

东南大学考试卷 (A 卷)

课程名称 数据库原理 考试学期 03-04-2 得分 _____
适用专业 计算机系 考试形式 开卷 考试时间长度 120 分钟

1. 什么是数据库系统？它有哪些重要组成部分？数据模型和数据语言有什么关系？
(10 分)
2. 在网状数据模型和关系数据模型中，如何表达两个记录型之间的 $m:n$ 关系？一般而言，三个实体型之间的多对多联系和这三个实体型两两之间的三个多对多联系等价吗？分别举例说明。(10 分)
3. 试回答下列有关数据库系统并发的问题： (12 分)
- (1) 什么叫并发？
 - (2) 为什么要并发？
 - (3) 并发会带来什么问题？
 - (4) 什么样的并发执行才是正确的？
 - (5) 如何避免并发所引起的问题？
 - (6) 既然目标可串行化调度比冲突可串行化调度多，我们为什么要强调冲突可串行化而非目标可串行化呢？
4. 试述提交规则和先记后写规则对更新事务的必要性。(8 分)
5. 设外关系 R 的物理块数为 b_R ，内关系 S 的物理块数为 b_S ， n_B 是可供连接的缓冲块数，简述应怎样改进连接操作的嵌套循环算法，使其总的访问物理块数为： (8 分)
- $$b_R + \lceil b_R / (n_B - 1) \rceil \times b_S$$
6. 试分析空值产生的原因。为了处理空值，DBMS 要做哪些主要工作？ (7 分)
7. 给定如下关系模式：
- $$R=(A,B,C) \quad S=(D,E,F)$$
- 设关系 $r(R)$ 和 $s(S)$ 已知，分别给出与下列表达式等价的元组关系演算表达式：
(3+3+4+5=15 分)
- (1) $\Pi_A(r)$

$$(2) \quad \sigma_{B=17}(r)$$

$$(3) \quad r \times s$$

$$(4) \quad \Pi_{A,F}(\sigma_{C=D}(r \times s))$$

8. 设学生关系 STUDENT，课程关系 COURSE 和学生选课关系 SC 的模式如下，并设计算机系的课程号以‘CS’开头，电机系的课程号以‘EE’开头，写出下列查询的 SQL 语句。(30 分)

STUDENT

SNO	SNAME	SEX	BDATE	HEIGHT
-----	-------	-----	-------	--------

COURSE

CNO	LHOUR	CREDIT	SEMESTER
-----	-------	--------	----------

SC

SNO	CNO	GRADE
-----	-----	-------

- (1) 查询不选 CS110 课的学生的姓名 (5 分)
- (2) 查询至少选修一门电机系课程的女生的姓名。(5 分)
- (3) 查询每个学生选课门数、最高成绩、最低成绩和平均成绩。(5 分)
- (4) 查询秋季学期有 2 门以上课程成绩为 90 分以上的学生的姓名。(7 分)
- (5) 查询所学每一门课程成绩均高于等于该课程平均成绩的学生的姓名及相应课程号。(8 分)

答题要点

1、要点：

- (1) 数据库系统面向数据密集型应用，以统一管理和共享数据为特征，具有存储、管理、维护数据的功能。
- (2) 数据库系统由应用程序、数据库管理系统、数据库、数据库管理员等组成。
- (3) 数据模型是用来描述现实世界数据的一组概念和定义；数据语言（数据库语言）包括数据定义、操作、查询等基本组成部分（可以以关系模型和 SQL 为例进一步展开）。

2、要点：

- (1) 网状数据模型中，通过 Link 记录来实现 M: N 关系，而关系模型中通过关系（表）来表示 M: N 关系，与网状数据模型相比，实现的是一种“软连接”；
- (2) 不等价；
- (3) 以供应商、零件、工程三者的关系为例，他们之间的一个三元多对多关系和三个两两之间的二元多对多关系是不同的，因为如果供应商 A 能够提供零件 B、A 又是工程 C 的供应商、而工程 C 又需要零件 B，并不意味着工程 C 中必须要使用供应商 A 提供的零件 B，也就是说三个两两之间的二元关系并不能决定他们之间的一个三元关系。

3、要点：

- (1) 是指 DBMS 可同时接纳多个事务，事务在时间上可以重叠执行；
- (2) 目的：①改善系统资源利用率；②改进响应时间；
- (3) 问题：①丢失更新；②读脏数据；③读值不可重复；
- (4) 可串行化；
- (5) 可采用封锁法、时间戳法等并发控制方法；
- (6) 目标可串行化的判断算法是 NP 完全问题，也没有保证目标可串行化的简单实用的规则；冲突可串行化覆盖了绝大部分可串行化的调度实例，测试算法简单、易实现。

4、要点：提交规则是保证后像在事务提交前写入非挥发存储器，这样即使事务进入提交阶段以后发生故障，仍可利用记录下来的后像重做更新，从而确保事务满足 ACID 原则；先记后写规则是指如果后像在事务提交前直接写入数据库，则必须在此之前将相应的前像写入运行记录 (Log)，以备当事务进入提交阶段之前发生故障时做 undo，使事务的执行满足 ACID 原则。

5、要点：DBMS 以物理块为基本存取单元， $n_B - 1$ 块缓冲区给外循环关系，1 块缓冲区给内循环关系，目的是尽量减少对内循环关系的扫描读取次数：

具体算法略。只需将书上原算法对外循环关系的每个元组将内循环关系读取扫描一遍，改为对外循环关系一次读取 $n_B - 1$ 个物理块到内存缓冲区，利用另一块缓冲区将内循环关系读取扫描一遍，在内存中仍利用两重循环进行两个关系元组间的两两比较。

6、要点：

- (1) 原因：①某些数据不确定；②某些数据当前仍不知道；③某些数据不存在；
- (2) DBMS 需要在查询处理中支持包括 True、False、Null 的三值逻辑；需要支持针对空值的实体完整性约束、引用完整性约束等。

7、要点：

- (1) $\Pi_A(r) = \{t(A) \mid t \in r\}$
- (2) $\sigma_{B=17}(r) = \{t \mid t \in r \wedge t.B=17\}$
- (3) $r \times s = \{t[ABCDEF] \mid t[ABC] \in r \wedge t[DEF] \in s\}$
- (4) $\Pi_{A,F}(\sigma_{C=D}(r \times s)) = \{t[AF] \mid t[ABC] \in r \wedge t[DEF] \in s \wedge t.C=t.D\}$

8、要点：

- (1) SELECT SNAME FROM STUDENT WHERE SNO NOT IN
(SELECT SNO FROM SC WHERE CNO='CS110');
- (2) SELECT SNAME FROM STUDENT, SC
WHERE STUDENT.SNO = SC.SNO AND
CNO LIKE 'EE%' AND SEX = '女';
- (3) SELECT SNO, COUNT(CNO), MAX(GRADE), MIN(GRADE), AVG(GRADE)
FROM SC
GROUP BY SNO;
- (4) SELECT SNAME FROM STUDENT
WHERE SNO IN
(SELECT SNO FROM SC
WHERE GRADE>90 AND CNO IN
(SELECT CNO FROM COURSE WHERE SEMESTER='秋')
GROUP BY SNO HAVING COUNT(*)>=2);
- (5) SELECT SNAME, CNO
FROM STUDENT, SC
WHERE STUDENT.SNO = SC.SNO AND
STUDENT.SNO NOT IN
(SELECT SNO FROM SC SCX
WHERE SCX.GRADE < (SELECT AVG(GRADE)
FROM SC SCY
WHERE SCY.CNO = SCX.CNO));

东南大学考试卷（A卷）

课程名称 数据库原理 考试学期 04-05-2 得分
适用专业 计算机系 考试形式 开卷 考试时间长度 120 分钟

1. 在网状数据模型和关系数据模型中，如何表达两个记录型之间的 $m:n$ 关系？关系模型的优点和存在的问题是什么？你认为是否会有另外一种数据模型像当初关系模型取代层次、网状模型那样取代关系模型而成为主流数据模型？为什么？(10%)

2. 给定如下关系模式：

$R=(A,B,C)$ $S=(D,E,F)$

设关系 $r(R)$ 和 $s(S)$ 已知，分别给出与下列表达式等价的元组关系演算表达式：(14%)

(1) $\sigma_{B=17}(r)$

(2) $\Pi_{A,F}(\sigma_{C=D}(r \times s))$

(3) $r * \bowtie_{C=D} s$

3. 假设有下列四个关系(36%)：

Student(sid, sname, did, sex, birthdate, gpa)

Course(cid, cname, preid, did, semester) /*preid 表示先修课的课程号*/

Enroll(sid, cid, grade)

Dept(did, dname, phdstunum) /*phdstunum 表示该系的博士生人数*/

试写出表达下列查询要求的 SQL 语句(必须用单条 SQL 语句表达)：

- (1) 用连接查询查秋季学期有一门以上课程考 90 以上的学生的姓名；
- (2) 用嵌套查询查秋季学期有一门以上课程考 90 以上的学生的姓名；
- (3) 用关联嵌套查秋季学期有一门以上课程考 90 以上的学生的姓名；
- (4) 查询只有一人选修的课程号；
- (5) 查询没有学生选修“数学分析”课程的系的名字和该系的博士生人数；
- (6) 查询所学每一门课程成绩均不低于该门课程平均成绩的学生的姓名及相应课程号；
- (7) 对计算机系所开的每门课程，查询其 cid, cname 以及选修学生的平均 gpa。

4. 为什么查询优化对关系数据库管理系统(RDBMS)来说特别重要？与网状、层次模型的数据库管理系统相比，RDBMS 的查询处理有什么本质的不同？试写出上题查询(1)的代数优化过程(10%)。

5. 试论述提交规则和先记后写规则对更新事务的必要性(10%)。

6. 试回答下列有关数据库系统并发的问题(10%)：

- (1) 什么叫并发？
- (2) 为什么要并发？
- (3) 并发会引起什么问题？

- (4) 什么样的并发执行才是正确的?
- (5) 如何避免并发所引起的问题?

7. 试分析分布式数据库系统出现的技术背景和应用背景。为什么虽然主流的数据库产品基本上都支持分布式数据库功能, 但实际应用中分布式数据库的成功案例并不多(10%)?

附加题(10%):

如果把 wait-die 策略修改为

```
if  ts( $T_A$ ) > ts( $T_B$ ) then  $T_A$  waits;  
    else{  
        rollback  $T_A$ ;  
        restart  $T_A$  with the same ts( $T_A$ );  
    }
```

- (1) 此策略能否避免死锁, 为什么?
- (2) 此策略有何主要问题? 如何改进?
- (3) 请将改进后的策略与标准的 wait-die 和 wound-die 策略进行比较。

答题要点

1. 要点:

- (1) 网状数据模型中, 通过 Link 记录来实现 M: N 关系, 而关系模型中通过关系 (表) 来表示 M: N 关系;
- (2) 优点: 关系模型中将现实世界中的实体以及实体之间的联系统一用关系 (表) 来表示, 与层次和网状数据模型中用物理指针表示实体间的联系相比, 关系模型实现的是一种“软连接”, 由于被查询的对象是关系, 查询结果还是关系, 因此可构成一个代数系统, 即关系代数。这使得我们可以用数学的方法来研究数据库中的问题, 加之关系模型概念非常简单, 又解决了查询效率的问题, 因此自 70 年代出现后很快就取代层次和网状数据模型而成为主流;
缺点: 以记录为基础, 不能很好面向用户和应用; 不能以自然的方式表达实体间联系; 语义贫乏; 数据类型太少, 不能满足应用需要;
- (3) 关系模型至今仍是主流, 是否会有新的数据模型取代它成为主流可根据自己的理解讨论, 这是一个开放的问题, 只要分析合理即可。

2. (1) $\delta_{B=17}(r) = \{t \mid t \in r \wedge t.B=17\}$

(2) $\Pi_{A,F}(\delta_{C=D}(r \times s)) = \{t[AF] \mid t[ABC] \in r \wedge t[DEF] \in s \wedge t.C=t.D\}$

(3) $r * \bowtie_{C=D} s = \{t[ABCDEFG] \mid (t[ABC] \in r \wedge t[DEF] \in s \wedge t.C=t.D) \vee (t[ABC] \in r \wedge \neg(t.C \in s[D]) \wedge t.D=NULL \wedge t.E=NULL \wedge t.F=NULL)\}$

3. Answers:

(1) SELECT sname

```
FROM Student S, Course C, Enroll E
WHERE S.sid=E.sid AND E.cid=C.cid AND E.grade>90 AND C.semester='秋季';
```

(2) SELECT sname

```
FROM Student S
WHERE S.sid IN ( SELECT E.sid
                  FROM Course C, Enroll E
                  WHERE E.cid=C.cid AND E.grade>90 AND C.semester='秋季' );
```

(3) SELECT sname

```
FROM Student S
WHERE EXISTS ( SELECT E.sid
                FROM Course C, Enroll E
                WHERE S.sid=E.sid AND E.cid=C.cid AND E.grade>90 AND
                  C.semester='秋季' );
```

(4) SELECT E.cid

```
FROM Enroll E
WHERE E.cid NOT IN (SELECT cid
                    FROM Enroll
```

WHERE sid \neq E.sid);

(5) SELECT dname, phdstunum
FROM Dept
WHERE did NOT IN (SELECT S.did
FROM Course C, Student S, Enroll E
WHERE C.cid = E.cid AND
E.sid = S.sid AND
C.cname = '数学分析');

SELECT D.dname, D.phdstunum
FROM Dept D
WHERE NOT EXISTS (SELECT S.sid
FROM Course C, Student S, Enroll E
WHERE S.did = D.did AND
C.cid = E.cid AND
E.sid = S.sid AND
C.cname = '数学分析');

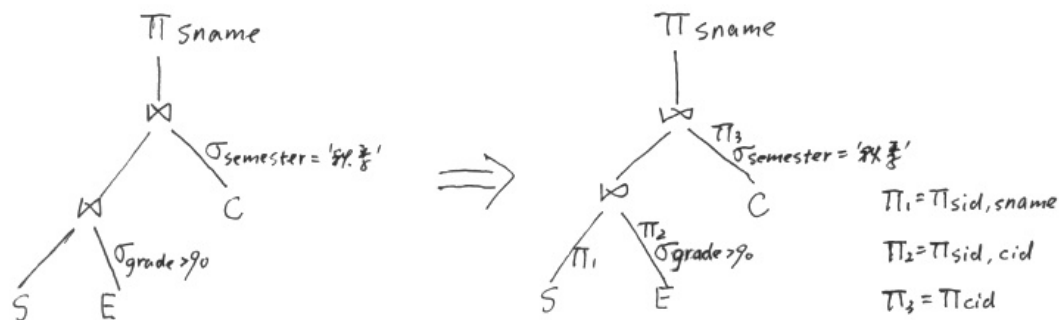
(6) SELECT S.sname, E.cid
FROM Enroll E, Student S
WHERE E.sid = S.sid AND
S.sid NOT IN (SELECT sid
FROM Enroll E1
WHERE grade < (SELECT AVG(grade)
FROM Enroll E2
WHERE E2.cid = E1.cid));

SELECT S.sname, E.cid
FROM Enroll E, Student S
WHERE E.sid = S.sid AND NOT EXISTS
(SELECT E1.cid
FROM Enroll E1
WHERE E1.sid = S.sid AND
grade < (SELECT AVG(grade)
FROM Enroll E2
WHERE E2.cid = E1.cid));

(7) SELECT C.cid, C.cname, AVG(S.gpa)
FROM Student S, Course C, Enroll E, Dept D
WHERE S.sid = E.sid AND E.cid = C.cid AND C.did = D.did AND D.dname='计算机系'
GROUP BY C.cid, C.cname;

4. 要点：因为关系模型仍然用表来表示实体间的联系，是一种“软连接”，查询时涉及大量

连接操作，因此必须优化以解决效率问题；本质不同：关系数据库系统采用非过程化的查询语言；代数优化过程：



5. 要点：提交规则是保证后像在事务提交前写入非挥发存储器，这样即使事务进入提交阶段以后发生故障，仍可利用记录下来的后像重做更新，从而确保事务满足 ACID 原则；先记后写规则是指如果后像在事务提交前直接写入数据库，则必须在此之前将相应的前像写入运行记录 (Log)，以备当事务进入提交阶段之前发生故障时做 undo，使事务的执行满足 ACID 原则。

6. 要点：

- (1) 是指 DBMS 可同时接纳多个事务，事务在时间上可以重叠执行；
- (2) 目的：①改善系统资源利用率；②改进响应时间；
- (3) 问题：①丢失更新；②读脏数据；③读值不可重复；
- (4) 可串行化；
- (5) 可采用封锁法、时间戳法等并发控制方法；

7. 要点：

(1) 背景：分布式处理技术的发展，当时网络带宽还不足，因此自然出现了由联网的多台计算机共同协作解决大量数据的存储、管理、查询的需求，通过将数据就近存放提高访问效率。

(2) 应用并不理想的原因：①系统实现和使用都比较复杂，从数据库设计角度看，物理上分布、逻辑上集中的数据库设计对开发人员要求较高；②难于管理，单个 DBA 很难维护大量物理上分布的数据库，多个 DBA 又很难协调，系统安全较难控制；③网络带宽和服务器性能的飞速提升，使当初的一些瓶颈不再成为障碍，基于集中式数据库服务器的三层（多层）信息系统架构成为流行。

附加题：

要点：(1) 能避免死锁，因为单向年轻事务等待年老事务，无环路。

(2) 年老事务申请锁时，有可能出现活锁。改进方法之一是重执 T_A 时附以新的时间戳。

(3) 可从调度公平性、是否会出现折腾等方面进行讨论。本题是开放性问题，只要观点合理，有想法就行。