

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)

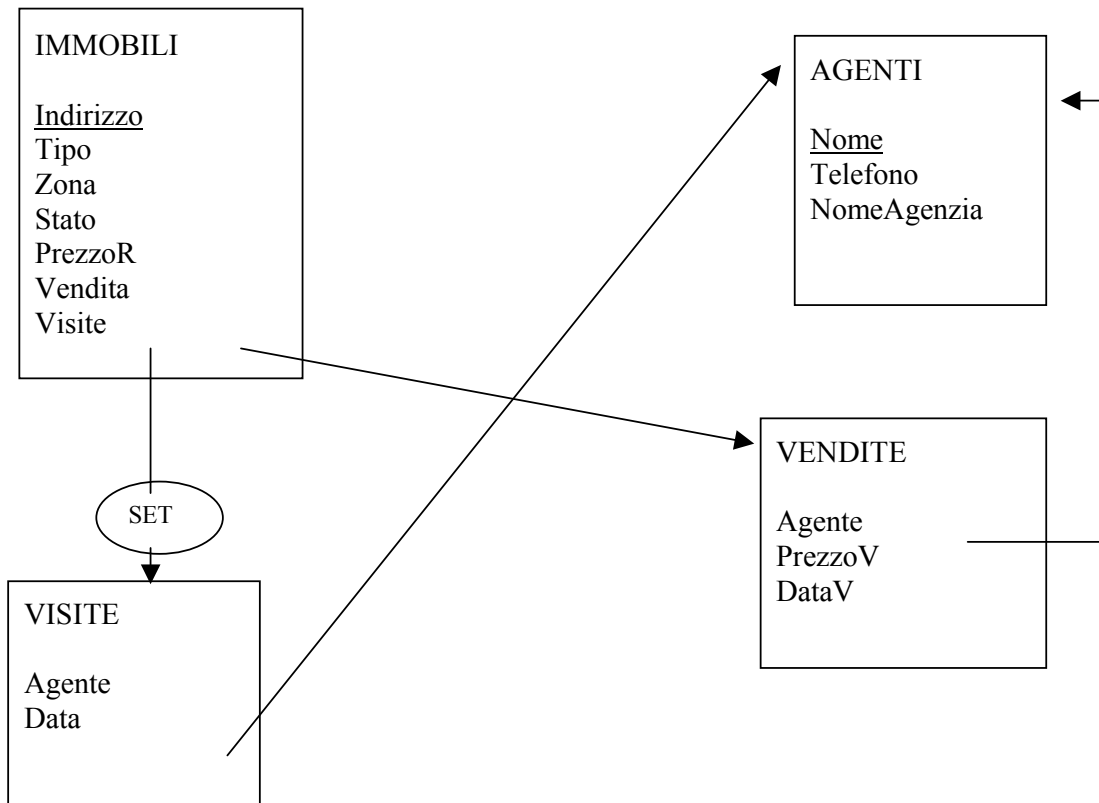
Relativo ad una base di dati di una catena di agenzie immobiliari:

- 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.

Immobili: tipo t_immobile (non ci sono archi entranti quindi non sono obbligata a definirlo ma assumendo di estendere lo schema successivamente specializzando Immobili decido di crearlo)
tabella: Immobili

Visite: Set_visite TABLE OF t_visite
Vendita: t_vendita

Non uso tipi riferimento perche' le visite e le vendite si riferiscono sempre ad un unico immobile, quindi la ridondanza e' bassa se non nulla.

Agente: tipo t_agente (ci sono archi entranti)
tabella: Agenti

Visite: tipo t_visita
No tabella (non ho tipi riferimento su t_visita e non devo gestirle in modo indipendente dagli immobili)
Agente: REF t_agente (riduco ridondanza, lo stesso agente puo' effettuare + visite)

Vendite: tipo t_vendita
No tabella (non ho tipi riferimento su t_vendite e non devo gestirle in modo indipendente dagli immobili)
Agente: REF t_agente (riduco ridondanza, lo stesso agente puo' effettuare + visite)

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.

```
CREATE TYPE t_agente AS OBJECT
(
  Nome VARCHAR(20),
  Telefono VARCHAR(10),
  NomeAgenzia VARCHAR(20));
```

```
CREATE TYPE t_visita AS OBJECT
(
  Agente REF t_agente,
  Data DATE);
```

```
CREATE TYPE t_vendita AS OBJECT
(
  Agente REF t_agente,
  PrezzoV NUMERIC(8,2),
  DataV DATE);
```

```
CREATE TYPE set_visite AS TABLE OF t_visita ;
```

```
CREATE TYPE t_immobile AS OBJECT
(
  Indirizzo VARCHAR(20),
  Tipo VARCHAR(20),
  Zona VARCHAR(20),
  Stato VARCHAR(20),
  PrezzoR NUMERIC(8,2),
  Vendita t_vendita,
  Visite set_visite);
```

```
CREATE TABLE Agenti OF t_agente
(PRIMARY KEY(Nome));
```

```
CREATE TABLE Immobili OF t_immobile
```



```
FROM Immobili j
WHERE ref(i) = ref(j)) v);
```

- 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).

```
SELECT i.Vendita.Agente.Nome, COUNT(*),
       SUM(i.PrezzoR) - SUM(i.Vendita.PrezzoV)
FROM Immobili i
WHERE i.Zona = 'residenziale'
GROUP BY i.Vendita.Agente.Nome
HAVING COUNT(*) >= 2;
```

- Determinare gli indirizzi degli immobili non ancora venduti, più cari per la zona e il tipo a cui si riferiscono.

```
SELECT i.Indirizzo
FROM Immobili i
WHERE i.Vendita = NULL AND
      NOT EXISTS (SELECT * FROM Immobili j
                  WHERE j.PrezzoR > i.PrezzoR AND
                        j.Zona = i.Zona AND j.Tipo = i.Tipo);
```

- Assegnare ad ogni immobile non ancora venduto il prezzo medio degli immobili relativi alla sua stessa zona e al suo stesso tipo.

```
UPDATE Immobili i
SET i.PrezzoR = (SELECT AVG(PrezzoR) FROM Immobili j
                WHERE j.Zona = i.Zona AND j.Tipo = i.Tipo)
WHERE i.Vendita = NULL;
```