浙江大学 2007－2008 学年春季学期

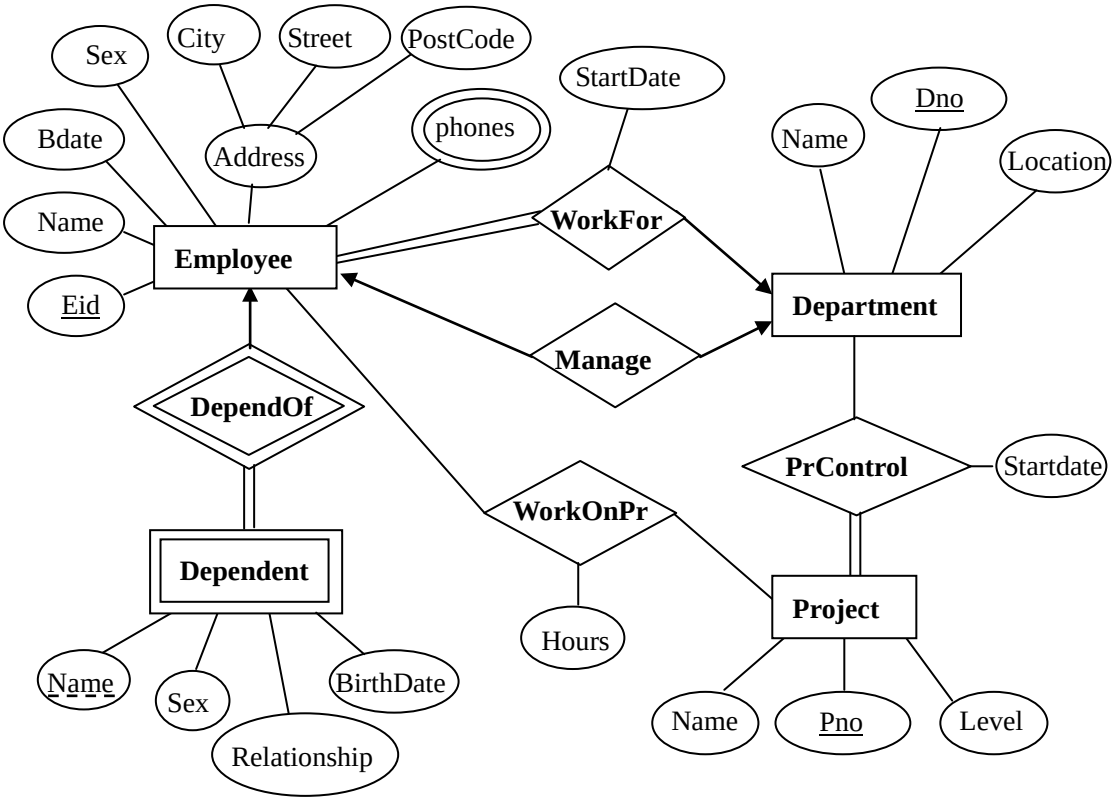
《数据库系统原理》课程期末试卷解答

开课学院：计算机学院，考试形式：闭卷，允许带_1 张 A4 纸笔记_入场

考试时间：2008 年 4 月 19 日,所需时间：120 分钟，任课教师_____

1. Entity-Relationship Model(15 points)

Transform the following ER diagram into a minimum number of relation schemas with necessary normalization. List all candidate keys for each relation schema.



Answer

Employee(Eid, name, Bdate, sex, city, street, postcode,startdate, dno);

E_phones(eid, phone_num);

Department(dno, name, location, manager);

Dependent(Eid, dname, sex, birthdate, relationship);

Project(pno, name, level);

ProControl(Pno, Dno, sartdate);

WorkOnPr(Eid, Pno, hours);

2. Relational Algebra (8 points)

Consider the following schemas:
Student(Sid, Name, Department);
ProjectTeam(TeamName, Sid).
The key attributes are underlined. Write relational algebra expressions for the following queries.

（1） Find the Names of students who are in the team with TeamName 'Red Star' .

（2） Find the Sid of students who have not been in any team yet.

Answer:

(1) Π_{name}(student $\bowtie$ ($\sigma_{teamname='Red\ Star'}$ (projectTeam)))

(2) Π_{sid}(student)- Π_{sid}(ProjectTeam)

3. SQL (28 points)

Consider the following relations about the online bookstore. Underlined attributes are keys of those relations. The attributes cid and isbn in Buy relation are foreign keys referring Customer and Book respectively.
Book(isbn, title, publisher, price, year);
Customer(cid, name, address, postcode);
Buy(bid, cid, isbn, NumberOfCopies, year, month, day). // customer cid buys a number of copies of the book isbn, i.t. NumberOfCopies ≥1

（1） List each customer's cid, the total number of copies of the books which he/she bought from the online bookstore.

（2） Find the name of customers who spent more than $1000 to buy books in year 2007.

（3） Find out the month during which the bookstore sold out the maximum number of books within year 2007.

（4） Find out the books which have never been bought from the bookstore.

（5） Cut down the price to the 50% of the books which were published before year 2003 and had not been bought any copies from the bookstore in the last three years.

Answer:
(1) select cid, sum(NumberOfCopies) from buy group by cid;
(2) select cid, name from customer where cid in
(select cid from buy, book where year=2007 and buy.isbn=book.isbn
Group by cid
having sum(price * NumberOfCopies) >= 1000) ;

(3) select month from buy where year=2007 group by month
Having sum(NumberOfCopies) >= all
(select sum(NumberOfCopies) from buy where year=2007 Group by month);

解法 2：select month from buy where year=2007 group by month
Having sum(NumberOfCopies) >=
(select max(TT.month_books) from
(select sum(NumberOfCopies) as month_books
from buy where year=2007 Group by month) as TT);

(4) select * from book where isbn not in (select isbn from buy);

(5) update book set price=price/2

Where year <2003 and isbn not in

(select isbn from buy where year between 2005 and 2007) ;

4. Armstrong Axiom(6 points)

(1) Use Armstrong's axioms to prove the decomposition rule. That is:

if $A \rightarrow BC$ holds, then $A \rightarrow B$ holds and $A \rightarrow C$ holds.

(2) Prove whether the following rule is true using Armstrong's axioms or reject it by counter example relations.

If ($A \rightarrow C$) and ($AB \rightarrow C$), then ($B \rightarrow C$)

Answer:

(1) According to Armstrong's rule1, we have: BC→B, BC→C,

Given $A \rightarrow BC$,

By Armstrong's rule3, we get: ∵$A \rightarrow BC$, BC→B ∴A→B

∵$A \rightarrow BC$, BC→C ∴A→C

(2) counter example

A	B	C
a1	b1	c1
a2	b1	c2
a1	b2	c1
a4	b2	c4

∴ the rule is not true.

5. Functional Dependencies and Normal Forms (15 points)

Consider the following relation schemas and functional dependencies:

R(A,B,C,D,E) with the following functional dependencies set:

F={ AB→C, BC→D, CD→E, DE→A }

Assume the above functional dependencies are the only ones that hold over R.

(1) Find all candidate keys of R.

(2) Identify whether R is in BCNF or 3NF or neither.

(3) If R is not in BCNF, decompose R into a collection of BCNF relations. Please explain that your decomposition is lossless-join.

(4) Determine if the decomposition of (3) is dependency preserving. If it is not, point out which dependencies are not preserved; if it is, explain it.

Answer:

(1) Candidate keys: (AB), (BC), (BDE)

(2) It is in 3NF.

(3) Decomposition: R1=(C, D, E), R2=(A, B, C, D)

R21=(A, C, D), R22=(B, C ,D)

∴ We get R1=(C, D, E), R21=(A, C, D), R22=(B, C ,D), all are in BCNF, and the decomposition is lossless-join.

(4) DE→A, AB→C, BC→E and AB→D are not preserved.

6. Concurrent schedule (10 points)

(1) Draw the precedence graphs for the schedule 1 and schedule 2 shown below.
(2) State which of the schedules are serializable. And if it is serializable, give out its equivalent serial schedule. Explain your answer briefly.

Schedule 1

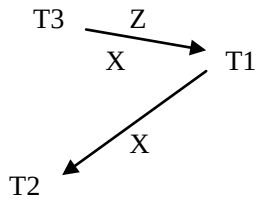
T1	T2	T3
Read (X)		
	Read (Y)	
		Read(X)
		Write(Z)
Read (Z)		
Write(X)		
	Read(X)	
	Write(Y)	

Schedule 2

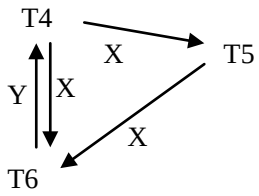
T4	T5	T6
Read(X)		
	Read(Y)	
	Write(X)	
		Write(X)
		Write(Y)
Read(Y)		

Answer:

(1) schedule1



schedule2



Z

(2) schedule1 is serializable. < T3, T1, T2 >
schedule2 is not serializable.

6. XML (16 points)

Answer:

(1) XML Document

```
<student-info>
  <student  sid='61001'>
    <sname> zhang san </sname>
    <sex> male </sex>
    <phones> 8795001 </phones>
    <phones> 134571001 </phones>
  </student>
```

```
<student    sid='61005'>
    <sname> Yao min </sname>
    <sex> female </sex>
</student>
<score    sid='61001' >
    <CourseName>Database Systems </CourseName>
    <Credits> 2.5 </Credits>
    <grade> 95 </grade>
</score>
<score    sid='61001' >
    <CourseName>Data Structure </CourseName>
    <Credits> 4.0 </Credits>
    <grade> 80 </grade>
</score>
<score    sid='61001' >
    <CourseName>Operating Systems </CourseName>
    <Credits> 4.5 </Credits>
    <grade> 78 </grade>
</score>
<score    sid='61005' >
    <CourseName>Discrete Math </CourseName>
    <Credits> 3.0 </Credits>
    <grade> 90</grade>
</score>

</student-info>
```

(2) DTD

```
<!DOCTYPE student-info [
    <!ELEMENT student-info (student, score )+ >
    <!ELEMENT student( sname, sex, phones* ) >
    <!ATTLIST student sid ID #REQUIRED>
    <!ELEMENT score (CourseName,credits, grade) >
    <!ATTLIST score    sid IDREFS #REQUIRED>
    <!ELEMENT sname (#PCDATA) >
    <!ELEMENT sex (#PCDATA) >
    <!ELEMENT phones (#PCDATA) >
    <!ELEMENT CourseName (#PCDATA) >
    <!ELEMENT credits(#PCDATA) >
    <!ELEMENT grade (#PCDATA) >
]>
```