Ö.3

Sunday, 10 September 2017 18:21

Uppgifter:

- 7.1.1[1] abc,
  - For 7.1.1b the text just ends mysteriously but should say ...' offending Movie tuple
- 7.1.5 abc,
- 7.2.1 c,
- 7.3.2 ab

7.1.1:

Movies(title, year, length, genre, studioName, producerC#)
MovieExec(name, address, cert#, netWorth)

A) ***The producer of a movie must be someone mentioned in MovieExec. Modifications to MovieExec that violate this constraint are rejected.***

- By default rejected

ALT1: Add foreign key declaration separate

```
ALTER TABLE Movies(
        producerC#,                    [cert# is here a primary key]
        FOREIGN KEY producerC# REFERENCES MovieExec(cert#)
);
```

ALT2: Declaring producerC# directly to reference #cert

```
producerC# INT REFERENCES MovieExec(cert#)
```

B) ***Repeat (A), but violations result in the deletion or update of the offending Movie tuple.***

- Just add the Cascade Policy - changes to the ref attributes are mimicked at the foreign key
  - ALT2: ..

```
        producerC# INT REFERENCES MovieExec(cert#)
        ON UPDATE CASCADE
```

ON UPDATE CASCADE
ON DELETE CASCADE

C) ***Repeat (A), but violations result in the producerC# in Movies being set to NULL. Offending Movies tuple.***

- Just add the Set-Null Policy - when a modification to the ref relation affects a foreign-key value, the latter is changed to NULL.
  - ALT2:..

    producerC# INT REFERENCES MovieExec(cert#)
    ON UPDATE SET NULL
    ON DELETE SET NULL

7.1.5:

Write following referential integrity constraints for the battleships DB (exc. 7.1.3) Use your assumptions about keys from that exercise, and handle all vilations by setting the ref attribute value to NULL.

Classes(class, type, country, numGuns, bore, displacement)
Ships(name, class, launched)
Battles(name, date)
Outcomes(ship, battle, result)

A) ***Every battle mentioned in Outcomes must be mentioned in Battles.***
  - Assuming all battles have unique values - prim key

ALTER TABLE Outcomes ADD CONSTRAINT battleOutcomesBattles FOREIGN KEY (battle) REFERENCES Battles(name);

B) ***Every ship mentioned in Outcomes, must be mentioned in Ships.***

ALTER TABLE Outcomes ADD CONSTRAINT sameShips FOREIGN KEY (ship) REFERENCES Ships(name);

C) *****Every class mentioned in Ships must be mentioned in Classes.***

ALTER TABLE Ships ADD CONSTRAINT sameClass FOREIGN KEY (class) REFERENCES Classes(class);

7.2.1)

Write the following constraints on the relation:
Movies(title, year, length, genre, studioName, producerC#)

C) The genre can only be drama, comedy, sciFi or teen.

- ALTER TABLE Movies
  ADD CONSTRAINT validGenres CHECK(genre IN ('drama', 'comedy', 'sciFi', 'teen'));

7.3.2)

Show how to alter the schemas of the "battleships" DB,
Classes(class, type, country, numGuns, bore, displacement)
Ships(name, class, launched)
Battles(name, date)
Outcomes(ship, battle, result)
, to have the following tuple-based constraints:

A) Class and country form a key for relation Classes.

ALTER TABLE Classes ADD CONSTRAINT newKey PRIMARY KEY(class, country);

B) Require the referential integrity constraint that every battle appearing in Outcomes also appears in Battles:

ALTER TABLE Outcomes
ADD CONSTRAINT matchingBattles FOREIGN KEY battle REFERENCES Battles(name);