FINAL EXAM: DATABASES ("BASES DE DATOS") – 6/6/08 – SCHEMA

Consider the following relational schema, which will be referred to as WORKING SCHEMA, which maintains information on a personal library:

AUTHOR(author_id: string(4), name: string(35), nationality: string(20))
PK: {author_id}
NNV: {name}

BOOK(book_id: string(10), title: string(80), year: integer, num_works: integer)
PK: {book_id} NNV: {num_works}

WORK(wk_code: integer, title: string(80), year: integer, topic: string(20))
PK: {wk_code}
NNV: {title}

FRIEND(num: integer, name: string(60), telephone: string(10))
PK: {num}
NNV: {name}

BORROWED(num: integer, book_id:string(10))
PK: {num,book_id}
FK: {num} → FRIEND
Restrictive delete and update
FK: {book_id} → BOOK
Restrictive delete and update

IS_IN(wk_code: integer, book_id:string(10))
PK: {wk_code,book_id}
FK: {wk_code} → WORK
Restrictive delete and update
FK: {book_id} → BOOK
On delete cascade
On update cascade

WAS_WRITTEN(wk_code: integer, author_id:string(4))
PK: {wk_code,author_id}
FK: {wk_code}→ WORK
Restrictive delete and update
FK: {author_id}→ AUTHOR
Restrictive delete and update

LOST(num: integer, book_id:string(10))
PK: {num,book_id}
FK: {num,book_id}→ BORROWED
Restrictive delete and update
Complete referential integrity

where the attributes and tables have the following meaning

Author: *for each author we store his/her identifier, his/her name and his/her nationality.*

Book: *for each book we store its identifier, the title if it has one, the year in which it was bought and the number of works it contains.*

Work: *for each work we store its code, the title, the year of creation and the topic.*

Friend: *for each friend we store his/her number, the name and the telephone.*

Borrowed: *each tuple of this relation represents the books that have been borrowed by friends.*

Is_in: *each tuple of this relation represents the works that are included in a book.*

Was_written: *each tuple of this relation represents the authors who have written a work.*

Lost: *represents the loans which have been lost.*

Additionally, the data must follow the **following properties**:

C1) The value of the attribute *num_works* in *Book* must always be equal to the number of tuples in *Is_in* where the book appears.

C2) Every book contains one work at least.

And consider the following extension of the previous schema. We will refer to this extension as database (DB). Empty cells represent null values:

Author		
Author_id	**Name**	**Nationality**
ADBI	Bioy Casares, Adolfo	Argentinian
ADGA	García Morales, Adelaida	Spanish
ADJA	James, Adobe	
ALBU	Budrys, Algis	
ALCA	Camús, Albert	French

Is_in	
Wk_code	**Book_id**
399	BOOK399
371	BOOK371
239	BOOK226
248	BOOK226
418	BOOK418

Was_written	
Wk_code	**Author_id**
399	ADBI
371	ADGA
239	ADJA
248	ALBU
418	ALCA

Book			
Book_id	**Title**	**Year**	**Num_Works**
BOOK399			1
BOOK371			1
BOOK226	Relatos que me asustaron		2
BOOK418			1

Work			
Wk_code	**Title**	**Topic**	**Year**
399	La invención de Morel		
371	El silencio de las sirenas	Novel	
239	El camino de Mictlantecutli	Tale	
248	El amo de los perros	Tale	
418	El extranjero	Novel	

Friend		
Num	**Name**	**Telephone**
1	Pepe Pérez Pérez	555 123 12
2	Isabel Peiró García	555 446 56
3	Marina Sánchez Vidal	555 789 89

Borrowed	
Num	**Book_id**
1	BOOK371
1	BOOK226
2	BOOK418

Lost	
Num	**Book_id**
2	BOOK418

Final exam: "Databases" – 6/06/08 – QUESTIONNAIRE TYPE A

This questionnaire has 14 questions; for each one we propose four possible answers. Only one of them is correct. The answer must be included in the answer sheet which has been handed with the exam. The maximum mark for the questionnaire is 3.5 points. The result is obtained through the formula: (Right – Wrong/3) × 0.25.

1. Given the working schema, which of the following options is FALSE?

a) An author can have written several works.

b) A work can be found in several books.

c) A book can have been borrowed and been lost.

d) A work cannot have two authors.

2. In the database BD, if we execute the instruction DELETE FROM Book WHERE book_id='BOOK371', which of the following options is TRUE?

a) The DBMS will reject the deletion.

b) The tuples Is_in(371,BOOK371), Borrowed(1, BOOK371) will also be deleted and the tuple Lost(1,BOOK317) will be inserted.

c) The tuples Is_in(371,BOOK371), Borrowed(1,BOOK371) will also be deleted.

d) The tuple Is_in(371,BOOK371) will also be deleted.

3. According two the ANSI/SPARC level architecture. Which of the following options is FALSE?

a) The internal schema describes the database in terms of the data model in use.

b) The external schema describes the several partial views for each user in the database.

c) Data independence ensures that the changes in a schema will not affect the upper schemas on data which is not used by the upper schemas.

d) The conceptual schema describes the information system from an organisational point of view which is independent to the data model the DBMS is used.

4. Which is the cardinality of the relation defined by the following expression in Relational Algebra? Borrowed ⋈ Lost.

a) Card(Borrowed)

b) Card(Book)

c) Card(Borrowed) * Card(Lost)

d) Card(Lost)

5. About physical database implementation, which of the following options is TRUE?

a) A hash file is appropriate to store relations where there is a big amount of null values (hash relations).

b) In an ordered file, indexes are useless.

c) In a disordered file, it is convenient to use indexes to ease the operations of record insertion and deletion.

d) In an ordered file, the insert operation is costlier than in a disordered file.

6. Considering the working schema, which information does the following relational algebra expression return?

(((BOOK[book_id,num_works] × BOOK[book_id,num_works]
((book_id,id),(num_works,num))) WHERE num_works < num) [book_id] ⋈ Book) [title]

a) Titles of the books which have the greatest number of works.

b) Titles of the books which have the smallest number of works.

c) Titles of the books which do not have the greatest number of works.

d) Titles of the books which do not have the smallest number of works.

7. How can we define the following constraint "Every author must have written at least one work" in standard SQL in the working schema?

a) This constraint is already covered by the definition of the working schema.

b) It can be defined through an attribute constraint over the attribute "author_id" in the relation *Was_written*.

c) It can only be defined through a general constraint (CREATE ASSERTION).

d) It can be defined either through a general constraint (CREATE ASSERTION) or with triggers (CREATE TRIGGER).

8. Which of the following relational algebra expressions do NOT obtain the name of the friends who have lost some book?

a) ((Friend[num] - Lost[num]) ⋈ Friend) [name]

b) (Lost[num] ⋈ Friend) [name]

c) ((Friend[num] ∩ Lost[num]) ⋈ Friend) [name]

d) (Lost ⋈ Borrowed ⋈ Friend) [name]

9. Which of the following statements about checkpoints is TRUE?

a) Before the recording of checkpoint in the logfile, all the transactions in execution are suspended and their locks are freed.

b) If the DBMS update mode is deferred the use of checkpoints in the logfile does not speed up the reconstruction of the database.

c) The checkpoint is used to recover the database after a loss of secondary memory which affects the database.

d) The updates on the database which have been performed by transactions which have been confirmed before a checkpoint are stored in disk.

10. Which of the following options is TRUE?

a) A database must handle concurrency problems.

b) Database technology allows data to only depend on user applications.

c) A database is a collection of data which is structured according to a model rules.

d) A database management system is composed of domains, relations, constraints and views.

11. According to the working schema, which of the following options is TRUE?

a) At least an author for each work must be known.

b) A book can be lost without being borrowed.

c) The attribute *num_works* for each tuple in *Book* does not need to match with the number of tuples in the relation *Is_in* where the book appears.

d) We can have authors in the relation *Author* which do not appear in the relation *Was_written.*

12. Given the database DB, what happens if in the relation *Is_in* we change the value *Book_id*=BOOK371 with *Book_id*=BOOK410?

a) The change will be done without any problem, and it will not generate any other cascade change.

b) The change violates the referential integrity and will not be allowed.

c) The change will be performed in cascade on the table *Book*.

d) It cannot be changed because this book is borrowed.

13. Choose which of the following options is TRUE:

a) Lock protocols are used to avoid incorrect updates due to the concurrent execution of transactions.

b) Lock protocols are used to recover database integrity, when this is lost because system failures.

c) Lock protocols are used to avoid deadlocks due the concurrent execution of transactions.

d) Lock protocols are used to achieve more efficiency in the concurrent execution of transactions.

14. Given the database DB, what happens if we execute the following transaction in a DBMS which ensures the atomicity of transactions?

START

INSERT INTO Was_written VALUES (399, 'ALMA')

INSERT INTO Author VALUES('ALMA', 'Maravillas, Alicia', 'Spanish')

COMMIT

a) The two new tuples are added without problems.

b) The first instruction fails, but the second one is executed.

c) If the foreign key {author_id}→ AUTHOR of *Was_written* were defined as DEFERRABLE, the two new tuples would be added without any problem.

d) If the foreign key {author_id}→ AUTHOR of *Was_written* were defined as INITIALLY DEFERRED, the two new tuples would be added without any problem.

FINAL EXAM: DATABASES – 6/6/08 – Problems

Given the working schema presented before, solve the following exercises in standard SQL:

1. Obtain the name of the friends who have borrowed exactly two books (0.75 points).

2. Obtain the name of the authors with unknown nationality and whose name starts by 'B' (0.5 points).

3. Obtain the name of the friends who have borrowed some book which contains some work of the author whose identifier is ALBU (0.75 points).

4. Obtain the name of the friends who have lost all the books they have borrowed (1 point).

5. Obtain the code and the title of the work (or works) which appears most frequently in books (the number of books where it appears is greatest) (1 point)

6. Obtain the number and the name of ALL the friends, also showing for each one the number of books which s/he has borrowed. (0.75 points)

7. Obtain the name of the friends who have never borrowed a book. (0.5 points)

8. Having into account that the value of the attribute *num_works* in the relation BOOK is a derived attribute whose value is equal to the number of works the book contains:

    a) Apart from the operation of UPDATING the attribute *book_id* in the relation IS_IN, please indicate the operations/events which may affect the value of this attribute. For each of these operations please specify the way in which the attribute *num_works* in the relation BOOK should be modified (0.5 points).

    b) Implement the trigger which corresponds to the operation of UPDATING the attribute *book_id* in the relation IS_IN (0.75 points).

SOLUTIONS TO THE QUESTIONNAIRE:

Question	Answer
1	d
2	a
3	a
4	d
5	d
6	c
7	d
8	a
9	d
10	c
11	d
12	b
13	a
14	d

SOLUTIONS TO THE PROBLEMS:

1.-

```
SELECT name FROM Friend a
WHERE 2=(SELECT COUNT(*) FROM Borrowed p
        WHERE a.num=p.num)
```

2.-

```
SELECT name
FROM Author
WHERE nationality IS NULL AND name LIKE 'B%';
```

3.-

```
SELECT name FROM Friend
  WHERE num IN (SELECT num FROM Borrowed p, Is_in e, Was_written  ee
              WHERE  p.book_id=e.book_id AND e.wk_code=ee.wk_code AND ee.author_id='ALBU')
```

4.-

```
SELECT name FROM Friend a
    WHERE NOT EXISTS (SELECT * FROM Borrowed p WHERE a.num=p.num AND
                    NOT EXISTS (SELECT * FROM Lost pr
                                WHERE p.num=pr.num AND p.book_id=pr.book_id))
      AND  EXISTS(SELECT * FROM Borrowed p WHERE a.num=p.num);
```

5.-

```
SELECT o.wk_code, o.title
FROM Work o, Is_in e
WHERE o.wk_code = e.wk_code
GROUP BY o.wk_code, o.title
HAVING COUNT(e.book_id) = (SELECT  MAX(COUNT(book_id))
                                   FROM is_in
                                   GROUP BY wk_code);
```

6.-

```
SELECT a. num, a.name, COUNT(p.book_id)
FROM Friend a LEFT JOIN Borrowed p ON p.num = a.num
GROUP BY a.num, a.name;
```

7.-

```
SELECT name
FROM Friend
WHERE num NOT IN (SELECT num FROM Borrowed);
```

8.-

a)

- Update *book_id* in IS_IN → subtract 1 from *num_works* and add 1 to *num_Works*, to the books which correspond to the old and new values, respectively, of *book_id.*
- Insert in IS_IN → add 1 to *num_works* in the book which corresponds to the new value of *book_id.*
- Delete from IS_IN → subtract 1 from *num_works* in the book which corresponds to the old value of *book_id.*
- Insert into BOOK → *num_works=0.*
- Update *num_works* in BOOK, forbidden.

b)

```
CREATE OR REPLACE TRIGGER modify_book_id_in_Is_in
AFTER UPDATE OF book_id ON IS_IN
FOR EACH ROW
WHEN NEW.book_id <> OLD.book_id
BEGIN
UPDATE BOOK SET num_works = num_works -1
WHERE book_id=:OLD.book_id;
UPDATE BOOK SET num_works = num_works +1
WHERE book_id=:NEW.book_id;
END;
```