**FINAL EXAM: DATABASES ("DATABASES") – 19/09/07 – SCHEMA**

Consider the following relational schema, which will be referred to as WORKING SCHEMA, which maintains information on Formula 1 races:

**TEAM** (**code**: dom1, **name**: dom3, **sponsor**: dom4, **country**: dom4, **coach**: dom2)
CP: {code}
VNN: {name}
VNN: {sponsor}
CAj: {coach} ->Technician (tcode)

**PILOT** (**pcode**: dom2, **name**: dom6, **address**: dom5, **age**: dom14, **team**: dom1, **country**: dom4)
CP: {pcode}
VNN: {name}
CAj: {team} ->Team (code)

**CAR_MODEL** (**model**: dom7, **design_year**: dom9, **team**: dom1)
CP: {model}
CAj: {team} ->Team (code)

**GRANDPRIX** (**name**: dom10, **year**: dom13, **start_date**:dom11, **end_date**:dom11, **location**: dom12)
CP: {name, year}

**TECHNICIAN:** (**tcode**: dom2, **name**: dom6, **address**: dom5, **team**: dom1, **country**: dom4)
CP: {tcode}
VNN: {name}
CAj: {team} ->Team (code) On Delete Cascade, On Update Cascade

**PARTICIPATES** (**name**: dom10, **year**: dom13, **pcode**: dom2, **classification**: dom14, **model**: dom7)
CP: {name, year, pcode}
CAj: {pcode} ->Pilot (pcode)
CAj: {name, year} ->Grandprix
CAj: {model} ->Car_model (model)

where the attributes and tables have the following meaning

**Team**
code: team identifier
name: name of the team
sponsor: main sponsor for the team
country: country of the team
coach: coach of the team

**Pilot**
pcode: code of the pilot
name: name of the pilot
address: pilot's address
age: age of the pilot
team: team in which he races
country: pilot's birth country

**Car_model**
model: car model
design_year: year in which the model is designed
team: team where the model belongs

**Grandprix**
name: name of the race
year: year when the grand prix takes place
start_date: Start date of the Grand Prix
end_date: End date of the Grand Prix
location: city where the Grand Prix takes place

**Technician**
tcode: code of the technician
name: name of the technician
address: technician's address
team: team in which he works
country: birth country of the technician

**Participates**
name: name of the race
year: year of the participation
pcode: code of the pilot who has participated
classification: final position/classification
model: car model being used by the pilot

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

| **Team** | | | | |
|---|---|---|---|---|
| code | name | sponsor | country | coach |
| Fer | Ferrari | Marlboro | Italy | D1 |
| Mac | MacLaren-Mercedes | Vodafone | United Kingdom | D2 |
| Ren | Renault | ING-direct | France | D3 |
| BMW | BMW Sauber F1 Team | Petronas | Germany | |

| **Pilot** | | | | | |
|---|---|---|---|---|---|
| pcode | name | address | age | team | country |
| P1 | Fernando Alonso | Oxford | 26 | Mac | Spain |
| P2 | Lewis Hamilton | Stevenage | 22 | Mac | United Kingdom |
| P3 | Giancarlo Fisichella | Monte Carlo | 27 | Ren | Italy |
| P4 | Heikki Kovalainen | Monte Carlo | 25 | Ren | Finland |
| P5 | Felipe Massa | Monte Carlo | 27 | Fer | Brazil |
| P6 | Kimi Raikkonen | Zúrich | 28 | Fer | Finland |

| **Car_model** | | |
|---|---|---|
| model | design_year | team |
| MP4-22 | 2006 | Mac |
| R27 | 2006 | Ren |
| F2007 | 2006 | Fer |
| F1-07 | 2006 | BMW |

| **Grandprix** | | | | |
|---|---|---|---|---|
| name | year | start_date | end_date | location |
| France GP | 2007 | 28/06/2007 | 01/07/2007 | Magny-Course |
| United Kingdom GP | 2007 | 06/07/2007 | 08/07/2007 | Silverstone |
| Europe GP | 2007 | 19/07/2007 | 22/07/2007 | Nurburgring |

| **Technician** | | | | |
|---|---|---|---|---|
| tcode | name | address | team | country |
| T1 | Mark Slade | Woking | Mac | United Kingdom |
| T2 | Phil Prew | Woking | Mac | United Kingdom |
| T3 | Aldo Costa | Maranello | Fer | Italy |
| D1 | Jean Tod | Maranello | Fer | |
| D2 | Ron Dennis | Woking | Mac | United Kingdom |
| D3 | Flavio Briatore | Monte Carlo | Ren | Italy |

| **Participates** | | | | |
|---|---|---|---|---|
| name | year | pcode | classification | model |
| France GP | 2007 | P1 | 7 | MP4-22 |
| France GP | 2007 | P6 | 1 | F2007 |
| Europe GP | 2007 | P1 | 1 | MP4-22 |
| Europe GP | 2007 | P5 | 2 | F2007 |
| Europe GP | 2007 | P2 | 9 | MP4-22 |

Blank cells represent null values.

Final exam: "databases" – 19/09/07 – 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) Choose the sentence which is FALSE:
   a) A car model can participates in a grand prix with different pilots.
   b) A pilot can belong to no team.
   c) A team can have no pilots.
   d) A pilot without team cannot participate in any grand prix.

2) Regarding physical implementation in databases, please choose the RIGHT answer:
   a) The insertion of a new record is more efficient in an ordered file than in a disordered one.
   b) If the retrieval of records is made in the order of a field is a frequent operation, then it is appropriate to use a hash file.
   c) A cluster for storing relations is appropriate if the execution of queries with group by is frequent.
   d) A hash file allows a very quick access to a record from the corresponding value of the hash field.

3) According to the working schema, which query solves the following expression in Relational Algebra?

   (Pilot ⋈ (Pilot[pcode] – (Participates WHERE classification = 1)[pcode]))[pcode, name]

   a) Code and name of the pilots which have scored once at the first position.
   b) Code and name of the pilots which have never scored at the first position.
   c) Code and name of the pilots which have always scored at the first position.
   d) Code and name of the pilots which have ever scored in a position which is not the first.

4) According to the working schema, which expression in relational algebra solves the following query?

   *"Obtain the car models which have not participated in any grand prix"*

   a) (Car_model – Participates)[model]
   b) Car_model – Participates[model]
   c) Car_model[model] – Participates[model]
   d) (Car_model – Grandprix((name, model)))[model]

5) Which constraint over the working schema will impose the following SQL instruction?

```
CREATE ASSERTION R1
CHECK (NOT EXISTS (SELECT * FROM Pilot Pi
                   WHERE NOT EXISTS (SELECT * FROM Participates Pa
                                     WHERE Pa.pcode=Pi.pcode)));
```

   a) Every pilot must participate in some grand prix.
   b) Every pilot must participate in every grand prix.
   c) No pilot can participate in any grand prix.
   d) There must be a pilot who has participated in all grand prixs.

6) Suppose that all the integrity constraints have been defined as DEFERRABLE INITIALLY IMMEDIATE and let us consider the transaction T1 which is executed over the database DB defined in the ORACLE DBMS:

```
TRANSACTION T1
    INSERT INTO Participates VALUES ('France GP', 2007, 'P1', 1, 'MP4-22');
    DELETE FROM Participates
        WHERE classification = 7 AND name = 'France GP';
    COMMIT;
END
```

Which of the following expressions is TRUE?

a) T1 will finish and will only insert the tuple ('France GP', 2007, 'P1', 1, 'MP4-22') in *Participates*, but nothing will be deleted.
b) T1 will fail, since one of the instructions violates an integrity constraint which hasn't been deferred and, either all instructions or none must be executed (atomicity property).
c) T1 will finish and will not insert the tuple ('France GP', 2007, 'P1', 1, 'MP4-22') in *Participates* but will delete all the tuples with *classification* = 7 and *name* = 'France GP'.
d) T1 will complete both instructions: it will insert the tuple ('France GP', 2007, 'P1', 1, 'MP4-22') in Participates and will delete all the tuples with classification = 7 and name = 'France GP'.

7) What will the result be after executing the following SQL instruction over the working schema?

```
CREATE ASSERTION r1
CHECK (NOT EXISTS
    (SELECT * FROM Grandprix G
    WHERE EXISTS
        (SELECT * FROM Participates P1, Participates P2
        WHERE G.name = P1.name AND G.year = P1.year AND P1.classification = 1 AND
            G.name = P2.name AND G.year = P2.year AND P2.classification = 1)));
```

a) An integrity constraint would be added to force every grand prix to have a winner.
b) An integrity constraint would be added to avoid two winners in a grand prix.
c) Nothing would happen since this constraint is already expressed by the constraints in the schema.
d) An integrity constraint would be added to avoid having a winner in any grand prix.

8) If we define a foreign key in *Participates* to *Grandprix* with a directive of "ON NULL DELETE", what would happen in the DB if we set the *year* to null at the "France GP" row in the table *Grandprix*?

a) The attribute *year* would be set to null at the "France GP" row in the table *Grandprix* and also in the rows of *Participates* which make reference to that row in *Grandprix*.
b) The row in *Participates* would be deleted and the attribute *year* would be set to null at the "France GP" row in the table *Grandprix*.
c) This operation cannot be done independently from the type of directive we define.
d) The row in *Grandprix* with the "France GP" would be deleted and the *year* would be set to null in the rows of *Participates* which make reference to that tuple of *Grandprix*.

9) Suppose that all the integrity constraints have been defined as DEFERRABLE INITIALLY IMMEDIATE and consider the transaction T1 which is executed over the database DB:

```
TRANSACTION T1
    SET ALL DEFERRED;
    INSERT INTO Team VALUES (‘UPV’, 'El Poli', 'Restaurant El Famós', 'Spain', 'JSB');
    INSERT INTO Technician VALUES (‘JSB’, 'Joseba Sededa Tos', 'Valencia', 'UPV', 'Spain');
    COMMIT;
END.
```

Which of the following expressions is TRUE?

a) T1 will finish and only the team will be inserted.
b) T1 will fail, since one of the instructions violates an integrity constraint.
c) T1 will finish and only the technician will be inserted.
d) T1 will complete both instructions and will finish correctly.

10) If for the foreign keys in *Team* to *Technician* and from *Technician* to *Team* we define the directive "ON DELETE CASCADE" (for both of them). What will happen if we execute the following instruction over the DB?

```
DELETE FROM TEAM WHERE code = 'BMW';
```

a) It won’t be allowed.
b) The *Team* 'BMW' will be deleted.
c) The *Team* 'BMW' will be deleted and its coach from the relation *Technician*.
d) The *Team* 'BMW' will be deleted, ant its car models and its coach from the relation *Technician*.

11) How can we define in ORACLE DBMS the integrity constraint “the age of a pilot cannot decrease”?

a) With a table constraint (a CHECK constraint for the attribute *age*).
b) With a trigger.
c) With the instruction CREATE ASSERTION as in standard SQL.
d) We cannot define integrity constraints in ORACLE.

12) Which of the following tools is used by the DBMS to ensure transaction atomicity and persistence?

a) Module for integrity checking.
b) Log file.
c) Module for trigger execution.
d) Hard disk.

13) A DBMS offers logical independence if:

a) It offers different implementation for the data structures of the underlying data model.
b) It allows the definition of external schemas.
c) The programs which access the database are independent from changes which are performed over the implementation of the structures in the physical schema.
d) The external schemas are not affected by modifications of the logical schema relative to data which they do not use.

14) What would happen if a DBMS does not use checkpoints for handling transactions and database recovery in front of failures?

a) The DBMS wouldn’t be able to ensure a correct behaviour.
b) The DBMS could only ensure a correct behaviour in non-concurrent environments.
c) The behaviour would be correct, but each database reconstruction from main memory failures would be very costly.
d) The behaviour would be correct, but each database reconstruction from secondary memory failures would be very costly.

**FINAL EXAM: DATABASES – 19/09/07 – Problems**

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

1) Obtain the code and the name of the pilots who have no team and they have not participated in any grand prix (**0.5 points**)

2) Obtain the code and the name of the youngest pilots who have participated in a grand prix. **(0.75 points)**.

3) Obtain the code and the name of the teams with the greatest number of pilots **(0.75 points)**.

4) Obtain the code and the name of all the pilots, also indicating how many different car models they have used in their participations in grand prixs. **(0.75 points)**.
5) Obtain the code and the name of the teams such that all their car models have the same design year. (We're only interested in teams which have at least one model) **(0.75 points)**.

6) Obtain the name, year and location of the grand prixs in which all the pilots who participate use a car of their team **(1 point)**.

7) Obtain the code and the name of the teams also showing how many pilots they have which have participated in more than two grand prixs (We're only interested in teams which have at least one time which complies with the condition).**(1 point)**.

8) Given the following integrity constraint: "*A pilot cannot participate in a grand prix with a car model which does not belong to his team"*
   a. Apart from the insertion into *Participates*, please enumerate other four operations which may violate the constraint**. (0.5 points)**
   b. Write a trigger to handle the operation of "Insertion into *Participates*". **(0.5 points)**

**SOLUTIONS TO THE QUESTIONNAIRE:**

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

**SOLUTIONS TO THE PROBLEMS**:

1) (**0.5 points**)

```
SELECT pcode, name
FROM Pilot
WHERE team IS NULL AND pcode NOT IN (SELECT pcode FROM Participates);
```

2) (**0.75 points**)

```
SELECT pcode, name
FROM Pilot
WHERE pcode IN (SELECT pcode FROM Participates) AND
            age = (SELECT MIN(age)
                   FROM Pilot
                   WHERE pcode IN (SELECT pcode FROM Participates));
```

3) (**0.75 points**)

```
SELECT code, name
FROM Team
WHERE code IN (SELECT team
               FROM Pilot
               GROUP BY team
               HAVING COUNT(*) = (SELECT MAX(COUNT(*))
                                  FROM Pilot
                                  GROUP BY team))
```

4) (**0.75 points**)

```
SELECT PI.pcode, PI.name, COUNT(DISTINCT PA. model)
FROM Pilot PI LEFT JOIN Participates PA ON PI.pcode=PA.pcode
GROUP BY PI.pcode, PI.name
```

5) (**0.75 points**)

```
SELECT code, name
FROM Team E
WHERE EXISTS (SELECT * FROM Car_model  C
              WHERE E.code = C.team AND
                    NOT EXISTS (SELECT * FROM Car_model  C1
                                WHERE E.code = C1.team AND
                                      C.design_year<> C1.design_year))
```

or

```
SELECT E.code, E.name
FROM Team E, Car_model  C
WHERE E.code = C.team
GROUP BY E.code, E.name
HAVING COUNT(DISTINCT C.design_year)=1
```

6) (**1 point**)

```
SELECT name, year, location
FROM Grandprix    GP
WHERE NOT EXISTS (SELECT *
                  FROM Participates PA, Pilot PI, Car_model C
                  WHERE PA.pcode=PI.pcode AND PA.model=C.model AND
                        PA.name=GP.name AND PA.year=GP.year AND
                        PI.team<>C.team)
```

7) (**1 point**)

```
SELECT E.code, E.name, COUNT(*)
FROM Team E, Pilot PI
WHERE E.code=PI.team AND  PI.pcode IN (SELECT PA.pcode FROM Participates PA
                                       GROUP BY pcode
                                       HAVING COUNT(*) >2)
GROUP BY E.code, E.name
```

or

```
SELECT E.code, E.name, COUNT(*)
FROM Team E, Pilot PI
WHERE E.code=PI.team AND
      2 < (SELECT COUNT(*) FROM Participates PA
           WHERE PI.pcode = PA.pcode)
GROUP BY E.code, E.name
```

8a) (**0.5 points**)

1. Update *team* on *Pilot*
2. Update *team*  on *Car_Model*
3. Update *model* on *Participates*
4. Update *pcode* on *Participates*

8b) (**0.5 points**)

```
CREATE TRIGGER insert_in_participates
    BEFORE INSERT ON Participates
    FOR EACH ROW
    WHEN (new.model is not null)
    DECLARE E1, E2 CHAR(20)
    BEGIN
      SELECT team INTO E1
      FROM Pilot WHERE pcode=:new.pcode;
      SELECT team INTO E2
      FROM Car_Model WHERE model=:new.model;
      IF E1<>E2
      THEN
          RAISE_APPLICATION_ERROR(-20000, 'A pilot cannot participate in a grand prix
                                           with a car model which is not from his team');
    END IF
    END
```