

**Sistemi Informativi - Soluzioni Prova Intermedia del 16/11/2000****Esercizio 1**

Considera il seguente schema di basi di dati relazionale:

**Opera**(Codice, Titolo, Genere, AnnoStesura)

**Persona**(Codice, Nome, Cognome, DataNascita, DataMorte)

**Autore**(CodiceOpera, CodicePersona, PosizioneListaAutori)

**Edizione**(Codice, CodiceOpera, Editore, AnnoEdizione)

**Libro**(NumeroCopia, CodiceEdizione, DataPrestito, DataRestituzione)

1.1) Descrivi il contenuto informativo della base di dati. In particolare: perché secondo te è stato necessario distinguere tra Edizione e Libro? Perché i dati relativi all'autore non sono incorporabili nella relazione Opera?

La base di dati rappresenta le informazioni coinvolte nella gestione di una biblioteca, per la quale si vuole tener traccia dei vari autori e delle opere da loro scritte, delle diverse edizioni delle suddette opere e delle copie di tali edizioni presenti in biblioteca e soggette a prestito.

E' stato necessario distinguere tra **Edizione** e **Libro** perché ad una edizione possono corrispondere più copie di libri presenti in biblioteca. I dati dell'autore non sono incorporabili nella relazione **Opera** perché un'opera può avere un numero arbitrario di autori.

**CONSIDERAZIONE:** la relazione **Libro** contiene attributi (DataPrestito e DataRestituzione) che si riferiscono evidentemente ad un oggetto fisico. Ne deriva che l'attributo NumeroCopia rappresenti il numero di copia della edizione (specificata in CodiceEdizione) presente nella biblioteca. Nel momento in cui n copie di un'edizione vengono acquisite dalla biblioteca, vengono inserite n tuple nella relazione **Libro** con valori progressivi (da 1 a n) per l'attributo NumeroCopia.

**CONSIDERAZIONE SOFISTICATA:** Quando una copia non è in prestito, DataPrestito e DataRestituzione sono posti a NULL. Nel momento in cui una copia viene data in prestito l'attributo DataPrestito viene aggiornato alla data corrente. Quando il libro viene restituito i due campi tornano necessariamente al loro valore iniziale NULL. L'attributo DataRestituzione non conterrà mai la data effettiva di restituzione (l'archivio non si presta ad una memorizzazione di informazioni storiche) ma evidentemente la data di restituzione presunta del libro.

1.2) Indica le chiavi primarie delle relazioni.

**Opera**(Codice, Titolo, Genere, AnnoStesura)

**Persona**(Codice, Nome, Cognome, DataNascita, DataMorte)

**Autore**(CodiceOpera, CodicePersona, PosizioneListaAutori)

**Edizione**(Codice, CodiceOpera, Editore, AnnoEdizione)

**Libro**(NumeroCopia, CodiceEdizione, DataPrestito, DataRestituzione)

1.3) Indica i vincoli di integrità referenziale. E' possibile definire un vincolo di integrità referenziale che imponga che solo gli autori vengano inseriti nella relazione Persona?

I vincoli di integrità referenziale sono:

- tra l'attributo CodiceOpera di **Autore** e la relazione **Opera**
- tra l'attributo CodicePersona di **Autore** e la relazione **Persona**
- tra l'attributo CodiceOpera di **Edizione** e la relazione **Opera**
- tra l'attributo CodiceEdizione di **Libro** e la relazione **Edizione**

Non è possibile esprimere il vincolo di integrità referenziale specificato perché tali vincoli si esprimono sempre tra un insieme di attributi della prima relazione e la chiave primaria della seconda relazione, e CodicePersona non è chiave primaria di **Autore**.

*(Lo scopo di questa domanda è quello di rendere esplicito il fatto che la relazione **Persona** non contiene esclusivamente autori. Questa considerazione sarà d'aiuto nello scegliere la risposta corretta dell'esercizio 2.1)*

1.4) Indica i campi sui quali è plausibile che si ammettano valori nulli.

Opera.Genere per le opere di genere indefinito

Opera.AnnoStesura per le opere in cui l'anno stesura sia dubbio

Persona.DataNascita per le persone di cui si ignora tale valore

Persona.DataMorte per le persone viventi

Libro.DataPrestito per i libri non in prestito

Libro.DataRestituzione per i libri non ancora restituiti (oppure, in ottemperanza alla "CONSIDERAZIONE SOFISTICATA" espressa sopra, per i libri di cui non si dispone di una data presunta di restituzione)

## Esercizio 2

Con riferimento allo stesso schema di basi di dati fornito nell'esercizio precedente:

2.1) Quale (o quali) delle seguenti interrogazioni SQL determina gli autori di nome 'Paolo'?

Questa va bene	<pre>SELECT DISTINCT Nome, Cognome FROM Persona, Autore WHERE Persona.Codice = Autore.CodicePersona AND       Persona.Nome = 'Paolo'</pre>
Questa cerca tra le persone senza verificare che siano anche autori	<pre>SELECT Nome, Cognome FROM Persona WHERE Nome = 'Paolo'</pre>
Questa non controlla che il nome sia 'Paolo'	<pre>SELECT DISTINCT Nome, Cognome FROM Persona WHERE Codice = ANY (SELECT CodicePersona                     FROM Autore)</pre>

2.2) Formula una interrogazione SQL che permetta di determinare i coautori di Montanelli (Montanelli incluso)

```
SELECT DISTINCT P.Nome, P.Cognome
FROM Persona, Autore, Persona as P, Autore as A
WHERE Persona.Codice = Autore.CodicePersona AND
      P.Codice = A.CodicePersona AND
      Persona.Cognome = 'Montanelli' AND
      Autore.CodiceOpera = A.CodiceOpera
```

Oppure nidificata:

```
SELECT DISTINCT Nome, Cognome
FROM Autore, Persona
WHERE CodicePersona = Codice AND
      CodiceOpera = ANY ( SELECT CodiceOpera
                          FROM Persona, Autore
                          WHERE Codice = CodicePersona AND
                                Cognome = 'Montanelli' )
```

2.3) Quale (o quali) delle seguenti interrogazioni SQL determina le opere di Pirandello in edizioni Mondadori?

<p>E' errato l'uso della particella ALL: così si cercano le edizioni Mondadori delle opere il cui codice coincide con TUTTE le opere di Pirandello. Darebbe una risposta esatta solamente se Pirandello avesse scritto una sola opera.</p>	<pre>SELECT DISTINCT Titolo FROM Opera, Edizione WHERE Opera.Codice = Edizione.CodiceOpera AND       Edizione.Editore = 'Mondadori' AND       Opera.Codice = ALL (         SELECT Autore.CodiceOpera         FROM Autore, Persona         WHERE Autore.CodicePersona = Persona.Codice AND               Persona.Cognome = 'Pirandello')</pre>
<p>Questa non è sintatticamente corretta: nella target list compare Titolo, che non è attributo di nessuna relazione della clausola From (Persona e Autore).</p>	<pre>SELECT DISTINCT Titolo FROM Persona, Autore WHERE Cognome = 'Pirandello' AND       Autore.CodicePersona = Persona.Codice AND       Autore.CodiceOpera = ANY(         SELECT Opera.Codice         FROM Opera, Edizione         WHERE Opera.Codice = Edizione.CodiceOpera AND               Edizione.Editore = 'Mondadori')</pre>
<p>Questa va bene e corrisponde ad un equi-join tra le quattro relazioni della clausola From (con le condizioni espresse dalle prime tre righe della clausola Where), più una selezione sui valori degli attributi Cognome ed Editore.</p>	<pre>SELECT DISTINCT Titolo FROM Persona, Autore, Opera, Edizione WHERE Persona.Codice = Autore.CodicePersona AND       Autore.CodiceOpera = Opera.Codice AND       Opera.Codice = Edizione.CodiceOpera AND       Cognome = 'Pirandello' AND       Editore = 'Mondadori'</pre>

2.4) Formula una interrogazione SQL che permetta di determinare gli autori di biografie di scrittori (si assuma che in una biografia il nome del personaggio coincida necessariamente con il titolo dell'opera)

```
SELECT DISTINCT Persona.Cognome
FROM Persona, Autore, Opera, Persona as Scrittore
WHERE Persona.Codice = Autore.CodicePersona AND
      Opera.Codice = Autore.CodiceOpera AND
      Opera.Titolo = Scrittore.Cognome
```

Oppure nidificando:

```
SELECT DISTINCT Persona.Cognome
FROM Persona, Autore, Opera
WHERE Persona.Codice = Autore.CodicePersona AND
      Opera.Codice = Autore.CodiceOpera AND
      Opera.Titolo = ANY (SELECT Cognome
                          FROM Persona)
```

In realtà, le due interrogazioni precedenti non tengono conto del fatto che nella relazione Persona potrebbero capitare tuple che non corrispondono a scrittori. Le seguenti sarebbero più corrette:

```
SELECT DISTINCT Persona.Cognome
FROM Persona, Autore, Opera, Persona as P, Autore as A
WHERE Persona.Codice = Autore.CodicePersona AND
      Opera.Codice = Autore.CodiceOpera AND
      Opera.Titolo = P.Cognome AND
      A.CodicePersona = P.Codice
```

Oppure nidificando:

```
SELECT DISTINCT Persona.Cognome
FROM Persona, Autore, Opera
WHERE Persona.Codice = Autore.CodicePersona AND
      Opera.Codice = Autore.CodiceOpera AND
      Opera.Titolo = ANY (SELECT Cognome
                          FROM Persona, Autore
                          WHERE Persona.Codice = Autore.CodicePersona)
```

### Esercizio 3

Rispondi alle seguenti domande:

3.1) La tabella qui sotto può essere una rappresentazione di una relazione nel modello relazionale dei dati? (Motiva la risposta: quali sono le proprietà che hai verificato?).

**Partita**

SquadraOspite	AltraSquadra	RetiOspite	RetiAltra
Juventus	Roma	0	0
Milan	Torino	1	2
Atalanta	Inter	2	3
Juventus	Roma	0	0

No, perché la tabella ha due righe uguali. Quando una tabella rispetta le seguenti 3 proprietà:

- 1) le righe sono tutte diverse
- 2) i nomi degli attributi sono distinti
- 3) le colonne hanno valori omogenei

allora può essere una rappresentazione grafica di una relazione nel modello relazionale. In tal caso l'ordine delle righe e delle colonne non è significativo.

3.2) Che differenza c'è tra uno schema di base di dati e un'istanza di base di dati?

Lo schema di basi di dati rappresenta la parte invariante dell'informazione, cioè la sua struttura. L'istanza di basi di dati rappresenta la parte dell'informazione che viene modificata con gli inserimenti e gli aggiornamenti, cioè una fotografia delle informazioni contenute nella base di dati ad un certo momento. Nel modello logico dei dati, per esempio, lo schema di base di dati è rappresentato dall'insieme degli schemi delle relazioni (nomi e attributi), mentre l'istanza di base di dati è rappresentata dall'insieme delle istanze delle relazioni (tuple).

3.3) Con riferimento allo schema di base di dati composto dalla sola relazione **Persona**(Codice, Nome, Cognome), quale (o quali) delle seguenti interrogazioni in algebra relazionale trova i nomi delle persone che hanno cognome 'Bianchi'?

Questa non è sintatticamente corretta: dopo aver eseguito una proiezione sull'attributo Nome, l'attributo Cognome non è più disponibile per una selezione.	$\sigma_{\text{Cognome}='Bianchi'} (\pi_{\text{Nome}} \text{Persona})$
Questa non contiene nemmeno un riferimento all'attributo Nome.	$\sigma_{\text{Cognome}='Bianchi'} (\pi_{\text{Cognome}} \text{Persona})$
Questa va bene: prima prende le righe inerenti ai vari 'Bianchi' e poi prende la colonna con i nomi.	$\pi_{\text{Nome}} (\sigma_{\text{Cognome}='Bianchi'} \text{Persona})$

3.4) Considera lo schema di base di dati così composto:

**Cavallo**(Codice, Nome)

**Genealogia**(CodiceGenitore,CodicePuledro)

Quale (o quali) delle seguenti interrogazioni in algebra relazionale trova il nome del cavallo genitore di 'Soldatino'?

<p>Questa va bene.</p>	$A = \rho_{C \leftarrow \text{Codice}, N \leftarrow \text{Nome}} (\text{Cavallo})$ $B = \sigma_{N='Soldatino'} (\text{Cavallo} \bowtie_{\text{Codice}=\text{CodiceGenitore}} \text{Genealogia} \bowtie_{\text{CodicePuledro}=C} A)$ $\pi_{\text{Nome}} B$
<p>Questa va bene.</p>	$A = \rho_{C \leftarrow \text{Codice}, N \leftarrow \text{Nome}} (\text{Cavallo})$ $B = \sigma_{\text{Nome}='Soldatino'} (\text{Cavallo} \bowtie_{\text{Codice}=\text{CodicePuledro}} \text{Genealogia} \bowtie_{\text{CodiceGenitore}=C} A)$ $\pi_N B$
<p>Questa non va bene: trova i figli di 'Soldatino' e non il genitore.</p>	$A = \rho_{C \leftarrow \text{Codice}, N \leftarrow \text{Nome}} (\text{Cavallo})$ $B = \sigma_{\text{Nome}='Soldatino'} (\text{Cavallo} \bowtie_{\text{Codice}=\text{CodiceGenitore}} \text{Genealogia} \bowtie_{\text{CodicePuledro}=C} A)$ $\pi_N B$