

Database Management Systems (COP 5725)

Spring 2017

Instructor:
Dr. Markus Schneider

TAs: Keke Zhai, Man Mohan Devineni

Exam 2 Solutions

Name:	
UFID:	
Email Address:	

Pledge (Must be signed according to UF Honor Code)

On my honor, I have neither given nor received unauthorized aid in doing this assignment.

Signature

For scoring use only:

	Maximum	Received
Question 1	25	
Question 2	39	
Question 3	18	
Question 4	18	
Total	100	

Question 1 (Relational Algebra) [25 points]

Consider the Sailors-Boats-Reserves DB described below (primary keys are underlined):

Sailor(sid, sname, rating, age)

Boat(bid, bname, color)

Reserve(sid, bid, date)

Write the following queries in Relational Algebra.

1. Find the ids of all sailors who have a rating of at least 8 or reserved a red boat at date "2017-03-03". [5 points]

$$\pi_{sid}(\sigma_{rating \geq 8}(\text{Sailor})) \cup \pi_{sid}(\sigma_{date = '2017-03-03' \wedge color = 'red'}(\text{Boat} \bowtie \text{Reserve}))$$

2. Find the names of all sailors with an age over 20 who have not reserved a red boat. [5 points]

$$\pi_{name}([\pi_{sid}(\sigma_{age > 20}(\text{Sailor})) - \pi_{sid}(\sigma_{color = 'red'}(\text{Boat} \bowtie \text{Reserve}))] \bowtie \text{Sailor})$$

3. Find the ids of all sailors whose rating is lower than the ratings of all sailors called Jack. [5 points]

$$\pi_{sid}(\text{Sailor}) - \pi_{S2.sid}(\sigma_{S2.rating \geq S.rating} [\rho_{S2}(\text{Sailor}) \times \sigma_{sname = 'Jack'}(\text{Sailor})])$$

4. Find the ids of the sailors with the oldest age. [5 points]

$$\pi_{sid}(\text{Sailor}) - \pi_{S2.sid}(\sigma_{S2.age < S.age} [\rho_{S2}(\text{Sailor}) \times \text{Sailor}])$$

5. Find the names of sailors who have reserved all boats that used to be reserved by Bob. [5 points]

$$\pi_{name}([\pi_{sid, bid}(\text{Reserve}) \div \pi_{bid}(\sigma_{name = 'Bob'}(\text{Reserve} \bowtie \text{Sailor}))] \bowtie \text{Sailor})$$

Question 2 (SQL) [39 points]

A) Consider the following table schemas (primary keys are underlined):

BOOK (BookId, Title, Publishername)

BOOK_AUTHORS (BookId, AuthorName)

PUBLISHER (Name, Address, Phone)

BOOK_COPIES (BookId, BranchId, No_Of_Copies)

BOOK_LOANS (BookId, BranchId, CardNo, DateOut, DueDate)

LIBRARY_BRANCH (BranchId, BranchName, Address)

BORROWER (CardNo, Name, Address, Phone)

Write SQL statements for the following queries:

1. List the number of copies of the book titled “The Lost Tribe” owned by each library branch and the corresponding library branch name. [4 points]

```
select BranchName, No_Of_Copies
from Book b, Book_Copies bc , Library_Branch lb
where b.BookId=bc.BookId and bc.BranchId=lb.BranchId and
      b.Title="The Lost Tribe";
```

2. List the names and addresses of all borrowers who have borrowed more than 5 books that are published by “ABC”. [5 points]

```
select br.name, br.address
from Borrower br, Book_Loans bl, Book b
where br.cardno=bl.cardno and bl.bookid=b.bookid and b.publishername="ABC"
group by br.cardno, br.name, br.address
having count(*)>5;
```

3. List the name and the phone number of the borrower who borrowed the book authored by “Stephen King” on date “2017-03-05”. [5 points]

```
select br.name, br.phone
from borrower br, book_authors ba, book_loans bl
where br.card_no=bl.card_no and bl.book_id=ba.book_id and
      ba.author_name="Stephen King" and bl.date_out="2017-03-05";
```

4. List the library branch name where the number of copies of the book titled “The Spring” is greater than the number of copies of the book titled “The Pleasure of Coding”. [6 points]

```

select a1.branch_name
from
  (select lb.branch_id, lb.branch_name, bc.no_of_copies
   from library_branch lb, book_copies bc, book bk
   where lb.branch_id=bc.branch_id and bc.book_id=bk.book_id and
         bk.title="The Spring") a1,
  (select lb.branch_id, lb.branch_name, bc.no_of_copies
   from library_branch lb, book_copies bc, book bk
   where lb.branch_id=bc.branch_id and bc.book_id=bk.book_id and
         bk.title="The Pleasure of Codeing") a2
where a1.branch_id=a2.branch_id and a1.no_of_copies>a2.no_of_copies;

```

B) Consider the following table schemas (primary keys are underlined):

Scientists(SSN, name)

AssignedTo(SSN, pcode)

Projects(pcode, name, hours)

Write SQL statements for the following queries:

5. List all the scientists' names, their projects' names, and the hours that scientist will spend on each project, in ascending order of project name, then scientist name. [4 points]

```

select s.name, p.name, p.hours
from scientists s, assignedto a, projects p
where s.ssn=a.ssn and a.pcode=p.pcode
order by p.name asc, s.name asc;

```

6. List the name of scientists, the sum of hours he/she will spend on all the projects he/she participates, with the total time greater than the average working hours of all scientists. (5 points)

```

select s.name, sum(p.hours)
from scientists s, assignedto a, projects p
where s.ssn=a.ssn and a.pcode=p.pcode
group by s.ssn, s.name
having sum(p.hours)>
(select avg(p.hours)
 from scientists s, assignedto a, projects p
 where s.ssn=a.ssn and a.pcode=p.pcode);

```

7. Find the project name that has the most number of scientists participated in and the corresponding number. [5 points]

```

select p.name, count(*)
from assignedto a, projects p
where a.pcode=p.pcode
group by p.pcode, p.name
having count(*) >= all
(select count(*)
 from assignedto a, projects p
 where a.pcode=p.pcode
 group by p.pcode, p.name);

```

8. Find the scientists that have participated in every project. [5 points]

```

select s.name from scientists s
where not exists
( (select projects.pcode from projects)
  minus
  (select a.pcode from assignedto a
   where a.ssn=s.ssn));

```

Question 3 (Relational Algebra and SQL) [18 points]

1. Let \mathcal{R} be a relation schema and R be a relation with respect to \mathcal{R} . Let \mathcal{S} be a relation schema and S be a relation with respect to \mathcal{S} . We assume that $\mathcal{R} \cap \mathcal{S} \neq \emptyset$. Show formally (without examples) that the outer-join operator $R \bowtie S$ is a derived Relational Algebra operator. That is, it can be expressed by the 5 (+1) basic Relational Algebra operators. You are allowed to make use of derived Relational Algebra operators that have already been shown to be derived in class. [6 points]

Let the relation T have the schema $\mathcal{S} - \mathcal{R}$. Let $T = \{\langle null, null, \dots, null \rangle\}$, that is, T is a relation with a single tuple and $|\mathcal{S} - \mathcal{R}|$ attributes, and all attribute values of this single tuple are *null*. Then we can define:

$$R \bowtie S = R \bowtie S \cup ((R - \pi_{\mathcal{R}}(R \bowtie S)) \times T)$$

2. Let $\mathcal{R} = \{A_1, A_2, \dots, A_n, C_1, \dots, C_k\}$ and $\mathcal{S} = \{B_1, B_2, \dots, B_m, C_1, \dots, C_k\}$ be the table schemas that correspond to the relation schemas in part 1. The attributes C_1, \dots, C_k are shared by both schemas. The A_i 's and B_j 's are supposed to be distinct from each other. Let R and S be tables with respect to the schemas \mathcal{R} and \mathcal{S} respectively. Provide a translation of your definition in part 1 into a corresponding SQL expression. The translation should be as near as possible to the Relational Algebra expression in part 1. [6 points]

```
CREATE TABLE T (B1 DT1, B2 DT2, ..., Bm DTm);
-- "DTj" means "data type j"

INSERT INTO T (B1, B2, ..., Bm) VALUES (null, null, ..., null);

(SELECT * FROM R NATURAL JOIN S)
UNION
(
  SELECT * FROM
    (
      (SELECT * FROM R)
      MINUS
      (SELECT A1, ..., An FROM R NATURAL JOIN S)
    )
  CROSS JOIN T
);
```

3. Consider a schema with two relations $R(A, B)$ and $S(B, C)$ where all values are integers. Make no assumptions about keys. Consider the following three relational algebra expressions:

A. $\pi_{A,C}(R \bowtie_{\sigma_{B=1}}(S))$

B. $\pi_A(\sigma_{B=1}(R)) \times \pi_C(\sigma_{B=1}(S))$

C. $\pi_{A,C}(\pi_A(R) \times \sigma_{B=1}(S))$

Two of the three expressions are equivalent (i.e., produce the same answer on all databases), while one of them can produce a different answer. Which query can produce a different answer? Give the simplest database instance you can think of where a different answer is produced. [6 points]

Query (C) is different. Let $R = \{(2, 4)\}$ and $S = \{(1, 3)\}$. Then query (a) and (b) produce an empty result while (c) produces $\{(2,3)\}$.

Question 4 (QBE) [18 points]

Consider the following database schema (primary key attributes are underlined):

Product (pid, name, price, category, maker-cid)

Purchase (buyer-ssn, seller-ssn, pid)

Company (cid, name, stock price, country)

Person(ssn, name, phone number, city)

Answer the following questions using QBE. Draw tables in your answers.

1. Find all people (print all information) who bought stuff from Joe or bought products from a company whose stock prices is more than \$50. [6 points]

Person	ssn	name	Phone number	city
P.	<u>_Id</u>			
P.	<u>_Id2</u>			
	<u>_Id3</u>	Joe		

Purchase	Buyer_ssn	Seller-ssn	pid
	<u>_Id1</u>	<u>_Id3</u>	
	<u>_Id2</u>		<u>_pid1</u>

Company	cid	name	Stock price	country
	<u>_cid1</u>		>50	

Product	pid	name	price	category	Maker_cid
	<u>_pid1</u>				<u>_cid1</u>

2. Find the names of the customers who have spent more than \$1000 on products made by company “MFC”. [6 points]

Person	ssn	name	Phone number	city
	_Id1	P._name		

Product	pid	name	price	category	Maker_cid
	_pid1		SUM.ALL._X1		_cid1

Company	cid	name	Stock price	country
	_cid1	MCF		

Purchase	Buyer_ssn	Seller_ssn	pid
	G._Id1		_pid1

Conditions
SUM.ALL._X1>1000

3. Since the stock price from company “Walmart” has increased, the product price from that company has to increase 10%. Update all the product price from that company by 10% of previous price. [6 points]

Company	cid	name	Stock price	country
	_cid1	Walmart		

Product	pid	name	price	category	Maker_cid
	_pid1		_p1		_cid1
U.	_pid1		_p1*1.1		_cid1