

Compito di Basi di dati

20 febbraio 2023

Esercizio 1:

Sia dato il seguente schema di una base di dati relazionale che registra informazioni relative ad un insieme di porti e di corse che li collegano:

Porto(*Città*, *Nazione*);

Banchina(*Numero*, *Porto*, *Lunghezza*);

Nave(*codNave*, *TipoNave*, *AnnoCostruzione*);

TipoNave(*NomeTipo*, *NumeroPasseggeri*, *QuantitàMerci*);

CorsaGiornaliera(*IdCorsa*, *GiornoSettimana*, *CittàPartenza*, *OraPartenza*, *CittàArrivo*, *OraArrivo*, *Nave*);

Si assuma che in ogni città sia presente (al più) un porto e che in ogni porto vi siano una o più banchine. Ogni banchina di un dato porto sia identificata da un numero (banchina numero 1 del porto di Trieste, banchina numero 2 del porto di Trieste e così via). Di ogni banchina si registri la lunghezza. Si assuma che ogni nave sia identificata da un codice e sia caratterizzata dal tipo e dall'anno di costruzione. Di ogni tipo di nave vengano specificati il numero di passeggeri e la quantità di merci che è in grado di trasportare. Infine, si assuma che ogni corsa sia identificata univocamente da un codice e sia caratterizzata dal giorno del settimana in cui si svolge, dalla città e dall'ora in cui inizia e da quelle in cui termina e dalla nave utilizzata (si assuma che, per una data corsa, venga utilizzata sempre la stessa nave).

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 SQL che permettano di determinare (senza usare l'operatore CONTAINS e usando solo se e quando necessario le funzioni aggregate):

- (a) il porto (i porti, se più d'uno) col minor numero di banchine;
- (b) le navi tali che ogni porto in cui arrivano è raggiunto solo da esse.

Formulare un'interrogazione in algebra relazionale che consenta di determinare i porti con una o due banchine, senza usare l'operatore di divisione e usando le funzioni aggregate solo se necessario.

Esercizio 2:

Sia dato il seguente insieme di requisiti relativi ad una base di dati per una palestra che offre un insieme di corsi ai propri iscritti (clienti della palestra).

- La base di dati dovrà tener traccia sia degli iscritti che degli istruttori. Degli uni e degli altri si vogliono registrare il codice fiscale, il nome, il cognome e un recapito telefonico. Di ogni iscritto si vuole conoscere anche l'età.
- Ogni corso è identificato univocamente dal proprio nome, ha un determinato prezzo e un numero massimo di partecipanti. Ogni corso è gestito da uno o due istruttori.
- Ogni iscritto può acquistare una o più tessere. Ogni tessera comprende un certo numero di corsi che, attraverso di essa, vengono acquistati ad un prezzo complessivo, che può variare da tessera a tessera, ma non può essere superiore alla somma dei prezzi dei singoli corsi.
- Ogni cliente della palestra può anche iscriversi direttamente a un corso, senza dover acquistare una tessera comprendente quel corso.
- Ogni tessera è identificata da un numero progressivo, univoco per ogni cliente. Tale numero può, però, ripetersi per due clienti 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.

Esercizio 3:

Si scriva il codice SQL per creare le tabelle **R** ed **S** sotto riportate, tenendo in considerazione l'eventuale opportuno ordine. Per quanto riguarda la prima tabella, si assuma che l'attributo **X** sia composto da una stringa di al più 5 caratteri alfanumerici e sia la chiave primaria, mentre l'attributo **Y** sia un numero decimale non-nullo, strettamente positivo. Per quanto riguarda la seconda tabella, si assuma che **W** sia un numero intero, chiave primaria della tabella, e che **Z** non possa contenere valori ripetuti, ma possa assumere il "valore" NULL, e sia una chiave esterna che fa riferimento alla chiave primaria di **R**.

Table 1: R	
<u>X</u>	Y
aa111	1.6
bb222	9.2
cc333	5.0

Table 2: S	
<u>W</u>	Z
1	aa111
2	bb222
3	NULL

Nel contesto delle transazioni concorrenti SQL, si specifichi in forma tabellare, per ogni livello di isolamento previsto dalla standard ANSI/ISO SQL-92, quali sono le anomalie ammesse.

Successivamente, si descriva l'anomalia di lettura inconsistente e, in riferimento alla tabella **R**, usando delle istruzioni SQL, si fornisca un esempio di schedule con due transazioni concorrenti nel quale si manifesti tale anomalia.

Esercizio 4:

Si illustrino le principali differenze tra B -alberi e B^+ -alberi.

Successivamente, dato l'elenco di chiavi:

2, 3, 4, 9, 6, 8, 10, 11, 12, 17, 16, 14

mostrare il B -albero di ordine $p = 4$ ottenuto inserendo un elemento dopo l'altro nell'ordine specificato, riportando la sequenza di alberi generata dal processo di inserimento.