

The relational schema below, which will be referred to as the WORKING SCHEMA, keeps information of the Nobel Prizes.

**CATEGORY** (*cat\_id*: varchar(1), *description*: varchar(30))

PK: {cat\_id}

NNV: {description}

**EDITION** (*cat\_id*: varchar(1), *year*: integer, *contribution*: varchar(80))

PK: {cat\_id, year}

FK: {cat\_id} → CATEGORY

**NOMINEE** (*cod*: integer, *name*: varchar(60), *country\_cod*: varchar(4), *birth\_year*: integer)

PK: {cod}

NNV: {name}

FK: {country\_cod} → COUNTRY

**AWARDED** (*cat\_id*: varchar(1), *year*: integer, *cod*: integer)

PK: {cat\_id, year, cod}

FK: {cat\_id, year} → EDITION      ON DELETE AND UPDATE RESTRICTIVE  
FULL REFERENCIAL INTEGRITY

FK: {cod} → NOMINEE      ON DELETE AND UPDATE RESTRICTIVE

**JUDGE** (*judge\_id*: integer, *name*: varchar(60))

PK: {judge\_id}

NNV: {name}

**COMMITTEE** (*cat\_id*: varchar(1), *year*: integer, *judge\_id*: integer, *function*: varchar(20))

PK: {cat\_id, year, judge\_id}

FK: {cat\_id, year} → EDITION      ON DELETE AND UPDATE RESTRICTIVE

FK: {judge\_id} → JUDGE      ON DELETE AND UPDATE CASCADE

**COUNTRY** (*country\_cod*: varchar(4), *name*: varchar(20))

PK: {country\_cod}

Where the relations have the following meaning:

**Category:** Type of prize. Identifier and description of each category. The description can be "Physics", "Chemistry", "Physiology or Medicine", "Peace", "Economics", or "Literature".

**Edition:** Stores the annual call for each category and the contribution for which the prize is awarded.

**Nominee:** people who have been nominated for the Nobel. A code is stored for each of them, along with their name, nationality and year of birth.

**Awarded:** Nominees who have been awarded the Nobel Prize in each of the editions.

**Judge:** people who can be part of one of the committees that select the winners of each edition.

**Committee:** stores the people who are part of each selection committee for each edition, indicating the function they occupy.

**Country:** Country code and name of the country.

CATEGORY	
cat_id	description
F	Physics
Q	Chemistry
M	Physiology or Medicine
L	Literature
E	Economy
P	Peace

EDITION		
cat_id	year	contribution
L	2007	Female experience
P	2007	Climate Change
F	1921	Photoelectric effect
Q	1911	Polonium and Radium Discovery
F	1903	Radiation phenomena

NOMINEE			
cod	name	country_cod	birth_year
1	Doris Lessing	UK	1919
2	Albert Einstein	GER	1879
3	Roger B. Myerson	USA	1951
4	Albert A. Gore	USA	1948
5	Marie Curie	FR	1867
6	Pierre Curie	FR	1859
7	Antoine H. Becquerel	FR	1852

AWARDED		
cat_id	year	cod
P	2007	4
L	2007	1
F	1921	2
Q	1911	5
F	1903	5
F	1903	6
F	1903	7

JUDGE	
judge_id	name
1	Berge Ragnar Furre
2	Ole Danbolt Mjos
3	Hans Jörnvall
4	Gunnar von Heijne

COMMITTEE			
cat_id	year	judge_id	function
P	2007	1	
P	2007	2	President
M	2001	3	Secretary
Q	2007	4	President

COUNTRY	
country_cod	name
USA	USA
UK	United Kingdom
GER	Germany
FR	France
ES	Spain

This questionnaire consists of 10 questions; for each one four answers are proposed, of which **only one** is correct. The answer should be included in the answer sheet provided separately. The maximum score for the questionnaire is 1 point. The score obtained is calculated with the formula **(Correct\_answers – Wrong answers/3) × 0,1**.

1. Given the working schema, which of the following statements is **FALSE**?
  - a) A nominated person may be awarded in several categories in different years.
  - b) A nominated person cannot be awarded for several categories in the same year.
  - c) Two nominees may be awarded for the same category in the same year.
  - d) One nominated person can be awarded in two categories.
  
2. Consider the database shown in the previous page. If we delete a tuple in *Judge* where judge\_id =1
  - a) The deletion will be done without any problem and without affecting any other tuple of any other relation.
  - b) The deletion will be done without any problem and the following tuples will also be deleted: {(cat\_id, 'P'), (year, 2007), (judge\_id, 1), (function, null)} and {(cat\_id, 'P'), (year, 2007), (judge\_id, 2), function, 'Chairman')} of the Committee relation.
  - c) It could not be done by violating a restriction on integrity.
  - d) The deletion would be done without any problem and the tuple {(cat\_id, 'P'), (year, 2007), (judge\_id, 1), (function, null)} of the Committee relationship would also be deleted.
  
3. Consider the database shown before, what happens if in the Awarded relationship the value year=2007 is changed to year=1911 in the tuple {(cat\_id, 'P'), (year, 2007), (cod, 4)}?
  - a) The change will be made without problems because the referential integrity is satisfied.
  - b) The change will not be made because the referential integrity is violated.
  - c) The change will be made and will be propagated in cascade to the Edition relation.
  - d) The change will be made and will be propagated in cascade to the Nominee relation.
  
4. Consider the database shown before, what happens if the following transaction is executed in a DBMS that guaranties the **atomicity** of the transactions?
 

```
BEGIN
      SET CONSTRAINT ALL DEFERRED;
      UPDATE Awarded SET cod = 21 WHERE cod = 1
      UPDATE Nominee SET cod = 21 WHERE cod = 1;
      COMMIT;
```

  - a) These updates cannot be performed because the foreign key in Awarded {cod} -> NOMINEE is defined as ON UPDATE RESTRICTIVE.
  - b) If the foreign key {cod} -> NOMINEE of Awarded relation was defined as INITIALLY IMMEDIATE the two tuples would be updated without problems.
  - c) The first instruction fails, but the second is executed.
  - d) If the foreign key {cod} -> NOMINEE of Awarded was defined as DEFERRABLE the two tuples would be modified without problems.
  
5. Given the working schema, which of the following statements is **TRUE**?
  - a) One person can be awarded and be a member of the selection committee in the same edition.
  - b) A country can only have one nominee.
  - c) A country can only have one winner.
  - d) A person can be awarded without having been nominated.

6. The three types of referential integrity (Weak, Partial and Complete) are equivalent...
- only when the foreign key has only one attribute.
  - only when all the attributes of the foreign key have a non-null value constraint.
  - when the foreign key has only one attribute or all its attributes have a non-null value constraint.
  - they are never equivalent.
7. If, on the working scheme, you want to add a constraint to ensure that no two countries have the same name, which of the following would you change in the scheme of the Country relation exclusively in order to ensure such restriction?
- Add PK: {name}.
  - Add Unique: {name}.
  - Add Unique: {name} and NNV: {name}.
  - Delete PK: {country\_cod} and add PK: {name}.
8. Given the working schema, what expression of Relational Algebra would solve the query "Name of the nominees who have received exactly one award"?
- $((\text{AWARDED}[\text{cod}] - (\text{AWARDED} \bowtie \text{AWARDED}((\text{cat\_id}, X), (\text{year}, Y)) \text{ WHERE } \text{cat\_id} \neq X \vee \text{year} \neq Y [\text{cod}]))) \bowtie \text{NOMINEE}) [\text{name}]$
  - $((\text{NOMINEE}[\text{cod}] - (\text{AWARDED} \bowtie \text{AWARDED}((\text{cat\_id}, X), (\text{year}, Y)) \text{ WHERE } \text{cat\_id} \neq X \vee \text{year} \neq Y [\text{cod}]))) \bowtie \text{NOMINEE}) [\text{name}]$
  - $((\text{AWARDED} \bowtie \text{AWARDED}((\text{cat\_id}, X), (\text{year}, Y)) \text{ WHERE } \text{cat\_id} = X \wedge \text{year} = Y [\text{cod}]) \bowtie \text{NOMINEE}) [\text{name}]$
  - $(\text{AWARDED} \bowtie \text{AWARDED}((\text{cat\_id}, X), (\text{year}, Y)) \text{ WHERE } \text{cat\_id} \neq X \vee \text{year} \neq Y [\text{cod}]) \bowtie \text{NOMINEE}) [\text{name}]$
9. The sentence GRANT... of SQL is...
- an instruction to remove permission to access objects in a database.
  - an instruction that allows to grant permission to access objects in a database.
  - an instruction that allows removing permission to access objects in a database and that can only be executed by the database administrator.
  - an instruction that allows granting permission to access objects in a database and that can only be executed by the database administrator.
10. Consider the definition of the Edition relation of the scheme, indicate which of the following definitions of the primary key is correct
- cat\_id PRIMARY KEY, year PRIMARY KEY DEFERRABLE
  - PRIMARY KEY (cat\_id CHAR(1) NOT NULL, year INTEGER NOT NULL)
  - CONSTRAINT PK\_Edition PRIMARY KEY (cat\_id, year) DEFERRABLE
  - CONSTRAINT PK\_Edition PRIMARY KEY (cat\_id, year) INITIALLY DEFERRED NOT DEFERRABLE

1) Solve the following queries in SQL:

- 1) Obtain the id and name of the judges who have participated in more than one committee **(0.25 points)**.
- 2) Obtain the code and name of the nominees who have not been awarded **(0.25 points)**.
- 3) Obtain the code and name of the nominees who have been awarded more than once, indicating the number of times they have been awarded **(0.5 points)**.
- 4) Obtain the code and name of the country that has the highest number of awards **(0.5 points)**.
- 5) Obtain the id and name of the judges who have been members of a committee in two consecutive editions with a known function **(0.5 points)**.
- 6) Obtain the code and name of the countries that have nominees and all those nominees have been awarded **(1 point)**.
- 7) For all editions with more than five judges on the committee, obtain the number of winners and the category and year of the edition **(1 point)**.

## ANSWERS

### P1

	A
1	B
2	D
3	B
4	D
5	A
6	C
7	B
8	A
9	B
10	C

### P2

1.

```
SELECT judge_id, name
FROM Judge J
WHERE 1 < (SELECT COUNT(*)
           FROM Committee C
           WHERE C.judge_id = J.judge_id) ;
```

2.

```
SELECT cod, name
FROM Nominee
WHERE cod NOT IN (SELECT cod FROM Awarded) ;
```

*Also*

```
SELECT N.cod, N.name
FROM Nominee N
WHERE NOT EXISTS (SELECT * FROM Awarded A WHERE N.cod = A.cod) ;
```

3.

```
SELECT A.cod, N.name, COUNT(*)
FROM Awarded A, Nominee N
WHERE A.cod = N.cod
GROUP BY A.cod, N.name
HAVING COUNT(*) > 1 ;
```

4.

```
SELECT C.country_cod, C.name
FROM Country C, Nominee N, Awarded A
WHERE C.country_cod = N.country_cod
      AND N.cod = A.cod
GROUP BY C.country_cod, C.name
HAVING COUNT(*) = (SELECT MAX(COUNT(*))
                   FROM Nominee N2, Awarded A2
                   WHERE N2.cod = A2.cod
                     AND N2.country IS NOT NULL
                   GROUP BY N2.country_cod) ;
```

5.

```
SELECT DISTINCT J.judge_id, J.name
FROM Judge J, Committee c1, Committee c2
WHERE J.judge_id = c1.judge_id
      AND J.judge_id = c2.judge_id
      AND c2.year = c1.year + 1
      AND c1.function IS NOT NULL
      AND c2.function IS NOT NULL ;
```

*Also*

```
SELECT DISTINCT J.judge_id, j.name
FROM Judge J, Committee C1
WHERE J.judge_id = C1.judge_id
      AND C1.function IS NOT NULL
      AND EXISTS (SELECT *
                  FROM Committee C2
                  WHERE C2.judge_id = J.judge_id
                    AND C2.funtion IS NOT NULL
                    AND C2.year = C1.year + 1) ;
```

6.

```
SELECT DISTINCT C.country_cod, C.name
FROM Country C, Nominee N
WHERE C.country_cod = N.country_cod
      AND NOT EXISTS (SELECT *
                      FROM Nominee N1
                      WHERE N1.country_cod = C.country_cod
                        AND N1.cod NOT IN (SELECT cod FROM Awarded)) ;
```

*Also*

```

SELECT C.country_cod, C.name
FROM Country C
WHERE NOT EXISTS (SELECT *
                  FROM Nominee N
                  WHERE N.country_cod = C.country_cod
                  AND NOT EXISTS (SELECT *
                                FROM Awarded A2
                                WHERE N.cod = A2.cod))

AND EXISTS (SELECT *
            FROM Nominee N
            WHERE N.country_cod = C.country_cod) ;

```

7.

```

SELECT E.cat_id, E.year, COUNT(A.cod)
FROM Edition E LEFT JOIN Awarded A ON A.cat_id=E.cat_id AND A.year=E.year
WHERE 5 < (SELECT COUNT(*)
          FROM Committee C
          WHERE C.cat_id = E.cat_id
          AND C.year = E.year)

GROUP BY E.cat_id, E.year ;

```

*Also*

```

SELECT E.cat_id, E.year, COUNT(A.cod)
FROM Edition E LEFT JOIN Awarded A ON A.cat_id=E.cat_id AND A.year=E.year
WHERE (E.cat_id, E.year) IN (SELECT cat_id, year
                            FROM Committee
                            GROUP BY cat_id, year
                            HAVING COUNT(*) >5)

GROUP BY E.cat_id, E.year

```