**FINAL EXAM: DATABASES ("BASES OF DATOS") – 12/09/05 – SCHEMA**

Once you had 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 a car race championship:

**Team** (teamcode: teamcode_dom, teamname: teamname_dom, country: country_dom)
PK: {teamcode}
NNV: {teamname}

**Driver** (num: num_dom, drivername: drivername_dom, birthdate: dom_ date, teamcode: teamcode_dom)
PK: {num}
NNV: {drivername, teamcode}
FK: {teamcode} → TEAM RESTRICTED Deletion, On update CASCADE

**Circuit** (circode: circode_dom, cirname: cirname_dom, city: city_dom, km: km_dom)
PK: {circode}

**Race** (year: year_dom, circode: circode_dom, date: date_dom, num_drivers: num_dom)
PK: {year, circode}
FK: {circode} → CIRCUIT
On delete CASCADE, On update CASCADE
num_drivers: default value=0

**Participates** (year: year_dom, circode: circode_dom, num: num_dom, points: points_dom)
PK: {year, circode, num}
FK: {num} → DRIVER
On delete CASCADE, On update CASCADE
FK: {year, circode} → RACE
RESTRICTED Deletion, RESTRICTED Update

**Assertion** Different_points: Check (in the same race, two drivers cannot obtain the same points unless both have 0 points).

where the attributes and tables have the following meaning

**Team:** description of the teams
- teamcode: code of the team
- teamname: name of the team
- country: country of the team

**Driver:** description of the drivers
- num: driver's number
- drivername: driver's name
- birthdate: birth date
- teamcode: code of the team which the driver belongs to

**Circuit:** description of the circuits where races take place
- circode: code of the circuit
- cirname: name of the circuit
- city: city where the circuit is located
- km: length of the circuit

**Race:** description of the races which take place each year and constitute the championship
- year: championship year
- circode: circuit where the race takes place
- date: date when the race takes place
- num_drivers: number of drivers who participate in a race. This number is calculated by reckoning the related rows in the table PARTICIPATES

**Participates:** races in which each driver participates and the points which the driver obtained in the race.
- year: championship year
- circode: circuit where the race takes place
- num: driver who participates
- points: points obtained by the driver in the race.

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

| Team | | |
|---|---|---|
| teamcode | teamname | country |
| 1 | McLaren | Reino Unido |
| 2 | Renault | Francia |
| 3 | Toyota | Japón |
| 4 | Williams | Alemania |
| 5 | Ferrari | Italia |

| Circuit | | | |
|---|---|---|---|
| circode | cirname | city | km |
| 1 | Albert Park | Melbourne | 5.3 |
| 2 | Montmeló | Montmeló | 4.6 |
| 3 | Montecarlo | Monaco | 3.3 |

| Driver | | | |
|---|---|---|---|
| num | drivername | birthdate | teamcode |
| 1 | Fernando Alonso | 1981 | 2 |
| 2 | Kimi Raikonen | 1979 | 1 |
| 3 | Juan Pablo Montoya | 1975 | 1 |
| 4 | Michael Schumacher | 1969 | 5 |
| 5 | Rubens Barrichello | 1972 | 5 |

| Race | | | |
|---|---|---|---|
| year | circode | date | num_drivers |
| 2005 | 1 | 6/3/2005 | 2 |
| 2004 | 2 | 9/5/2004 | 0 |
| 2005 | 2 | 8/5/2005 | 1 |

| Participates | | | |
|---|---|---|---|
| year | circode | num | points |
| 2005 | 1 | 1 | 10 |
| 2005 | 2 | 1 | 8 |
| 2005 | 1 | 2 | 8 |

**FINAL EXAM: "DATABASES" – 12/09/05 – Questionnaire Type A**

**1.** During the execution of a transaction T, a main memory loss occurs. Which of the following statements is TRUE?

**a)** Only transaction T will have to be executed again.
**b)** All the confirmed transactions since the latest checkpoint will have to be remade
**c)** It is necessary to recover the most recent backup and repeat all the transactions since the backup date.
**d)** It is necessary to recover the most recent backup and repeat all the confirmed transactions since the backup date.

**2.** In the definition of the relation *Race* in the working schema, please indicate which of the following definitions of primary key is correct in SQL.

**a)** CONSTRAINT CP_Race PRIMARY KEY (year, circode).
**b)** year PRIMARY KEY, circode PRIMARY KEY DEFERRABLE.
**c)** PRIMARY KEY (year NOT NULL, circode NOT NULL).
**d)** CONSTRAINT CP_Race PRIMARY KEY (year, circode) INITIALLY DEFERRED NOT DEFERRABLE.

**3.** Assume that U1, U2 and U3 are users (not owners) of the database represented by the previous working schema, and assume that only the following authorisations have been granted,

```
GRANT INSERT ON Participates TO U2 WITH GRANT OPTION;
GRANT SELECT ON Circuit TO U1;
```

Which of the following statements is FALSE?

**a)** U3 could insert a tuple in *Participates* if s/he is authorised by U2.
**b)** U3 could read *Circuit* if s/he is authorised by U1.
**c)** U1 could only read *Circuit*.
**d)** Only U2 can insert a tuple in *Participates*.

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

**a)** A checkpoint confirms the transactions which are previous to the latest failure.
**b)** A checkpoint records on disk all the updates of the transactions which appear as confirmed in the logfile since the last checkpoint.
**c)** A checkpoint undoes the changes (already recorded) of the transactions which were interrupted since the last checkpoint.
**d)** A checkpoint records on disk all the updates of all the transactions since the last checkpoint.

**5.** Given a DBMS whose transactions comply with the isolation property, which of the following situations is IMPOSSIBLE?

**a)** A transaction begins while there is another transaction running.
**b)** A transaction has modified data which other non-confirmed transactions have read previously.
**c)** A transaction locks some data which other non-confirmed transactions have read (and released) previously.
**d)** A transaction T1 reads some data which T2 has modified, but T2 has not been confirmed.

**6.** In the context of a database which complies with the ANSI/SPARC architecture, which of the following statements is FALSE?

**a)** The external level describes the views that several users have of the database.
**b)** The logical independence guarantees that the application programs cannot be affected by changes on the logical schema of data which they do not use.
**c)** A binding in compilation time guarantees that a change in the logical schema will not require the recompilation of the application.
**d)** A change in the physical schema will not force a change in the source code of the applications which use the affected data.

**7.** On the DB, after the execution of the command: DELETE PARTICIPATES WHERE num=1

**a)** All the tuples of the relation *Participates* are deleted.
**b)** No tuple is deleted because the foreign key *FK:{year, circode}→RACE* has the constraint *RESTRICTED Deletion*.
**c)** The two tuples of the relation *Participates* whose driver has number 1 are deleted.
**d)** The two tuples of the relation *Participates* whose driver has number 1 are deleted, and, additionally, the driver is deleted.

**8.** Consider the transaction T1 which is executed over the DB in the Oracle DBMS:

```
TRANSACTION T1
    SET CONSTRAINT ALL IMMEDIATE;
    INSERT INTO PARTICIPATES VALUES('2005', 2,  2, 10);
    INSERT INTO PARTICIPATES VALUES('2005', 2, 4, 10);
COMMIT
```

**a)** The transaction will add two tuples to the relation *Participates*.
**b)** The transaction will not add any tuple.
**c)** The transaction will add the first tuple.
**d)** The transaction will add the second tuple.

**9.** On the proposed schema, which will be the maximum and minimum cardinality with the following query?

$$(\text{Team} \triangleright\!\triangleleft \text{Driver})\ [\text{num}] \triangleright\!\triangleleft \text{Participates}$$

**a)** The minimum cardinality is 0 and the maximum is Card (Driver) × Card(Participates).
**b)** The minimum and maximum cardinalities match with the minimum and maximum cardinalities of the relation Participates.
**c)** The minimum cardinality is 0 and the maximum is Card(Driver).
**d)** The minimum cardinality is Card(Driver) and the maximum is Card(Driver) × Card(Participates).

**10.** Given the proposed extension of the database, would it be possible to execute the following action: UPDATE Circuit SET circode=10 WHERE circode=1 ?

**a)** Yes, and the tuple Circuit(1, "Albert Park", "Melbourne", 5.3) will be modified.
**b)** No, never.
**c)** Yes, and the tuple Race(2005, 1, "6/3/2005", 16) will be modified on cascade.
**d)** Yes, and the tuples Race(2005, 1, "6/3/2005", 16), Participates(2005, 1, 1, 10) and Participates(2005, 1, 2, 8) will be modified.

**11.** Which constraint over the working schema would the following SQL command imply?

```
CREATE ASSERTION R1
CHECK (
NOT EXISTS (SELECT *
      FROM Race P1, Circuit C1
      WHERE P1.circode = C1.circode AND
            EXISTS (SELECT *
                  FROM Race P2, Circuit C2
                        WHERE P2.circode = C2.circode AND P1.year = P2.year
                         AND P1.circode <> P2.circode
                         AND C1.city = C2.city)));
```

**a)** Two races cannot take place in the same circuit.
**b)** Two races cannot take place in the same city the same year.
**c)** Two races cannot take place on two different circuits the same year.
**d)** All the races in the same year must take place in the same city.

**12.** Which query over the working schema corresponds to the following expression in Relational Algebra?

Circuit ▷◁ ((Race[circode, year]((circode,circuit),(year,any)) × Race[circode, year]) where circode = circuit ∧ year ≠ any) [circode]

**a)** Circuits where two or more races have taken place.
**b)** Circuits where exactly two races have taken place.
**c)** Circuits where no race has taken place.
**d)** Circuits where at least one race has taken place.

**13.** Which of the following statements is **FALSE**?

a) A team can participate with several drivers in a race.
b) A team can have no drivers.
c) There can be races in which no driver has participated.
d) A driver can score points in a race without participating in it.

**14.** Which integrity constraint does the following command correspond to?

```
CREATE ASSERTION Rest CHECK (
        NOT EXISTS (SELECT *
                    FROM Race R, Team T
                    WHERE 2 < (SELECT COUNT(*) FROM Participates P, Driver D
                               WHERE P.year=R.year AND P.circode=R.circode
                               AND P.num=D.num AND D.teamcode=T.teamcode));
```

a) Indicates that a driver cannot participate twice in the same race (same circuit and year) with different teams.
b) Indicates that a driver cannot participate in two different races with different teams.
c) Indicates that more than two drivers of the same team cannot participate in the same race.
d) Does not indicate any constraint which is not already covered by the schema.

**FINAL EXAM: DATABASES – 12/09/05 – Problems**

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

1. Obtain, for EVERY circuit, its code, name, city and the number of races which have taken placed there. (0.5 points)
2. Obtain the code and the name of the teams which have a driver born after 01/01/1985. (0.5 points)
3. Obtain the code (*num*) and the name of the drivers which have obtained more than 5 points in at least three of the races which have run. (0.5 points)
4. Obtain the code and the name of the circuits which have a length that is greater than the average length of all circuits. (0.5 points)
5. Obtain the code and the name of the circuits with at least 10 drivers in every race. (0.75 points)
6. Obtain the code (*num*) and the name of the drivers which have participated in at least 6 races during the year 2004. (0.75 points)
7. Obtain the code of the circuit which held the race with the greatest number of drivers in the year 1990. (0.75 points)
8. Obtain the code (*num*) and the name of the drivers who have participated in most races. (1 points)
9. El attribute *num_drivers* of the relation *Race* is a derived attribute which is obtained by counting the drivers who participate in that race.
    a. Enumerate the events over the database which can affect the value of the derived attribute. Please indicate the relation which is affected by each event, and the attribute if needed. (0.75 points)
    b. Design a trigger in SQL to adequately update *num_drivers* for each insertion in the relation *Participates*. (0.5 points)

## SOLUTIONS TO THE QUESTIONNAIRE:

| | |
|---|---|
| 1 | B |
| 2 | A |
| 3 | B |
| 4 | B |
| 5 | D |
| 6 | C |
| 7 | C |
| 8 | C |
| 9 | B |
| 10 | B |
| 11 | B |
| 12 | A |
| 13 | D |
| 14 | C |

## SOLUTIONS TO THE PROBLEMS:

1.

```
SELECT C.circode, C.cirname, C.city, COUNT(P.circode)
FROM Circuit C LEFT JOIN Race P ON C.circode = P.circode
GROUP BY C.circode, C.cirname, C.city;
```

2.

```
SELECT DISTINCT E.teamcode, E.teamname
FROM Team E, Driver P
WHERE P.teamcode=E.teamcode AND P.birthdate > '01/01/1985';
```

3.

```
SELECT P.num, P.drivername
FROM Driver P
WHERE 3 <= (SELECT COUNT(*) FROM Participates Pa
            WHERE P.num = Pa.num AND Pa.puntos > 5);
```

4.

```
SELECT circode, cirname
FROM Circuit
WHERE km > (SELECT AVG(km) FROM Circuit);
```

5.

```
SELECT C.circode, C.cirname
FROM Circuit C
WHERE NOT EXISTS (SELECT * FROM Race P WHERE P.circode=C.circode
                                        AND P.num_drivers <= 10)
      AND EXISTS (SELECT * FROM Race P WHERE P.circode=C.circode);
```

Another option, without using the derived attribute, is:

```
SELECT C.circode, C.cirname
FROM Circuit C
WHERE NOT EXISTS (SELECT * FROM Race P WHERE P.circode=C.circode
                  AND 10 >= (SELECT COUNT(*) FROM Participates Pa
                             WHERE Pa.year=P.year AND Pa.circode=P.circode))
      AND EXISTS (SELECT * FROM Race P WHERE P.circode=C.circode);
```

6.

```
SELECT P.num, P.drivername
FROM Driver P
WHERE 6 <= (SELECT COUNT(*) FROM Participates Pa
            WHERE P.num = Pa.num AND Pa.year = 2004);
```

7.

```
SELECT DISTINCT P.circode
FROM Race P
WHERE P.year = 1990 AND P.num_drivers = (SELECT MAX(num_drivers)
                                         FROM Race
                                         WHERE Year = 1990);
```

Another option, without using the derived attribute, is

```
SELECT P.circode
FROM Participates P
WHERE P.year = 1990
GROUP BY P.circode
HAVING COUNT(*)  >= ALL (SELECT COUNT(*) FROM Participates P
                         WHERE  P.year = 1990 GROUP BY P.circode);
```

8.

```
SELECT P.num, P.drivername
FROM Driver P, Participates Pa
WHERE P.num = Pa.num
GROUP BY P.num, P.drivername
HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM Participates Pa2
                        GROUP BY Pa2.num);
```

9.

a. The operations that must be controlled are:
- Insert into *Participates* ➔ recalculate *num_drivers*
- Delete from *Participates* ➔ recalculate *num_drivers*
- Update circode in *Participates* ➔ recalculate *num_drivers*
- Update year in *Participates* ➔ recalculate *num_drivers*
- Insert into *Race* ➔ the value of *num_drivers* must be 0 (the by-default value is not sufficient to ensure this).
- Update *num_drivers* in *Race* ➔ must be forbidden

b.

```
CREATE TRIGGER Insert_into_Participates
AFTER INSERT ON Participates
REFERENCING NEW ROW AS nueva
FOR EACH ROW
BEGIN ATOMIC
        UPDATE Race SET num_drivers= num_drivers + 1
                WHERE year=nueva.year AND circode=nueva.circode;
END
```

or, alternativaly,

```
CREATE TRIGGER Insert_into_Participates
AFTER INSERT ON Participates
REFERENCING NEW ROW AS nueva
FOR EACH ROW
BEGIN ATOMIC
        DECLARE Total AS INTEGER;
        SELECT COUNT(*) INTO Total
                FROM Participates
                WHERE year=nueva.year AND circode=nueva.circode;
        UPDATE Race SET num_drivers= Total
                WHERE year=nueva.year AND circode=nueva.circode;
END
```