

The University of British Columbia
Computer Science 304

Midterm Examination
March 7, 2007

Time: 50 minutes
Instructor: Rachel Pottinger

Total marks: 35

Name **ANSWER KEY** _____ Student No _____
(PRINT) (Last) (First)

Signature _____

This examination has 3 doublesided pages.

Check that you have a complete paper.

This is a closed book, closed notes exam. No books or other material may be used.

Answer all the questions on this paper.

Give very **short but precise** answers.

State any assumptions you make

Work fast and do the easy questions first. Leave some time to review your exam at the end.

Good Luck

Question	Mark	Out of
1.a		5
1.b		5
2.a		5
2.b		5
2.c		5
2.d		5
2.e		5
TOTAL		Out of 35

1. {10 marks}

This question is isomorphic to parts a and b from Question 1 from practice midterm 3

The marks relation was defined based on the following SQL statement:

```
CREATE TABLE marks
  (studentID CHAR(20) NOT NULL,
   courseID CHAR(20) NOT NULL,
   courseType CHAR(20),
   score REAL,
   PRIMARY KEY (studentID, courseID)
)
```

For each of the following relational calculus queries, determine if there exists an equivalent relational algebra statement. If your answer is positive, give such a statement; otherwise, just state that no such statement exists. (You may get part marks if you write down in English the correct meaning of each query.)

- a) $\{ \langle N \rangle \mid \exists x, y (\langle N, x, \text{'CPSC'}, z \rangle \in \text{marks} \text{ and } \langle N, y, \text{'CPSC'}, r \rangle \in \text{marks} \text{ and } x \neq y) \}$

(5 points) *Students who received marks from at least 2 distinct CPSC courses.*

$\pi_{A.studentID} (\sigma_{A.studentID = B.studentID \wedge A.courseType = \text{'CPSC'} \wedge B.courseType = \text{'CPSC'} \wedge A.courseID <> B.courseID}$

$[\rho A (\text{marks}) \times \rho B (\text{marks})]$

- b) $\{ \langle N \rangle \mid \forall x (\langle r, x, \text{'CPSC'}, s \rangle \in \text{marks} \Rightarrow \langle N, x, \text{'CPSC'}, t \rangle \in \text{marks}) \}$

(5 points) *Recipients who received marks from every 'CPSC' type of course.*

$\pi_{studentID, courseID}(\text{marks}) /$

$\pi_{courseID} (\sigma_{courseType = \text{'CPSC'}} (\text{marks}))$

2. {25 marks} The following relations keep track of airline flight information:
 Flights: (fno: integer, origin: string, destination: string, distance, integer, departs: timestamp, arrives: timestamp, price: integer)
 Aircraft(aid: integer, aname: string, cruisingrange: integer)
 Certified(eid: integer, aid: integer)
 Employees(eid: integer, ename: string, salary: integer)

Note that the employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft, and only pilots are certified to fly. Write each of the following queries in SQL:

- a. {5 marks} List in reverse alphabetical order all pilots who are certified to fly some Airbus plane

*Select distinct ename
 From employees e, aircraft a, certified c
 Where c.eid = e.eid AND c.aid = a.aid and a.aname like '%Airbus%'
 Order by ename desc*

- b. {5 marks} Find the name(s) and salary(salaries) of the pilot(s) who is(are) certified to fly the largest number of planes

*create view multi(pilot,numplanes) as
 select eid as pilot, count(*) as numplanes
 from certified
 group by eid;*

*select ename, salary
 from employees, multi m
 where eid = m.pilot and numplanes = (select max(numplanes) from multi)*

- c. {5 marks} Compute the difference between the average salary of a pilot and the average salary of all employees (including pilots).

*This is question 5.3.10 from the book
 Create view pilotaverage(avgsal) as
 Select avg(e.salary) as avgsal
 From employees e
 Where e.eid in (select distinct c.eid from certified c)*

*Create view averagesal(avgsal) as
 Select avg (salary) as avgsal
 From employees e*

*Select p.avg – a.avg
 From pilotaverage p, averagesal a*

Note: need to be careful on selection of pilot salary to make sure that each pilot's salary is only counted once. While the "distinct" on c.eid is unnecessary, just performing the join won't work, nor will just taking the distinct value of the salaries.

- d. {5 marks} For each plane that has at least six pilots, find the name of the plane and the average salary of the pilots who are certified to fly it

Create view planesalaries(average, aid) as

Select average(salary), aid

From certified c, employees e

Where c.aid = e.aid

Group by c.aid

Having count(c.aid) > 5

Select average, a.aname

From aircraft a, planesalaries p

Where p.aid = a.aid

Theoretically, you shouldn't group by aname, since it's not a key. However, we decided to let that one go by. Can't include aname in select initially – not an aggregation or a grouping attribute.

- e. {5 marks} Find the set of origins and destinations that can be reached by two hops that can't be reached by a direct flight (e.g., list "Pittsburgh" "Honolulu" if there's no direct flight between Pittsburgh and Honolulu *and* there's a flight from "Pittsburgh" to some city "X" and then a flight from "X" to "Honolulu")

Select a.origin, b.destination

From Flights a, flights b

Where a.destination = b.origin

Except (or minus)

Select origin, destination

From flights