Course ID: COMP 3005A
Course Name: DBMS
Assignment #: 1

Due: 05-Oct-2023 at 11:59 PM
Instructor: Abdelghny Orogat

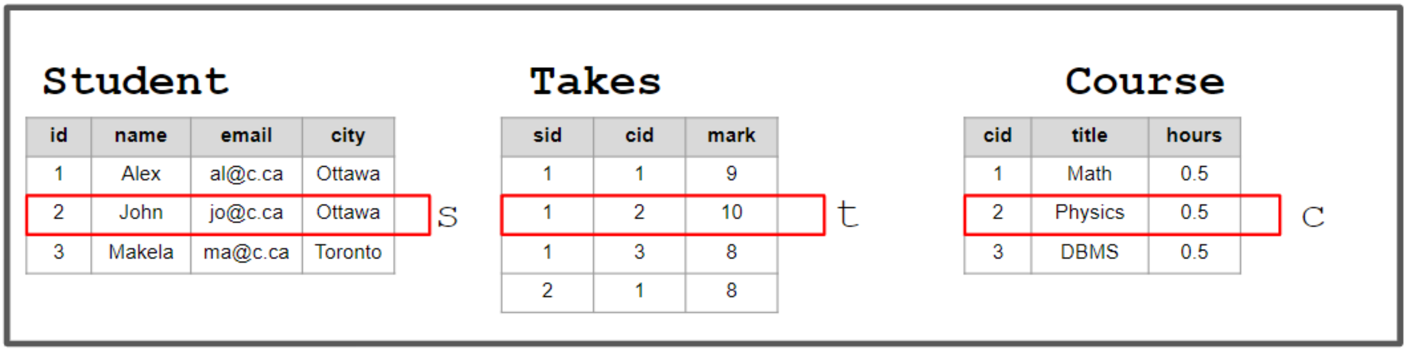
Name: Vidun Jayakody
ID: 101224988

Instructions:

• The accepted format for your submission is pdf only and .txt for SQL queries.

Question (1):

Consider the following relational database. Show the output result of the following queries



Student

id	name	email	city
1	Alex	al@c.ca	Ottawa
2	John	jo@c.ca	Ottawa
3	Makela	ma@c.ca	Toronto

Takes

sid	cid	mark
1	1	9
1	2	10
1	3	8
2	1	8

Course

cid	title	hours
1	Math	0.5
2	Physics	0.5
3	DBMS	0.5

(a) π_{email} ($\sigma_{name='John'}$ (Student)) [1 mark]

email
jo@c.ca

(b) Student $\bowtie_{Student.id=Takes.sid}$ Takes [1 mark]

id	name	email	city	sid	cid	Mark
1	Alex	al@c.ca	Ottawa	1	1	9
1	Alex	al@c.ca	Ottawa	1	2	10
1	Alex	al@c.ca	Ottawa	1	3	8
2	John	jo@c.ca	Ottawa	2	1	8

(c) Student $\bowtie_{\texttt{Student.id=Takes.sid}}$ Takes **[1 mark]**

id	name	email	city	sid	cid	Mark
1	Alex	al@c.ca	Ottawa	1	1	9
1	Alex	al@c.ca	Ottawa	1	2	10
1	Alex	al@c.ca	Ottawa	1	3	8
2	John	jo@c.ca	Ottawa	2	1	8
3	Makela	ma@c.ca	Toronto			

(d) ($\pi_{sid,cid}$ Takes) / (π_{cid} Course) **[1 mark]**

sid
1

Question (2):

Consider the following relational database instance where

- For the “Employee” relation,
  - the “EManager” column refers to the EID of the manager of the current employee. For example, the employee (22222, Einstein) is the manager of (12121, Wu), (32343, El Said), (45565, Katz), and (98345, Kim), and so on.
  - the “EDepart” column refers to a department in the “Department” relation. For example, the department’s id of (12121, Wu) is 2. This id=2 refers to the “Accounting” department in the “Department” relation.

.

Employee

EID	EName	ESalary	EManager	EDepart
22222	Einstein	95000	Null	1
12121	Wu	90000	22222	2
32343	El Said	60000	22222	3
45565	Katz	75000	22222	4
98345	Kim	80000	22222	5
76766	Crick	72000	98345	5
10101	Srinivasan	65000	98345	5
58583	Califieri	62000	98345	5
83821	Brandt	92000	98345	5
15151	Mozart	40000	98345	5
33456	Gold	87000	12121	2
76543	Singh	80000	12121	2

Department

DID	DName	DLocation	DBudget
1	CEO	Toronto	10,000
2	Accounting	Ottawa	100,000
3	Media	Ottawa	30,000
4	Research	Ottawa	20,000
5	Production	Ottawa	500,000

Write relational algebra and tuple-relational-calculus expressions to solve the following questions, and show the resulting table for each example. The first example is solved to show you how to answer each question

(a) Return the employees whose salaries are less than \$80,000. **[0 marks]**

Answer

Relational Algebra Expression:

$\sigma_{ESalary<80000}$ (Employee)

Tuple Relational Calculus Expression:

$\{e.* \mid e \in Employee \text{ AND } e.ESalary<80000\}$

Returned Result:

Employee

EID	EName	ESalary	EManager	EDepart
32343	El Said	60000	22222	3
45565	Katz	75000	22222	4
76766	Crick	72000	98345	5
10101	Srinivasan	65000	98345	5
58583	Califieri	62000	98345	5
15151	Mozart	40000	98345	5

(b) Return employees who are working in department number 5. **[3 marks]**

Relational Algebra Expression:

$\sigma_{EDepart=5}(Employee)$

Tuple Relational Calculus Expression:

$\{t.* \mid t \in Employee \wedge t.EDepart = 5\}$

EID	EName	ESalary	EManager	EDepart
98345	Kim	80000	22222	5
76766	Crick	72000	98345	5
10101	Srinivasan	65000	98345	5
58583	Califieri	62000	98345	5
83821	Brandt	92000	98345	5
15151	Mozart	40000	98345	5

(c) What is the salary of the employee "Kim"? **[3 marks]**

Relational Algebra Expression:

$\pi_{ESalary}(\sigma_{EName="Kim"}(Employee))$

Tuple Relational Calculus Expression:

$\{t.ESalary \mid t \in Employee \wedge t.EName = "Kim"\}$

ESalary
80000

(d) List pairs of employees' names and their departments' names. **[3 marks]**

Relational Algebra Expression:

$\boldsymbol{\pi_{EName,DName}(Employee \bowtie_{EDepart=DID} Department)}$

Tuple Relational Calculus Expression:

$\{e.EName, d.DName \mid e \in Employee \wedge d \in Department \wedge e.EDepart = d.DID\}$

EName	DName
Einstein	CEO
Wu	Accounting
El Said	Media
Katz	Research
Kim	Production
Crick	Production
Srinivasan	Production
Califieri	Production
Brandt	Production
Mozart	Production
Gold	Production
Singh	Production

(e) List employees' names only for those who work in the Accounting department. **[3 marks]**

Relational Algebra Expression:

$$\pi_{EName}\left(\sigma_{DName="Accounting"}\left(Employee \bowtie_{Employee.EDepart=Department.DID} Department\right)\right)$$

Tuple Relational Calculus Expression:

$$\{e.EName \mid e \in Employee \wedge \exists d(d \in Department \wedge e.EDepart = d.DID \wedge d.DNAme = "Accounting")\}$$

EName
Gold
Singh
Wu

(f) In relational algebra, joining a relation with itself can lead to ambiguity about which table we're referencing. To address this, we can rename one of the relations before executing the join operation. If you have a relation `R` with two attributes `a` and `b`, which can be written as `R(a,b)`, you can rename it and its columns using the expression $\rho_{S(a->c,\ b->d)}(R)$ to generate a new relation with the same structure and the same content but with different name `S(c,d)`.

Take the renaming operator into your consideration, solve the following query:

For employees only who have managers, list the employees' names with their corresponding managers' names. **[3 marks]**

Relational Algebra Expression:

$$\rho_{Manager(EID\rightarrow MID,EName\rightarrow MName,ESalary,EManager\rightarrow MManager,EDepart\rightarrow MDepart)}(Employee)$$

$$\pi_{\text{Employee.EName, Manager.MName}}\left(\text{Employee} \bowtie_{\text{Employee.EManager=Manager.MID}} \text{Manager}\right)$$

Tuple Relational Calculus Expression:

$$\{ (e."\{EName\}, m.\{MName\}) \mid e \in "\{Employee\}, m \in "\{Employee\} \wedge e."\{EManager\} = m."\{EID\} \}$$

EmployeeName	ManagerName
Wu	Einstein
El Said	Einstein
Katz	Einstein
Kim	Einstein

Crick	Kim
Srinivasan	Kim
Califieri	Kim
Brandt	Kim
Mozart	Kim
Gold	Wu
Singh	Wu

Question (3):

Compared to Relational Algebra (RA), Tuple Relational Calculus (TRC) doesn't focus on operators or the sequence of operations. When crafting a query in TRC, it's essential to establish tuple variables (either free or bound) for the relations, which then help in defining the desired output relation. However, TRC allows one variable to be defined on multiple relations, and this helps us to write queries equivalent to the set operators (e.g., ∪, ∩, and -). For example, given two relations: R and S, the TRC query `{t.* | t ∈ R ∧ t ∈ S}` is equivalent to the RA query `R ∩ S`, which returns the intersection between two relations.

(a) Write the TRC query that is equivalent to the following RA queries

(i) R ∪ S **[1 mark]**

$$\{\boldsymbol{t}.*|\ (\boldsymbol{t} \in \boldsymbol{R}) \vee (\boldsymbol{t} \in \boldsymbol{S})\}$$

(ii) R - S **[1 mark]**

$$\{\boldsymbol{t}.*|\ \boldsymbol{t} \in \boldsymbol{R} \wedge \neg(\boldsymbol{t} \in \boldsymbol{S})\}$$

(iii) S - R **[1 mark]**

$$\{\boldsymbol{t}.*|\ \boldsymbol{t} \in \boldsymbol{S} \wedge \neg(\boldsymbol{t} \in \boldsymbol{R})\}$$

Question (4):

For this question, upload the SQL queries in a (.txt) file.

Using the university database schema discussed in class, answer the following:

(a) The primary key for the advisor relation is s_id. Suppose a student can have more than one supervisor. Would s_id still be a primary key in advisor? If yes, why? If not, what would be a suitable primary key? **[1 mark]**

No, **`s_id`** can't be a primary key. We'd need a composite key of (**`s_id`**, **`i_id`**) for uniqueness. This allows each student to be linked to several instructors in the advisors table.

(b) The primary key for prereq is both attributes course_id and prereq_id. Why wouldn't only course_id work as the primary key? **[1 mark]**

Only using `course_id` wouldn't work because a course can have multiple prerequisites. To uniquely identify each prerequisite for a course, both `course_id` and `prereq_id` are needed as the primary key.

(c) Given the existing schema of teaches, two or more instructors can teach the same section. How can the primary key be changed to restrict a section to one instructor only? **[1 mark]**

To restrict a section to only one instructor in the teaches relation, we'd need to make the combination of `course_id`, `sec_id`, and semester (which identifies a unique section) the primary key. This ensures each section can be associated with just one instructor.

(d) Create a new course ("Aces of Databases") with ID ("COMP5118") in the Computer Science department ("Comp. Sci.") with 0 credit hours. **[1 mark]**

```
INSERT INTO course (course_id, title, dept_name, credits)
VALUES ('COMP5118', 'Aces of Databases', 'Comp. Sci.', Null);
```

Note: 0 is not a valid credit hours because of the `check (credits > 0)` constraint in the provided database.

(e) Create a section 'A' for this course in the Winter of 2020 with no known location or time, yet. **[1 mark]**

```
INSERT INTO section
(course_id, sec_id, semester, year, building, room_number, time_slot_id)
VALUES ('COMP5118', 'A', 'Winter', 2020, NULL, NULL, NULL);
```

(f) One student with ID 12345 cannot take this course because of violating the prerequisite requirements (didn't pass COMP3005). Unregister this student from the new section **[1 mark]**

```
DELETE FROM takes
WHERE ID = 12345 AND course_id = 'COMP5118' AND sec_id = 'A'
AND semester = 'Winter' AND year = 2020;
```