2020-2021 学年第 2 学期

（2021 春季）

《数据管理技术》

期末考试卷

班级_____学号_____

姓名_____成绩_____

2021 年 6 月 23 日

Problem 1: Single-Choice Questions (10 points).

20. In two phase locking, transaction obtains lock and release it in (respective order):

a) Shrinking phase & Initial Phase
b) Running phase & Completion Phase
c) Growing phase & Shrinking phase Answer
d) Initial phase & Completion Phase
e) None of the above

Consider following transactions

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

Which of the following schedules below is the correct serialization (i.e., an equivalent non-serial schedule) of the above?

a) T1 - >> T2 ->> T3
b) T2 ->> T3 ->> T1
c) T1 ->> T3 ->> T2 **Answer**
d) T3 ->> T2 ->> T1
e) This is a non-serializable schedule.

The database system must take special actions to ensure that transactions operate properly without interference from concurrently executing database statements. This property is referred to as

a) Atomicity
b) Consistency
c) Isolation Answer
d) Durability
e) All of the above

Which property is guaranteed by the two-phase locking protocol?

(a) serial schedules
(b) x serializable schedules
(c) recoverable schedules

(d) avoiding cascading rollback

Which is not an advantage of a DBMS over file-based systems:
a. Control of data redundancy
b. Improved data consistency
c. Easier data sharing.
d. Improved security.
e. Improved performance. ANSWER

Select the relational expression that could possibly return the following result:
ac
12
23

a. $\sigma_{a<c}(\pi_{a,\,c}\,R)$ ANSWER

b. $\pi_{a<c}(\pi_{a,\,c}\,R)$

c. $\pi_{a<2}\,R$

d. $\sigma_{a,c}R$

e. $\pi_{a,c}(\sigma_{a=c}\,R)$

A foreign key always refers to a primary key in the same table. True or False?

False (*)

(e) (3 pts) Briefly (using one sentence please) explain the meaning of the following trigger.

```
CREATE TRIGGER FooTrigger
AFTER UPDATE OF salary ON Emp
REFERENCING
  OLD AS OldTuple,
  NEW AS NewTuple
WHEN ((OldTuple.salary > NewTuple.salary) AND
     (OldTuple.name = 'Tom Smith'))
```

```
    UPDATE Emp

SET salary = OldTuple.salary

    WHERE  SSN = NewTuple.SSN
  FOR EACH ROW;
```

1. In relational algebra, R ($\bowtie_{R.B<S.C}$) S can be rewritten as: 【 C 】

A. $R \times_{R.B<S.C} S$

B. $\Pi_{R.B<S.C} (R \times S)$

C. $\sigma_{R.B<S.C} (R \times S)$

D. None of the above

2. Consider a relation R(A, B, C, D, E), in which A, B is the primary key. R has the following functional dependencies: (A, B) → (C, D, E) A→C

1) R is in at least 1st normal form because: 【 B 】

A. There are no partial dependencies
B. There are no multivalued attributes
C. There are dependencies on a non-candidate key attribute
D. There are no transitive dependencies
E. It is not in 1NF.

2) R is not in 2nd normal form because: 【 A 】

A. There are partial dependencies
B. There are multivalued attributes
C. There are dependencies on a non-candidate key attribute
D. There are transitive dependencies
E. It is already in 2NF.

3) After transforming R into 2nd normal form, the resulting relations would be: 【 D 】

A. It is already in 2nd normal form.
B. R1 (A, B, C, D) & R2 (A, E)
C. R1 (A, C, E) & R2 (B, D)
D. R1 (A, B, D, E) & R2 (A, C)
E. None of the above.

3. Which of these statements about serializable schedules is true? 【 C 】

A. Every serializable schedule is recoverable.

B. Every serializable schedule contains no conflicting actions.
C. Every 2PL schedule is serializable.
D. None of the above.

Problem2: Indicate true or right of each statement (5 points)

Suppose that the Take_course relation was created as follows:

```
CREATE TABLE Take_course(
  courseID INT REFERENCES course_ID ON DELETE SET NULL,
  stuID INT REFERENCES stu_ID ON DELETE CASCADE )
```

Note: course_ID is the key of Course relation; stu_ID is the key of Student relation.

1. If we try to delete a tuple from Take_course, the tuple is not deleted. Instead, courseID is set to NULL. 【 False 】

2. If we delete a tuple from Take_course, any tuples in student referred to by this tuple are also deleted. 【 False 】

3. If we delete a tuple from Course, some attributes of Take_course may have their values set to NULL. 【 True 】

4. If we try to insert a tuple into Student, with an ID that is not referred to in Take_course, the operation is rejected. 【 False 】

5. If we try to insert a tuple into Take_course, with an ID that does not exist in Student, the operation is rejected. 【 True 】

Problem 3: Answer the questions (15 points)

1. How many categories can SQL statements be divided into? Define them..

2. Please make a comparison study for the two consistency models --BASE and ACID.

3. When will a concept be modeled as an attribute, or an entity set, during ER modelling?

4. What are the different types of Normalization?

5. what does “database integrity” mean in database? List at least 3 integrities that you know.

Problem 4: Relational algebra and SQL (30 Points)

The following database contains information about actors, plays, and roles performed.

Actor (actor_id, name, year_born)

Play (play_id, title, author, year_written)

Role (actor_id, character_name, play_id)

Where:

• Actor is a table of actors, their names, and the year they were born. Each actor has a unique actor_id, which is a key.

• Play is a table of plays, giving the title, author, and year written for each play. Each play has a unique play_id, which is a key.

• Role records which actors have performed which roles (characters) in which plays. Attributes actor_id and play_id are foreign keys to Actor and Play respectively. All three attributes make up the key since it is possible for a single actor to play more than one character in the same play.

➢ Write **relational algebra** expressions for the following queries:

1. Find the actors' name who have participated in all the plays.

Π_{name} (($\Pi_{actorid,\ playid}$(Role) ÷ Π_{playid}(Play)) ⋈ actor)

2. Find the pairs of actor's name and play's title where the actor's year_born equals the play's year_written.

$\Pi_{name,title}$ (Actor $\bowtie_{actor.year\_born=play.year\_written}$ play)

3. Find the actors' name who have participated in more than 2 plays (including 2).

Π_{name} ($\Pi_{actorid}$ (Role1 r1 $\bowtie_{r1.playid<>r2.playid\ and\ r1.actorid=r2.actorid}$ Role2 r2) ⋈ Actor)

➢ Write **SQL expressions** for the following requests:

1. Write a SQL query that returns the number of actors who have performed in three or more different plays written by the author "Great Writer"

这答案似乎有误欸

```
SELECT count(r.actor_id)
FROM Role r, Play p
WHERE p.author = 'Great Writer' AND p.play_id = r.play_id GROUP BY r.actor_id
HAVING count(DISTINCT r.play_id) > 2
```

2. Write a SQL query that returns the names of all actors who have performed some play by the author "Great Writer" and have never performed in any play written by author "Nobody". The list

should not contain duplicates but does not need to be ordered.

SELECT DISTINCT a.name FROM Actor a, Play p, Role r WHERE p.author = 'Great Writer' AND p.play_id = r.play_id AND r.actor_id = a.actor_id AND a.actor_id NOT IN (SELECT r2.actor_id FROM Role r2, Play p2 WHERE p2.author = 'Nobody' AND p2.play_id = r2.play_id)

3. Write a SQL query that returns a list of the first play of each actor. (actor_id, play_id)

这答案也怪怪的，好吧是题目怪怪的

解法 1. Select palyid, actorid from play p, role r where p.playid=r.playid and yearwritten =(select min(yearwritten) from play p1 where p1.playid=p.playid)

解法 2. Select palyid, actorid from play p, role r where p.playid=r.playid and not exists (select * from play p2 where p2.playID=p.playID and p2.到岗时间<p.到岗时间）

4. Write a SQL query that returns the play that most actors have performed.

Select playid from role group by playid having count(actor_id) > all (select count(actor_id) from role group by playid)

得加distinct吧

5. Write a SQL statement to list all the actors' name and the plays' title they were in. (note: some actors may have no records in Role table).

Select name, title from play p inner join (actor a left outer join role r on a.actor_id= r.actor_id) on p.play_id=r.play_id

6. Write a SQL statement to delete all the actors' records from Actor table who have not attend any play.

Delete from actor where actor_id not in (select actor_id from role)

7. Write a SQL statement to grant the select privilege of table Play to user Hank.

Grant select on play to Hank

Problem 5: Transaction (10 points)

There are three transactions:

$R_1(X)$, $R_1(Y)$, $W_1(X)$, $R_1(Z)$, $W_1(X)$

$R_2(Y)$, $W_2(Y)$, $R_2(X)$, $W_2(Z)$

$R_3(Z)$, $R_3(X)$, $R_3(Y)$

Here is a schedule as follows:

$S_1(X)$, $R_1(X)$, $S_2(Y)$, $R_2(Y)$, $X_2(Y)$, $W_2(Y)$, $R_2(X)$, $X_2(Z)$, $U_2(X)$, ...

For $S_1(X)$: S means shared lock; 1 stands for transaction 1; X is the object that is locked.
Also: R is read operation; W is write operation; X is exclusive lock; U is unlock.

1. What is wrong with the above example schedule. Explain why.

2. Write a 2PL schedule for the above transactions.

Problem 6: Recovery (10 points)

Consider the following log:

⟨START T⟩ ⟨T,A,10,11⟩ ⟨START U⟩ ⟨U,B,20,21⟩

⟨T,C,30,31⟩ ⟨U,D,40,41⟩ ⟨COMMIT U⟩

⟨T,A,10,11⟩: T is the transaction, A is the modified object, 10 is the old or new value, 11 is the new value.

After a crash:

1. What values might/must have been changed?

2. How does the recovery manager get the database back to a consistent state?

Discuss for Undo-, Redo-logging.

1.

Undo:

• A might have had its value changed on disk.

• B must have had its value changed on disk.

• C might have had its value changed on disk.

• D must have had its value changed on disk.

Redo:

• A cannot have had its value changed on disk.

• B might have had its value changed on disk.

• C cannot have had its value changed on disk.

• D might have had its value changed on disk.

2.

Undo:

Starting from the end of the log:

• Ignore changes of transaction U altogether.

- **Write value 30 for C.**
- **Write value 10 for A.**

Redo:

Starting from the beginning of the log:

- **Ignore changes of transaction T altogether.**
- **Write value 21 for B.**
- **Write value 41 for D.**

Problem 7: Normal Form (10 points)

Consider the relation with schema R(A,B,C,D,E,F) and the following functional dependencies (FDs): A→BC, D→AF

1. What are the candidate keys of this relation?

The candidate key is DE

2. Is relation R in BCNF? If it is, explain why it is. If it is not, explain why not and give a decomposition of R into a collection of relations that are in BCNF.

No. R is in BCNF if, for every non-trivial FD X→Y, we have that X+ = all keys. Or, in other words FD X→Y violates BCNF if X ≠X+ ≠all attributes of R.

In this case, A→BC violates BCNF since A+ = ABC ≠ABCDEF. So we split R into R1(ABC) and R2(ADEF).

The only non-trivial FD in R1 is A→BC, and A+ = ABC, so R1 is in BCNF.

R2 has a non-trivial dependency D→AF that violates BCNF because D+ = ADF ≠ADEF. So we split R2 into R21(DAF) and R22(DE). Both of these are in BCNF since they have no non-trivial dependencies that are not superkeys.

3. In what normal forms is relation R?

in 2NF because all candidate keys have only one attribute

It is not in 3NF, because in D → E neither is D a superkey, nor is E part of any candidate key.

Problem 8: Database modelling (10 points)

Design a database for a bank that implements the following requirements. The database you design should store information about customers, accounts, branches and employees.

• **Customer**: Customers are identified by their ID（身份证）. For each customer we store a name, multiple phone numbers (one or more), and an occupation.

• **Account**: Accounts are identified by an account number and the branch they belong to. For each account we store a balance and the type of account (e.g., individual account, corporate account, stock account, …).

– An account belongs to one or more customers. A customer can have any number of accounts.

– An account belongs to exactly one branch. Obviously, branches can have multiple accounts (branches are not required to have accounts).

• **Branch**: A branch is identified by a unique branch code. For each branch we want to store a location and number of employees.

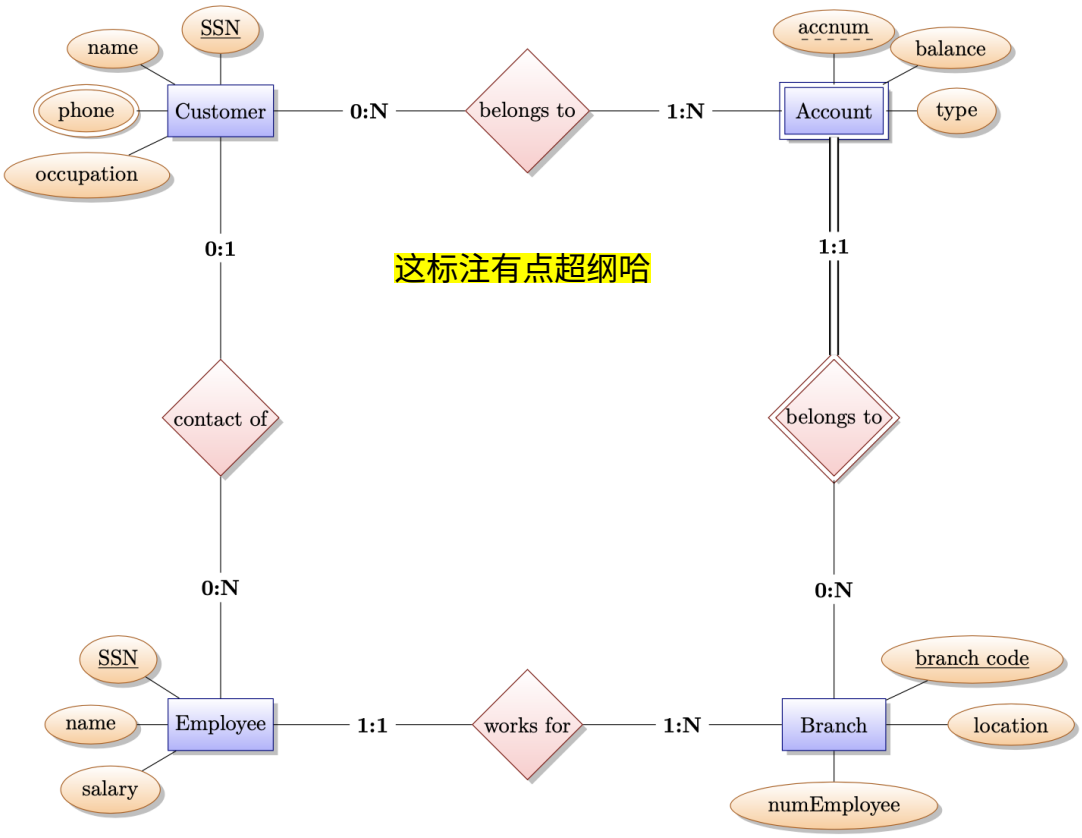
• **Employee**: Employees are identified by their ID. For each employee we store a name and salary.

– An employee works for exactly one branch. Branches have one or more employees.

– An employee is the contact for zero or more customers. Every customer has at most one employee as a contact.

Questions:

1. Draw an ER diagram for this application. Be sure to mark the multiplicity of each relationship of the diagram. Decide the key attributes and identify them on the diagram. Please state all assumptions you make in your answers.



2. Translate your ER diagram into a relational schema. Specify the key of each relation in your schema.