Corso di Basi di Dati 1

Secondo Compitino

21 Dicembre 2001

Si consideri lo schema relazionale

CLIENTE(NumCC, Nome, Via, Città, Telefono, TipoCC, ScadCC)
ALBERGO(TelefonoA, Nome, Via, Città, Categoria)
CAMERE(TelefonoA, TipoC, Num, Prezzo)
PRENOTAZIONE(NumCC, TelefonoA, TipoC, DataA, DataP, Num)

relativo ad una base di dati di prenotazioni alberghiere. Gli attributi NumCC, TipoCC, ScadCC rappresentano, rispettivamente, numero, tipo e data di scadenza della carta di credito di un cliente (che si suppone unica). L'attributo TipoC rappresenta la tipologia di camera (esempi di possibili valori sono: doppia/singola/tripla/suite). Nella relazione PRENOTAZIONE gli attributi DataA e DataP rappresentano la data di arrivo e la data di partenza, mentre l'attributo Num rappresenta il numero di camere di quella tipologia prenotate.

1. (Indicativamente, punti 4) Specificare le clausole `PRIMARY KEY` e `FOREIGN KEY` del comando SQL per la creazione della tabella PRENOTAZIONE, tenendo presente che:
   (a) gli attributi NumCC, TelefonoA, TipoC, DataA sono chiave primaria;
   (b) gli attributi NumCC e (TelefonoA, TipoC) sono chiavi esterne, e che si vogliono i seguenti comportamenti in caso di violazioni dell'integritá referenziale:
   - se si cancella un cliente si vogliono cancellare anche le sue prenotazioni;
   - si vogliono propagare eventuali modifiche al numero di carta di credito dei clienti;
   - non si possono cancellare nè modificare tipologie di camere per cui vi sono prenotazioni;
2. (Indicativamente, punti 5) Definire l'asserzione corrispondente al vincolo di integrità che nello stesso albergo una camera doppia non può costare meno di una singola.
3. Formulare in SQL le seguenti interrogazioni:
   (a) (indicativamente, punti 5) determinare nome e indirizzo dell'albergo di Firenze di seconda categoria in cui le camere doppie hanno prezzo minimo;
   (b) (indicativamente, punti 5) determinare gli alberghi di Firenze in cui hanno effettuato prenotazioni sia clienti di Milano che clienti di Roma, ma nessun cliente di Genova;
   (c) (indicativamente, punti 7) determinare gli alberghi di Firenze che hanno prenotazioni per l'8 Dicembre 2001 da almeno due clienti differenti, in cui il prezzo delle camere doppie è inferiore al prezzo medio delle camere doppie in alberghi di Firenze della stessa categoria;
   (d) (indicativamente, punti 7) per ogni prenotazione che comprende tipologie di camere differenti, determinare il nome del cliente, l'indirizzo dell'albergo, la data di arrivo, il numero totale di camere prenotate, i giorni di permanenza, l'importo totale dovuto.

- CREATE TABLE R
  (spec_col$_1$, ..., spec_col$_n$)
  [, PRIMARY KEY (*Lista Nomi Colonne*)]
  [, FOREIGN KEY (*Lista Nomi Colonne*)
  REFERENCES R'
  [ON DELETE {NO ACTION | RESTRICT | CASCADE |
  SET NULL | SET DEFAULT}]
  [ON UPDATE {NO ACTION | RESTRICT | CASCADE |
  SET NULL | SET DEFAULT}]]
  [, FOREIGN KEY ...]
  [CHECK *Condizione*];
  ...
  [CHECK *Condizione*];
  );

  spec_col$_i$ = *Nome Colonna$_i$ Dominio$_i$* [DEFAULT *Valore Default*] [NOT NULL] [PRIMARY KEY] [UNIQUE]
  [CHECK *Condizione*], $\quad 1 \leq i \leq n$

  CREATE ASSERTION *Nome Vincolo*
  CHECK *Condizione*

- DROP TABLE R
- ALTER TABLE R ADD COLUMN spec_col;
  ALTER TABLE R ALTER COLUMN spec_col;
  ALTER TABLE R DROP COLUMN nome_colonna;
- SELECT $R_{i_1}.C_1, R_{i_2}.C_2, \ldots, R_{i_n}.C_n$
  FROM $R_1, R_2, \ldots, R_k$
  WHERE F;
  - C BETWEEN v_1 AND v_2
  - C IN $(v_1, \ldots, v_n)$ e C IN sq
  - C LIKE pattern, con pattern stringa di caratteri che può contenere i caratteri speciali % e_
  - EXISTS sq, NOT EXISTS sq
  - ANY sq, ALL sq
  - UNION, INTERSECT, EXCEPT tra query
  - funzioni di gruppo: MAX, MIN, SUM, AVG, COUNT - per le prime quattro è $f(nome_colonna)$, per COUNT ci sono le due varianti COUNT(*) e COUNT(DISTINCT nome_colonna)
  - l'ordine delle clausole (se presenti) è SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY
- INSERT INTO R [$(C_1, \ldots, C_n)$]
  { VALUES $(e_1, \ldots, e_n)$ | sq}
- DELETE R [alias] [WHERE F];
- UPDATE R [alias]
  SET C_1 = {e_1 | NULL }, ..., C_n = {e_n | NULL }
  [WHERE F];
- CREATE VIEW *Nome* [(*Lista Nomi Colonne*)] AS Q
  WITH [[LOCAL | CASCADED] CHECK OPTION];