

Correzione del 12/07/2021

1. Dato il seguente schema logico relazionale¹

codificare le seguenti richieste in linguaggio SQL.

- ```
a) CREATE TABLE pubblicazioni(
 codice NUMERIC NOT NULL PRIMARY KEY,
 titoli VARCHAR(200) NOT NULL,
 data_stampa DATE NOT NULL,
 autore VARCHAR(100) NOT NULL REFERENCES autori(nome)
 editore NUMERIC NOT NULL REFERENCES editori(codice));

b) ALTER TABLE librerie
 ADD CONSTRAINT nome_citta_uk
 UNIQUE(nome,citta)

c) SELECT lib.citta SUM (v.copie_vendute)
 FROM librerie lib, vendite v,
 WHERE v.libreria = lib.codice
 AND v.data BETWEEN '2020-01-01' AND '2020-12-31';
 -- EXTRACT (GROUP BY lib.citta);

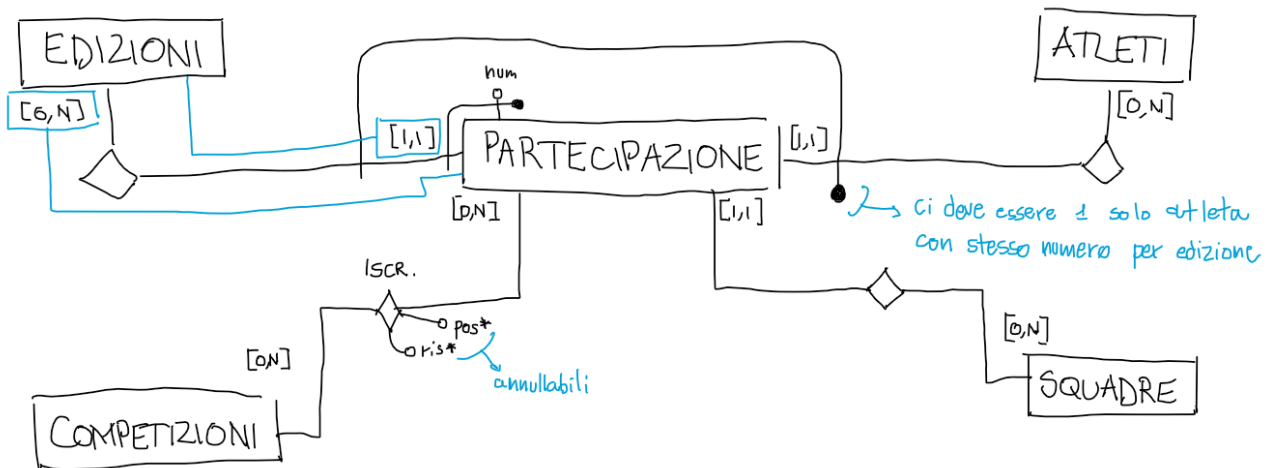
d) SELECT e.nome, COUNT(a.nome) AS numero_autori_in_vita
 FROM editori e, autori a, pubblicazioni p,
 WHERE e.codice=p.editore
 AND a.nome=p.autore
 AND a.morte IS NULL
 AND p.data >= '2000-01-01'
 GROUP BY e.nome
 ORDER BY e.nome;

e) DELETE FROM editori
 WHERE codice NOT IN(SELECT editore FROM pubblicazioni);
 -- WHERE NOT EXISTS(SELECT * FROM pubblicazioni p, editori e
 WHERE p.editorie=e.codice);
```

2. Mostrare lo schema concettuale Entità-Relazione per un database che codifica informazioni relative a varie edizioni di una manifestazione sportiva. Si richiede di modellare le informazioni seguenti.

- Ogni *edizione* della manifestazione sportiva è identificata dall'anno di svolgimento; si tiene anche traccia delle date di inizio e fine dell'edizione.
- Gli *atleti* sono identificati da un codice; per essi si registrano nome, cognome e data di nascita.
- Le *competizioni* sportive sono identificate da un codice e caratterizzate da un nome, anche esso univoco; si considerano solo competizioni individuali.
- Le *squadre*, identificate da un codice, sono dotate di un nome, anche esso univoco.
- Per ogni *edizione* della manifestazione, si registrano le informazioni seguenti.
  - La *partecipazione* degli atleti a quell'edizione: ogni atleta può partecipare a più edizioni, ma al più una volta per ogni edizione; la partecipazione è identificata dall'edizione e da un numero progressivo (il numero di pettorina dell'atleta in quella edizione); ogni partecipazione è associata ad una sola squadra (un atleta può far parte di squadre diverse, ma in edizioni diverse).
  - Ogni partecipante ad una edizione può *isciversi* ad una o più competizioni distinte; un atleta che partecipa a più edizioni può iscriversi più volte alla stessa competizione.
  - Per ogni iscrizione si possono registrare il *risultato*, cioè una misura della prestazione ottenuta, e la *posizione* in classifica; entrambi sono opzionali.

Tradurre lo schema ER in uno schema logico relazionale, codificando opportunamente i vincoli dello schema.



```

Edizioni (annopk, inizio, fine)
Atleti (codicepk, nome, cognome, nascita)
Squadre (codicepk, nomeuk)
Competizioni (codicepk, nomeuk)
Partecipazione ([edizionefk, num]pk, [edizionefk(edizioni), atletafk(atleti)]uk, squadrafk)
Iscrizioni (competizionefk(competizioni), [edizionefk, numfk](partecipazioni)pk, ris*, pos*)
--associazione molti a molti va codificata
--gli attributi associati sono automaticamente PRIMARY KEY e FOREIGN KEY

```

3. Rispondere brevemente alle seguenti domande.

- (a) Data la relazione  $R(X)$ , sotto quali condizioni l'operatore di proiezione  $\pi_Y(R)$  è ben definito? Sotto quali condizioni tale espressione dell'algebra relazionale è equivalente all'istruzione SQL `select Y from R`?
  - (b) Spiegare brevemente in cosa consistono le anomalie dette *lettura sporca* e *lettura non ripetibile*.
  - (c) Spiegare brevemente come si costruisce il grafo dei conflitti a partire da uno schedule di un insieme di transazioni; per cosa può essere utile tale grafo?
- 
- a) `proj_Y(R)`      $Y$  è un sottoinsieme di  $X$  -> per dire quando è definito  
in SQL la eliminazione dei duplicati non c'è in automatico, cosa che invece è presente  
nell'algebra relazionale -> quando  $Y$  è una superchiave di  $R$
  - b) anomalia delle transazioni non correttamente serializzate,  
-> (sporca) la transazione ha letto un dato da un'altra transazione che non ha ancora fatto commit (abort) (viol. isolamento)  
-> (nonrip) rilettura del dato ma il valore cambia perché un'altra transazione ha fatto commit (viol. isolamento)
  - c) v. serializzazione in base ai conflitti