

## Solution Final CSC 380 S1 2018-2019

### Question 1: 10pts



### Question 2: 10 pts

A

<u>A1</u>	B1	C1	R1
-----------	----	----	----

C

<u>C1</u>	A1
-----------	----

B

<u>B1</u>	<u>C1</u>	U1
-----------	-----------	----

S

<u>A1</u>	<u>B1</u>	<u>C1</u>
-----------	-----------	-----------

### **Question 3: 8 pts**

1. List all the course names that professor 'Smith' taught in Fall of 2007.

$\pi_{\text{crsname}}(\sigma_{\text{profname}='Smith'}(\text{Professor}) \sigma_{\text{semester}='f2007'}(\text{Taught}) \text{Course})$

2. Return those professors who have taught 'csc6710' but never 'csc7710'.

$\pi_{\text{ssn}}(\sigma_{\text{crscode}='csc6710'}(\text{Taught})) - \pi_{\text{ssn}}(\sigma_{\text{crscode}='csc7710'}(\text{Taught}))$

3. Return those professors who have taught both 'csc6710' and 'csc7710'.

$\pi_{\text{ssn}}(\sigma_{\text{crscode}='csc6710'}(\text{Taught})) \cap \pi_{\text{ssn}}(\sigma_{\text{crscode}='csc7710'}(\text{Taught}))$

4. Return those courses that have been taught by all professors.

$\pi_{\text{crscode}, \text{ssn}}(\text{Taught}) / \pi_{\text{ssn}}(\text{Professor})$

### **Question 4: 8 pts**

Given relational schema:

Sailors (sid, sname, rating, age)

Reservation (sid, bid, date)

Boats (bid, bname, color)

1. For each rating find the age of the youngest sailor with age  $\geq 18$

SELECT S.rating, MIN (S.age)

FROM Sailors S

WHERE S.age  $\geq 18$

GROUP BY S.rating

2. Find sid's of sailors who've reserved both a red and a green boat:

SELECT R.sid

FROM Boats B, Reserves R

WHERE R.bid=B.bid

AND B.color='red'

AND R.sid IN (SELECT R2.sid

FROM Boats B2, Reserves R2

WHERE R2.bid=B2.bid

AND B2.color='green')

3. Find the number of reservations for each red boat.

SELECT B.bid, COUNT(\*) AS scout

FROM Boats B, Reserves R

WHERE R.bid=B.bid

AND B.color='red'  
GROUP BY B.bid

4. Find sailors whose rating is greater than every sailor called "Ali". Use ALL.

```
SELECT sid
FROM Sailors
WHERE rating > ALL (SELECT rating
                    FROM Sailors
                    WHERE sname='Ali')
```

### **Question 5: 4 pts**

#### **First Normal Form**

- Assume the **key** is Name, Project, Task.
- Is EMPLOYEE in 1NF ?

No composite or Multivalued attributes.

#### **Second Normal Form**

- List all of the functional dependencies for EMPLOYEE.
- Are all of the non-key attributes dependant on *all* of the key ?
- Split into two relations EMPLOYEE\_PROJECT\_TASK and EMPLOYEE\_OFFICE\_PHONE. EMPLOYEE\_PROJECT\_TASK (Name, Project, Task)
- EMPLOYEE\_OFFICE\_PHONE (Name, Office, Floor, Phone)

#### **Third Normal Form**

- Assume each office has exactly one phone number.
- Are there any transitive dependencies ?
- Where are the modification anomalies in EMPLOYEE\_OFFICE\_PHONE ?
- Split EMPLOYEE\_OFFICE\_PHONE.

EMPLOYEE\_PROJECT\_TASK (Name, Project, Task)

EMPLOYEE\_OFFICE (Name, Office, Floor)

EMPLOYEE\_PHONE (Office, Phone)