

Database Systems (資料庫系統)

Name: _____

Midterm exam, November 12, 2008

Time: 9:20 AM ~ 12:20 AM

Student ID: _____

I herewith state that I understand and will adhere to the following academic integrity: "I will not use or attempt to use any unauthorized assistance, material, or study aids in this midterm examination. If I violate academic integrity, I may receive a grade of 0."

Signature

(2 points)

PARTS	PART Total	Score
Name, ID, Signature	2	
A.	24	
B	20	
C	27	
D	16	
E	16	
Total	105	

PART A: ER Diagram (24 Points)

(1) (12 points) In the space below, draw an E/R diagram that best represents the following scenario. Indicate keys, many-one relationships, weak entity sets, and other features of E/R diagrams covered in class, as appropriate.

- There are hospitals, with a name and address. Address is unique.
- There are rooms in hospitals, with a room number.
- There are people, with a name and social-security number.
- Some people are doctors, and doctors have a specialty; some people are patients, and patients have an age.
- An operation involves a patient and a doctor, and takes place in a hospital room. An operation has a date and time.

For full credit, do not make operations an entity set; rather, make it a relationship.

- (2) (12 points) Translate the above ER diagram into relational tables. Use SQL **create_table** command to define all the relational tables.

PART B: Relational Algebra (20 Points)

Consider the following four relations to store data about animals in a zoo:

Animals(idNumber, type, cageNumber)

TypeKeepers(name, type)

CageKeepers(name, cageNumber)

idNumber is a key for Animals, and cageNumber is a key for Cages. TypeKeepers lists people responsible for caring for a particular type of animal, while CageKeepers lists people responsible for caring for a certain cage. For each of these two, both attributes form the key; i.e. it is possible for the same person to care for multiple types/cages, and it is possible for the same type/cage to be cared for by several people.

(1) (6 points) Consider the following sequence of relational algebra statement:

$$\text{Result} = (\text{Animals} \bowtie \text{CageKeepers}) \cap (\text{Animals} \bowtie \text{TypeKeepers})$$

Given the following sets of data for each of the relations, show (in the space on the next page) the relation that will be stored into Result table (**show the schema and all the tuples that will be in the relation**).

Animals

idNumber	type	cageNumber
1	Zebra	10
2	Monkey	20
3	Monkey	10
4	Kangaroo	30
5	Zebra	40
6	Monkey	40

CageKeepers

name	cageNumber
Ann	10
Chris	20
Chris	30
Bob	40
Bob	10

TypeKeepers

name	Type
Ann	Zebra
Bob	Monkey
Ann	Kangaroo
Chris	Kangaroo

- (2) (6 points) State briefly in English what the query from (1) is asking.
- (3) (8 points) Write a relational algebra expression to find all pairs of animals of two different types in the same cage. For each pair of animals satisfying this requirement, your expression should return each animal's idNumber and type, as well as the cageNumber of the cage they are in. For example, if Animal 1 of type Zebra and Animal 2 of type Monkey are both in cage 10, then the result of your expression should include a tuple (1, Zebra, 2, Monkey, 10) (the attributes could be in a different order). The **same animal pair should not appear twice** in your result.

PART C: Query Equivalences (27 Points)

Each row of the table below shows two queries. In the blank third column of the table write “YES” if the two queries are equivalent, and “NO” if they are not equivalent. Remember that two queries are equivalent if they always return exactly the same answer on all databases. All queries refer to relations R(A,B) and/or S(A,B).

- In both relations R and S, attribute A is a key and attribute B is not a key.
- No attributes are permitted to contain NULL values.
- Do not make any other assumptions about the data.

To lessen the score for guessing, you will earn 3 points for each correct answer, **-1 points for each incorrect answer**, and 0 points for each answer left blank.

Query 1	Query 2	Equivalent
$\pi_A(R \cup S)$	$\pi_A(R) \cup \pi_A(S)$	
$\pi_A(R - S)$	$\pi_A(R) - \pi_A(S)$	
$\sigma_{R.A=S}(R)$	$\sigma_{R.A=S}(\pi_{R.A,R.B}(R \bowtie S))$	
$\pi_{R.A}(\sigma_{R.A=S.A}(R \times S))$	<i>select R.A from R,S where R.A=S.A</i>	
$\pi_{R.B}(\sigma_{R.A=S.A}(R \times S))$	<i>select R.B from R,S where R.A=S.A</i>	
$\pi_A(R) - \pi_{R1.A}(\sigma_{R1.A > R2.A}(\rho_{R1}(R) \times \rho_{R2}(R)))$	<i>select min(A) from R</i>	
$\pi_A(R) - \pi_{R1.A}(\sigma_{R1.A < R2.A}(\rho_{R1}(R) \times \rho_{R2}(R)))$	<i>select max(A) from R</i>	
<i>select B from R group by B</i>	<i>select distinct B from R</i>	
$R/\pi_A(S)$	$\pi_A(R) - \pi_A((\pi_A(R) \times \pi_A(S)) - R)$	

PART D: SQL (16 Points)

The relations below represent information about the 15 members of the UN Security Council and the resolutions (res) that they voted on.

Members(country-name, status)
Votes(res, country-name, vote)

The status of a country is either 'permanent' or 'elected'. A vote can be either 'yes', 'no', or 'abstain'.

- (1) (8 points) In order for a resolution to pass, there must be at least 9 'yes' votes from among the 15 members of the council. In addition, there must be no veto; that is, none of the five permanent members of the council vote 'no' on that resolution. We wish to write a query to find the set of resolutions that passed.

- (2) (8 points) write a query to list each permanent member of the council, along with the number of vetos ('no' votes) they have cast. We must not forget to consider the possibility that a permanent member cast no vetos.

PART E: More SQL (16 Points)

- (1) (8 points) Consider a SQL table T(A int). Assume there are no NULL values. Make no assumptions about keys. We execute the following SQL query:

```
select sum(B)
from T, (select distinct A, 1 as B from T) as T2
where T.A = T2.A and T2.A = 3
```

In one sentence, state in English what this query computes. The correct answer is simple and brief.

- (2) (8 points) Consider a SQL table T(A int, B int, C int). Assume there are no NULL values. Suppose the following SQL query always returns an empty result.

```
select *
from ( select T1.A, T1.B, T2.C
      from T as T1, T as T2
      where T1.A = T2.A) as T3
where (T3.A,T3.B,T3.C) not in (select * from T)
```

Assuming we know nothing else about T, what does this behavior tell us about the table T? The correct answer is simple and brief.