

Consider the following relational schema, that will be referred to as WORKING SCHEMA, which maintains information about a massive casting call:

Actor(act_code: d1, name: d2, age: d3)

PK: {act_code}

NNV: {name}

Panel_member (mem_code: d1, name: d2, speciality: d8)

PK: {mem_code}

NNV: {name}

Role(role_code: d3, description: d4, duration: d5)

PK: {role_code}

NNV: {description, duration}

Performance(role_code: d3, act_code: d1, date:d10)

PK: {role_code, act_code}

NNV: {date}

FK: {role_code} → Role ON DELETE CASCADE, ON UPDATE CASCADE

FK: {act_code} → Actor ON UPDATE CASCADE

Scoring_form(form_code: d6, score: d7, role_code: d3, act_code: d1, pmember: d1)

PK:{form_code}

NNV: {pmember}

FK: { pmember } → Panel_member f(pmember)= mem_code
ON UPDATE CASCADE

FK: {role_code, act_code} → Performance
PARTIAL referential integrity
ON DELETE CASCADE, ON UPDATE CASCADE

Where the relation **Performance** has the following meaning: “The actor with code *act_code* has played the role *role_code* on day *date*”

The relation **Scoring_form** represents that “There is a form with code *form_code*, filled in by the *pmember* with code *pmember*, with a *score*, about the performance of actor *act_code* as role *role_code*”.

Consider the following extension of the previous schema. We will refer to this extension as database (DB). **Null values are represented by ‘?’**

Actor			Panel_member			Role		
act_code	name	age	mem_code	name	speciality	role_code	description	duration
456	Luisa	18	222	Marta	critic	1	Girl	20
678	María	21	333	Pablo	director	2	Boy	17
123	Juan	32	444	Aurora	?	4	Friend	3

Performance			Scoring_form				
role_code	act_code	date	form_code	score	role_code	act_code	pmember
1	456	3-9-18	6	6	?	123	444
4	456	3-9-18	2	5	4	456	333
2	123	3-9-18	4	6	2	123	444
			5	8	1	456	222

Circle the correct answer for each question.

This test penalizes students' incorrect answers with *negative points* (1/3) to discourage guessing.

1. Consider the foreign key in the relation Scoring_form
FK: { pmember } \rightarrow Panel_member f(pmember)= mem_code
The only operations that may violate the referential integrity are:
 - a) Insert a tuple or update the primary key of Scoring_form, and delete a tuple or update the primary key of Panel_member.
 - b) Delete a tuple or update pmember in Scoring_form, and Insert a tuple or update the mem_code in Panel_member.
 - c) Insert a tuple or update the pmember in the Scoring_form relation, and delete a tuple or update the mem_code in the Panel_member relation.
 - d) None of the above is true
2. Which statement is TRUE?
 - a) UNI: {a,b} is equivalent to UNI: {a} AND UNI: {b}
 - b) UNI: {a,b} is equivalent to UNI: {a} OR UNI: {b}
 - c) UNI: {a,b} is equivalent to PK: {a,b}
 - d) None of the above
3. How can we include the following constraint: "No score in the Scoring_form table can be greater than 10"?
 - a) By adding an attribute constraint to the Score attribute.
 - b) By adding a table constraint into the Scoring_form table.
 - c) By adding a static integrity constraint into the Scoring_form table.
 - d) All the above are true.
4. Which statement referring to the working schema is TRUE?
 - a) There cannot be two Panel members with the same name.
 - b) All actors have at least one performance.
 - c) It is possible to have a scoring_form with no score.
 - d) All the Scoring forms have a role_code.
5. Complete the content of the tables after the execution of the following sentence
DELETE FROM Role WHERE role_code=2

Answers:

- 1.- c
- 2.- d
- 3.- d
- 4.- c
- 5.-

Actor			Panel_member			Role		
act_code	name	age	mem_code	name	speciality	role_code	description	duration
456	Luisa	18	222	Marta	critic	1	Girl	20
678	María	21	333	Pablo	director	2	Boy	17
123	Juan	32	444	Aurora	?	4	Friend	3

Performance			Scoring_form				
role_code	act_code	date	form_code	score	role_code	act_code	pmember
1	456	3-9-18	6	6	2	123	444
4	456	3-9-18	2	5	4	456	333
2	123	3-9-18	4	6	2	123	444
			5	8	1	456	222