

Question 1. [7 MARKS]**Part (a)** [5 MARKS]

Indicate whether each statement is True or False by circling the appropriate answer.

TRUE ☒ FALSE All integrity constraints represent a foreign key constraint.

TRUE ☒ FALSE Theta joins are natural joins that include a select operator on a condition.

☒ TRUE FALSE The assignment operator is not usually used to update the content of an existing relation.

TRUE ☒ FALSE A natural join on two tables with no common attributes results in an empty relation.

☒ TRUE FALSE If a relational algebra query has a select operator followed by a project operator, you cannot always swap the positions of the operators to get the same resulting relation.

Part (b) [2 MARKS]

Briefly explain what is meant by ‘dangling’ tuples. Show an example with some small tables.

Must explain dangling tuples and show example - see lecture slides.

Recall this schema, which we have used many times in class.

Relations

Student(sID, surName, firstName, campus, email, cgpa)
 Course(dept, cNum, name, breadth)
 Offering(oID, dept, cNum, term, instructor)
 Took(sID, oID, grade)

Integrity constraints

Offering[dept, cNum] \subseteq Course[dept, cNum]
 Took[sID] \subseteq Student[sID]
 Took[oID] \subseteq Offering[oID]

Question 2. [4 MARKS]

Part (a) [2 MARKS]

Consider this constraint:

$Proom(cNum1, cNum2, term) :=$

$$\Pi_{O1.cNum, O2.cNum, O1.term} \sigma_{\begin{array}{l} O1.cNum < O2.cNum \\ O1.dept = O2.dept = 'CSC' \\ O1.instructor = O2.instructor \\ O1.term = O2.term \end{array}} [(\rho_{O1} Offering) \times (\rho_{O2} Offering)]$$

$$\sigma_{\begin{array}{l} P1.cNum1 = P2.cNum1 \\ P1.cNum2 = P2.cNum2 \\ P1.term \neq P2.term \end{array}} [(\rho_{P1} Proom) \times (\rho_{P2} Proom)] = \emptyset$$

Define an instance of Offering that violates the constraint.

Solution:

oID	dept	cNum	term	instructor
o1	CSC	343	termA	Sina
o2	CSC	443	termA	Sina
o3	CSC	343	termB	Diane
o4	CSC	443	termB	Diane

Part (b) [2 MARKS]

Write the following constraint using relational algebra: If a student takes a course taught by Horton, they cannot take a course taught by Gries.

Solution:

Any solution that did something like the following:

Find all students taking Horton Courses.

Find all students taking Gries Courses.

Intersection of the two above should be empty.

Question 3. [8 MARKS]

Write a query in relational algebra to find the following: Consider the student(s) who received the highest grade over all breadth courses (a course where 'breadth' is true). Of those students, find sIDs of the ones who study on the 'St. George' campus.

You should break up your query into steps using the assignment operator. Adding commentary will help you understand your answer and can help us grade your answer.

Solution: this is one possible way to solve this

– All offerings of breadth courses.

$Breadth(oID) := \Pi_{oID} \sigma_{breadth=true}(Course \bowtie Offering)$

–Students who have taken Breadth courses

$Breadthers(sID, oID, grade) := \Pi_{sID, oID, grade}(Breadth \bowtie Took)$

–Students who don't have the highest grade over all breadth courses

$NotHighest(sID, oID) := \Pi_{B1.sID, B1.oID} \sigma_{B1.grade < B2.grade}[(\rho_{B1} Breadthers) \times (\rho_{B2} Breadthers)]$

–Students with highest grade over all breadth courses

$Highest(sID) := \Pi_{sID}(\Pi_{sID, oID} Breadthers - NotHighest)$

– Answer: The students from above that go to St. George.

$Answer(sID) := \Pi_{sID} \sigma_{campus=St.George}[Highest \bowtie Student]$

Question 4. [8 MARKS]

For this question, you will write SQL queries using a version of the Restaurants schema from Assignment 1.

Relations

Restaurant(name, owner, capacity, country)
 Patron(PID, name, birthday)
 Dish(DID, name, dietary)
 Reservation(RID, PID, rname, date)
 Order(RID, DID, number)
 Rating(PID, rname, rating, comment)

Integrity constraints

Reservation[PID] \subseteq Patron[PID]
 Reservation[rname] \subseteq Restaurant[name]
 Order[RID] \subseteq Reservation[RID]
 Order[DID] \subseteq Dish[DID]
 Rating[PID] \subseteq Patron[PID]
 Rating[rname] \subseteq Restaurant[name]

Part (a) [3 MARKS]

In our schema, users can leave **comments** on their restaurant rating, and the comment text can be null. Write a query in SQL to find, for each Patron who has made a rating for a restaurant (where the comment text is not null), their name and the number of restaurants they have made a comment on. Report the Patron's name and the number of restaurants. Organize the output in non-increasing order by the number of restaurants.

Solution:

```
SELECT name, count (rname)
FROM Patron, Rating
WHERE Patron.pid = Rating.pid
AND comment IS NOT NULL
GROUP BY Patron.name
ORDER BY count(rname) DESC;
```

Part (b) [2 MARKS]

Write a query in SQL that finds the names and owners of all restaurants that have a lower capacity than the restaurant named 'Red Lobster'.

Solution:

```
SELECT name, owner
FROM Restaurant
WHERE capacity <
  (SELECT capacity
   FROM Restaurant
   WHERE name = 'Red Lobster');
```

The following query is supposed to print the number of pairs of dishes which have the same dietary restriction. It runs but does not always give the correct output.

```
SELECT count(*)

FROM (

    SELECT D1.DID, D2.DID

    FROM Dish D1, Dish D2

    WHERE D1.dietary = D2.dietary

    AND D1.DID <> D2.DID

) as DietaryPairs;
```

Part (c) [1 MARK]

Suppose that Dish has these values. What will be the output of the query?

DID	name	dietary
1	'veggie burger'	'veg'
2	'salad'	'veg'
3	'lasagna'	'normal'
4	'quinoa'	'gf'
5	'pistachio'	'gf'

Solution:

```
count
-----
4
```

Part (d) [1 MARK]

Generalizing to any dataset, explain what is wrong with the output of this query.

Solution: It counts every pair twice.

Part (e) [1 MARK]

Fix the query by making the smallest change that you can. Write your corrections directly on the query text above.

Solution: One option is:

```
add    AND D1.DID < D2.DID
(or change the <> to <)
```

Another solution is to do `count(*)/2`

Question 5. [8 MARKS]

Suppose we have the following tables from a Twitter database:

Follows:

a	b
sina	kanyewest
sina	RonConwayFacts
diane	LilaFontes
diane	swcarpentry
diane	mfeathers
diane	sina
michelle	sina
michelle	diane
michelle	Jeff

(9 rows)

Profile:

id	name	location
alan	catman	Ottawa
sina	superman	
diane	superwoman	Toronto
michelle	rockstar	Montreal

(4 rows)

Tweets:

id	userid	content
123	alan	hellow twitter
125	alan	bye twitter
126	alan	hellow twitter
128	alan	bye twitter
476	sina	hellow twitter
553	diane	hellow twitter

(6 rows)

Show the result of running each of the following queries. If a table is produced, include the column names. If the query generates an error, explain.

Solutions

```
SELECT a, count(*)
FROM Profile RIGHT JOIN Follows
ON a = id GROUP BY a;
```

-- Output:

a	count
michelle	3
diane	4
sina	2

(3 rows)

```
SELECT P.id, count(Follows.b) AS followers
From Profile P Join Follows
On P.ID=Follows.b
Group by(P.ID) HAVING count(Follows.b) > 1;
-- Output:
```

id	followers
sina	2

(1 row)

Here are the tables again, for easy reference:

Follows:

a	b
sina	kanyewest
sina	RonConwayFacts
diane	LilaFontes
diane	swcarpentry
diane	mfeathers
diane	sina
michelle	sina
michelle	diane
michelle	Jeff

(9 rows)

Profile:

id	name	location
alan	catman	Ottawa
sina	superman	
diane	superwoman	Toronto
michelle	rockstar	Montreal

(4 rows)

Tweets:

id	userid	content
123	alan	hellow twitter
125	alan	bye twitter
126	alan	hellow twitter
128	alan	bye twitter
476	sina	hellow twitter
553	diane	hellow twitter

(6 rows)

```
SELECT P.id, count(T.content) AS number
FROM Profile P JOIN Tweets t
  On T.userid = P.id
AND P.location='Montreal';
```

```
Select Tweets.content
From Tweets Join Profile
On Tweets.userid = Profile.ID
And Profile.location IN
(select location from profile
 where name='catman');
```

-- Output:

```
ERROR: column "p.id" must appear in the GROUP BY clause or be used in an aggregate function
LINE 1: SELECT P.id, count(T.content) AS number
                                hellow twitter
                                bye twitter
                                hellow twitter
                                bye twitter
                                (4 rows)
```

