

Esercizio 1:

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

FILM(CodiceFilm, Titolo, Regista, Anno);

ATTORE(CodiceAttore, Cognome, Nome, Sesso, DataNascita, Nazionalità);

INTERPRETAZIONE(Film, Attore, Personaggio)

Ogni film sia identificato univocamente da un codice e sia caratterizzato da un titolo, da un regista e dall'anno in cui è uscito. Per semplicità, si assuma che ogni film sia diretto da un unico regista e che un regista sia identificato univocamente dal suo Cognome. Non si escluda la possibilità che film diversi abbiano lo stesso titolo (è questo il caso, ad esempio, dei remake). Ogni attore sia identificato univocamente da un codice e sia caratterizzato da un nome, un cognome, un sesso, una data di nascita e una nazionalità. Più attori possano recitare in uno stesso film e un attore possa recitare in più film. Per semplicità, si assuma che in un film un attore possa interpretare un solo personaggio.

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 hanno recitato in tutti i film diretti dal regista Antonioni;
- gli attori che hanno recitato in al più due film diretti dal regista Allen;
- le coppie di attori tali che esista un film in cui il primo ha recitato e il secondo no, e viceversa.

$$a. \text{ FILM_ANTO} \leftarrow \pi_{\text{FILM}} \left(\sigma_{\text{REGISTA} = \text{ANTONINI}} \left(\text{FILM} \bowtie \text{INTERPRETAZIONE} \right) \right)$$

FILM = CODICEFILM

$$\text{REQUISITI}(\text{FILM}, \text{ATTORE}) \leftarrow \text{FILM_ANTO} \times \pi_{\text{CODICEATTORE}}(\text{ATTORE})$$

$$\text{STATO_DI_FATTO} \leftarrow \pi_{\text{FILM}, \text{ATTORE}}(\text{INTERPRETAZIONE})$$

$$\text{NO_GOOD} \leftarrow \text{REQUISITI} - \text{STATO_DI_FATTO}$$

$$S \leftarrow \pi_{\text{CODICEATTORE}}(\text{ATTORE}) - \pi_{\text{CODICEATTORE}}(\text{NO_GOOD})$$

AL PIÙ DUE = TUTTI - ALMENO 3

$$b. \text{ FILM_ATT_ALLEN} \leftarrow \pi_{\text{FILM}, \text{ATTORE}} \left(\sigma_{\text{REGISTA} = \text{ALLEN}} \left(\text{FILM} \bowtie \text{INTERPRETAZIONE} \right) \right)$$

codiceFilm = Film

$$\text{FILM_ATT_ALLEN_1}(\text{FILM1}, \text{ATTORE1}) \leftarrow \text{FILM_ATT_ALLEN}$$

$$\text{FILM_ATT_ALLEN_2}(\text{FILM2}, \text{ATTORE2}) \leftarrow \text{FILM_ATT_ALLEN}$$

$$\text{NO_GOOD} \leftarrow \pi_{\text{ATTORE}} \left(\sigma_{\text{FILM} < \text{FILM1} \text{ AND } \left(\text{FILM_ATT_ALLEN} \bowtie_{\text{ATTORE} = \text{ATTORE1}} \text{FILM_ATT_ALLEN_1} \right) \bowtie_{\text{ATTORE} = \text{ATTORE2}} \text{FILM_ATT_ALLEN_2} } \right)$$

$$S \leftarrow \pi_{\text{CODICEATTORE}}(\text{ATTORE}) - \text{NO_GOOD}$$

$$c. \text{ SX} \leftarrow \pi_{\text{ATTORE}, \text{ATTORE1}} \left(\sigma_{\text{ATTORE} < \text{ATTORE1}} \left(\text{INT} \times \text{INT1} \right) \Rightarrow \text{quelli dove l'attore a sx ha recitato e dx no} \right)$$

AND FILM <> FILM1

$$\text{DX} \leftarrow \pi_{\text{ATTORE}, \text{ATTORE1}} \left(\sigma_{\text{ATTORE} > \text{ATTORE1}} \left(\text{INT} \times \text{INT1} \right) \Rightarrow \text{quelli dove l'attore a dx ha recitato e sx no} \right)$$

AND FILM <> FILM1

$$\text{ATTORI_FILM_DIVERSI} \leftarrow \text{SX} \cup \text{DX}$$

$$\text{TUTTI_ATTORI}(\text{ATTORE}, \text{ATTORE1}) \leftarrow \sigma_{\text{ATTORE} < \text{ATTORE1}} \left(\pi_{\text{CODICEATTORE}}(\text{ATTORE}) \times \pi_{\text{CODICEATTORE1}}(\text{ATTORE1}) \right)$$

$$\text{NO_GOOD} \leftarrow \text{TUTTI_ATTORI} - \text{ATTORI_FILM_DIVERSI}$$

A1	F1	A1	F1
A1	F2	A1	F2
A1	F3	A1	F3
A2	F1	A2	F1
A2	F2	A2	F2
A2	F4	A2	F4
A3	F1	A3	F1

~~ATTORI_STESSI_FILM $\leftarrow \sigma_{\text{ATTORE} <> \text{ATTORE1} \text{ AND FILM} = \text{FILM1}} (\text{INT} \times \text{INT1})$~~

~~STATO_DI_FATTO $\leftarrow \sigma_{\text{ATTORE} <> \text{ATTORE1}} (\text{INT} \times \text{INT1})$~~

~~ATTORI_DIVERSI $\leftarrow \text{STATO_DI_FATTO} - \text{ATTORI_STESSI_FILM}$~~

A1 A2

A1 A3

A2 A1

A2 A3

A3 A1

A3 A2

C. $\text{INTERPRETAZ_VERA} \leftarrow \pi_{\text{FILM}, \text{ATTORE}} (\text{INTERPRETAZ})$

$\text{NON_INTERPRETAZ} \leftarrow (\pi_{\text{CODICEFILM}(\text{FILM}) \times \pi_{\text{CODICEATTORE}(\text{ATTORE})}) - \text{FILM_ATTORE}$

$\text{A1_A2_NO}(\text{FILM}, \text{ATTORE}, \text{NATTORE}) \leftarrow \sigma_{\text{FILM} = \text{CODICEFILM}} (\text{INTERPRETAZ_VERA} \times \text{NON_INTERPRETAZ})$

$\text{A1_NO_A2}(\text{FILM2}, \text{ATTORE2}, \text{NATTORE2}) \leftarrow \text{A1_A2_NO}$

$S \leftarrow \sigma_{\text{ATTORE} < \text{ATTORE2}} (\pi_{\text{ATTORE}, \text{ATTORE2}} (\sigma_{\text{ATTORE} = \text{NATTORE2} \text{ AND NATTORE} = \text{ATTORE2}} (\text{A1_A2_NO} \times \text{A1_NO_A2})))$

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).

a.

```
SELECT CODICEATTORE
FROM ATTORE A
WHERE NOT EXIST ( SELECT *
                  FROM INTERPRETAZIONE, FILM F
                  WHERE CODICEFILM = F.FILM AND
                        REGISTA = ANTONIONI AND
                        NOT EXIST ( SELECT *
                                  FROM INTERPRETAZIONE
                                  WHERE ATTORE = A.ATTORE
                                  AND FILM = F.FILM
                                )
                  )
```

b.

```
CREATE VIEW FILM_ALLEN (FILM, ATTORE) AS
SELECT FILM, ATTORE
FROM INTERPRETAZIONE, FILM
WHERE FILM = CODICEFILM AND
      REGISTA = ALLEN
```

```
SELECT CODICEATTORE
FROM ATTORE A
WHERE NOT EXIST ( SELECT *
                  FROM FILM_ALLEN I1 I2 I3
                  WHERE I1.FILM < I2.FILM AND
                        I2.FILM < I3.FILM AND
                        A.CODICEATTORE = I1.ATTORE AND
                        A.CODICEATTORE = I2.ATTORE AND
                        A.CODICEATTORE = I3.ATTORE AND
                  )
```

```

C. SELECT I1.ATTORE AND I2.ATTORE
FROM INTERPRETAZIONE I1 I2
WHERE I1.ATTORE < I2.ATTORE AND
      I1.ATTORE NOT IN (SELECT ATTORE
                        FROM INTERPRETAZIONE
                        WHERE FILM = I2.FILM
                        )
      AND I2.ATTORE NOT IN (SELECT ATTORE
                            FROM INTERPRETAZIONE
                            WHERE FILM = I1.FILM
                            )

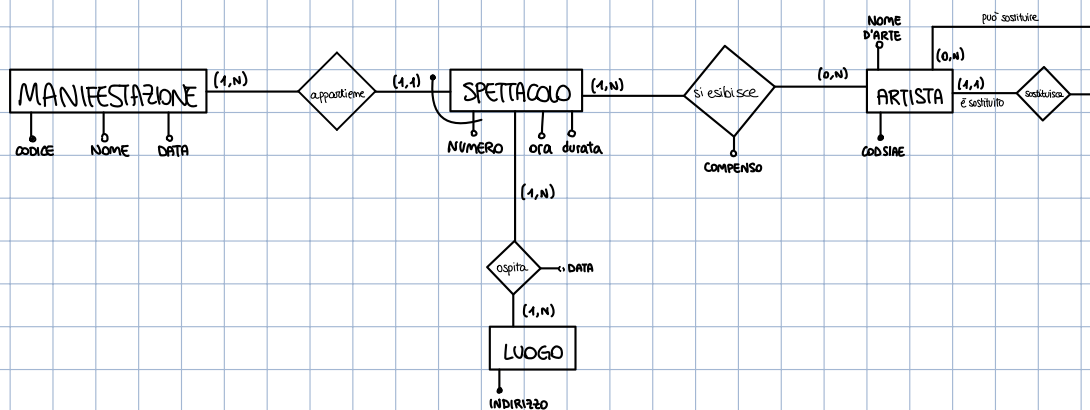
```

Esercizio 3:

Si vuole realizzare una base di dati per gestire le manifestazioni artistiche offerte durante l'estate da una data città sulla base del seguente insieme di requisiti.

- Una manifestazione, identificata univocamente da un codice, è caratterizzata da un nome e da una data, e consiste di uno o più spettacoli.
- Ogni spettacolo è identificato da un numero, univoco all'interno della manifestazione alla quale appartiene, ed è caratterizzato dall'ora di inizio e dalla durata.
- Durante uno spettacolo, si esibiscono uno o più artisti (un artista si può esibire al massimo una volta durante lo stesso spettacolo), ricevendo un certo compenso. Uno stesso artista si può esibire in più spettacoli di una data manifestazione. Un artista è identificato univocamente dal codice SIAE e è caratterizzato da un nome d'arte. Per ogni artista, si deve indicare necessariamente un altro artista che lo sostituisca in caso di indisponibilità; un'artista può essere indicato come sostituto di più artisti.
- Per ospitare gli spettacoli vengono adibiti opportuni luoghi. In una certa data, un luogo può ospitare al massimo tre diversi spettacoli, sia della stessa manifestazione che di manifestazioni differenti.

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.



VINCOLI AZIENDALI: in OSPITA, per una certa DATA al massimo 3 spettacoli diversi

Esercizio 4:

Si svolgono i seguenti due punti.

1. Con riferimento allo standard ANSI/ISO SQL-92, si illustrino le differenze fra i due livelli di isolamento *Read Committed* e *Repeatable Read*, per quanto riguarda le anomalie permesse o meno da essi.
2. Siano T_1 e T_2 due transazioni sottomesse simultaneamente al sistema. Le istruzioni che le compongono vengono eseguite secondo il seguente ordine temporale:

T_1	T_2
start transaction;	
select stipendio from Dipendente where id = 'D03';	
	start transaction;
	update stipendio from Dipendente set stipendio = stipendio - 100 where id = 'D03';
	commit;
select stipendio from Dipendente where id = 'D03';	
commit;	

Assumendo che il valore dell'attributo *stipendio* per la tupla con *id* = 'D03' sia inizialmente pari a 2000 si spieghi, giustificando la risposta, quali risultati possono produrre le due *select* di T_1 nei diversi livelli di isolamento previsti dallo standard ANSI/ISO SQL-92.

1. A differenza del *Read Committed*, il *Repeatable Read* non soffre dell'anomalia della lettura inconsistente

2. fino a *Read Committed* legge 1900 perché

READ con LOCK e RILASCIO
IMMEDIATO

select stipendio from Dipendente
where id = 'D03'

```
start transaction;
update stipendio from Dipendente
set stipendio = stipendio - 100
where id = 'D03';
commit;
```

invece con *SERIALIZABLE* e *REPEATABLE READ* legge 2000 perché c'è LOCK 2PL stretto in lettura e quindi una volta che T_1 acquisisce in lettura, non rilascia la risorsa fino al commit, perciò T_2 non fa l'update finché T_1 non fa commit

Esercizio 5:

Si consideri un file contenente 100.000.000 record di dimensione prefissata pari a 500 byte, memorizzati in blocchi di dimensione pari a 4096 byte in modo unspanned. La dimensione del campo chiave primaria V sia 14 byte; la dimensione del puntatore a blocco P sia 6 byte. Si chiede di confrontare fra loro le seguenti soluzioni, in termini di numero medio di accessi a blocco e di dimensione dell'indice.

- (a) Ricerca basata su un indice multilivello statico ottenuto a partire da un indice secondario costruito sul campo chiave primaria V .
- (b) Ricerca basata su un B -albero, con campo di ricerca il campo chiave primaria V , puntatore ai dati di dimensione pari a 7 byte e puntatore ausiliario di dimensione pari a 6 byte, assumendo che ciascun nodo del B -albero sia pieno al 70%.
- (c) Ricerca basata su un B^+ -albero, con campo di ricerca il campo chiave primaria V , puntatore ai dati di dimensione pari a 7 byte e puntatore ausiliario di dimensione pari a 6 byte, assumendo che ciascun nodo del B^+ -albero sia pieno al 70%.

M = 100.000.000 record

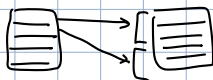
dim = 500 byte

B = 4096 byte

V = 14 byte

P = 6 byte

a. indice secondario: $bfr_i = \left\lfloor \frac{4096}{20} \right\rfloor = 204 \text{ record}$



$$mb_i = \left\lceil \frac{100.000.000}{204} \right\rceil = 490197 \text{ blocchi}$$

indice multilivello: $fd = bfr_i = 204$

2° livello: $\left\lceil \frac{490197}{204} \right\rceil = 2403$

ho 4 livelli \rightarrow 5 accessi

3° livello: $\left\lceil \frac{2403}{204} \right\rceil = 12$

4° livello: $\left\lceil \frac{12}{204} \right\rceil = 1$

b. V = 14 byte

$P_R = 7$ byte

P = 6 byte

$\sigma = 0,7$

$$p \cdot P + (p-1)(V + P_R) \leq B$$

$$6p + 21p - 21 \leq 4096 \Rightarrow p = 152$$

$$p_{\text{effettivo}} = 152 \cdot 0,7 = 106$$

livello	modi	dati	puntatori
Radice	1	105	106
1	106	11130	11236
2	11236	1179780	1191016
3	1191016	125056680	

\Rightarrow ho 4 livelli + 1 accesso \Rightarrow 5 accessi

dim. indice = 1202350
 \hookrightarrow \neq modi

c. con B+ albero

$$p \cdot P + (p-1)(V) \leq B \rightarrow 20p - 14 \leq 4096 \Rightarrow p = 205 \Rightarrow p = 143$$

$$p_{\text{LEAF}}(P_R + V) + P \leq B \rightarrow 21p_{\text{LEAF}} + 6 \leq 4096 \Rightarrow p_{\text{LEAF}} = 143 \Rightarrow p_{\text{LEAF}} = 136$$

livello	modi	dati	puntatori
Radice	1	142	143
1	143	20306	20449
2	20449	2903758	2924207
3	2924208	397692288	

\nearrow
 $\times p_{\text{LEAF}}!$

ho 4 livelli \Rightarrow 5 accessi

dim. indice = 2944801