

Esercizio 1

Si consideri il seguente schema relazionale:

IMMOBILI(CodI, Indirizzo, Tipo, Zona, Stato, PrezzoR)

AGENTE(CodA, Nome, Telefono, NomeAgenzia)

VISITE(CodA, CodI, Data)

VENDITE(CodA, CodI, PrezzoV, DataV)

Relativo ad una base di dati di una catena di agenzie immobiliari:

- Nella relazione IMMOBILI, l'attributo Tipo memorizza il tipo dell'immobile (appartamento, villetta, magazzino, ecc.), l'attributo Zona memorizza la zona in cui è situato l'immobile (centro, residenziale, periferia, ecc.), l'attributo Stato memorizza lo stato dell'immobile (buono, ottimo, ordinato, ecc.), l'attributo PrezzoR memorizza il prezzo di vendita in Euro fissato per l'immobile.
- Nella relazione VISITE, l'attributo Data memorizza la data della visita.
- Nella relazione VENDITE, l'attributo PrezzoV memorizza il prezzo in Euro a cui l'immobile è stato venduto, e DataV la data di vendita. Tale prezzo può essere inferiore al prezzo richiesto.

Supponiamo che il campo Nome in Agente e il campo Indirizzo in Immobili siano chiavi secondarie.

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 corrispondente, motivando le scelte effettuate.
3. Presentare gli statement di creazione dello schema logico, utilizzando, a scelta, SQL-99 o Oracle. Si consiglia di utilizzare Oracle in presenza di tipi collezione.
4. Supponendo che le tabelle contengano un certo numero di tuple, presentare gli statement per aggiornare la base di dati nel caso in cui l'agente Rossi abbia venduto l'immobile situato in Via XX Settembre 25, al prezzo di 300.000 in data 2/3/03.
5. Presentare gli statement SQL per le seguenti interrogazioni:
 - Determinare gli indirizzi degli appartamenti in buono stato, situati in zona residenziale, venduti ad un prezzo inferiore a quello inizialmente richiesto.
 - Determinare il nome e il telefono degli agenti che hanno venduto solo villette in zona residenziale.
 - Determinare il nome dell'agente che ha venduto un immobile al prezzo più alto tra tutte le vendite effettuate.
 - Determinare gli agenti che hanno visitato tutti i magazzini situati in zona periferica.
 - Per gli agenti che hanno venduto (almeno) due immobili di tipo diverso in zona residenziale, determinare il codice di tali agenti, il numero di vendite in zone residenziali effettuate, e la somma fatta perdere all'agenzia per la vendita di tali immobili (cioè lo scarto tra i prezzi di vendita richiesti e quelli effettivamente ottenuti mediante la vendita).
 - Determinare gli indirizzi degli immobili non ancora venduti, più cari per la zona e il tipo a cui si riferiscono.
 - Assegnare ad ogni immobile non ancora venduto il prezzo medio degli immobili relativi alla sua stessa zona e al suo stesso tipo.

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 AS Var | NEW AS Var}]  
[FOR EACH {ROW | STATEMENT}]  
[WHEN Condizione ]  
Comandi SQL  
con
```

- Evento: INSERT, DELETE, UPDATE [OF Lista Attributi]
- Condizione: clausola WHERE SQL
- Se l'azione comprende piu' di un comando: BEGIN ATOMIC ... END;