Primo compitino di Basi di Dati; Evoluzioni Recenti
14 Aprile 2003

Esercizio 1
Si consideri il seguente schema relazionale, relativo ai voli effettuati da Milano Malpensa:

VOLI(CodiceVolo, Data, Compagnia, Destinazione, Ora_partenza, Cod_Aereo, Miglia)
IMBARCHI(Volo, Passeggero)
PASSEGGERI(Matricola, Cognome, Nome, Indirizzo, Nazione, Miglia_acc)
ESIGENZE(Passeggero, Esigenza)
AEREI(Cod_Aereo, Tipo, Anno_costruzione)

La tabella Esigenze contiene per ogni passeggero le esigenze particolari dello stesso (ad esempio, se e' vegetariano). Ogni esigenza e' una stringa. I passeggeri possono avere una tessera in cui accreditare le miglia corrispondenti ad ogni volo. Il campo Miglia_acc rappresenta il totale miglia accumulato.

Si richiede di:

1. Progettare uno schema concettuale ad oggetti equivalente allo schema relazionale precedente, motivando le scelte effettuate.
2. Tradurre lo schema concettuale in uno schema logico relazionale ad oggetti per Oracle, motivando le scelte effettuate.
3. Presentare gli statement di creazione della parte di schema logico relativa all'entita' VOLO utilizzando il sistema dei tipi di Oracle.
4. Supponendo che le tabelle contengano un certo numero di tuple, presentare gli statement per inserire nella base di dati un nuovo volo Alitalia, con codice AZ507 e destinazione Parigi, in partenza alle ore 10:00 del 14/3/03. Per il volo si utilizza l'aereo con codice X45Y. Il volo permette inoltre di accumulare 600 Miglia. Le matricole dei passeggeri di tale volo solo le seguenti: AB23,CD25,GT87.
5. Presentare gli statement SQL per le seguenti interrogazioni:
   - Determinare gli anni di costruzione degli aerei utilizzati per voli con destinazione Parigi.
   - Determinare il nome e il cognome dei passeggeri vegetariani sul volo AZ507.
   - Per ogni tipo di aereo, determinare il numero di voli che utilizzano aerei di quel tipo, sono gia' stati effettuati e non hanno avuto a bordo passeggeri italiani.
6. Si supponga che i passeggeri vengano distinti in passeggeri con carta di accumulo miglia e passeggeri senza carta. Si supponga inoltre di volere mantenere informazioni sulle miglia accumulate solo per i passeggeri con la carta. Si discuta come si dovrebbe cambiare lo schema logico per tenere traccia di questa situazione, in SQL-99 e in Oracle.

Esercizio 2
In riferimento allo schema relazionale proposto nell'esercizio 1, si progettino i trigger in SQL-99 per ottenere i seguenti comportamenti (si richiede di progettare ALMENO un trigger di tipo FOR EACH STATEMENT):

1. Mantenere aggiornato il campo Miglia_acc della tabella passeggeri, a fronte di inserimenti e cancellazioni dalla tabella IMBARCHI.
2. Reagire agli inserimenti nella tabella IMBARCHI, impedendo al passeggero con codice ZW65 di imbarcarsi sul volo AZ507, evitando di fare il rollback della transazione.
3. Reagire all'inserimento di ogni nuovo passeggero di nazionalita' Indiana, inserendo nelle esigenze per quel passeggero l'informazione relativa al fatto che non puo' mangiare carne di mucca.

Tipi in Oracle

CREATE TYPE Nome AS OBJECT
(Dichiarazioni_attributi)
[FINAL | NOT FINAL];

CREATE TYPE Nome UNDER Nome_superclasse
(Dichiarazioni_attributi)
[FINAL | NOT FINAL];

Default: NOT FINAL

CREATE TYPE Nome AS TABLE OF Tipo;

CREATE TYPE Nome AS VARRAY(Numero) OF Tipo;

CREATE TABLE Nome OF Nome_Tipo
[({Vincoli chiave primaria, chiave esterna, not null, su campi nel tipo |
SCOPE FOR NomeCampoREF is NomeTabella})]
[NESTED TABLE NomeCampoNested STORE AS NomeTabella];

CREATE TABLE Nome
(Dichiarazioni_attributi)
[NESTED TABLE NomeCampoNested STORE AS NomeTabellaStorage];

con

- NomeCampoNested: nome del campo della tabella di tipo TABLE.
- NomeTabellaStorage: nome tabella utilizzata per la memorizzazione della tabella nested.
- I tipi utilizzati in Dichiarazioni_attributi possono essere:
  - Tipi atomici
  - Tipi definiti dall'utente
  - REF Nome [SCOPE IS Nome_tabella]
    La clausola SCOPE IS puo' essere inserita solo negli statement di CREATE TABLE.
    La clausola puo' anche essere aggiunta ai campi delle tabelle tipate (SCOPE FOR).

Funzioni OR in Oracle

- ref( ): dato un oggetto di un certo tipo, restituisce l'identificatore per quell'oggetto
- deref( ): dato un identificatore, restituisce l'oggetto puntato
- value( ): prende un alias di relazione e restituisce l'oggetto tupla associato (utilizzando il costruttore opportuno)
- TABLE( ): data una valori di tipo collezione, restituisce la tabella corrispondente al contenuto di tale valore.

Trigger SQL-99

CREATE TRIGGER Nome
{BEFORE|AFTER} Evento ON Relazione
[REFERENCING {OLD AS Var | NEW AS Var | OLD TABLE AS Var | NEW TABLE AS Var}]
[FOR EACH {ROW | STATEMENT}]
[WHEN Condizione]
Comandi SQL

con

- Evento: INSERT, DELETE, UPDATE [OF Lista Attributi]
- Condizione: clausola WHERE SQL