

### Esercizio 1 (punti 8)

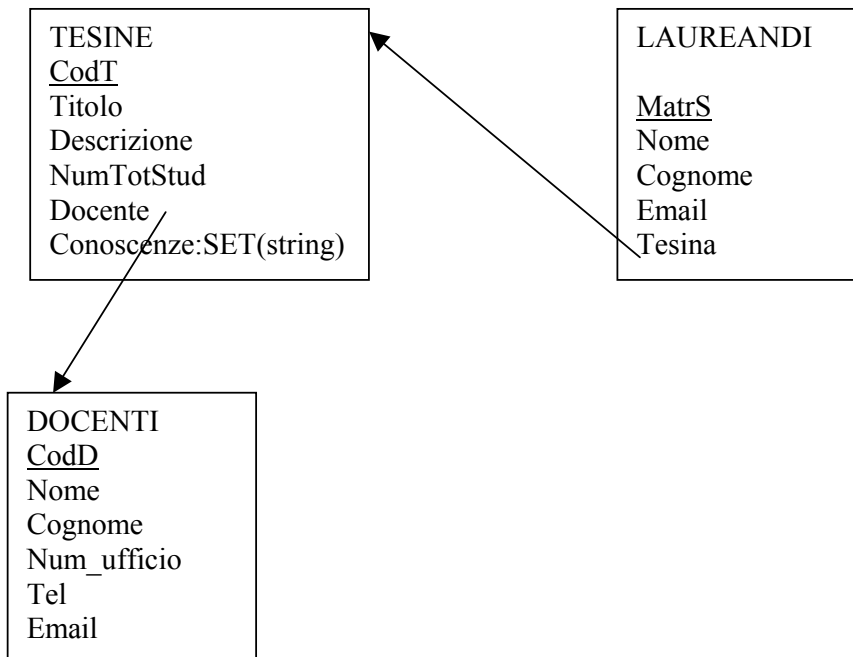
Si consideri il seguente schema relazionale, relativo alle gestione delle tesine da assegnare per la nuova laurea in Informatica:

TESINE(CodT, Titolo, Descrizione, CodD, Num\_Tot\_Stud)  
DOCENTI(CodD, Nome, Cognome, Num\_Ufficio, Tel, Email)  
LAUREANDI(MatrS, Nome, Cognome, Email, CodT)  
CONOSCENZE(CodCon, Descrizione)  
RICHIEDE(CodT, CodCon)

Nello schema precedente, la tabella CONOSCENZE contiene la descrizione delle varie conoscenze che lo svolgimento delle tesine può richiedere. Il campo Num\_Tot\_Stud in TESINE indica il numero totale di studenti che svolgono la tesina in questione. Campi con lo stesso nome rappresentano chiavi e chiavi esterne. Il campo Descrizione in CONOSCENZE è chiave secondaria.

Si richiede di:

1. Progettare uno schema concettuale ad oggetti equivalente allo schema relazionale precedente, motivando le scelte effettuate.



Non ho creato istanza per conoscenze in quanto rappresentata da un'unica stringa (non ha molta importanza come entità in se').

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

Tesine: tipo t\_tesina (arco entrante)  
Tabella Tesine  
Docente REF t\_docente  
(un docente potrebbe proporre più tesine, la scelta del tipo riferimento riduce i duplicati)  
Conoscenze: Set\_conoscenze TABLE OF VARCHAR(20)

Laureandi: no tipo (no archi entranti)  
Tabella Laureandi  
Tesina REF t\_tesina (la tesina ‘nasce’ prima dello studente quindi ha senso mantenerla in una tabella ed utilizzare in Laureandi un tipo riferimento, inoltre la stessa tesina puo’ essere associata a + studenti).

Docenti tipo t\_docente (arco entrante)  
Tabella Docenti

3. Presentare gli statement di creazione della parte di schema logico relativa all’entita’ TESINE utilizzando il sistema dei tipi di Oracle.

```
CREATE TYPE Set_conoscenze AS TABLE OF VARCHAR(20);
```

```
CREATE TYPE t_tesina  
(CodT VARCHAR(4),  
Titolo VARCHAR(30),  
Descrizione VARCHAR(100),  
NumTotStud NUMERIC(2),  
Docente REF t_docente,  
Conoscenze Set_conoscenze);
```

```
CREATE TABLE Tesine OF t_tesina  
(PRIMARY KEY (CodT),  
SCOPE FOR Docente IS Docenti)  
NESTED TABLE Conoscenze STORE AS tab_con;
```

4. Supponendo che le tabelle contengano un certo numero di tuple, presentare gli statement SQL per eseguire le seguenti operazioni:
  - o Inserire tra le conoscenze della tesina identificata da CodT=14, la conoscenza “basi di dati testuali”.

```
INSERT INTO TABLE( SELECT t.Conoscenze FROM Tesine t  
WHERE t.CodT= '14')  
VALUES ('basi di dati testuali');
```

- Determinare nome e cognome dei docenti le cui tesine non sono state scelte da alcun studente.

```
SELECT d.Nome, d.Cognome
FROM Docenti d
WHERE ref(d) NOT IN (SELECT l.Tesina.Docente
                    FROM Laureandi l);
```

- Determinare il numero medio di studenti che svolgono le tesine assegnate da Marina Ribaldo che richiedono come conoscenze “basi di dati”.

```
SELECT AVG(NumStud)
FROM
  (SELECT l.Tesina.CodT, COUNT(*) AS NumStud
   FROM Laureandi l
   WHERE l.Tesina.Docente.Nome = 'Marina' AND l.Tesina.Docente.Cognome =
   'Ribaldo' AND 'basi di dati' IN (SELECT * FROM TABLE(l.Tesina.Conoscenze))
   GROUP BY l.Tesina.CodT);
```