UNIVERSITY OF VICTORIA

EXAMINATION #3 (SQL & Transactions)

VERSION A

CSC 370: Database Systems

14 Nov 2022

13.00 - 13.45 UTC-7

(0 hours, 45 minutes)

This examination consists of ten equally-weighted multiple choice questions. You should record your solutions in the provided bubble sheet. Each question has a single best solution; if you record more than one solution for the same question, you will receive a score of zero on that question. If you answer x *questions correctly, then your grade on the exam will be* x*/10, i.e., you must answer at least five questions correctly to pass. This exam is closed-book: you are welcome to bring with you empty pages and a single-sided A4/US letter note sheet, but you cannot bring other notes nor electronic devices to your desk. Please confirm immediately after the exam starts that you have all 5 pages and ten questions.*

Sections: A01, A02, A03

CRN's: 10874, 10875, 14303

Instructor: Mr. Yichun Zhao

Data		
x	**y**	**z**
1	3	2
2	4	3
3	5	NULL

Question 1

You are provided with the table "Data", in which *y* is the primary key. Which of the three provided queries could plausibly insert at least one new tuple to the table? There are **no syntax errors** in the options.

(a) INSERT INTO Data (SELECT x, y+1, z FROM Data); // this creates duplicates for y

(b) INSERT INTO Data(x) (SELECT z FROM Data); // no guarantee y has default values which are distinct

(c) INSERT INTO Data(y) (SELECT z FROM Data LIMIT 1);

Question 2

You are provided with the table "Data", in which *y* is the primary key. Which of the four provided queries could remove all tuples (and just the tuples) from the table? There are **no syntax errors** in the options.

(a) DELETE FROM Data WHERE y != NULL; // should use `IS NOT NULL`

(b) DROP TABLE Data; // this deletes the meta data as well

(c) DELETE FROM Data WHERE x < 100 and 1;

Question 3

Which of the scenarios below best describes an example of the *Atomicity* property being **violated**?

(a) The data from a committed transaction is lost after a power outage

(b) A transaction that contains two queries commits after executing just one of them

(c) A transaction causes non-deterministic (i.e., random) behaviour

(d) A transaction is aborted before it completes

Question 4

Given the relations:

adult(id, age)
adult_backup(id, age), where adult_backup.id is a foreign key to adult.id

Currently, the two tables have exactly the same data. Which of the behaviours written below is correct regarding the effect of this trigger?

```
CREATE TRIGGER update_after
AFTER UPDATE ON adult
FOR EACH ROW
        UPDATE adult_backup SET id = NEW.id, age = NEW.age
                WHERE id = OLD.id AND age = OLD.age;
```

(a) **This trigger ensures the tuples in both tables are being updated accordingly.**

(b) This trigger makes sure `adult_backup` is always exactly a backup copy of `adult`.
// does not consider insertion / deletion

(c) This trigger is invalid.

Question 5

Which of the following statements creates an index on the table R created below when using a MySQL database? There are **no syntax errors** in the options.

```
CREATE TABLE R(a int, b int, c int, d int);
```

(a) ALTER TABLE R ADD PRIMARY KEY(c);

(b) ALTER TABLE R MODIFY B INT UNIQUE;

(c) CREATE INDEX idx ON R(d);

(d) all of the above

Question 6

Given relation R with the following data:

x	y
1	5
2	0

Two transactions, T1 and T2, execute concurrently:

T1	T2
BEGIN TRANSACTION;	BEGIN TRANSACTION;
UPDATE R SET y = y-5 where x=1;	SELECT SUM(y) from R;
UPDATE R SET y = y+5 where x=2;	COMMIT;
COMMIT;	

Which isolation level is the **most strict** that can be applied to T2, if the SELECT query returns the value 0?

(a) READ UNCOMMITTED // clearly dirty read happens for T2 before T1 commits

(b) REPEATABLE READ

(c) SERIALISABLE

(d) READ COMMITTED

Question 7

Given the following example query:

```
SELECT * FROM a_table WHERE id > 100 ORDER BY c;
```

The relation is defined as:

a_table(id, a, b, c) where no indexes exist.

Given the following possible improvements of run time of this query. Which of the option is correct?

1. Make id the primary key.
2. Create a two-column index on (id, c).
3. Create a two-column index on (c, id).
4. Create an index on c.

(a) 1. and 2. have the same effect for this query. // 2 is the better option

(b) 2. and 3. have the same effect for this query. // 2 is the better option

(c) 3. and 4. have the same effect for this query.

Question 8

Given the following relations:

Instructor(v_number, specialization)
Class(code, name)
Teaches(v_number, code, semester)

and given the following query:

```
SELECT c.name FROM
Class as c, Teaches as t
WHERE c.code=t.code
GROUP BY c.name
HAVING COUNT( v_number ) >= 2;
```

Which of the following is equivalent to the query above?

(a)
```
SELECT c.name FROM
        Teaches JOIN Class ON (Class.code = Teaches.code)
WHERE COUNT( v_number) >= 2;
```

(b)
```
SELECT c.name FROM
        (SELECT name, COUNT(v_number) as c FROM
                Teaches NATURAL JOIN Class
        GROUP BY name ORDER BY COUNT(v_number) DESC)
AS c WHERE c.c >= 2;
```

(c)
```
SELECT c.name
FROM Teaches as t NATURAL JOIN Class as c
GROUP BY c.name ORDER BY COUNT(v_number) DESC LIMIT 2;
```

Question 9

The following example query is executed in a newly created empty database in MySQL:

```
CREATE TABLE J (a INT PRIMARY KEY, b INT NOT NULL);
```

Which of the following options is **invalid,** meaning not being able to be executed or throwing out error(s) when it is executed? There are **no syntax errors** in the options.

(a) CREATE TABLE K (c INT PRIMARY KEY, d INT, FOREIGN KEY(d) REFERENCES J(b)); // no constraint on J(b) being unique to ensure referential integrity

(b)
```
CREATE TABLE K (c INT UNIQUE KEY, d INT );
ALTER TABLE K ADD FOREIGN KEY (d) REFERENCES K(c);
```

(d)
```
CREATE TABLE K (c INT PRIMARY KEY, d INT );
ALTER TABLE J ADD FOREIGN KEY (a) REFERENCES K(c);
```

Question 10

A pair of tables and a SQL query are specified below. Indicate which of the specified tuples would be returned by this query if executed on these tables using MySQL with default setting

S		
t	**u**	**v**
1	NULL	'a'
2	4	'b'
3	2	'a'
4	5	'b'
5	3	'a'
6	2	'b'

R		
x	**y**	**z**
1	1	'Alice'
1	2	'Eve'
1	3	'Eve'
2	4	'Carol'
3	5	'Eve'
4	6	'Bob'

```
SELECT t,u,v,y,z FROM
S RIGHT OUTER JOIN R ON (R.x = S.u);
```

(a) (NULL,NULL,NULL,3,'Eve')

(b) (2,4,'b',6,'Bob')

(c) Both (a) and (b) would be returned

(d) Neither (a) or (b) would be returned