Problem 1 (Relational algebra)

Following are relational schemas of a project management (PM) database:
project(pid, name, budget, tid)
employee(eid, name, age, gender)
team(tid, title, size, leader)
member(eid, tid, date)

Note:
project is identified by **pid**, and described by **name** ,**budget and tid, w**here **tid** is the primary key of the team that takes on the project.
employee is identified by **eid**, and described with **name**, **age** and **gender**
team is identified by **tid**, and described by **title** , **size,** and **leader**, where **leader** is the **eid** of an employee who leads the team.
member records the **date** when an employee(**eid**) participates in the a team(**tid**).

Please answer following questions.
(1) Transform following query to a SQL statement without nested subquery.

```
select eid, name
from employee as E
where age <20 and gender = 'F' and eid =some
        (select eid from member as M
         where M.eid = E.eid and date >= '2017-01-01' and tid in
        ( select tid from team as T where title='Cloud Migration'));
```

(2) Transform above query to an equivalent relational algebra expression.
(3) Write a relational algebra expression to find out the name of employees who is the member of all teams.

Problem 2 (SQL Query)

For the PM database in **problem 1,** please write SQL statements to fulfill following query requests.
(1) Find out the projects with max size of project team.
(2) Find out employees who is the member of only one team.
(3) Find out teams that takes on max number of projects.

Problem 3 (SQL View)

(1) For the PM database in **problem 1**, please define a view named **project_statistics** to capture the information of the total number and average age of all team members for each project.
(2) Please write a SQL statement to grant the select priviledge on the view above to all database users.

Problem 4 (SQL Transaction)

For the PM database in **problem 1**, please write a SQL transaction that moves **employee** "e001" from **team** "t001" to **team** "t002". Note: when an employee becomes a member of a team, the **size** of the team should be increased by 1, and vise versa.

Problem 5 (SQL trigger)

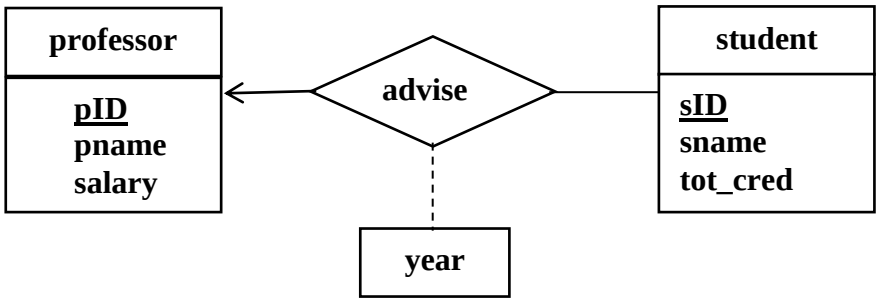
For the PM database in **problem 1**, please write a SQL trigger to implement following functionality: if an employee becomes a member of a team (i.*e.,* inserting a record into **member** table), then the **size** of the **team** is automatically increased by 1.

Problem 6 (E-R Model)

Meeting Preservation System has a database to store information about **meeting rooms**, **telephone lines**, **employees**, and **meetings**, etc. A meeting room is identified by name, and described by capacity, condition, and state. A telephones line is identified by no#, and described by area code, phone number and passcode. An employee is identified by eid, and described by name, age, gender, department, email, and a couple of phones. Every meeting has a unique name, and is described by start time, end time. A meeting preserves a meeting room, and has a host who is also an employee. A meeting may also preserve a telephone line for offsite participants to call in. An employee may participate in several meetings.

(1) Draw an ER diagram that captures this information.
(2) Transform the ER diagram into relational schemas.

Problem 7 (Relational Database Design)



Above is an E-R diagram that depicts one-to-many relationship **advise** between two entity sets **professor** and **student**. The candidate key of the entity sets are underlined in the diagram. If the E-R diagram is transform to following relation schema:

advisor(pID, pname, salary, year , sID, sname, tot_cred).

Please answer following questions:

(1) Give out the functional dependency set F that holds on relation **advisor**
(2) List the candidate key(s) of relation **advisor**.
(3) Explain that relation **advisor** is not in BCNF.
(4) Explain the pitfalls of relation **advisor**.
(5) Prove that decomposing of relation **advisor** into two relation schemas **professor(pID, pname, salary)** and **student(sID, sname, tot_cred, pID, year)** is lossless join decomposition.

Problem 8(Relational Formalization)

For the relation schema R(A, B, C, D, E) with the functional dependencies set F={A→B,A→C, C→D,C→E, A→D,D→E},

(1) Compute the Canonical Cover of F.
(2) Compute the Closure of attribute set {B,D} .
(3) Decompose the relation R into a collection of BCNF relations, and give out the functional dependency set for each relation.
(4) Explain whether above decomposition is dependency preserving or not.

Problem 1（XML）

Following DTD describes XML documents containing information about components of software project. Assuming there is a document “components.xml” corresponding to this DTD.

```
<!DOCTYPE components [
    <!ELEMENT components (component+)>
    <!ELEMENT component (name, functionality, complexity, sub-component*)>
    <!ELEMENT sub-component (component, context)>
    <!ATTLIST component cid    ID    #REQUIRED>
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT functionality (#PCDATA)>
    <!ELEMENT complexity (#PCDATA)>
    <!ELEMENT context (#PCDATA)>
]>
```

1) Please write a query in XPath on “components.xml” that returns the name of all components that has at least 5 sub-components.
2) Please write a query in XQuery that returns the **cid** of components that has the same complexity as the component named “merge-sort”.
3) Please design relational schema to capture the information described by this DTD.

Problem 2（B+ -tree）

1) Construct a B+-tree (n=4) for the following bulk of index entries:
   （7, 37, 67, 97, 127, 157, 187, 217）
   Assume that the tree is initially empty, and the bottom-up B+-tree construction method is used for bulk loading the above index entries.
2) For the constructed B+-tree in 1）, draw the B+ -tree after operation “insert 83”。
3) For the constructed B+-tree in 1）, draw the B+- tree after operation “delete 187”
4) Assume that the B+-tree (n=4) contains 20000 index items, please estimate the height of the tree.

Problem 3 (Query Optimization)

Consider following relational schema and SQL statement:

book(bid: **char(20)**, title: **char(20)**, author: **char(20)**, price: **integer**)
reader(rid: **integer**, name: **char(20)**, age: **integer**; city: char**(20)**)
borrow(rid: **integer**, bid: **char(20),** borrow_date **date,** return_date **date**)

```
select reader.rid
from reader, borrow, book
where reader.rid=borrow.rid and borrow.bid = book.bid and
        book.title =’Deep Learning’ and
        borrow_date between ’2016-01-01’ and ’2016-06-30’
```

1) Identify a relational algebra tree (or a relational algebra expression if you prefer) that reflects the order of operations that an optimizer would choose.
2) What indexes might be of help in processing this query? Explain briefly.

Problem 4 (Query Cost Estimation)

For the relational schema defined in **problem 4**, there are following assumptions:

- **reader** table has 1000 records. **book** table has 1000 records. **borrow** table has 79900 records.
- The ***title*** attribute of **book** is unique.
- The value of attribute ***borrow_date*** is uniformly distributed between ‘2012-01-01’ and ‘2016.12.31’.
- Every reader borrowed some books. Every book was borrowed with the same chance.
- Attribute with ‘integer’ type needs 4 bytes. Attribute with ‘date’ type needs 8 bytes.
- The block size is 4K bytes.
- There are 32 blocks in buffer for performing relational operations.
- **book** table and **borrow** table both are sorted on ***bid*** attribute.

1) Estimate the size (i.e. number of records) returned by the SQL statement given in problem 4.
2) Estimate the number of blocks of relation **book** and **borrow** respectively.
3) Estimate the best cost for evaluating **book**⋈ **borrow** using Merge Join method. (Hint:（a）The cost is measured by the number of blocks transferred to main memory and the times to seek disk. （b）In order to get the best cost, please consider how many blocks in buffer should be allocated to relation **book** and **borrow** respectively)

Problem 5 (Buffer Manager)

DBMS generally buffer disk blocks in memory. Assuming that the maximum number of blocks that can be held in buffer is 4. Initially the buffer is empty. Consider the following schedule with assumption that each of the data items A, B, C, D and E lies on distinct blocks.

r1(A) w2(B) r1(C) w3(A) c1 w2(D) w3(E) c2 r4(D) w5(C) r5(B) w4(A)
where: ***ri(X)*** means transaction ***Ti*** read data ***X***.

wi(X) means transaction ***Ti*** write data ***X***.

ci means transaction ***Ti*** commits.

1) Figure out the first operation where an existing block in buffer must be dropped in order that another block can be read in.
2) Assume that we are using an LRU buffer-replacement policy. What block will be dropped from buffer at the time of the operation figured out in 1)
3) From the perspective of data consistency, can DBMS simply drop the block found in 2) at that time, forgetting its value, or must do something else before? Why?
4) When ***c1*** occurs in the schedule above, does this force out the blocks with the data item updated by ***T1***?

Problem 6 (Two-Phase Locking Protocol)

Consider the following schedule that is generated by the two-phase locking protocol.

r1(A) r2(B) r2 (C) w1(B) w4(C) r4(D) w5(A) w5(D) w3(D) r1(E)

Where: ***ri(X)*** means transaction ***Ti*** read data ***X***.

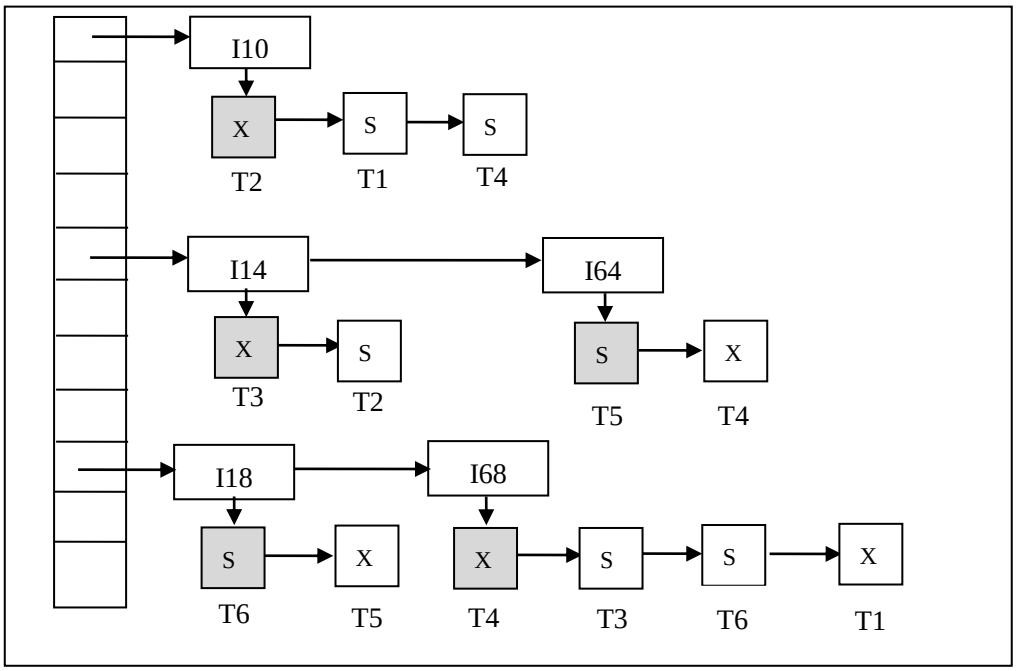
wi(X) means transaction ***Ti*** write data ***X***.

1) Draw the precedence graph of the schedule.
2) In the schedule, which transaction is the first to obtain its final lock? Which transaction is the last to obtain its final lock?
3) Is the schedule generated by the strict two-phase locking protocol? Explain briefly.

Problem 7 (Lock Manager)

The following figure shows an instance of lock table. There are 6 transactions (T1-T6) and 5 data items (I10, I14, I64, I18, I68). Granted locks are filled (black) rectangles, while waiting requests are empty rectangles. The lock mode(**S** or **X**) is marked in the rectangles.

1) Which transactions are involved in deadlock?
2) In order to break the deadlock, which transaction (victim) should be rolled back?
3) Draw the lock table after the victim transaction of 2) is rolled back.



Problem 8 (Aries Recovery Method)

Aries recovery method is used in a DBMS. Following figure is a log file just after the DBMS crashes. The log file consists of 14 log records with LSN from 6001 to 6014. The PrevLSN and UndoNextLSN values in the log records are not shown in the figure. Assuming that the last completed checkpoint is the log record with LSN 6008. Please answers following questions about the ***Aries*** recovery process.

1) From which log record does the Analysis Pass start?
2) When the Analysis Pass is complete, what is the content of the DirtyPageTable?
3) From which log record does the Redo Pass start?
4) Which transactions should be Undone?
5) Which log records is the start point of the Undo Pass? Which log record is the end point of the Undo Pass?

6001： <T1 begin>

6002： <T1 , 1001.1, 110, 111>

6003： <T2 begin>

6004： <T2 , 1001.2, 120, 122>

6005： <T2, 1002.1, 210, 211>

6006： <T3, 1003.1, 310, 311 >

6007： <T2 , 1003.2, 320, 322 >

6008： checkpoint

Txn	LastLSN
T1	6002
T2	6007
T3	6006

PageID	PageLSN	RecLSN
1001	6004	6002
1002	6005	6005
1003	6007	6006

6009： <T1 commit >

6010： <T3, 1004.1, 410, 411 >

6011： <T4 begin >

6012： <T4，1003.3, 330, 333>

6013： <T4，1004.2, 420, 422>

6014： <T3 commit >

浙江大学 2005－2006 学年春夏季学期

《数据库系统原理》课程期末考试试卷

开课学院：软件学院，考试形式：闭，允许带 1 张 A4 纸笔记 入场

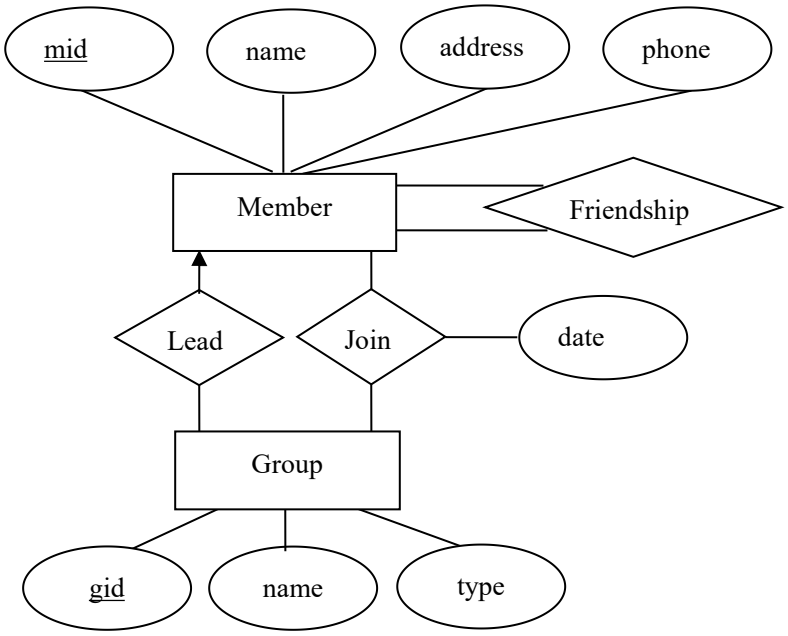
考试时间：2006 年 4 月 16 日,所需时间： 120 分钟

考生姓名：_____学号：_____专业：_____

题序	一	二	三	四	五	六	总 分
得分							
评卷人							

1. Entity-Relationship Model(20 points)

Transform the following ER diagram into a minimum number of BCNF relations. List all candidate keys for each relation.



2. SQL (30 points, 6 points per part)

Relation Person is defined as below:

```
create table Person
(     id char(2) not null,
      name char(8) not null,
      age smallint,
      sex char(1) not null,
      spouse char(2),
      primary key(id),
      foreign key(spouse) references Person
            on update cascade
            on delete set NULL,
      check (sex in {'F', 'M'})
)
```

Following is an instance of the relation Person:

id	name	age	sex	spouse
01	Zhao	20	M	02
02	Qian	21	F	01
03	Sun	22	F	04
04	Li	20	M	03
05	Wang	24	M	08
06	Zhang	24	M	
07	Liu	24	F	
08	Ling	25	F	05
09	Jiang	25	M	

1) Write a relational algebra expression to find the names of all female persons who are older than 22.
2) What's the output of the following SQL query on the Person table above?

```
select age, count(*)
from Person as P1
where    P1.sex=' M' and
exists (select * from Person P2    where P1.spouse=P2.id )
group by age
order by age
```

3) What's the output of executing the following two SQL statements on the Person table above?

```
delete from Person where name='Wang';
select count(spouse) from Person;
```

4) Write a SQL query to find the men who has no spouse and whose age is above the average age of all men.
5) Write a SQL query to find the pairs of husband and wife whose age sum is the greatest among all pairs of husband and wife.

3. Relational Formalization (20 points, 5 points per part)

Consider the following relation schema and functional dependencies:
R(A,B,C,D,E) with the following functional dependencies set
F={AB→C, C→D, A→E}
Assume the above functional dependencies are the only ones that hold over R.

1) Find all candidate keys.
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. The decomposition must be lossless-join. Show each step of the decomposition.
4) Whether the decomposition of 3) is dependency preserving or not? Why?

4. Deadlock Detection (10 points)

In a DBMS there is a process that awakes periodically to do deadlock detection for the purpose of deadlock resolution. The interval of "deadlock detection" can be configured by DBA. Please explain the influence of the interval length of the "deadlock detection" to the database system.

5. Two Phase Locking Protocol (10 points)

Following is a schedule for transactions T1,T2,T3, and T4:

T1	T2	T3	T4
write(A)			
		write(A)	
	write(C)		
			write(B)
write(C)			
			write(A)

1) Draw the precedence graph of the schedule.
2) If the schedule is conflict serializable?
3) Can the schedule be generated by the two phase locking protocol?

6. XML and XPath(10 points)

Consider the following XML document containing book information:

```
<bookstore>
    <book ISBN="7-04-011049-0">
        <title>Dadatbase System Concepts</title>
        <author>Abraham Silberschatz</author>
        <author>Henry F. Korth </author>
        <author>S.Sudarshan </author>
        <price>59.5</price>
    </book>
    <book ISBN="0-06-662099-6">
        <title>Good to Great</title>
        <author>Jim Collins </author>
        <price>88.9</price>
    </book>
    <book ISBN="1-55860-622-x">
        <title>Data on the Web</title>
        <author>Serge Abiteboul</author>
        <author>Peter Buneman</author>
        <author>Dan Suciu</author>
        <price>120</price>
    </book>
    <book ISBN="0-13-629239-9">
        <title>Object-Oriented Database Management</title>
        <author>Alfons Kemper</author>
        <author>Guido Moerkotte</author>
        <price>135</price>
    </book>
</bookstore>
```

1) Write the result of the following XPath expression on the document above:

   /bookstore/book[price>100]/title/text()

2) Write an XML DTD so that the document above conforms to the DTD.

浙江大学 2005－2006 学年夏季学期

《数据库系统原理》课程期末考试试卷

开课学院：计算机 ，考试形式：闭卷，允许带 1 张 A4 纸笔记 入场

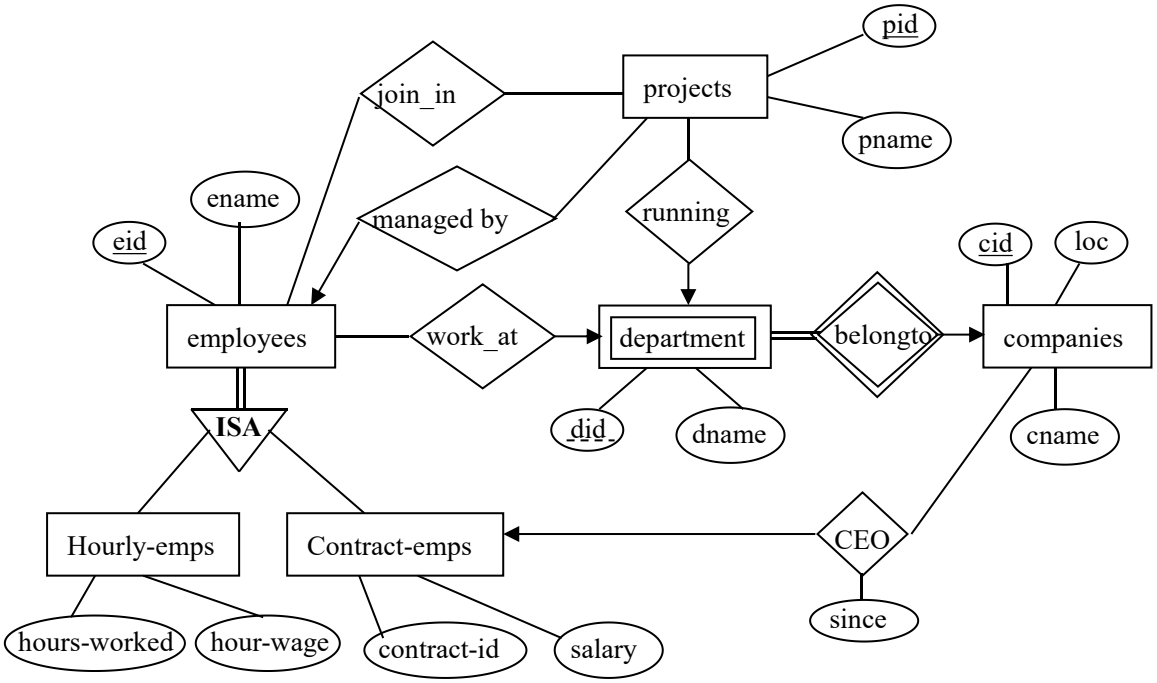
考试时间：2006 年 06 月 29 日,所需时间：120 分钟，任课教师_____

考生姓名：_____学号：_____专业：_____

题序	一	二	三	四	五	六	总 分
得分							
评卷人							

1. Problem 1. Entity-Relationship Model(20 points)

Please translate the E-R diagram below into a set of minimum relational tables. And specify the primary key for each table.



Problem 2. Relational Algebra & SQL Queries (30 points, 6 points each)

Consider a database schema with the following relations:
Student (sid, name, age, sex, phone, email), //sid is primary key
Prof (pid, name, age, sex, position, phone, email), //pid is primary key
Course (cid, pid, title, credits, room_no),//(cid, pid) is primary key
Enroll (student_sid, course_cid, score), //(student_sid, course_cid) is primary key

1) Write a relational algebra query that finds all students who are enrolled in courses taught by the professor whose name is 'Silberschatz'.
2) Write a SQL QUERY that finds out the name and phone number of all students who have class in room number 'A-305'.
3) Write a SQL QUERY that finds out the students who have enrolled more than 10 courses.
4) Write a SQL QUERY that finds out the students who get the highest score for the course title 'Introduction to Database Systems'.
5) Write SQL statements that find out the students whose credits sum is greater than that of the average among all students.

Problem 3. SQL Assertion(10 points)

Please write a SQL assertion to indicate that the functional dependency CD→E holds over the relation R(A,B,C,D,E), where AB is the primary key of R.

Problem 4. Relational Formalization (20 points, 5 points per part)

Consider the the relation schema R(A,B,C,D,E) with the following functional dependencies set F={A→B, B→A, B→C, D→E }

1) Find all candidate keys.
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. The decomposition must be lossless-join. Show each step of the decomposition.
4) Whether the decomposition of 3) is dependency preserving or not? Why?

Problem 5. Strict Two Phase Locking Protocol (10 points)

Please figure out the advantages and the disadvantages of the strict two phase locking protocol compared with the basic two phase locking protocol.

Problem 6. XML and XPath(10 points)

Following is an XML DTD:

```
<!DOCTYPE message [
        <!ELEMENT message (to+, from, cc*, subject, text)>
        <!ELEMENT to (#PCDATA)>
        <!ELEMENT from (#PCDATA)>
```

```
    <!ELEMENT cc (#PCDATA)>
    <!ELEMENT subject (#PCDATA)>
    <!ELEMENT text (#PCDATA)>
]>
```

1) Please give an XML document that conforms to the above DTD and contains at least two **message** elements.
2) Suppose that msg.xml is an XML document that conforms to the above DTD. Please write an XPath expression to find out all messages from "***Dave"*** to "***Marry"***.

浙江大学 2007－2008 学年春季学期

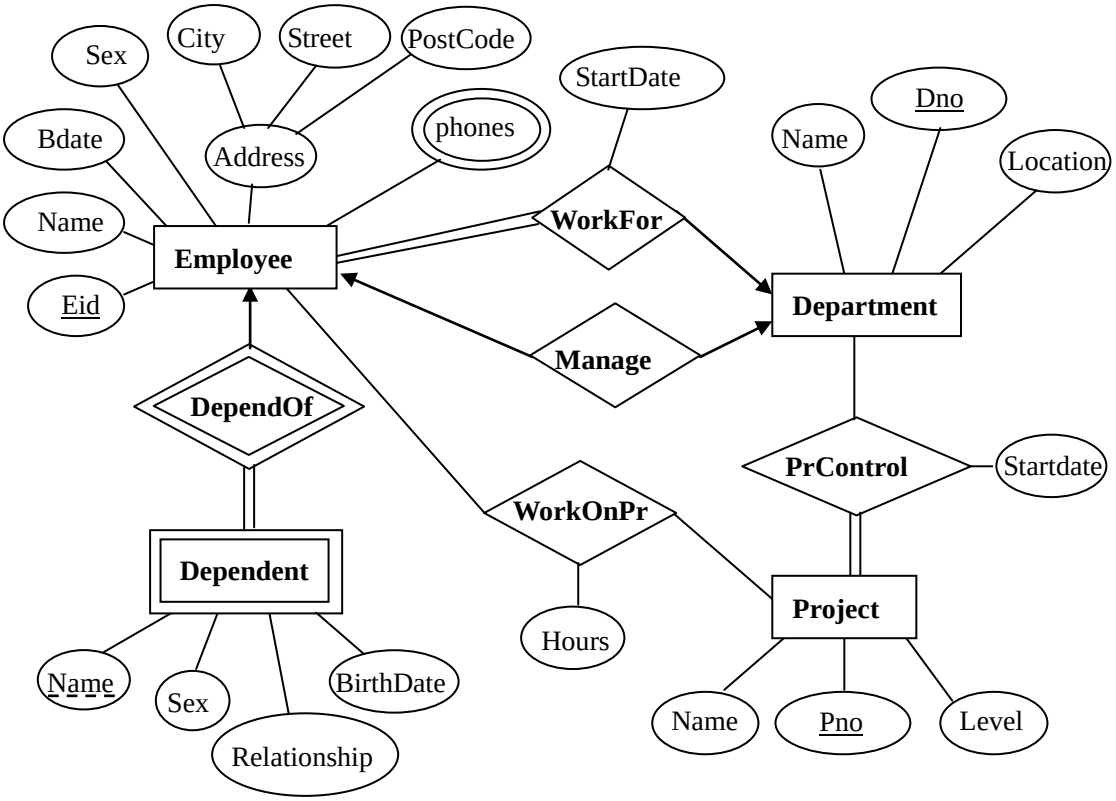
《数据库系统原理》课程期末试卷解答

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

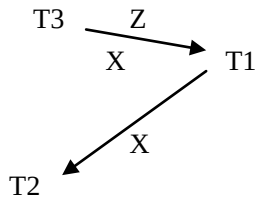
T1	T2	T3
Read (X)		
	Read (Y)	
		Read(X)
		Write(Z)
Read (Z)		
Write(X)		
	Read(X)	
	Write(Y)	

Schedule 2

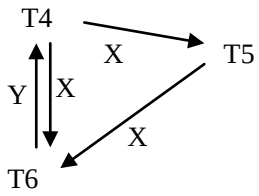
T4	T5	T6
Read(X)		
	Read(Y)	
	Write(X)	
		Write(X)
		Write(Y)
Read(Y)		

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) >
]>
```

浙江大学 2007－2008 学年春季学期

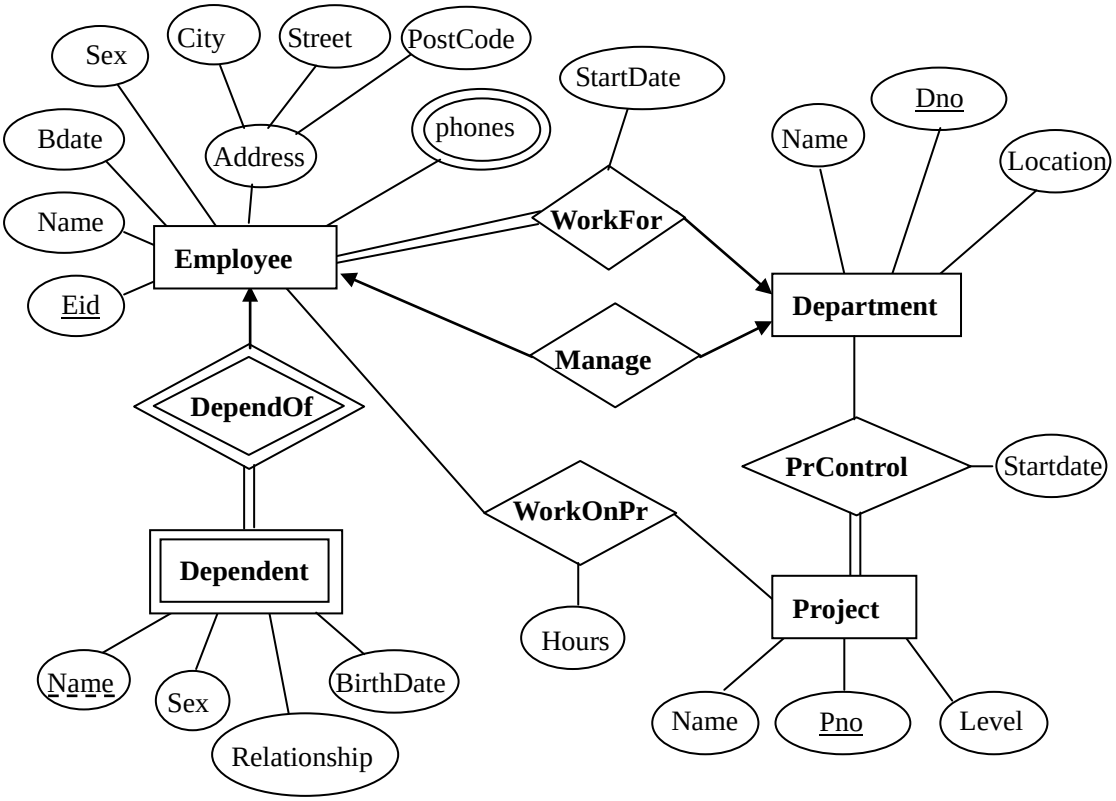
《数据库系统原理》课程期末试卷解答

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

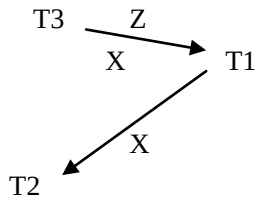
T1	T2	T3
Read (X)		
	Read (Y)	
		Read(X)
		Write(Z)
Read (Z)		
Write(X)		
	Read(X)	
	Write(Y)	

Schedule 2

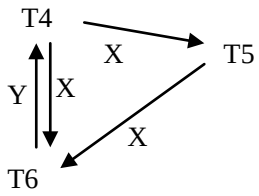
T4	T5	T6
Read(X)		
	Read(Y)	
	Write(X)	
		Write(X)
		Write(Y)
Read(Y)		

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) >
]>
```

浙江大学 2008–2009 学年 春 季学期

《数据库系统原理》课程期末考试试卷

参考答案

1. Relational Algebra (10 points, 5 points each)

(1) $\prod_{name}((\sigma_{gender='F'}(Employee)) \bowtie EmpSkill \bowtie (\sigma_{job='DEV'}(JobSkill)))$

(2) $\prod_{name}(Employee \bowtie (EmpSkill \div \prod_{skill}(\sigma_{job='DEV'}(JobSkill))))$

2. SQL Query (20 points, 5 points each)

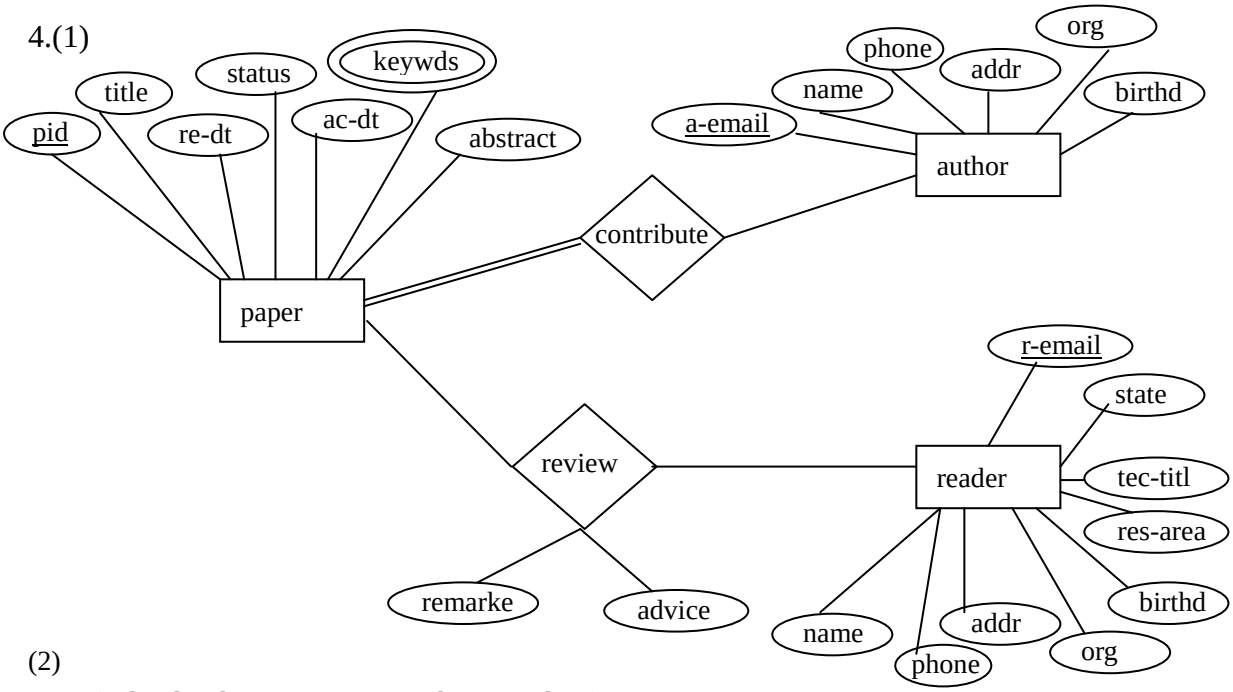
```
(1) select * from Employee where id not in ( select id from EmpSkill );
(2) select A.job from JobSkill A, JobSkill B
        where A.job=B.job and A.skill='Java' and B.skill='C++';
(2)' ( select job from JobSkil    whereskill='Java' )
      Intersect
    ( select job from JobSkil    whereskill='C++' );
(3) select name from Employee where id in ( select id from EmpSkill
                                            group by id having count(skill) >= all
                                          (select count(skill) from EmpSkill group by id ) );
(4)   select * from Employee E where not exists
          ( (select skill from JobSkill where job='DEV' or job='UAT' )
          except
          (select skill from EmpSkill K where K.id=E.id) );
(4)'   select * from Employee E where not exists
          (select * from JobSkill J where (job='DEV' or job='UAT') and not exists
          (select * from EmpSkill K where K.id=E.id and K.skill=J.skill )
```

3. Embedded SQL (10 points)

```
(1) select skill from EmpSkill where id= :id
(2) OPEN skill_cursor
(3) FETCH skill_cursor INTO :skill
(4) SQLCA.STATE = '02000'
(5) CLOSE skill_cursor
```

4. E-R Model (20 Points, 10 points each)

4.(1)



(2)

Paper(<u>pid</u>, title, abstract, status, rec-date, acc-date)

Paper-keywords(<u>pid, kword</u>);

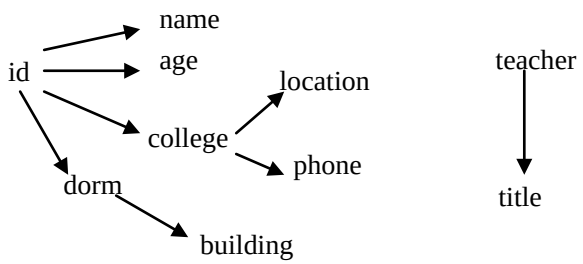
Author(<u>a-email</u>, name, birthdate, phone, address, org)

Reader(<u>r-email</u>, name, birthdate, phone, address, org, tech-title, res-area, state)

Contribute(<u>pid, a-email</u>);

Review(<u>pid, r-email</u>, remark, advice);

5. Relational Formalization (20 points, 5 points each)



(1) candidate key = (id, teacher, year)

(2) Neither BCNF and 3NF, because the left side of all function dependencies in F are not super-key, e.g. id→name, id→ age, dorm→building, …

(3) R1=(id, name), R2=(id, age), R3=(dorm, building), R4=(id, dorm), R5=(college, location), R6=(college, phone), R7=(id, college), R8=(teacher, title), R9=(id, teacher, year);

(4) the above decomposition is dependency preserving, because all function dependencies in F are preserved.

6. XML (20 points, 5 points each)

(1) relational schemas:

Project(pid, pname, budget, from, to, manager, members);

Developer(did, dname, age);

(2)

```
<!DOCTYPE research-proj [
    <!ELEMENT research (project +, developer + )>
        <!ELEMENT project (pname, budget, from, to )>
            <!ATTLIST project
                pid    ID       # REQUIRED
                members        IDREFS       # REQUIRED >
        <! ELEMENT pname (#PCDATA)>
        <! ELEMENT budget (#PCDATA)>
        <! ELEMENT from (#PCDATA)>
        <! ELEMENT to (#PCDATA)>
        <!ELEMENT developer(dname,age)>
            <!ATTLIST developer
                did    ID       # REQUIRED >
        <! ELEMENT dname (#PCDATA)>
        <! ELEMENT age (#PCDATA)>
        ]>
```

(3) Xiao Zhao
Xiao Sun

(4)

```
<talent>
    <dname>Xiao Sun</dname>
    <pname>Unstructured Data Management</pname>
</talent>
```

诚信考试　沉着应考　杜绝违纪

浙江大学 2008–2009 学年 春 季学期

《 数据库系统原理 》课程期末考试试卷

开课学院：计算机学院，考试形式：闭卷，允许带一张含内容 A4 纸入场

考试时间：2009 年 4 月 13 日，所需时间：120 分钟，任课教师_____

考生姓名：_____学号：_____专业：_____

题序	一	二	三	四	五	六	总 分
得分							
评卷人							

1. Relational Algebra (10 points, 5 points each)

Consider the following relational schemas:

Employee (id, name, age, gender)

EmpSkill(id, skill)

JobSkill (job, skill)

Following are instances of these relations:

Employee:

id	**name**	**age**	**gender**
E01	John	20	M
E02	Mean	22	F
E03	Smith	24	M
E04	David	30	M
E05	Mary	18	F

EmpSkill:

id	**skill**
E01	Java
E01	C++
E02	C++
E03	TEST
E04	Finance

JobSkill:

job	**skill**
DEV	Java
DEV	C++
BA	FX
BA	Equity

UAT	C++
UAT	Finance

In addition, we define that ***"an employee fits a job"*** if and only if the employee has all skills that are needed by the job.

Please write relational algebra expressions for the following queries.

1) Find the name of female employees who have at least one skill needed by the "DEV" job.
2) Find the names of employees who fit the "DEV" job.

2. SQL Query (20 points, 5 points each)

Consider the relational schemas given **in problem 1**, please write SQL statements to meet the following requests.

1) Find the employees who have not any skills.
2) Find the jobs that need at least "Java" and "C++" skills.
3) Find the names of employees who have the maximum number of skills among all employees.
4) Find the employees who fit both the "DEV" and "UAT" jobs.

3. Embedded SQL (10 points)

Based on the schemas defined in **problem 1**, the following embedded SQL program accepts the id of an employee as input , and output all skills of the employee. Please fill in the blanks of the program.

```
main( )
{       EXEC SQL INCLUDE SQLCA;
        EXEC SQL BEGIN DECLARE SECTION;
        char id[10]; char skill [20];
        EXEC SQL END DECLARE SECTION;
        EXEC SQL CONNECT  TO  skill_db  USER use1 USING password1;
        EXEC SQL DECLARE skill_cursor CURSOR for _____①_____;
        printf("please input employee id :");
        scanf ("%s", id);
        EXEC SQL _____②_____;
        for (; ;)
        {       EXEC SQL _____③_____;
                if (_____④_____) break;
                printf( "%s\ n", id);
        }
```

EXEC SQL _____⑤_____;

}

4. E-R Model (20 Points, 10 points each)

A publishing company needs to use database to store paper reviewing information. Each paper is identified by *pid*, with *title, author names, abstract, key words, status , received date* , and *accepted date* as attributes; Authors and readers are all identified by *email account*, with *name, birthday, phone, address*, and *organization* as attributes; Readers are additionally described by *technical title, research area,* and *state;* Each paper is written by a couple of authors and reviewed by several readers. Each reader gives *remark* and *advice* (accepted or not accepted) for the paper that he reviews.

1) Draw an ER diagram that captures this information.
2) Please transform the ER diagram of 1) into a set of formalized relational schemas.

5. Relational Formalization (20 points, 5 points each)

Consider relational schema R (id, name, age, college, location, phone, building, dorm, teacher, title, year,) with functional dependencies set

F={id→name; id→ age; id→college; college→location; college→phone; id→dorm; dorm→building; teacher→title},

Please answer the following questions:

1) List all candidate keys for the relation schema 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. The decomposition must be lossless-join. Show each step of the decomposition.
4) Is the above decomposition dependency preserving? Why?

6. XML (20 points, 5 points each)

Consider the following XML document containing research project information.

```
<research>
    <project pid="p01"  manager="d001"  members="d001 d002">
        <pname>Native XML Database</pname>
        <budget>100</budget>
        <from>2008-01-01</from>
        <to>2008-12-01</to>
    </project>
    <project pid="p02"  manager="d003"  members="d003 d002 ">
```

```
        <pname>Unstructured Data Management</pname>
        <budget>4000</budget>
        <from>2009-01-01</from>
        <to>2010-12-01</to>
    </project>
    <developer  did="d001">
        <dname>Xiao Zhao</dname>
        <age>30</age>
    </developer>
    <developer  did="d002">
        <dname>Xiao Qian</dname>
        <age>20</age>
    </developer>
    <developer  did="d003">
        <dname>Xiao Sun</dname>
        <age>25</age>
    </developer>
</research>
```

1) Give relational schemas that can store the data contained in the XML documents like the one above.
2) Write an XML DTD that the document above conforms.
3) Write the result of the following XPath expression on the document above.

   /research/developer[age>20]/dname/text()

4) Write the result of the following XQuery statement on the document above.

```
for     $p  in  /research/project,
        $d  in  /research/developer
where   $p/@manager = $d/@did  and $d/age <30
return  <talent>  { $d/dname  $p/pname  }</talent
```

浙江大学 2009–2010 学年 春 季学期

《 数据库系统原理 》课程期末考试试卷

开课学院：计算机学院，考试形式：闭卷，允许带一张含内容 A4 纸入场

考试时间：2010 年 4 月 28 日，所需时间：120 分钟，任课教师_____

考生姓名：_____学号：_____专业：_____

题序	一	二	三	四	五	总 分
得分						
评卷人						

1. Relational Algebra (20 points)

1) Consider two relations *r(A,B)* and *s(A,B)*. You would like to compute their intersection $r \cap s$, but unfortunately suppose you only have four relational algebra operators at your disposal: σ, $\prod$, x, and $\bowtie$ (natural join). Is it possible to compute $r \cap s$ using just these four operators?
If so, show the simplest equivalent expression you can come up with. If not, briefly explain why not.

2) Consider a relation schema $R(A,B,C)$. Write a relational algebra expression that returns an empty result if and only if the functional dependency $A \rightarrow B$ holds on R. Please write a single algebraic expression.

2. SQL Query (20 points)

1) Consider a table ***T(A char(1))*** .
Write a SQL query to find the largest number of duplicated values in T. (For example, if T contains three A's, six B's, five C's, and four E's, then your query should return the number 6.) Your query will be graded on simplicity as well as on correctness.

2) Consider a SQL table ***T(A int unique, B int).*** Assume there are no NULL values. As specified, attribute A is a key.
Consider the following three SQL queries:

```
Q1: select B from T
      where B >= some (select B from T) ;
Q2: select B from T as T1
      where B > all (select B from T as T2 where T2.A < > T1.A) ;
Q3: select max(B) from T ;
```

Which of the queries above are equivalent ? Please show a smallest single instance of

T you can come up with that demonstrates your answer.

3. Database Design (20 points)

Consider the following relational schemas describing books, publishers, readers and reader ratings of the books:

```
Book(bid, title, author, price) // bid is a primary key
Reader(name, age, gender, profession) // name is a primary key
Rating(name, bid, date, score) // (name,bid) is a primary key
Publisher(pid, pname, location, phone) //pid is a primary key
PublishedBy(bid, pid, year) //bid is a primary key
```

1) Draw a E-R diagram from which these relational schemas could have been produced. Your diagram should be fully connected, and it should be as detailed as possible from the information you have.

2) Please make necessary formalization of the relational schemas above, to get a minimum number of relation schemas.

3) Write SQL data definition statements for the relation schemas from the step 2), and give necessary integrity constraints on them.

4. Relational Formalization (20 points)

Consider relational schema R (A, B, C, D, E, F) with functional dependencies set

F={A→B; A→C; C→D; E→D }.

Please answer the following questions:

1) List all candidate keys for the relation schema 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. The decomposition must be lossless-join. Show each step of the decomposition.
4) Is the above decomposition dependency preserving? Why?

5. XML (20 points)

Consider the following XML document containing company information.

```
<company>
    <division Name="DEV1">
        <manager Name="Emp010">
            <office>201</office>
            <salary>8000</salary>
        </manager>
        <engineer Name="Emp201">
            <office>202</office>
            <salary>6000</salary>
        </engineer>
        <engineer Name="Emp205">
            <office>201</office>
            <salary>4000</salary>
        </engineer>
```

```
            <intern Name="EmpT01", Supervisor="Emp201">
                  <college>ZJU</college>
            </intern>
            <intern Name="EmpT03", Supervisor="Emp205">
                  <college>PKU</college>
            </intern>
      </division>
      <division Name="SALES">
            <manager Name="Emp150">
                  <office>301</office>
                  <salary>5000</salary>
            </manager>
            <intern Name="EmpT34", Supervisor="Emp150">
                  <college>ZJUT</college>
            </intern>
      </division>
</company>
```

1) Give relational schemas that can store the data contained in the XML documents like the one above.
2) Write an XML DTD that the document above conforms.
3) Write the result of the following XPath expression on the document above.

   /company/division/*/@Name[./salary>5000]
4) Write an XPath expression to find the names of all Managers who share an office with an Engineer.
5) Write the result of the following XQuery statement on the document above.

```
for     $t  in  /company/division/intern,
        $e  in  / company/division/engineer
where   $t/@Supervisor = $e/@Name  and $e/salary <=5000
return  <MakeupSup> {$e/../@Name   $e/@Name} </MakeupSup>
```

浙江大学 2011－2012 学年秋季学期

《数据库系统设计》课程期末考试试卷

课程号：__21120301___，开课学院：计算机_，　任课教师：_____

考试试卷：√A 卷、B 卷

考试形式：√闭、开卷，允许带 1 张 A4 纸笔记入场

考试日期：2011 年 11 月 7 日，考试时间：120 分钟

诚信考试，沉着应考，杜绝违纪。

考生姓名：_____学号：_____所属院系：_____

题序	一	二	三	四	五	总 分
得分						
评卷人						

Problem 1 : Index and Query (20 points)

Consider the following relational schemas:

Create table emp(id char(20) primary key,
name char(30),
age int,
salary real,
address char(50),
dname char(30) ,
foreign key (dname) references dept);

Create table dept(dname char(30) primary key,
location char(50));

Assume table ***emp*** has 6000 records, which are stored sequentially based on their ids; table ***dept*** has 200 records, which are stored sequentially based on their ***dnames***; each (data or index) block can hold 2K (2048) bytes, in which 48 bytes are reserved for the block header. The attribute with *real* type is 8 bytes and *int* type is 4bytes. No records span two or more blocks. Each block pointer(which points to a block) is 6 bytes. Each record pointer (which points a data record) is 8 bytes (includes a block pointer + 2bytes offset within a block). Index entries do not span blocks.

(1) Suppose we want to build a conventional primary index INDEX1 on ***emp.id*** using a dense

index. What is the minimum number of blocks needed for INDEX1?

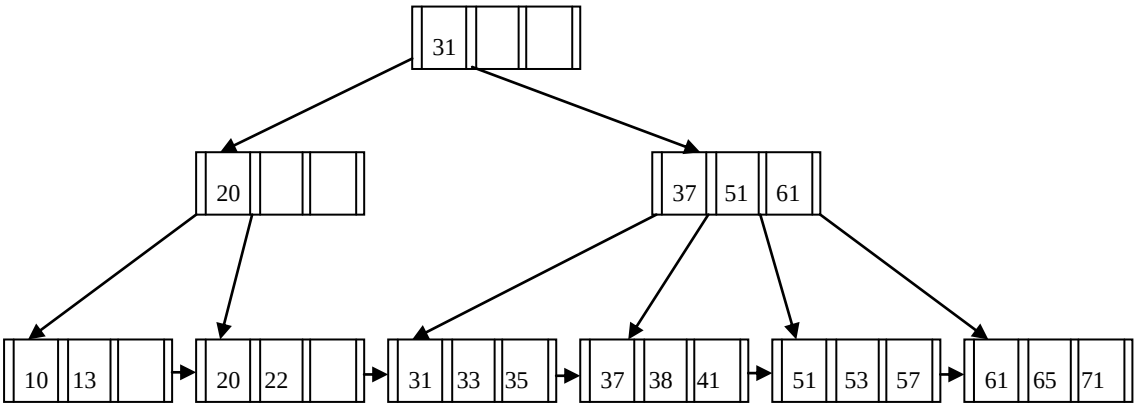
(2) For the same primary index INDEX1 on ***emp.id***, we now want to use a sparse index. What is the minimum number of blocks needed for INDEX1?

(3) Suppose we want to build a secondery index INDEX2 on ***dept.location***. What type of INDEX2 makes sense, sparse or dense? What is the minimum number of blocks needed for INDEX2?

(4) Suppose now we want to do the natural join *emp* ⋈ *dept*. The available memory buffer size is 10 blocks. Estimate the cost for the join, and explain the algorithm to get the lowest cost.

Problem 2 : B+ Tree (24 points)

For the following B+ tree (n=4):



Answer the following questions:

(1) Draw the B+ tree after insert an index item with key ‘55’ to the above tree.

(2) Draw the B+ tree after delete an index item with key ‘13’ from the original tree.

(3) Suppose the B+ tree contains 900 index items, please estimate the height of the B+ tree. (Consider the tree with half-filled and full-filled respectively.)

(4) Suppose that the B+ tree contains 900 index items, please estimate the size (i.e. the number of nodes) of the B+ tree. (Consider the tree with half-filled and full-filled respectively.)

Problem 3. Query Optimization (12 points)

Suppose we have the database tables R(ID, A) and S(ID, B) in which IDs are keys, and we would like to execute the following query:

```
SELECT A, B
FROM R, S
WHERE R.ID = S.ID AND A = 'a1' AND B = 'b1' ;
```

We have the following statistics and assumptions:

R stored contiguously in b_r = 1000 blocks;

S stored contiguously in b_s = 600 blocks;

V (R, ID) = 15000; // V(R, ID) is the number of distinct values that appear in the

relation R for attribute ID .

V (R, A) = 50;

V (S, ID) = 10000;

V (S, B) = 100;

Assume that the values in R.A and S.B are uniformly distributed, in which value 'c1' and 'b1' exist. The available memory buffer size is 30 blocks.

Please answer the following questions:

(1) Please give out an execution plan with the lowest cost for the query at the relational-algebra level.

(2) Estimate the cost for the query with your optimized algorithm.(Note: the cost is measured with the number of blocks transferred between disk and memory buffer and the seek times of disk. As usual, do not count the cost to write out the final query result.)

Problem 4: Concurrent Control (24 points)

There are two concurrent schedules:

*S*1 : *w*1(*Y*);*w*2(*X*); *r*3(*X*); *r*1(*Z*);*w*4(*Z*); *r*4(*X*);*r*2(*Y*).

*S*2 : *r*1(*X*); *r*3(*X*);*r*1(*Z*);*r*4(V);*w*3(*Z*);*w*2(*Y*);*w*2(*X*); *w*4(*V*); *r*3(*Y*).

For each of the above schedules *S1* and *S2,* answer the following questions respectively:

(1) Draw the precedence graph for the schedule.

(2) Is the schedule conflict-serializable? If yes, describe all the conflict-equivalent serial schedules. (You do not need to explicitly write out the serial schedules, just use the form like < T1, T3, T2 > .) If no, explain why it is not.

(3) Could the schedule be created by the 2PL protocol? If yes, put the lock and unlock actions (e.g. use ‘Ls(Q)’ for Lock-s(Q), ‘Lx(Q)’ for Lock-x(Q), ‘UL(Q)’ for Unlock(Q)) into the schedule to show it can be generated by 2PL. If not, explain briefly why.

Problem 5 Crash Recovery (20 points)

Consider the following transaction log of a database system. Suppose the system crashes just after the last log record.

1) < T1 START >
2) < T1, A, 20, 30 >
3) < CKPT (T1) >
4) < T2 START >
5) < T1, A, 30, 50 >
6) < T2, B, 52, 66 >
7) < T3 START >
8) < T3, E, 0, 15 >
9) < T2 COMMIT >
10) < CKPT (T1,T3) >

11) < T3, E, 15, 40 >
12) < T1, C, 45, 90 >
13) < T1 COMMIT >
14) < T3 ABORT >
15) < T4 START >
16) < T4, D, 39, 0 >

Assume the log entries are in the format <T_{id}, Variable, Old value, New value >.
Please answer each of the following questions:

(1) What are the values of A, B, C, D and E in the database just after system crash?
(2) Which transactions should be undone? Which transactions should be redone?
(3) What are the start and end points for undo and redo processes respectively?
(4) What are the values of A, B, C, D and E after system recovery ?

Problem 1: Buffer Manager (12 points, 3 points per part)

Consider buffering of disk blocks in memory under the assumption that, to begin with, the buffer is empty, and that the maximum number of blocks that can be held in buffer is four. Assume in the following schedule that each of the data items A, B, C, D, E, F accessed in the schedule lies on distinct blocks.

$R_1(A,1)$ $R_2(B,2)$ $W_1(A,3)$ $R_3(C,4)$ $W_2(B,5)$ C_2 $W_3(C,6)$ $R_4(D,7)$

$R_5(E,8)$ $W_5(E,9)$ $R_6(A,3)$ $R_3(F,10)$ $W_3(F,11)$ $W_4(D,12)$

Where: R_i (X, V) means transaction T_i read value V from data X.

W_i (X, V) means Transaction T_i write value V to data X.

C_i means transaction T_i commits.

1) Name the first operation where an existing block in buffer must be dropped in order that another block can be read in.
2) Blocks in buffer are called dirty if they have been updated in buffer but not yet written back out to their place on disk. What are the dirty blocks in buffer at the time of the operation named in 1)?
3) Assume that we are using an LRU buffer-replacement policy, and that each block is in use only for duration of the read or write that access it. What block will be dropped from buffer at the time of the operation named in 1)
4) Can we simply drop the block mentioned in 3) at that time, forgetting its value, or must some other event take place first? Why?

Problem 2: B+ -Tree (16 points, 4 points per part)

1) Construct a B+-tree (fan-out rate is 4) for the following set of key values: (5,10,15,20,25,30,35,40, 45, 50,55) .
   Assume that the tree is initially empty and values are added in ascending order.
2) For the B+-tree constructed in 1), show the form of the tree after the operation "delete 40".
3) Assume that the B+-tree (fan-out rate is 4) contains 10000 index items, please estimate the height of the B+-tree.
4) Assume that the B+-tree ((fan-out rate is 4)) contains 10000 index items, please estimate the size (i.e. the number of nodes) of the B+-tree.

Problem 3: Query Optimization (10 points, 5 points per part)

Consider the following relational schema and SQL query:

account(**account_no: char(10)**, customer_name: char(10), branch: char(20), balance: integer)

access (**serial char(10)**, account_no: char(10), amount: integer,
access_branch: char(20), access_date: date)
note: the account_no of table access *references to table account*.

```
select  account.account_no, account.customer_name
from    account , access as A1, access as A2
where account.account_no = A1.account_no and
        A1.account_no= A2.account_no and
        A1.access_branch =A2.access_branch and
        A1.access_date=A2.access_date and
        account.branch='Shanghai branch' and
        A1.amount<=1000 and A2.amount>=63500
```

1) Identify a relational algebra expression that reflects the order of operations that a query optimizer would choose after algebra optimization.
2) What indexes might be of help in processing this query? Explain briefly.

Problem 4: Query Size Estimation (10 points)

For the relational schema defined in problem 3, there are following assumptions:

- **account** table has 10,000records
- **access** table has 1,000,000 records
- **branch** attribute has 100 distinct values that are uniformly distributed.
- The value of **access_date** attribute is between '2001-01-01' and '2010-12-31'
- The value of **amount** attribute is between 1 and 100,000 that are uniformly distributed.

Please estimate the size (i.e. number of records) returned by the SQL statement given in the **problem 3**. (Write out your answer step by step).

Problem 5: Query Cost Estimation (10 points)

For the relational schema defined in **problem 3**, assume that:

- The **account** table has 10,000 records
- The **access** table has 1,000,000 records
- The file system support 4K byte blocks.
- There are 50 buffer pages (blocks) available in memory for operating join.
- The attribute with 'integer' type needs 4 bytes.
- The attribute with 'date' type needs 4 bytes.

Please estimate the best cost for evaluating **account** $\bowtie$ **access** with Block Nested-Loop Join method. (Hint: the cost is measured with the number of blocks transferred to main memory and the times to seek disk.)

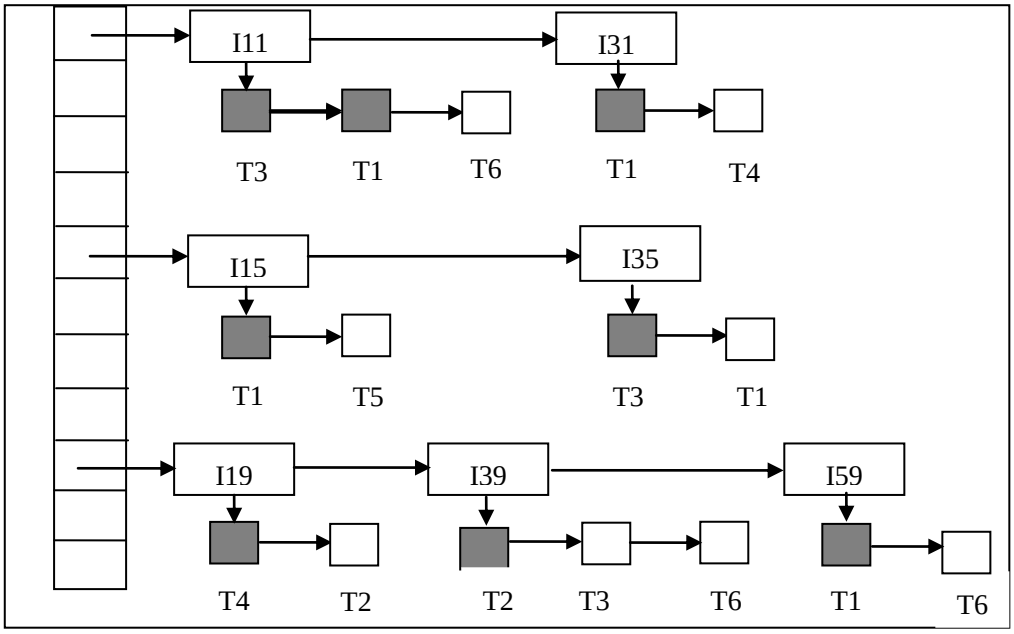
Problem 6: Two Phase Locking protocol (10 points)

Prove that in a schedule where each transaction obeys the Two Phase Locking protocol, it is possible to move all the read and write actions of the transaction with the first unlock action forward to the beginning of the schedule without passing any conflicting actions.

Problem 7: Deadlock Handling (12 points, 4 points per part)

The following figure shows an instance of a lock table. There are 6 transactions (T1-T6) and 7 data items (I11, I31, I15, I35, I19, I39, I59). Granted locks are filled (black) rectangles, while waiting requests are empty rectangles.

1) Which transactions are involved in deadlock?
2) In order to break the deadlock and release most lock resources, which transaction (victim) should be rolled back?
3) Please draw the lock table after the victim transaction of 2) is rolled back.



Problem 8: Crash Recovery (20 points, 4 points per part)

Consider the following log sequence of transactions T1, T2, T3, T4 and T5. Supposing the system crashes just after the last log record. Please answer each of the following questions:

1) What are the values of A, B , C and D in the database after system crash?
2) Which transactions should be undone? Which transactions should be redone?
3) What are the start and end points for undo and redo processes respectively?
4) What are the values of A, B , C and D after system recovery?
5) If the system crashes again while the system is doing database recovery from

previous crash, how does the DBMS handle such kind situation?

[1] <T1 **start**>
[2] <T1, A, 10, 20>
[3] <T2 **start**>
[4] <T2, B, 30, 40>
[5] <T1, C, 50 ,60>
[6] <T3 **start**>
[7] <checkpoint { T1,T2,T3}>
[8] <T3, D, 70, 80>
[9] <T2 **commit**>
[10] <T3, B, 40, 50>
[11] <T1 **commit**>
[12] <checkpoint {T3}>
[13] <T3, A, 20, 30>
[14] <T4 **start**>
[15] <T4, C, 60, 70>
[16] <checkpoint {T3,T4}>
[17] <T3 **commit**>
[18] <T5 **start**>
[19] <T5 , A, 30, 40>

浙江大学 2015－2016 学年秋季学期

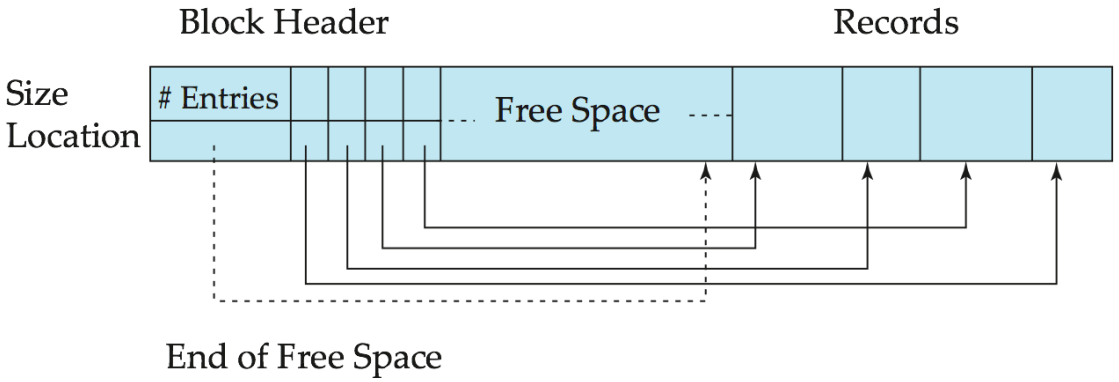
《数据库系统设计》课程期末考试试卷-参考答案

Problem 1: Variable-length Records (10 points)

1)

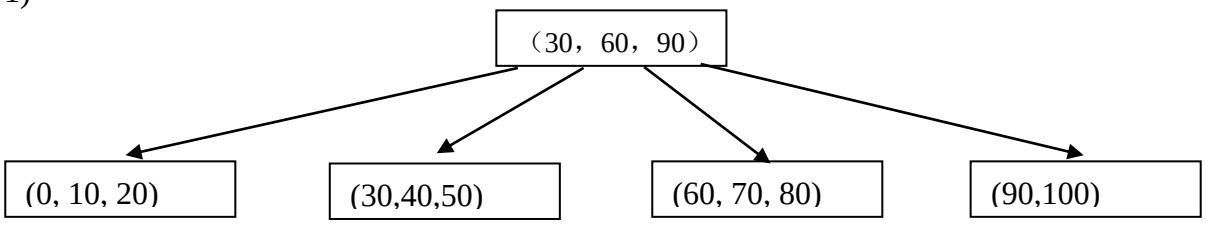
1001	15,10	60		0001	Bill Gates	
0	4	8	10	14	15	
1002	15,12	60	27,20	0000	Bob Williams	111, State Street, MA.
0	4	8	10	14	15	27
1001	15,12			0011	John Harvard	
0	4	8	10	14	15	

2)

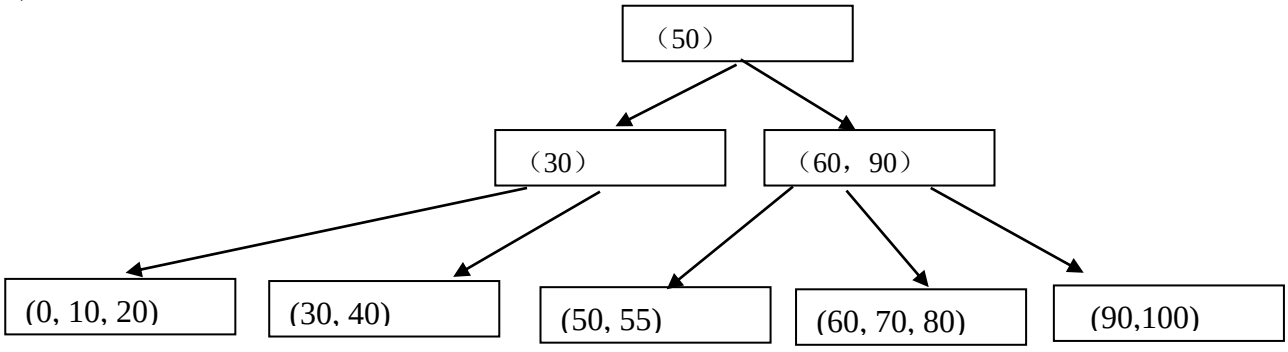


Problem 2: B+ -Tree (16 points, 4 points per part)

1)



2)



3）min height : $\log_4(10000)$ = 7 (up) max height: log 2 (10000/2) +1 = 13 (low)
4) min size: 4443 maz size: 9995

Problem 3: Cost Estimation (18 points, 9 points per part)

1) 100000*1/4*1/50*80000/1000000=100000*1/4*1/50*8/100=40

2) Size of account record=44

Number of account record per block=4096/44 = 93

Number of blocks of account= 10000/93=108

Size of access record=28

Number of account record per block=4096/28 = 146

Number of blocks of account= 100000/146=685

br/M-2*bs+br = 108/98*685+ 108 = 2*685+108 =1478 block transfers

2*br/(M-2) seeks = 2*108/98 = 4 seeks

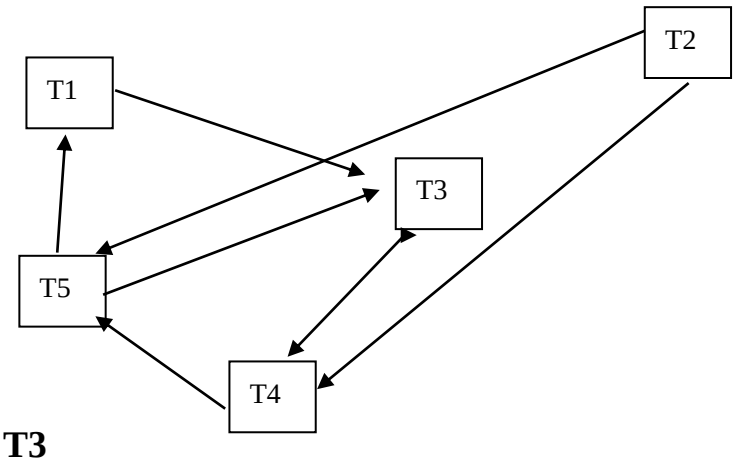
Problem 4: Materialized View (10 points, 5 points per part)

1) π account.account_no, account.customer_name, access.access_date(
δ branch_name=’Hangzhou’(account)⋈ δ ammount>=50000(access))

2)
π account.account_no, account.customer_name, access.access_date(
δ branch_name=’Hangzhou’(account)⋈ δ ammount>=50000(access∪S))
=
π account.account_no, account.customer_name, access.access_date(
δ branch_name=’Hangzhou’(account)⋈(δ ammount>=50000(access)∪ δ ammount>=50000(S)))
=
π account.account_no, account.customer_name, access.access_date(
δ branch_name=’Hangzhou’(account)⋈(δ ammount>=50000(access))
∪
π account.account_no, account.customer_name, access.access_date(
δ branch_name=’Hangzhou’(account)⋈(δ ammount>=50000(S))
=V ∪V’, 其中:
V’ =π account.account_no, account.customer_name, access.access_date(
δ branch_name=’Hangzhou’(account)⋈(δ ammount>=50000(S))
=π account.account_no, account.customer_name, access.access_date(
V”⋈(δ ammount>=50000(S))
V”=δ branch_name=’Hangzhou’(account)

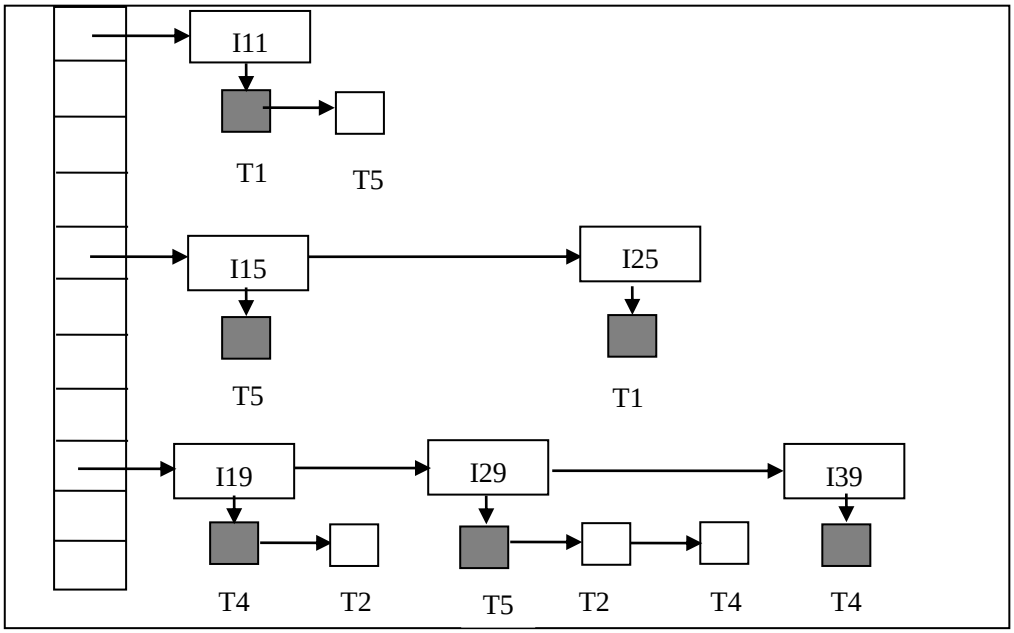
Problem 5: Deadlock Handling (12 points, 4 points per part)

1） T1，T3 ，T4， T5



2） T3

3）



Problem 6: Transactions (10 points)

要点为：

由于事务遵循 2PL，事务持有的锁并不是用好即释放，而是要等到在 shrinking 阶段才释放，因此，对于访问频繁、冲突概率高的数据操作（更新分行的总现金余额），要放在冲突概率低的数据操作（更新某账户的余额）之后执行，这样才能总体减少锁的等待时间，提高并发度，从而提高事务的吞吐率。

Problem 7: Crash Recovery (16 points, 4 points per part)

1） redo : T1，T3
undo : T4
2） start point of redo phase: [12]
end point of undo phase: [10]
3） A 33 B 33 C 22 D 22
4）

[17] <T4, D, 22>
[18] <T4 abort>

Problem 8: Aries Recovery Method (8 points)

Aries recovery method is widely used in industrial DBMSs. Please answer following questions about ***Aries***.

1) What contents are in the checkpoint log of ***Aries***?
   Dirty Page Table
   Active Transaction List
2) Why checkpoint operation of ***Aries*** puts less side effects on normal transaction processing of DBMS?
   Doesn't output dirty pages in buffer to disk during checkpointing.

浙江大学 2015－2016 学年秋季学期

《数据库系统设计》课程期末考试试卷

课程号：21120302，开课学院：计算机学院

考试试卷：√A 卷、B 卷（请在选定项上打√）

考试形式：闭、开卷（请在选定项上打√），允许带 1 张 A4 纸笔记 入场

考试日期：2015 年 11 月 13 日，考试时间：120 分钟

诚信考试，沉着应考，杜绝违纪。

考生姓名：_____学号：_____所属院系：_____

题序	一	二	三	四	五	六	七	八	总 分
得分									
评卷人									

Problem 1: Variable-length Records (10 points)

Variable-length records arise in database systems in several ways. Following SQL statement creates a table ***person***, in which ***name*** and ***address*** are variable-length attributes. In addition, ***age*** and ***address*** could be empty.

```
create table person
(   id char(4) primary key,
    name varchar(20) not null,
    age smallint,
    address varchar(20) )
```

Following is an instance of ***Person*** table with 3 variable-length records.

id	name	age	address
1001	Bill Gates	60	NULL
1002	Bob Williams	50	111, State Street, MA.
1003	John Harvard	NULL	NULL

1) Show by example how to implement the representation of variable-length records.
2) Show by example how to store variable-length records in a block with slotted-page structure.
   (Assume that smallint type is 2 bytes, and block size is 1024 bytes.)

Problem 2: B+ -Tree (16 points, 4 points per part)

1) Construct a B+-tree (n=4) for the following bulk of index entries:
   (0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
   Assume that the tree is initially empty, and the bottom-up B+-tree construction method is used to bulk loading the above index entries.
2) For the constructed B+-tree, draw the form of the tree after operation "insert 55".
3) Assume that the B+-tree (n=4) contains 10000 index items, please estimate the height of the B+-tree.
4) Assume that the B+-tree (n= 4)) contains 10000 index items, please estimate the size (i.e. the number of nodes) of the B+-tree.

Problem 3: Cost Estimation (18 points, 9 points per part)

Consider the following relational schema and SQL query:

account(account_no: char(10), customer_name: char(10),
branch: char(20), balance: integer)
access (serial char(10), account_no: char(10), access_date: date, amount: integer)
note: account_no of table ***access*** *references to table* ***account****.*

select account.account_no, account.customer_name
from account , access
where account.account_no = access.account_no **and** year(access_date)=2012
and branch='Hangzhou' **and** amount between 10000 and 89999.

1) Estimate the size (i.e. number of records) returned by above SQL statement.
2) Estimate the cost, in the best case, for evaluating **account** ⋈ **access** with Block Nested-Loop Join method.

Assumptions:

- account ***table*** has 10,000 records; ***access*** table has 100,000 records
- ***branch*** attribute has 50 distinct values.
- The value of a***ccess_date*** attribute is between '2011-01-01' and '2014-12-31'
- The value of ***amount*** attribute is between 1 and 1,000,000.
- The values of all attributes are uniformly distributed.
- The integer type needs 4 bytes.
- The date type needs 4 bytes.
- The file system supports 4K byte blocks.
- There are 100 buffer blocks available in memory for evaluating the join operation.

Problem 4: Materialized View (10 points, 5 points per part)

A materialized view is a view whose contents are computed and stored. Materialized views are important for improving query performance in some applications. Consider following materialized view ***big_deal*** over the tables given in problem 3:

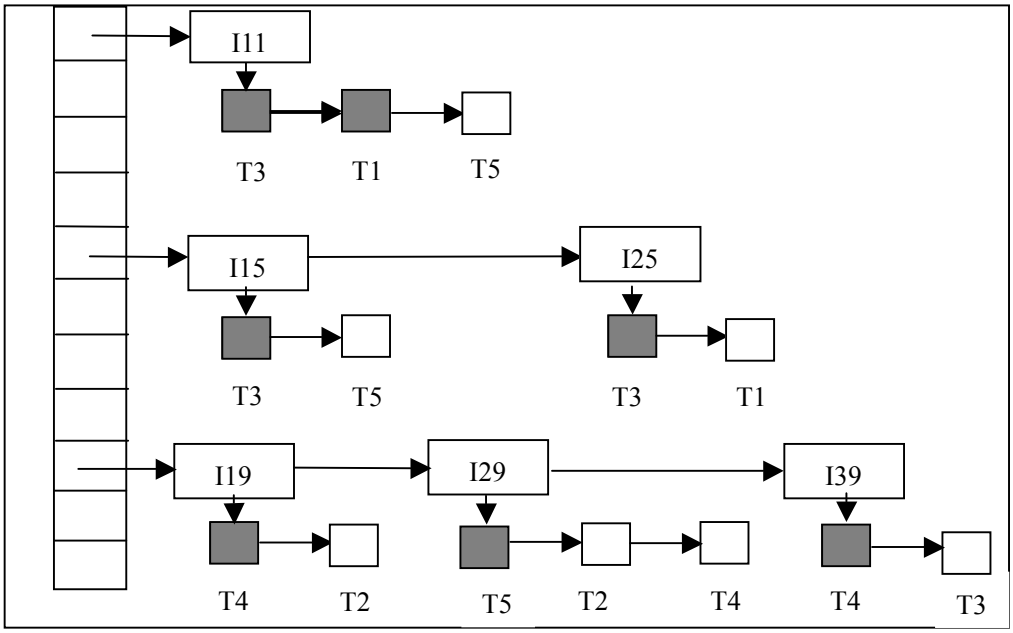
```
create materialized view big_deal(account_no, customer_name, access_date) as
select   account.account_no, account.customer_name, access.access_date
from     account , access
where    account.account_no = access.account_no and branch_name='Hangzhou'
         and amount >=50000;
```

1) Please identify a relational algebra expression that reflects the definition of the materialized view above.
2) If inserting a set ***S*** of records into the **access** table, explain how to incrementally maintain the materialized view ***big_deal***?

Problem 5: Deadlock Handling (12 points, 4 points per part)

The following figure shows an instance of a lock table. There are 5 transactions (T1-T5) and 6 data items (I11, I15, I19, I25, I29, I39). Granted locks are filled (black) rectangles, while waiting requests are empty rectangles.

1) Which transactions are involved in deadlock?
2) In order to break the deadlock and release most lock resources, which transaction (victim) should be rolled back?
3) Please draw the lock table after the victim transaction of 2) is rolled back.



Problem 6: Transactions (10 points)

In banking application, "Deposit & Withdraw" transactions have two operations:
operation_A updates the cash balance at a branch(a common item)
operation_B updates the individual account balance(a private item)
Explain how you can increase concurrency (and throughput) by ordering the operations of the "Deposit & Withdraw" transaction. Assume that all transactions follow the two-phase locking protocol.

Problem 7: Crash Recovery (16 points, 4 points per part)

Consider the following log sequence of transactions T1, T2, T3, and T4. Supposing the system crashes just after the last log record. Please answer each of the following questions:

1) Which transactions should be redone? Which transactions should be undone?
2) What is the start point of redo phase? What is the end point of undo phase?
3) What are the values of A, B, C and D after recovery?
4) What log records are added to log file during recovery?

[1] <T1 **start**>
[2] <T1, A, 11, 22>
[3] <T2 **start**>
[4] <T2, B, 11, 22>
[5] <T3 **start**>
[6] <T3, C, 11, 22>
[7] <checkpoint { T1,T2,T3}>
[8] <T1, D, 11 ,22>
[9] <T2 **commit**>
[10] <T4 **start**>
[11] <T3, B, 22, 33>
[12] <checkpoint { T1, T3,T4}>
[13] <T1 **commit**>
[14] <T3, A, 22, 33>
[15] <T4, D, 22, 33>
[16] <T3 **commit**>

Problem 8: Aries Recovery Method (8 points)

Aries recovery method is widely used in industrial DBMSs. Please answer following questions about ***Aries***.

1) What contents are in the checkpoint log of ***Aries***?
2) Why checkpoint operation of ***Aries*** puts less side effects on normal transaction processing of DBMS?

浙江大学 2018–2019 学年春季学期

《数据库系统》课程课堂测试一

(Quiz 1 for Database Systems)

考生姓名：_____ 学号：_____ 专业：_____ 得分：_____

Problems concerning the basic concepts of database systems

1. **According to the levels of data abstraction, database systems have several schemas, which are as follows:**

Physical schema, **Logical schema**, and **View schema**.

2. **List the names of data model used in database systems, and explain briefly how or when to use it.**

(i) **E-R model:** It is used in the conceptual database design, which describes the real world objects with entities, and relationships among these objects.

(ii) **Relational model:** It is used in the logical database design. It is the basis of the most DBMS products in the database world for more than 30 years, and it uses tables to represent both data and the relationships among those data.

(iii) **Object-oriented model:** It is heavily influenced by object-oriented programming language and can be understood as an attempt to add DBMS functionality to a programming language environment. It can be regarded as extending the E-R model with notions of encapsulation, methods and object identity, and support complex data structure. The model can be used in logical database design in most cases, and can also be used in conceptual design step.

(iv) **Object-relational model:** It extends the relational model with features of object-oriented model.

(v) **Semistructured-data model:** It is used in the logical database design steps. In this model, XML is widely used to represent semistructured data.

(vi) **Network model:** Data are represented by collections of records, and relationships among data are represented by links, which can be viewed as pointers. The records in the database are organized as collections of arbitrary graphs. It was only used in old database systems.

(vii) **Hierarchical model:** It is similar to the network model in the sense that data and relationships among data are represented by records and links, respectively. But it differs from the

network model in that the records are organized as collections of trees rather than arbitrary graphs. It was used only in old database systems.

3. **What are the physical data independence and logical data independence in database system?**

Physical Data Independence: the ability to modify the physical schema without changing the logical schema, as well as application programs, that is, the modification in the physical schema does not affect the logical schema, there application program need not be changed.

Logical data independence: Protect application programs from changes in *logical* structure of data, i.e., the modification in the logical schema does not affect the view schema, there application program need not be changed.

浙江大学 2018–2019 学年春夏季学期

《数据库系统》课程课堂测试二

(Quiz 2 for Database Systems)

考生姓名：_____ 学号：_____ 专业：_____ 得分：_____

Consider the following relation schemas and then answer the subsequent problems. Note that the key attributes in the relation schemas are underlined.

***Student*(*Sid*, *Name*, *Age*)**
***Project*(*ProjectName*, *Sid*, *Score*)**

Problem 1. Relational algebra

(1) Find the names of students who are in the project with project name 'MiniSQL'.

$\Pi_{name}((student) \bowtie (\sigma_{projectName='MiniSQL'}(project)))$

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

$\Pi_{sid}(student) - \Pi_{sid}(project)$

(3) Find the names of students who are the youngest.

Method 1: $\Pi_{name}(student) - \Pi_{name}(\sigma_{student.age>st2.age}(student \times (\rho_{st2}(student))))$

Method 2: Temp $\leftarrow g_{\min(Age)}(student)$;

$\Pi_{name}(\sigma_{age=minage}(student \times (\rho_{T(minage)}(\text{Temp}))))$

Problem 2. Write SQL statement for the following queries.

(1) Find the names of students who get score more than 85 in the project.

```
SELECT Name
FROM Student S, Project P
WHERE S.Sid = P.Sid and Score > 85
```

(2) Find the names of students who get the maximum score in each project.

Method 1:
```
SELECT Name
FROM Student S, Project P
WHERE S.Sid = P.Sid and Score = (SELECT max(Score)
                                 FROM Project
                                 WHERE ProjectName = P.ProjectName)
```

Method 2:
```
SELECT Name
FROM Student
WHERE Sid in (SELECT Sid
              FROM Project P
              WHERE Score >= ALL (SELECT Score
                                  FROM Project
                                  WHERE ProjectName = P.ProjectName))
```

Method 3:
```
SELECT Name
FROM Student S, Project P
WHERE S.Sid = P.Sid and (ProjectName, Score) in (SELECT ProjectName,
                                                        max(Score)
                                                 FROM Project
                                                 GROUP BY ProjectName)
```

浙江大学 2018–2019 学年春季学期

《数据库系统》课程课堂测试三

(Quiz 3 for Database Systems)

考生姓名：_____ 学号：_____ 专业：_____ 得分：_____

Problem 1. Consider a SQL table ***T*(*A int unique*, *B int*).** Assume there are no NULL values. As specified, attribute *A* is a key. Consider the following three SQL queries:

Q1: Select *B* From *T*
Where *B* >= some (Select *B* From *T*);

Q2: Select *B* From *T* as T_1
Where *B* > all (Select *B* From *T* as T_2 where T_2.A < > T_1.A);

Q3: Select max(*B*) From *T*;

Which of the queries above are equivalent? Please show a smallest single instance of *T* you can come up with that demonstrates your answer.

Case 1: Q2 and Q3 are equivalent, if the maximum of *B* is only once.
Case 2: None are equivalent, if the maximum of *B* is multiple times.

A	B
1	5
2	7
3	3

Case 1

Q1: 5
7
3
Q2: 7
Q3: 7

A	B
1	7
2	5
3	7

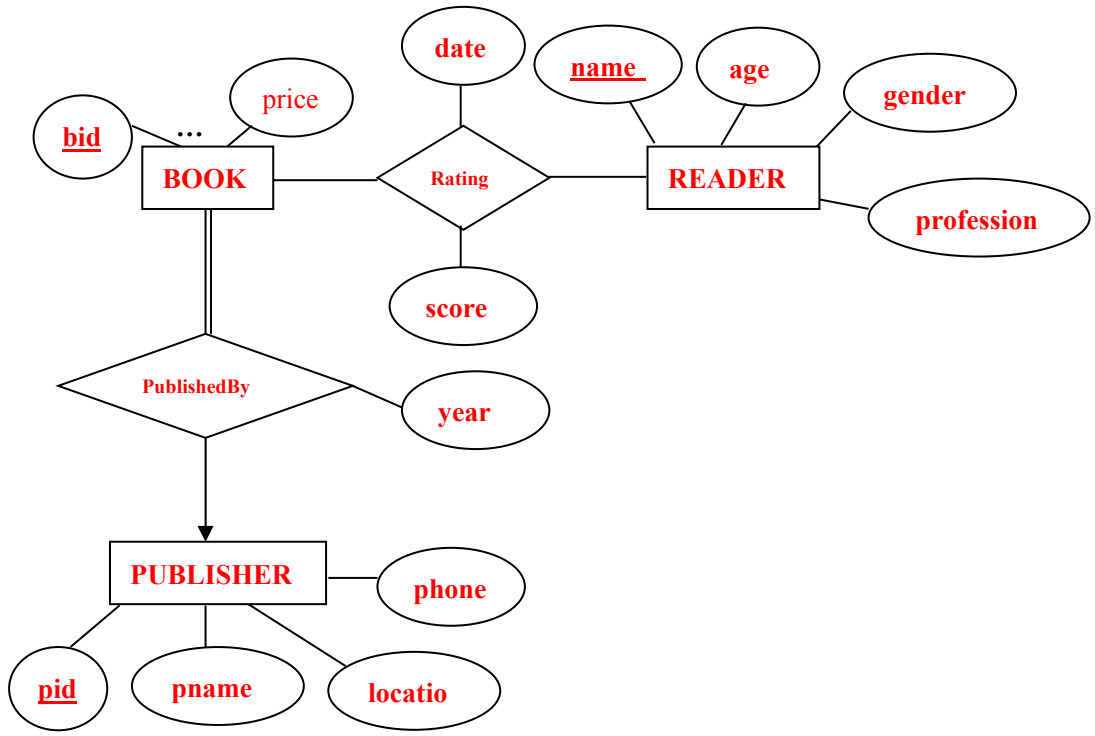
Case2

Q1: 5
7
3
Q2:
Q3: 7

Problem 2. Consider the following relational schemas describing ***books***, ***publishers***, ***readers***, and reader ratings of the books:

***Book*(*bid*, *title*, *author*, *price*)** // ***bid* is a primary key**
***Reader*(*name*, *age*, *gender*, *profession*)** // ***name* is a primary key**
***Rating*(*name*, *bid*, *date*, *score*)** // **(*name*, *bid*) is a primary key**
***Publisher*(*pid*, *pname*, *location*, *phone*)** // ***pid* is a primary key**
***PublishedBy*(*bid*, *pid*, *year*)** // ***bid* is a primary key**

(1) Draw an E-R diagram from which these relational schemas could have been produced. Your diagram should be fully connected, and it should be as detailed as possible from the information you have.



(2) Please make necessary formalization of the relational schemas above, to get a minimum number of relation schemas.

Book(*bid*, *title*, *author*, *price*, *pid*, *year*) // *bid* is a primary key

Reader(*name*, *age*, *gender*, *profession*) // *name* is a primary key

Rating(*name*, *bid*, *date*, *score*) // (*name*, *bid*) is a primary key

Publisher(*pid*, *pname*, *location*, *phone*) // *pid* is a primary key

(3) Write SQL data definition statements for the relation schemas from the issue/step (2), and give necessary integrity constraints on them.

```
Create table Publisher(pid        char(10) primary key,
                       pname      varchar(30),
                       location   varchar(50),
                       phone      varchar(20));

Create table Book(bid        char(10) primary key,
                  title      varchar(50),
                  author     varchar(50),
                  price      real,
                  pid        char(10),
                  year       date,
```

```
                    Foreign key(pid) references publisher);

Create table Reader(name          varchar(15) primary key,
                    age           int,
                    gender        char(1) not null,
                    profession    varchar(30),
                    check (gender in ('M', 'F')));

Create table Rating(name      varchar(15),
                    bid       char(10),
                    date      date,
                    score     real,
                    primary key (name, bid),
                    foreign key (name) references reader,
                    foreign key (bid) references book);
```

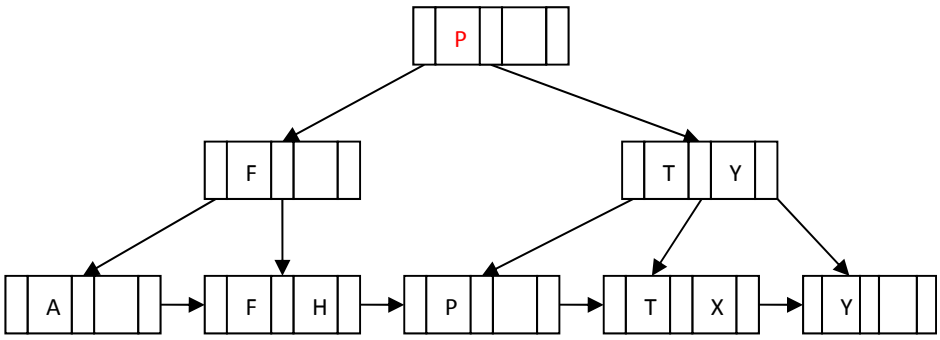
浙江大学 2018–2019 学年春季学期

《数据库系统》课程课堂测试四

(Quiz 4 for Database Systems)

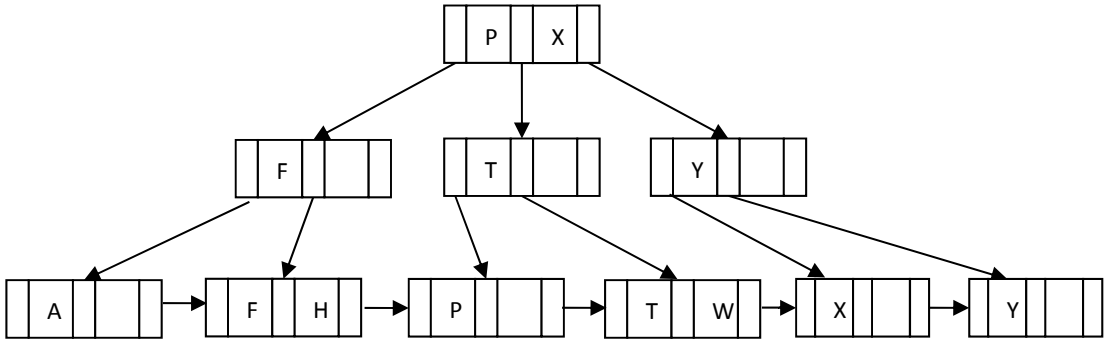
考生姓名：_____ 学号：_____ 专业：_____ 得分：_____

Problem 1. For the following B+ tree (n=3):



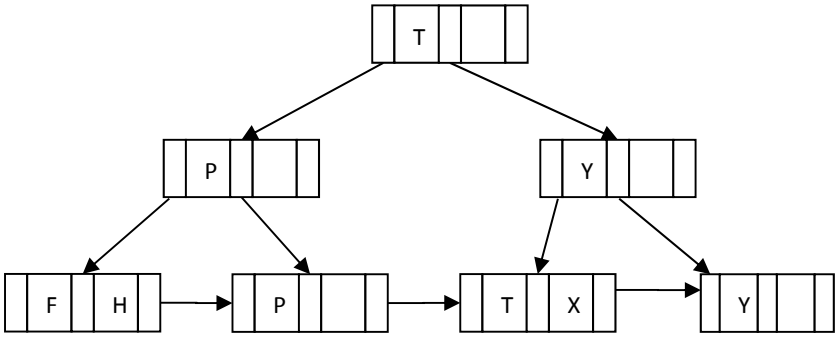
(1) Draw the B+ tree after insert an index item with key 'W' to the given tree.

Solution:



(2) Draw the B+ tree after delete an index item with key 'A' from the original tree.

Solution:



(3) Assume that the B+ tree contains 1000 index items, please estimate the height of the B+ tree.

$$\left\lceil log_3 \frac{1000}{2} \right\rceil + 1 \leq \text{height} \leq \left\lceil log_{\lceil \frac{3}{2} \rceil} 1000 \right\rceil + 1$$

$$7 \leq \text{height} \leq 11$$

(4) Assume that the B+ tree contains 1000 index items, please estimate the size (i.e. the number of nodes) of the B+ tree.

$$\text{size} \geq \left\lceil \frac{1000}{2} \right\rceil + \left\lceil \left\lceil \frac{1000}{2} \right\rceil /3 \right\rceil + \left\lceil \left\lceil \left\lceil \frac{1000}{2} \right\rceil /3 \right\rceil /3 \right\rceil + \cdots + 1 = 755$$

$$\text{size} \leq 1000 + \lceil 1000/2 \rceil + \lceil \lceil 1000/2 \rceil /2 \rceil + \cdots + 1 = 2001$$

$$755 \leq \text{size} \leq 2001$$

Problem 2. Consider the following relational schema and SQL query:

product(pid: char(10), name: char(20), producer: char(20), price: integer)
customer(cid: integer, name: char(20), age: integer; city: char(20))
order(cid: integer, pid: char(10))

select customer.name, product.name
from customer, order, product
where customer.cid= order.cid and product.pid = order.pid
customer.city =‘Hangzhou’ and product.price>=200;

(1) Identify a relational algebra tree (or a relational algebra expression if you prefer) that reflects the order of operations that a decent query optimizer would choose.

$$\prod_{\text{customer.name,product.name}} (\sigma_{\text{city}='\text{Hangzhou}'}(\text{customer}) \bowtie \text{Order} \bowtie \sigma_{\text{price}\geq 200}(\text{product}))$$

(2) What indexes might be of help in processing this query? Explain briefly.
Since the pid (in product and order) and cid (in customer and order) is crux of the join, it’s helpful to create B+tree indexes in these attributes. (Different answer is OK if it makes sense.)

浙江大学 2018–2019 学年春季学期

《数据库系统》课程课堂测试五

(Quiz 5 for Database Systems)

考生姓名：_____ 学号：_____ 专业：_____ 得分：_____

Problem 1. Consider the following relational schema and SQL query:

product(pid: char(10), name: char(20), producer: char(20), price: integer)
customer(cid: integer, name: char(20), age: integer; city: char(20))
order(cid: integer, pid: char(10))

assume that: The *product* table has 100,000 records
The *order* table has 2,000,000 records
The file system support 4K byte blocks.
There are 60 buffer pages (blocks) for operating join.
The attribute with 'integer' type needs 4 bytes.

Please estimate the best cost for evaluating **product** ⋈ **order** with Block Nested-Loop Join method.

The *product* table needs $(10 + 20 + 20 + 4) * 100{,}000/4K = 1350$ blocks.
The *order* table needs $(4 + 10) * 2{,}000{,}000/4K = 7000$ blocks.
Both of them is larger than 60 buffer pages.
If *product* is the outer relation, we need $\lceil 1350/(60-1) \rceil * 7000 + 1350 = 162350$ disk accesses.
If *order* is the outer relation, we need $\lceil 7000/(60-1) \rceil * 1350 + 7000 = 167650$ disk accesses.

Problem 2. Consider the following log sequence of transactions.

1. <T_1 start>
2. <T_1, A, 1, 2>
3. <T_2 start>
4. <T_2, B, 3, 4>
5. <T_3 start>
6. <T_3, C, 5, 6>
7. <checkpoint{T_1, T_2, T_3 } >
8. <T_1 commit>
9. < T_3, A, 2, 7>
10. < checkpoint{T_2, T_3 }>
11. < T_2, B, 4, 8>
12. < T_3 commit >
13. <T_4 start>
14. < T_4, A, 7, 9>
15. <T_5 start>
16. < T_5, C, 6, 10>

Supposing the system crashes just after the last log record. Please answer each of the following questions:

(1) What are the values of *A*, *B*, and *C* in the database after system crash?

A = 9; B=8; C=10

(2) Which transactions should be undone? Which transactions should be redone?

Undone: T2, T4, T5;

Redone: T3

(3) What are the start points for undo and redo processes respectively?

Undo: 16.

Redo: 10.

(4) What are the values of *A*, *B* and *C* after system recovery?

A=7; B=3; C=6