

Esercizio 3

Si vuole sviluppare una base di dati per gestire i ristoranti di un certo insieme di città italiane, da riportare in una guida turistica.

Ogni città è caratterizzata da un nome e dalla regione in cui si trova. Ogni città è divisa in zone, ognuna con un codice e un nome. Zone con lo stesso codice possono appartenere solo a città distinte, eventualmente con nomi diversi.

I ristoranti possono essere ristoranti veri e propri oppure essere i ristoranti di un albergo.

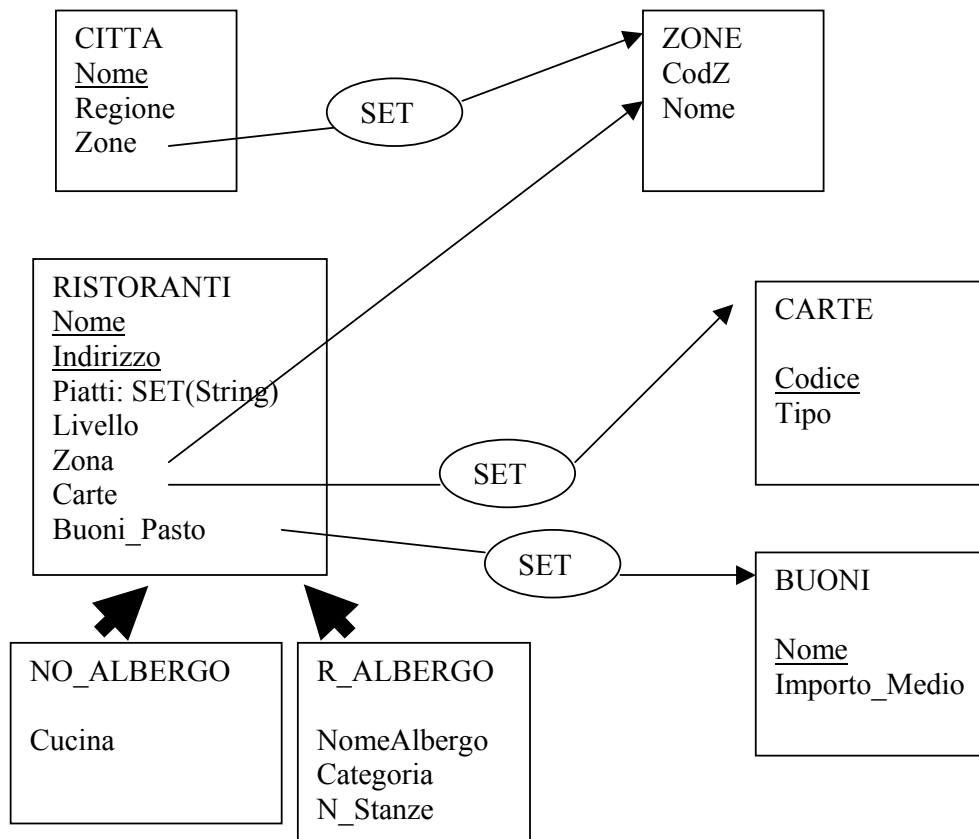
In entrambi i casi, si vuole mantenere il nome del ristorante, un codice identificativo, un indirizzo, un insieme (anche vuoto) di piatti per i quali il ristorante è rinomato, il prezzo medio di una cena in tale ristorante e il livello di qualità del ristorante. Ogni ristorante risiede in una certa zona. Per i ristoranti di alberghi, si vuole inoltre memorizzare il nome dell'albergo, la categoria dell'albergo e il numero di stanze.

Ogni ristorante accetta un certo insieme di carte di credito e di buoni pasto. Per ogni carta di credito, si vuole memorizzare il tipo della carta e un codice identificativo. Per ogni buono pasto, si vuole memorizzare il nome del buono e il suo importo medio.

Ogni ristoranti che non risiede in un albergo propone un certo tipo di cucina. Per ogni tipo di cucina si vuole memorizzare un codice identificativo e una descrizione.

Si richiede di:

1. Progettare uno schema concettuale ad oggetti per il precedente dominio applicativo, motivando le scelte effettuate.



Vincolo esplicito: identificativo Zona: Codice + Nome Citta (non riesco a rappresentarlo)
Nota: scelgo di rappresentare la cucina con solo la descrizione (dimentico il codice). Se avessi voluto mantenere entrambe le informazioni avrei dovuto creare un'entita' Cucina (non fatto per rendere piu' semplice lo schema).

2. Tradurre lo schema concettuale in uno schema logico relazionale ad oggetti corrispondente, motivando le scelte effettuate.

Citta: no tipo (no archi entranti)

Tabella: Citta

Zone: Set_zona TABLE OF REF t_zona (supponendo che una stessa zona possa essere utilizzata per citta' diverse, il tipo REF riduce ridondanza)

Zona: tipo t_zona (archi entranti)

Tabella Zone (archi entranti da più tabelle)

Carte tipo t_carta (archi entranti)

Tabella Carte (tipo riferito in Ristoranti come REF)

Buoni tipo t_buono (archi entranti)

Tabella Buoni (tipo riferito in Ristoranti come REF)

Ristoranti: tipo t_ristorante (no archi entranti ma sottotipi)

Tabella Ristoranti

Piatti Set_piatti TABLE OF String

Carte Set_carte TABLE OF REF t_carta

(la stessa carta puo' essere associata a ristoranti diversi)

Buoni Set_buoni TABLE OF REF t_buono

(lo stesso buono puo' essere associato a ristoranti diversi)

Zona REF t_zona (nella stessa zona + ristoranti)

Ristoranti_no_albergo: tipo t_no_alb sottotipo t_ristorante

Tabella Ristoranti_no_alb

Ristoranti_albergo: tipo t_alb sottotipo t_ristorante

Tabella Ristoranti_alb

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.

Scelgo Oracle

```
CREATE TYPE t_zona AS OBJECT
```

```
(Cod VARCHAR(4),
```

```
Nome VARCHAR(15));
```

```
CREATE TYPE Set_zona AS TABLE OF REF t_zona;
```

```
CREATE TYPE t_carta AS OBJECT
(Tipo VARCHAR(15),
Codice VARCHAR(10));
```

```
CREATE TYPE t_buono AS OBJECT
(Nome VARCHAR(15),
Importo_medio NUMERIC(4,2));
```

```
CREATE TYPE Set_piatti AS TABLE OF VARCHAR(20);
CREATE TYPE Set_carte AS TABLE OF REF t_carta;
CREATE TYPE Set_buoni AS TABLE OF REF t_buono;
```

```
CREATE TYPE t_ristorante AS OBJECT
(Nome VARCHAR(20),
Indirizzo VARCHAR(20),
Piatti Set_piatti,
Zona REF t_zona,
Carte Set_carte,
Buoni Set_buoni)
NOT FINAL;
```

```
CREATE TYPE t_no_alb UNDER t_ristorante
(Cucina VARCHAR(20));
```

```
CREATE TYPE t_alb UNDER t_ristorante
(Nome_Albergo VARCHAR(20),
Categoria VARCHAR(3),
N_Stanze NUMERIC(2));
```

```
CREATE TABLE Citta
(Nome VARCHAR(20) PRIMARY KEY,
Regione VARCHAR(20),
Zone Set_Zona)
NESTED TABLE Zone STORE AS tab_zone;
```

```
CREATE TABLE Zone OF t_zona;
```

```
CREATE TABLE Ristoranti OF t_ristorante
(PRIMARY KEY(Nome,Indirizzo))
NESTED TABLE Piatti STORE AS tab_piatti,
NESTED TABLE Carte STORE AS tab_carte,
NESTED TABLE Buoni STORE AS tab_buoni;
```

In Ristoranti posso anche inserire istanze dei sottotipi di t_ristorante

```
CREATE TABLE Carte OF t_carta
(PRIMARY KEY(Codice));
```

```
CREATE TABLE Buoni OF t_buono
```

(PRIMARY KEY(Nome));

4. Presentare gli statement di inserimento per ogni tabella generata.

Presento quelli per ristorante (tipo + complesso)

```
INSERT INTO Ristoranti
VALUES ('Da Vittorio', 'Via XXX', Set_piatti('pesce', 'primi'), '1', Set_carte(), Set_buoni());
```

inserisco le carte (per i buoni e' simile)

```
INSERT INTO TABLE(SELECT Carte FROM Ristoranti
                    WHERE Nome = 'Da Vittorio' AND Indirizzo = 'Via XXX')
SELECT ref(c) FROM carte c WHERE c.tipo IN ('Carta si');
```

5. Presentare gli statement SQL per le seguenti interrogazioni:

- a. Determinare per ogni città il numero di ristoranti.

```
SELECT c.Nome, COUNT(*)
FROM Ristoranti r, Citta c, TABLE(c.Zone) w
WHERE r.zona = value(w)
GROUP BY c.Nome;
```

- b. Determinare per ogni città il numero di ristoranti che servono spaghetti all'astice.

```
SELECT c.Nome, COUNT(*)
FROM Ristoranti r, Citta c, TABLE(c.Zone) w
WHERE r.zona = value(w) AND
      'spaghetti all'astice' IN (SELECT * FROM TABLE(r.Piatti))
GROUP BY c.Nome;
```

- c. Determinare per ogni ristorante il numero di carte di credito accettate.

```
SELECT r.Nome, r.Indirizzo, COUNT(*)
FROM Ristoranti r, TABLE(r.Carte) c
GROUP BY r.Nome, r.Indirizzo;
```

6. Modificare le informazioni relative al ristorante "Da Vittorio", inserendo come piatto rinomato gli spaghetti all'astice.

```
INSERT INTO TABLE(SELECT Piatti FROM Ristoranti WHERE Nome = 'Da Vittorio'
AND Indirizzo = 'Via XXX')
VALUES ('spaghetti all'astice');
```