

# Compito di Basi di dati

22 gennaio 2019

## Esercizio 1:

Sia dato il seguente schema relazionale relativo ai prodotti offerti da un dato fornitore ai suoi clienti:

*PRODOTTO*(*Cod\_prodotto*, *Cod\_prodotto*, *Tipologia*, *Prezzo\_Unitario*);

*CLIENTE*(*Cod\_cliente*, *Telefono*, *Email*, *Settore*);

*FORNISCE*(*Cod\_cliente*, *Cod\_prodotto*).

Si assuma che ogni prodotto sia identificato univocamente da un codice e sia caratterizzato dal produttore, dalla tipologia e dal prezzo unitario. Si assuma che un produttore possa produrre più prodotti e che il fornitore possa offrire più prodotti della stessa tipologia. Si assuma che ogni cliente sia identificato univocamente da un codice e sia caratterizzato da un recapito telefonico, un recapito di posta elettronica e un settore di attività. Si assuma, infine, che il fornitore possa fornire più prodotti ad uno stesso cliente e che uno stesso prodotto possa essere fornito a più clienti. Non si escluda la possibilità che vi siano dei prodotti che non vengono richiesti da alcun cliente e dei clienti ai quali non viene fornito alcun prodotto.

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):

- i clienti a cui vengono forniti solo prodotti di un'unica tipologia;
- il cliente (i clienti se più di uno) a cui viene fornito il maggior numero di prodotti;
- le coppie (produttore, cliente) tali che il fornitore fornisca a quel cliente tutti e soli i prodotti di quel produttore.

## 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 consideri il seguente schema relazionale:

*autore*(*codice\_fiscale*, *nome*, *cognome*)

*libro*(*codice\_isbn*, *titolo*, *anno*)

*ha\_scritto*(*codice\_fiscale\_autore*, *codice\_isbn\_libro*)

CE: *ha\_scritto.codice\_fiscale\_autore* → *autore*

CE: *ha\_scritto.codice\_isbn\_libro* → *libro*

Tenendo presente che ogni libro ha almeno un autore e che, in generale, può avere più autori:

- si elenchino le operazioni sulla base di dati che possono comportare una violazione di tale vincolo di integrità;
- si fornisca del codice SQL che permetta di mantenere tale vincolo tramite trigger, limitatamente ad una delle operazioni precedentemente individuate.

#### Esercizio 4:

Si vuole realizzare una base di dati per gestire la stagione concertistica di una data città sulla base del seguente insieme di requisiti.

- La stagione concertistica si articola in una serie di concerti. Ogni concerto è identificato univocamente da un codice. Ogni concerto è contraddistinto da un titolo e da una descrizione. Si assuma che vi possano essere più concerti col medesimo titolo. Ogni concerto comprende un certo insieme di pezzi musicali, in generale di autori diversi.
- Ogni pezzo musicale ha un codice, un titolo e uno o più autori, ciascuno caratterizzato da un codice univoco, un nome, un cognome e una data di nascita. Lo stesso pezzo può essere eseguito in più concerti.
- Ogni concerto è eseguito da un'orchestra. Ogni orchestra ha un nome, che la identifica univocamente, un direttore e un insieme di orchestrali. Ogni direttore è caratterizzato da un codice fiscale, un nome, un cognome e una data di nascita, e può dirigere più orchestre. Ogni orchestrale ha un codice fiscale, un nome e un cognome, suona uno o più strumenti e può far parte di più orchestre. Un orchestrale non può dirigere un'orchestra e un direttore non può partecipare ad un'orchestra come orchestrale.
- Ogni concerto è tenuto in una sala di un teatro, in una certa data. Ogni teatro è identificato univocamente dal suo nome e contiene una o più sale. Ogni sala è identificata univocamente dal suo nome all'interno di un determinato teatro e ha una determinata capienza. Non si esclude la possibilità che vi siano sale con lo stesso nome in teatri diversi.

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 5:

Si stabilisca se i seguenti schedule appartengono o meno a 2PL, 2PL stretto, TS, CSR e VSR.

1.  $s_1 : r_3(z), r_1(x), w_4(z), r_4(y), w_2(x), r_2(x), r_3(y), w_1(x), w_4(y);$
2.  $s_2 : r_2(z), w_1(t), w_3(z), r_2(x), w_4(t), r_1(y), w_2(z), w_3(y), r_1(x), w_4(z), w_2(x);$
3.  $s_3 : r_4(y), w_1(x), r_1(y), w_3(t), r_2(t), w_2(x), r_2(y), w_4(y), r_1(z), w_4(x), r_4(t), w_3(z).$

[FACOLTATIVO] Si dimostri che CSR è contenuto in VSR.