

Compito di Basi di dati

22 gennaio 2020

Esercizio 1:

Sia dato il seguente schema relazionale relativo a film e attori:

FILM(*CodiceFilm*, *Titolo*, *Regista*, *Anno*);

ATTORI(*CodiceAttore*, *Nome*, *Cognome*, *Sesso*, *DataNascita*, *Nazionalità*);

INTERPRETAZIONE(*Film*, *Attore*, *Ruolo*).

Si assuma che ogni film sia identificato univocamente da un codice e sia caratterizzato da titolo, regista e anno di uscita. Per semplicità, si assuma che ogni film sia diretto da un unico regista e ogni regista sia identificato univocamente dal suo cognome. Si ammetta la possibilità che vi siano film diversi con lo stesso titolo (questo è il caso, ad esempio, dei remake), ma si escluda la possibilità che due film con lo stesso titolo escano lo stesso anno. Si assuma che la base di dati non contenga film privi di attori (ad esempio, film di animazione), ossia che in ogni film reciti almeno un attore.

Ogni attore sia identificato univocamente da un codice e sia caratterizzato da nome, cognome, sesso, data di nascita e nazionalità. Si assuma che più attori possano recitare in un dato film e che un attore possa recitare in più film. Infine, si assuma che, in ogni film, un attore possa svolgere più di un ruolo.

Definire preliminarmente le chiavi primarie, le eventuali altre chiavi candidate e, se ve ne sono, le chiavi esterne delle relazioni date. Successivamente, formulare opportune interrogazioni in algebra relazionale che permettano di determinare (senza usare l'operatore di divisione e usando solo se necessario le funzioni aggregate):

- gli attori che non hanno recitato in alcun film di Ken Loach;
- per ogni regista, il film (i film se più di uno) da lui/lei diretto col minor numero di attori;
- gli attori che hanno recitato in tutti i film di Steven Spielberg e in almeno un film di un altro regista.

Esercizio 2:

Con riferimento all'Esercizio 1, formulare opportune interrogazioni in SQL che permettano di determinare quanto richiesto (senza usare l'operatore CONTAINS e usando solo se e quando necessario le funzioni aggregate).

Esercizio 3:

Si vuole realizzare una base di dati per la gestione di informazioni cinematografiche relative a film, registi e attori sulla base del seguente insieme di requisiti.

- Si assuma che sia i registi che gli attori siano identificati univocamente da nome e cognome, e siano caratterizzati dalla loro data di nascita e dalla loro nazionalità. Si assuma anche che l'insieme dei registi e l'insieme degli attori non siano necessariamente disgiunti. Si tenga traccia delle relazioni moglie/marito e genitore/figlio sull'insieme unione dell'insieme dei registi e dell'insieme degli attori (col termine figlio indichiamo genericamente figli di sesso maschile e figli di sesso femminile).
- Ogni film sia caratterizzato da titolo, anno di uscita, nazione in cui è stato prodotto, produttore, regista e attori. L'insieme dei film possa includere film d'animazione, privi di attori. Si assuma che ogni film venga prodotto in un'unica nazione, da un unico produttore, e abbia un unico regista. Non si escluda la possibilità che vi siano film diversi con lo stesso titolo (questo è il caso, ad esempio, dei remake), ma si escluda la possibilità che due film con lo stesso titolo escano lo stesso anno. Si tenga traccia delle nazioni (in generale, più di una) dove un film è stato girato.
- Infine, si tenga traccia del numero di film prodotti ogni anno in ogni nazione.

Si definisca uno schema Entità-Relazioni che descriva il contenuto informativo del sistema, illustrando con chiarezza le eventuali assunzioni fatte. Lo schema dovrà essere completato con attributi ragionevoli per ciascuna entità (identificando le possibili chiavi) e relazione. Vanno specificati accuratamente i vincoli di cardinalità e partecipazione di ciascuna relazione. Si definiscano anche eventuali regole di gestione (regole di derivazione e vincoli di integrità) necessarie per codificare alcuni dei requisiti attesi del sistema.

Esercizio 4:

Siano dati i seguenti schedule:

- $s_1 : r_2(x), r_1(y), w_1(y), r_2(y), w_1(x), w_2(y)$;
- $s_2 : r_1(x), r_1(y), w_2(z), w_1(z), r_2(x), w_3(x), w_3(z)$;
- $s_3 : r_1(x), w_2(z), w_1(z), w_3(z), w_3(y), w_1(y)$.

Si stabilisca se appartengono o meno a VSR, CSR, 2PL, 2PL stretto e TS. Nel caso di schedule appartenenti a VSR, si forniscano tutti gli schedule seriali ad esso equivalenti; lo stesso nel caso di schedule appartenenti a CSR.

Esercizio 5:

Si consideri il seguente schema relazionale:

MUSICISTA(*nickname*, *eta*, *strumento*)

- PK : *nickname*
- vincolo NOT NULL: *eta*, *strumento*

GENERE(*gid*, *nome*)

- PK : *gid*

PIACE(*mid*, *gid*)

- PK : (*mid*, *gid*)
- FK : *mid*- > *MUSICISTA.nickname*
- FK : *gid*- > *GENERE.gid*

SUONA_CON(*mid1*, *mid2*)

- PK : (*mid1*, *mid2*)
- FK : *mid1*- > *MUSICISTA.nickname*
- FK : *mid2*- > *MUSICISTA.nickname*

1. Scrivere il codice SQL corrispondente allo schema relazionale dato.
2. Si consideri il vincolo: “nessun musicista può suonare con un altro musicista a cui piace l’Indie”.
 - 2.1 Elencare quali operazioni e su quali tabelle possono violare questo vincolo.
 - 2.2 Si scelga almeno una delle operazioni individuate al punto precedente e si scriva il codice SQL di un trigger che eviti la/e violazione/i provocata/e da tale operazione.