

Cognome:

Nome:

Matricola:

## Basi di dati – Corso di Laurea in Informatica Umanistica

Esercitazione pre-verifica del 21.03.2007

Parte 1 – modello relazionale, SQL (A)

Si consideri il seguente schema di base di dati letteraria:

### TABLE Autori

{Codice integer **PRIMARY KEY**,  
Cognome char(30) **NOT NULL**,  
Nome char(20) **NOT NULL**,  
Sesso char(1),  
AnnoNascita integer,  
AnnoMorte integer,  
Nazionalità char(20)}

### TABLE Romanzi

{Codice integer **PRIMARY KEY**,  
Titolo char(40) **NOT NULL**,  
Autore integer **NOT NULL REFERENCES** Autori(Codice),  
Protagonista integer **REFERENCES** Personaggi(Codice),  
AnnoPubblicazione integer}

### TABLE PersonaggiNeiRomanzi

{Romanzo integer **NOT NULL**  
**REFERENCES** Romanzi(Codice),  
Personaggio integer **NOT NULL**  
**REFERENCES** Personaggi (Codice),  
**PRIMARY KEY**(Romanzo, Personaggio)}

### TABLE Personaggi

{Codice integer **PRIMARY KEY**,  
Cognome char(30) **NOT NULL**,  
Nome char(20) **NOT NULL**,  
Sesso char(1)}

1) Elencare gli autori dell'ottocento, ordinati per nazionalità, che hanno scritto romanzi nel cui titolo compare un termine con radice "tempesta" (algebra oppure SQL, punti 6)

Tab1 = (Autori JOIN Romanzi ON Autori.Codice=Romanzi.Autore)

Tab2 =  $\sigma$  Autori.AnnoNascita  $\geq$  1800 AND Autori.AnnoNascita  $<$  1900 AND Romanzi.Titolo LIKE '\*tempest\*' (Tab1)

Tab3 = ( $\pi$  Autori.Cognome, Autori.Nome, Autori.Nazionalità (Tab2))

Risultato = ORDER BY Autori.Nazionalità, Autori.Cognome (DISTINCT(Tab3))

---

```
SELECT DISTINCT Autori.Cognome, Autori.Nome, Autori.Nazionalità
FROM Autori JOIN Romanzi ON Autori.Codice=Romanzi.Autore
WHERE Autori.AnnoNascita  $\geq$  1800 AND Autori.AnnoNascita  $<$  1900 AND
      Romanzi.Titolo LIKE '*tempest*'
ORDER BY Autori.Nazionalità, Autori.Cognome
```

2) Elencare i romanzi degli autori russi viventi, con il relativo anno di pubblicazione (algebra oppure SQL, punti 6)

Tab1 = (Autori JOIN Romanzi ON Autori.Codice=Romanzi.Autore)

Tab2 =  $\sigma$  Autori.Nazionalità = 'russa' AND Autori.AnnoMorte IS NULL (Tab1)

Risultato = ( $\pi$  Romanzo.Titolo, Romanzo.AnnoPubblicazione, Autore.Cognome (Tab2))

---

```
SELECT Romanzo.Titolo, Romanzo.AnnoPubblicazione, Autore.Cognome
FROM Autori JOIN Romanzi ON Autori.Codice=Romanzi.Autore
WHERE Autori.Nazionalità = 'russa' AND Autori.AnnoMorte IS NULL
ORDER BY Autore.Cognome, Romanzo.AnnoPubblicazione
```

Cognome:

Nome:

Matricola:

3) Elencare i personaggi dei romanzi il cui protagonista è Maigret (algebra o SQL, punti 7)

Tab1 = (Personaggi AS Protagonisti JOIN Romanzi ON Personaggi.Codice=Romanzi.Protagonista  
JOIN PersonaggiNeiRomanzi ON Romanzi.Codice=PersonaggiNeiRomanzi.Personaggio  
JOIN Personaggi ON Personaggi.Codice=PersonaggiNeiRomanzi.Personaggio)

Tab2 =  $\sigma_{\text{Protagonisti.Cognome} = \text{'Maigret'}}(\text{Tab1})$

Risultato =  $\text{DISTINCT}(\pi_{\text{Personaggi.Cognome, Personaggi.Nome}}(\text{Tab2}))$

---

```
SELECT DISTINCT Personaggi.Cognome, Personaggi.Nome
FROM Personaggi AS Protagonisti JOIN Romanzi ON Personaggi.Codice=Romanzi.Protagonista
JOIN PersonaggiNeiRomanzi ON Romanzi.Codice=PersonaggiNeiRomanzi.Personaggio
JOIN Personaggi ON Personaggi.Codice=PersonaggiNeiRomanzi.Personaggio
WHERE Protagonisti.Cognome = 'Maigret'
```

4) Elencare gli autori che hanno pubblicato romanzi a distanza di oltre 30 anni l'uno dall'altro (algebra o SQL, punti 7)

Tab1 = (Autori JOIN Romanzi AS Rom1 ON Autori.Codice=Rom1.Autore  
JOIN Romanzi AS Rom2 ON Autori.Codice=Rom2.Autore)

Tab2 =  $\sigma_{\text{Rom1.AnnoPubblicazione} - \text{Rom2.AnnoPubblicazione} > 30}(\text{Tab1})$

Tab3 =  $\text{DISTINCT}(\pi_{\text{Autori.Cognome, Autori.Nome, Autori.Nazionalit\`a}}(\text{Tab2}))$

Risultato = ORDER BY Autori.Cognome (Tab3)

---

```
SELECT DISTINCT Autori.Cognome, Autori.Nome
FROM Autori JOIN Romanzi AS Rom1 ON Autori.Codice=Rom1.Autore
JOIN Romanzi AS Rom2 ON Autori.Codice=Rom2.Autore
WHERE Rom1.AnnoPubblicazione - Rom2.AnnoPubblicazione > 30
ORDER BY Autori.Cognome
```

5) Modificare lo schema del database letterario per rappresentare le case editrici dei romanzi (punti 7)

```
TABLE Romanzi
{Codice integer PRIMARY KEY,
Titolo char(40) NOT NULL,
Autore integer NOT NULL REFERENCES Autori(Codice),
Protagonista integer REFERENCES Personaggi(Codice),
CasaEditrice integer REFERENCES CaseEditrici(Codice),
AnnoPubblicazione integer}
```

```
TABLE CaseEditrici
{Codice integer PRIMARY KEY,
Nome char(30) NOT NULL,
Citt\`a char(20),
Nazionalit\`a char(20)}
```