FINAL EXAM: DATABASES ("BASES DE DATOS") – 13/09/04 – SCHEMA

When you have finished your questionnaire, you can copy your answers to the following table. In this way, once finished your exam, you will be able to calculate the result you obtained (Right – Wrong/3) × 0,25.

1	2	3	4	5	6	7	8	9	10	11	12	13	14

Consider the following relational schema, which will be referred to as WORKING SCHEMA, which maintains information on country houses:

Player (ID: ID_dom, name:name_dom, team_name:name_team_dom)
PK:{ID}
FK:{team_name} →TEAM
RESTRICTED deletion, On update CASCADE
NNV:{name}
Stadium (st_name: st_name_dom, city:city_dom, capacity: capacity_dom)
PK: {st_name, city}
NNV:{capacity}
Category (catnum: category_dom, name:name_dom)
PK: {catnum}
NNV:{name}
Team (team_name:name_team_dom, city: city_dom, category: category_dom)
PK:{team_name}
FK:{category} → CATEGORY
f(category) = catnum
RESTRICTED deletion, On update CASCADE
NNV:{city, category}
Match (local:name_team_dom, visitor: name_team_dom, date:date_dom, st_name:st_name_dom, city: city_dom)
PK: {local, visitor}
FK:{local} → TEAM
f(local) = team_name
On delete CASCADE, On update CASCADE
FK:{visitor} → TEAM
f(visitor) = team_name
On delete CASCADE, On update CASCADE
FK:{st_name, city} → STADIUM
Partial Referential Integrity
RESTRICTED deletion, On update CASCADE
NNV:{date, city}

where the attributes and tables have the following meaning

Player: *ID:* player identifier
name: player name
team_name: player team

Stadium: *st_name:* stadium name
city: city name where the stadium is located
capacity: capacity of the stadium

Category: *catnum:* category number
name: category name

Team: *team_name:* team name
city: city of the team
category: category in which the team plays

Match: *local*: name of the local team
visitor: name of the visitor team
date: date of the match
st_name: name of the stadium where the match takes place
city: city in which the match is played

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

Player		
ID	name	team_name
1	Stelea	Salamanca
2	Almunia	Albacete
3	Cocu	Barcelona
4	Ochoa	Alavés
5	Alex	

Stadium		
st_name	city	capacity
Mestalla	Valencia	49674
Carlos Belmonte	Albacete	16141
Mendizorroza	Vitoria	19238
Nou Estadi	Valencia	27900
Helmántico	Salamanca	17341
Camp Nou	Barcelona	98187

Category	
catnum	name
1	Primera división
2	Segunda división

Team		
team_name	city	category
Elche	Elche	2
Levante	Valencia	1
Salamanca	Salamanca	2
Albacete	Albacete	2
Barcelona	Barcelona	1
Alavés	Vitoria	1
Valencia	Valencia	1

Match				
local	visitor	date	st_name	city
Salamanca	Albacete	16-10-04	Helmántico	Salamanca
Valencia	Levante	26-09-04	Mestalla	Valencia
Alavés	Barcelona	05-03-05	Mendizorroza	Vitoria
Levante	Valencia	19-03-05	Nou Estadi	Valencia

FINAL EXAM: "DATABASES" – 11/06/04 – Questionnaire Type A

1. According to the database schema, which of the following expressions in Relational Algebra, answer the question: "obtain the name of the stadium with highest capacity"?

a) (Stadium ⋈ Stadium((city, CIT), (capacity, CAP)))[st_name] − (Stadium WHERE capacity = 0)[st_name]

b) ((Stadium × Stadium((st_name, NAM), (city, CIT), (capacity, CAP))) WHERE CAP > capacity) [NAM]

c) Stadium[st_name] − ((Stadium × Stadium((st_name, NAM), (city, CIT), (capacity, CAP))) WHERE CAP > capacity) [st_name]

d) ((Stadium × Stadium((st_name, NAM), (city, CIT), (capacity, CAP))) WHERE CAP > capacity) [NOM] − (Stadium(st_name, NAM)) [NAM]

2. Assume that all the integrity constraints have been defined as DEFERRABLE INITIALLY IMMEDIATE and consider a transaction T1 which is executed over the database DB in the Oracle DBMS:

```
TRANSACTION T1
        INSERT INTO Stadium VALUES ('Heliópolis', 'Sevilla', 70000);
        INSERT INTO Match VALUES ('Salamanca', 'Valencia', 05-01-05,
                                  'Heliópolis', NULL);
        COMMIT;
COMMIT
```

Which of the following statements is TRUE?

a) T1 will only insert the tuple {(local, *'Salamanca'*), (visitor, *'Valencia'*), (date, *05-01-05*), (st_name, *'Heliópolis'*), (city, *NULL*)} in Match.

b) T1 will fail, since all the constraints are defined as DEFERRABLE, which means that either all are execture or none is executed (transaction atomicity property).

c) T1 will only insert the tuple {(st_name,*'Heliópolis'*), (city,*'Sevilla'*),(capacity, *70000*)} in the Stadium relation.

d) T1 will not fail because the constraint of **Partial Referential Integrity** defined over the foreign key in Match is met.

3. When will an integrity constraint be checked if it has been defined as NOT DEFERRABLE INITIALLY IMMEDIATE?

a) After each instruction that may affect the constraint, but it can be changed in such a way that it would be checked at the end of the transaction.

b) After each transaction, but it cannot be changed in such a way that it would be checked after each instruction that may affect the constraint.

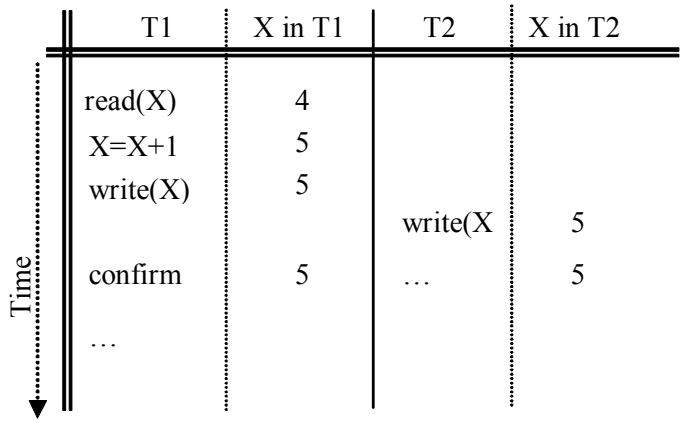
c) This option is forbidden.

d) After each instruction that may affect the constraint, and it cannot be changed in such a way that it would be checked at the end of the transaction.

4. Which of the following statements related to the concept of *binding* is FALSE?

a) Can take place when the program execution begins
b) The applications are less efficient as more frequent the binding is
c) Data independence is greater the sooner the binding is
d) When the binding takes place, data independence is lost

5. What can be stated about a database management system (DBMS) which process transactions T1 and T2 as the following figure shows:



a) The DBMS does not ensure transaction atomicity.
b) The DBMS does not ensure transaction isolation.
c) The DBMS does not ensure transaction consistency.
d) The DBMS does not ensure transaction durability.

6. Which of the following statements about *checkpoints* is TRUE?

a) Are used when secondary memory is lost.
b) Can speed fail recovery which involve main memory losses.
c) The transaction execution is not suspended during the recording of a checkpoint.
d) Can speed fail recovery in front of secondary memory losses which affect the database.

7. According to the working schema definition, which of the following statements is TRUE?

a) Every match is played in the city of one of the playing teams.
b) A team can belong to a city which has no stadiums.
c) Every team must have at least one player.
d) A team cannot play against itself.

8. According to the working schema definition, which of the following statements is TRUE?

a) We cannot delete a match from the database if the stadium where it takes place is in relation *Stadium*.
b) We cannot delete a team if it plays a match.
c) We cannot delete a player if the team to which he belongs is the only one in its category.
d) We cannot delete a stadium if there are matches played in that stadium.

9. If we modify the attribute *city* in a match Valencia-Levante and we assign the value "Albacete". Which of the following statements is TRUE?

a) We cannot make this modification because none of the teams which take part in the match are from the city "Albacete".
b) The referential integrity over the foreign key {*st_name, city*} is violated because there is no stadium in the city "Albacete" with name "Mestalla".

c) The referential integrity is not violated because it is partial and there is no stadium called "Mestalla".

d) If the referential integrity was weak, the change would be performed with no problems.

10. Given the definition of the following view:

```
CREATE VIEW Big_Cities AS
    SELECT DISTINCT E.city FROM Team E, Team F
    WHERE E.city=F.city AND E.category='1' AND F.category='1' AND
            E.team_name<>F.team_name WITH CHECK OPTION;
```

The following query obtains, for any database extension:

```
SELECT DISTINCT P.date FROM Big_Cities C, Match P
WHERE C.city = P.city;
```

a) The dates of the matches in cities which have two or more teams in "primera división".

b) The dates of the matches in cities which have exactly two teams in "primera división".

c) The dates of the matches in cities which have one or more teams in "primera división".

d) The dates of the matches in cities which have only one team in "primera división".

11. If we perform the following instruction:

```
INSERT INTO Match(local, visitor, date, city)
        VALUES('Levante', 'Levante', '11/9/2004', 'Valencia');
```

a) It wouldn't be allowed, because there are two stadiums in 'Valencia' and we wouldn't know where the match is played.

b) It wouldn't be allowed, because a team cannot play against itself.

c) It wouldn't be allowed, because the type of referential integrity does not allow having null values in *st_name*.

d) It would be allowed.

12. Which constraint does the following SQL instruction impose over the working schema?

```
CREATE ASSERTION R2
CHECK (NOT EXISTS (SELECT * FROM Team E
                   WHERE 22 < (SELECT COUNT(*)
                               FROM PLAYER J
                               WHERE J.team_name=E.team_name)))
```

a) A team cannot have more than 22 players.

b) There cannot be more than 22 teams with players.

c) A team cannot have less than 22 players.

d) There cannot be less than 22 teams with players.

13. Which set of instructions can violate the constraint R2 defined in the previous question?

a) Insert into the relation Player and modify the attribute *team_name* in the relation Player.

b) Insert and delete into/from the relation Player and modify the attribute *team_name* in the relation Player.

c) Insert into the relation Player, modify the attribute *team_name* in the relation Player and modify the attribute *team_name* in the relation Team

d) Insert and Delete in the relation Player, modify the attribute *team_name* in the relation Player and Delete in the relation Team

14. Which of the following SQL statements would be valid to define a Primary Key constraint on relation Match?

a) CREATE TABLE Match
(local name_team_dom CONSTRAINT pk1 PRIMARY KEY,
visitor name_team_dom CONSTRAINT pk2 PRIMARY KEY,
…)

b) CREATE TABLE Match
(local name_team_dom,
visitor name_team_dom,
…
CONSTRAINT pk PRIMARY KEY(local,visitor))

c) CREATE TABLE Match
(local name_team_dom,
visitor name_team_dom,
…
CONSTRAINT pk UNIQUE(local, visitor))

d) The solutions a) and b) are both valid since the Primary Key can always be defined as an attribute constraint or as a relation constraint.

SOLUTIONS TO THE QUESTIONNAIRE:

	Answers Type A/E	Answer Type B/F	Answer Type C/G	Answer Type D/H
1	C	D	A	B
2	C	D	A	B
3	D	A	B	C
4	C	D	A	B
5	B	C	D	A
6	B	C	D	A
7	B	C	D	A
8	D	A	B	C
9	B	C	D	A
10	A	B	C	D
11	D	A	B	C
12	A	B	C	D
13	A	B	C	D
14	B	C	D	A

FINAL EXAM: DATABASES – 11/06/04 – Problems

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

1. Obtain the name of the teams without players. (0.5 points)
2. Obtain the name of the cities where there are matches between teams of the same category. (0.5 points)
3. Obtain the name of the city where there have been more matches. (0.75 points)
4. Obtain the name of the teams playing in the category "Primera división", which have the highest number of players. (0.75 points)
5. Obtain the name of the categories in which no team in that category has played any match (according to the database). (0.75 points)
6. Assuming that in the database there is at least one stadium, obtain the name of the teams which have played a match in all the stadiums. (1 point)
7. Obtain the name of the cities where more than 5 matches have been played, and that have more than 2 stadiums. Also show the number of matches per city. (1 point)

8. Assume that we add to the relation *Team* the derived attribute *Nplayers*, whose value is obtained by counting the number of players which belong to each team.
    a. Apart from the insertion into the relation *Player*, enumerate the instructions over the database which can entail the update of the value of the derived attribute. (0.5 points)
    b. Design a trigger in standard SQL to control the insertion into *Player*. (0.75 points)

SOLUTIONS TO THE PROBLEMS:

1. SELECT E.team_name FROM Team E
   WHERE E.team_name NOT IN
   (SELECT J.team_name FROM Player J)

2. SELECT DISTINCT P.city FROM Match P, Team E1, Team E2
   WHERE P.local = E1.team_name AND P.visitor = E2.team_name AND
   E1.category = E2.category

3. SELECT P.city FROM Match P
   GROUP BY P.city
   HAVING COUNT(*) >= (SELECT MAX(COUNT(*)) FROM Match P1
   GROUP BY P1.city)

4. SELECT E.team_name FROM Team E, Category C, Player J
   WHERE E.category = C.catnum AND C.name = "Primera división" AND
   J.team_name = E.team_name
   GROUP BY E.team_name
   HAVING COUNT(J.ID) >= ALL(
   SELECT COUNT(J1.ID) FROM Team E1, Category C1, Player J1
   WHERE E1.category = C1.catnum AND C1.name = "Primera división"
   AND J1.team_name = E1.team_name
   GROUP BY E1.team_name)

5. SELECT C.name FROM Category C
   WHERE NOT EXISTS (SELECT * FROM Match P, Team E
   WHERE (P.local = E.team_name OR
   P.visitor = E.team_name) AND
   E.category = C.catnum)

6. SELECT E.team_name FROM Team E
   WHERE NOT EXISTS (SELECT * FROM Stadium Es
   WHERE NOT EXISTS (
   SELECT * FROM Match P
   WHERE P.st_name = Es.st_name AND
   P.city = Es.city AND
   (P.local = E.team_name OR
   P.visitor = E.team_name)))

7. SELECT P.city, COUNT(*) FROM Match P
   WHERE 2 <= (SELECT COUNT(*) FROM Stadium Es WHERE Es.city = P.city)
   GROUP BY P.city
   HAVING COUNT(*) >= 5

8. a)
   – Insert into *Player*
   – Delete from *Player*
   – Update *team_name* in *Player*

   Additionally:
   – Insert into *Team* ➔ the value *Nplayers* must be zero
   – Update *Nplayers* in *Team* ➔ forbidden

   b) In standard SQL:

```
CREATE TRIGGER Insertion_on_ Player
AFTER INSERT ON Player
REFERENCING NEW ROW AS mynew
FOR EACH ROW
WHEN mynew.team_name IS NOT NULL

BEGIN ATOMIC
        UPDATE Team SET Nplayers= Nplayers+1
                WHERE team_name=mynew.team_name;
END
```