新操作不同，包含基本表主键的视图是允许更新的。
 - 在嵌入式 SQL 中，当 SELECT 语句的查询结果是多个元组时，要用游标机制把多个元组一次性全部传送给主语言程序处理。
 - 在 SQL 的连接操作中，右外连接是返回右表所有数据和左右表连接字段相等的记录。
 - SQL3 中用 WITH RECURSIVE 子句来支持递归查询的有限形式。
- 设 U 是关系模式 R 的属性集， F 是 R 上成立的只涉及 U 中属性的函数依赖集，以下推理规则中正确的是 (D)。
 - 自反性：若 $Y \subseteq X \subseteq U$ ，则 $Y \rightarrow X$ 。
 - 增广性：若 $X \rightarrow Y$ ， $W \rightarrow Z$ ，则 $XW \rightarrow YZ$ 。
 - 分解性：若 $X \rightarrow Y$ ， $Z \subseteq Y$ ，则 $ZX \rightarrow Y$ 。
 - 伪传递性：若 $X \rightarrow Y$ ， $WY \rightarrow Z$ ，则 $WX \rightarrow Z$ 。
- 设关系模式 $R(A, B, C)$ ， $F=\{A \rightarrow B, B \rightarrow A, C \rightarrow A\}$ ，请判断这是第几范式 (B) 的模式。
 - 1NF
 - 2NF
 - 3NF
 - BCNF
- 对于关系模式 $R(ABC)$ ， F 是 R 上成立的 FD 集， $F=\{A \rightarrow B, B \rightarrow C\}$ ，在 F 的闭包中，左部为 (AB) 的函数依赖有 (C)。
 - 2 个
 - 4 个
 - 8 个
 - 16 个
- 在有关 ER 模型的叙述中，正确的是 (C)。
 - 在 ER 模型中，联系用于描述联系的特征，不拥有属性。
 - 在 ER 模型中，弱实体通常用双线矩形框表示，参与联系时部分参与。
 - 弱实体的主键的一部分或者全部从其强实体中获得。
 - 子类与超类具有继承性特点，这种继承性是通过有不同的实体标识符实现的。

4. 在 SQL 语句中，条件表达式中进行字符串模糊匹配的操作符是 (LIKE) (%)
5. 在 ER 模型中，参与一个联系的实体集个数，称为该联系的 (元数或者度数) (关系模式中属性的个数称为元数，元组的个数称为基数)

三、简答题（本题 12 分）

设有关系模式 $R\{A, B, C, D, E\}$ ，其上的函数依赖集为： $F=\{A \rightarrow C, C \rightarrow D, B \rightarrow C, DE \rightarrow C, CE \rightarrow A\}$ ；g

- (1) AE 和 BE 是 R 的候选码吗？试分别进行说明。（4 分）**从 F 中看出，候选码至少包含 BE（因为它们不依赖于任何属性），而 $(BE)^+ = ABCDE$ ，所以 BE 是 R 的唯一候选码。**

- (2) 判断 $P = \{AD, AB, BC, CDE, AE\}$ 是否为无损连接分解？（5 分）

R_i	A	B	C	D	E
AD	a_1		a_3	a_4	
AB	a_1	a_2	a_3	a_4	
BC		a_2	a_3	a_4	
CDE	a_1		a_3	a_4	a_5
AE	a_1		a_3	a_4	a_5

其中没有全 a 行，由此判定 P 不具有无损连接性。

- (3) 将 R 分解为 BCNF，并具有无损连接性。（3 分）

考虑 $A \rightarrow C$ ，因为 ABCDE 不是 BCNF ($A \rightarrow C$ 的左部不是候选码 BE)，将 ABCDE 分解为 AC 和 ABDE，AC 已是 BCNF。进一步分解 ABDE，选择 $B \rightarrow D$ ，把 ABDE 分解为 BD 和 ABE，此时 BD 和 ABE 均为 BCNF。所以 $R = \{AC, BD, ABE\}$ 。或者 $\{ABE, BC, CD\}$ 或者 $\{BE, BC, AE, AC, CD\}$ 或者 $\{AC, BE, BD, AD, AE\}$

四、设计题（本题 34 分）

假设教学数据库中有 4 个关系：

学生关系 S (S#, SNAME, AGE, SEX)

选课关系 SC (S#, C#, SCORE)

课程关系 C (C#, CNAME, T#)

教师关系 T (T#, TNAME, TITLE)

1. 根据要求，写出 **SQL 语句**：（本小题共 18 分）

- (1) 创建选课关系 SC 表（S#和 C#为 CHAR(4)类型，SCORE 为 INT 类型，注意要指出主键和外键）。（3 分）

```
CREATE TABLE SC(
    S# CHAR(4) NOT NULL,
    C# CHAR(4) NOT NULL,
    SCORE INT,
    PRIMARY KEY(S#, C#),
    FOREIGN KEY(S#) REFERENCES S(S#),
```

FOREIGN KEY(C#) REFERENCES C(C#)

);

- (2) 从课向关系 C 中插入一个课程元组 ('C0315', 'VC++', 'T8')。(2 分)。

INSERT INTO C

VALUES('C0315', 'VC++', 'T8');

- (3) 在表 SC 中修改 C4 课程的成绩, 当成绩小于等于 70 分时提高 5%, 当成绩大于 70 分时提高 4% (用一个 UPDATE 实现) (2 分)。

UPDATE SC

SET SCORE=SCORE*CASE

WHEN SCORE>70 THEN 1.04

ELSE 1.05

END

WHERE C#='C4';

- (4) 对删除所有分数为空的 SC 记录 (2 分)。

DELETE FROM SC WHERE SCORE IS NULL;

- (5) 检索 WANG 同学不上的课程号和课程名 (3 分)

**SELECT C.C#, C.CNAME FROM C WHERE C.C# NOT IN (
SELECT SC1.C# FROM S AS S1, SC AS SC1
WHERE S1.SNAME='WANG' AND S1.S#=SC1.S#);**

- (6) 检索所授每门课程的平均成绩均大于 80 分的教师姓名, 并把检索到的值送往另一个已存在的表 FACULTY(TNAME)中 (6 分)。

1、INSERT INTO FACULTY(TNAME)

SELECT DISTINCT TNAME

FROM (SELECT TNAME, C.C#, AVG(SCORE)

FROM T,C,SC

WHERE T.T#=C.T# AND C.C#=SC.C#

GROUP BY TNAME, C.C#)

AS RESULT(TNAME,C#,AVG_SCORE) AS X

WHERE 80<ALL(SELECT AVG_SCORE

FROM RESULT AS Y

WHERE Y.TNAME=X.TNAME);

2、INSERT INTO FACULTY(TNAME)

SELECT TNAME FROM T

WHERE NOT EXISTS(

SELECT * FROM C WHERE C.T# = T.T# AND NOT EXISTS(

SELECT C# FROM SC WHERE SC.C# = C.C# GROUP BY SC.C# HAVING

AVG(SCORE) > 80))

3、INSERT INTO FACULTY(TNAME)

**SELECT T.TNAME FROM T
WHERE ALL(SELECT AVG(SC.SCORE) FROM SC,C WHERE SC.C# = C.C# AND C.T# =
T.T# GROUP BY C.C#) > 80**

2. 请用元组表达式表达下列查询：（本小题共 6 分）

- (1) 检索这样的学生学号与姓名，该学生所学课程包含学号为 S4 的同学所学所有课程。
（3 分）

$\{t | (\exists u)(S(u) \wedge (\forall v)(SC(v) \wedge (v[1]='s1' \Rightarrow (\exists w)(SC(w) \wedge w[1]=u[1] \wedge w[2]=v[2]))) \wedge t[1]=u[1] \wedge t[2]=u[2]))\}$

- (2) 检索至少有学号为 S2 和 S4 的学生选修的课程与课程号。（3 分）

$\{t | (\exists u)(\exists v)(SC(u) \wedge SC(v) \wedge u[1]='S2' \wedge v[1]='S4' \wedge u[2]=v[2] \wedge t[1]=u[2])\}$

3. 检索选修 LIU 老师课程的学生学号和姓名：（本小题共 10 分）

- (1) 用关系代数表达式表示该查询，并画出语法树（5 分）。

$\pi_{S\#,SNAME}(\sigma_{TNAME='LIU'}(S \bowtie SC \bowtie C \bowtie T))$

- (2) 根据启发式的查询语句优化算法，写出优化后的关系代数表达式，并画出语法树（5 分）。

$\pi_{S\#,SNAME} \left(\sigma_{S.S\#=SC.S\#} \left(S \right. \right.$
 $\left. \left. \times \pi_{SC.S\#} \left(\sigma_{SC.C\#=C.C\#} \left(SC \times \pi_{C.C\#} \left(\sigma_{C.T\#=T.T\#} \left(C \times \pi_{T\#} \left(\sigma_{TNAME='LIU'}(T) \right) \right) \right) \right) \right) \right) \right)$

五、综合题（本题 28 分）

上海某旅行社设计了一个小型国内旅游系统，其中涉及实体信息有：

旅游线路：路线号，起点，终点，天数

旅游班次：班次号，出发日期，回程日期，报价

旅游团：团号，团名，人数，联系人，地址，电话

游客：游客编号，姓名，性别，年龄，地址，电话

导游：导游编号，姓名，性别，年龄，地址，电话，语种，业绩

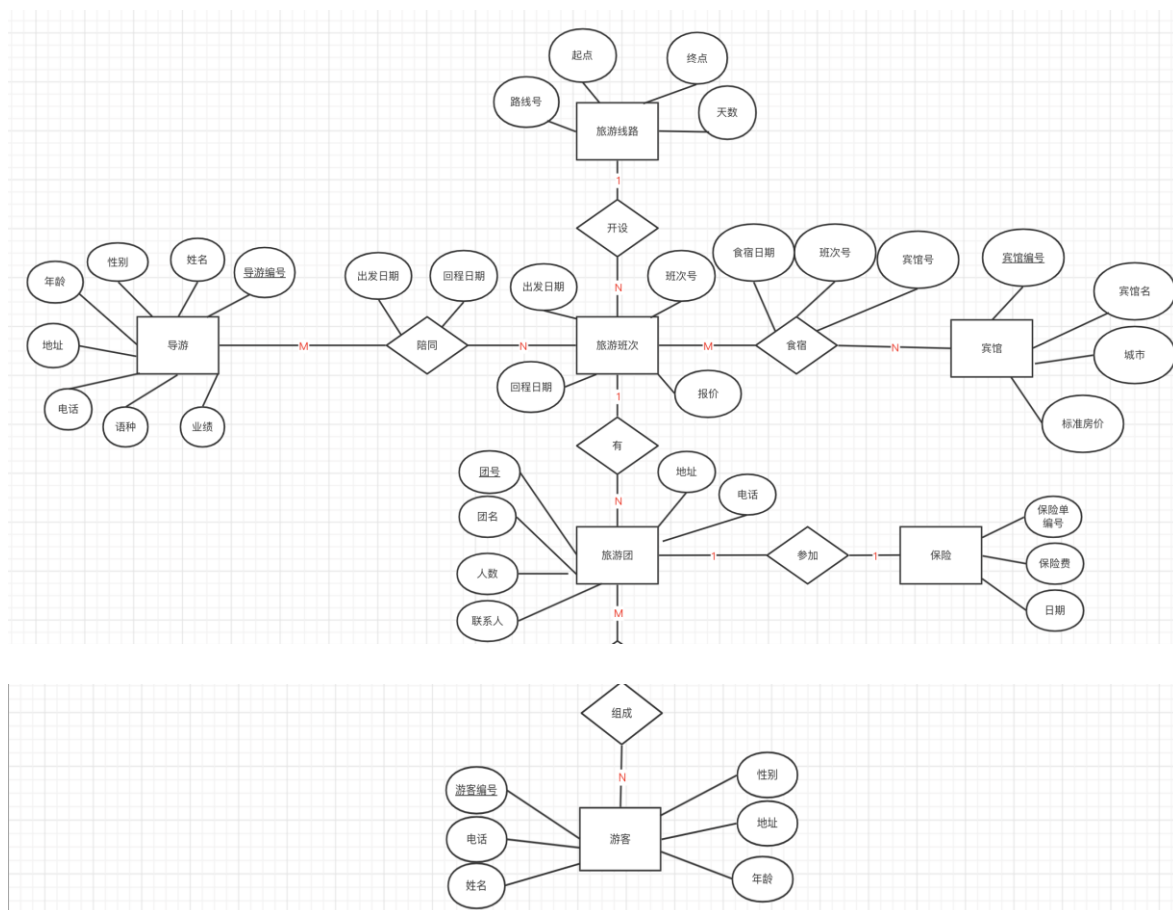
宾馆：宾馆编号，宾馆名，城市，标准房价

保险单：保险单编号，保险费，投保日期

其中，旅游线路与旅游班次存在开设关系，一条旅游线路可以开设多个旅游班次。一个旅

游班次可能有多个导游陪同，一个导游在不同时间也可以陪同不同的旅游班次，因此陪同关系具有出发和回程日期属性。旅游班次和宾馆存在食宿关系，一个旅游班次可能会在不同日期入住不同宾馆，一个宾馆也可以接待不同的旅游班次。同时一个旅游班次会有一个或多个旅游团，一个旅游团只属于一个旅游班次。旅游团和游客存在组成关系，一个旅游团由多个游客组成，每个游客也可以参加不同旅游班次的旅游团。为了保证游客的利益，每个旅游团也都会参加一份保险，每份保险也只属于一个旅游团。

1. 试为该数据库系统设计一个 ER 图，并在图上注明属性、联系类型、实体标识符。（6 分）



2. 将 ER 图转换为关系模式集，并标出每个关系模式的主键和外键。（10 分）

导游（导游编号，姓名，性别，年龄，地址，电话，语种，业绩）

旅游线路（路线号，起点，终点，天数）

旅游班次（班次号，出发日期，回程日期，报价，路线号）

保险单（保险单编号，保险费，投保日期）

旅游团（团号，团名，人数，联系人，地址，电话，班次号，保险单编号）

游客（游客编号，姓名，性别，年龄，地址，电话）

宾馆（宾馆编号，宾馆名，城市，星级，标准房价）

陪同（导游编号，班次号，出发日期，回程日期）

食宿（班次号，宾馆号，食宿日期）

组成（旅游团号，游客）

3. 讨论该数据库中存在的冗余信息，并给出改进方案，使其符合 3NF。（4 分）

对于陪同关系中，开始日期就是旅游班次的出发日期，回程日期就是旅游班次的回程日期，因此存在（导游编号，班次号）->（出发日期，回程日期），（班次号）->（出发日期，回程日期），存在局部依赖。

应该改成：陪同（导游编号，班次号）

（将旅游团中的人数去掉（也算冗余信息））

4. 根据以上的关系模式集进行论述：如果没有组成关系，直接在游客表中增加一个旅游团号字段存储所有该游客参与的旅游团（即多个旅游团号可以用预定义的特殊分隔符隔开，例如该游客参加的旅游团的团号是“A”，“B”，“C”，预定义的分隔符是“;”，则保存为“A;B;C”），这样设计相比于初步设计在插入和查询上的优势与劣势是什么（4 分）

插入时容易造成类别名称不一致：例如 a，A 等字段值大小写原因造成多种写法，实际指的是同一个旅游团。

但是在插入某个游客参加的旅游团时，不需要表连接操作，插入速度比较快。

查询时，如果查询某个游客参与的旅游团，可以减少表连接操作，查询速度会很快。

但是如果要查询 A 旅游团的游客，可能需要看每一个游客的这个字段，还要进行字符串解析，分析该游客是否属于 A 旅游团，速度就会慢很多。

5. 写出 **SQL 语句**：查询最热门的旅游班次，即该旅游班次参与的人数最多，返回班次号和参与人数。（4 分）

1、with temp as(

Select 旅游班次.班次号 as id, sum(组成.游客) as peopleNum from 旅游班次, 旅游团, 组成 where 旅游班次.班次号 = 旅游团.班次号 and 旅游团.团号 = 组成.旅游团号 group by (旅游班次.班次号))

select temp.id,max(temp.peopleNum) from temp;

2、Select 旅游班次.班次号 as id, sum(组成.游客) as peopleNum from 旅游班次, 旅游团, 组成 where 旅游班次.班次号 = 旅游团.班次号 and 旅游团.团号 = 组成.旅游团号 group by (旅游班次.班次号) order by peopleNum desc limit 1;

3、Select 旅游班次.班次号 as id, sum(组成.游客) as peopleNum from 旅游班次, 旅游团, 组成 where 旅游班次.班次号 = 旅游团.班次号 and 旅游团.团号 = 组成.旅游团号 group by (旅游班次.班次号) having peopleNum >= All(Select sum(组成.游客) from 旅游班次, 旅游团, 组成 where 旅游班次.班次号 = 旅游团.班次号 and 旅游团.团号 = 组成.旅游团号 group by (旅游

班次.班次号))

小测验

教学数据中有三个基本表:

S (S#, SNAME, AGE, SEX)

SC (S#, C#, GRADE)

C (C#, CNAME, Teacher-name)

- 1、查询老师“刘明”教的学生的sname和age, 用两种语句写
- 2、查询没有选老师“刘明”的课学生的sname, 用两种语句写
- 3、查询课程“C01”的成绩比“C02”的成绩好的学生的学号和姓名
- 4、查询学过“wang”同学学过的至少一门课程的同学的学号和姓名
- 5、统计老师“刘明”教的学生中男生和女生的数量 (一条语句)
- 6、将S1表中的数据插入S中, S1的表结构同S的表结构相同
- 7、删除c#为” c4”的课程成绩中小于60分的记录
- 8、将c#为” c4”的课程成绩小于50分的同学的成绩增加5%

1 SELECT SNAME, AGE FROM S WHERE S.S# IN (SLEECT SC.S# FROM SC WHERE SC.C# IN FROM (SELECT C# FROM C WHERE C.CNAME = 'LIUMING'))

2 SELECT SNAME FROM S WHERE S.S# IN (SLEECT SC.S# FROM SC WHERE SC.C# not IN FROM (SELECT C# FROM C WHERE C.CNAME = 'LIUMING'))

3 SELECT S#, SNAME FROM S WHERE S.S# IN (SELECT S# FROM SC AS SC_1 WHERE SC.C# = 'C01' AND SC_1.C# = 'C02' AND SC_1.GRADE > SC.GRADE)

4

5 SELECT COUNT(*) FROM S WHERE S.S# IN (SLEECT SC.S# FROM SC WHERE SC.C# IN FROM (SELECT C# FROM C WHERE C.CNAME = 'LIUMING')) GROUP BY SEX

6 INSERT INTO S VALUES (SELECT * FROM S1)

7 DELETE FROM C WHERE C.C# IN (SELECT C# FROM SC WHERE SC.C# = 'C4' AND SC.GRADE < 60);

8 UPDATE C SET C# = CASE WHEN C# IN (SELECT C# FROM SC WHERE SC.C# = 'C4' AND SC.GRADE < 50) THEN C.C#*1.05 END