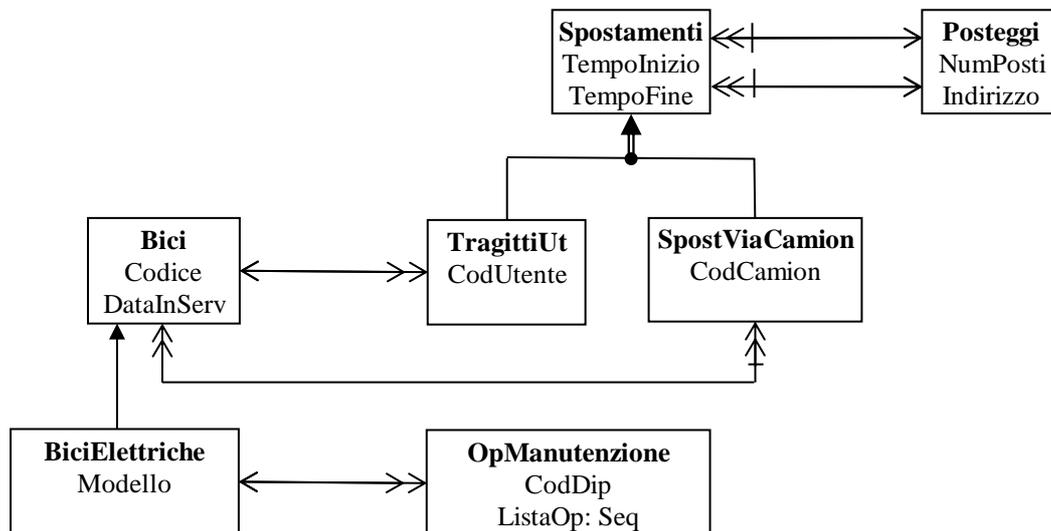


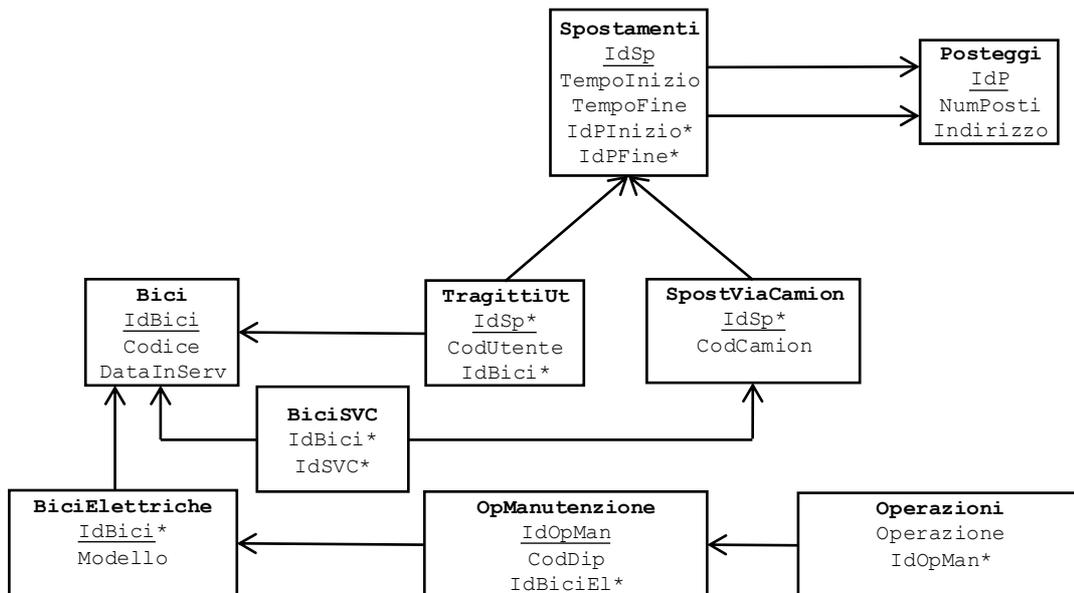
Primo compito di Basi di Dati - 9/4/2018 – testo 1 – soluzioni – v1.0

- 1) Una città ha un servizio di bici pubbliche basato su posteggi: un utente arriva a un posteggio, si identifica tramite tessera, ritira una bici, e la lascia ad un posteggio diverso. Quando un posteggio è troppo pieno, un camion ritira un certo numero di bici e le deposita ad un posteggio vuoto. La società che gestisce le bici vuole tenere traccia di tutti i loro spostamenti, tramite utente o tramite camion. Di ogni bici interessano un codice e la data di inizio servizio. Di ogni posteggio interessa conoscere il numero di posti, l'indirizzo, gli spostamenti che sono iniziati dal posteggio, e gli spostamenti terminati sul posteggio. Gli spostamenti sono di due tipi: tragitti utente e spostamenti via camion. Per ogni tragitto utente interessa sapere in che momento e da quale posteggio è iniziato, e in che momento e in quale posteggio si è concluso, il codice dell'utente e la bicicletta spostata. Se era uno spostamento via camion interessano ugualmente ora e posteggio di inizio e conclusione, ma in più interessa l'insieme delle biciclette trasportate e il codice che identifica il camion che ha effettuato il trasporto. Alcune biciclette sono elettriche. Per queste bici interessano anche il modello e l'insieme delle operazioni di manutenzione effettuate. Per tali operazioni interessa la data, la bici coinvolta, il codice del dipendente che l'ha effettuata, e una lista di operazioni (si tratta di un elenco arbitrario di termini).
- a) Si disegni lo schema concettuale della base di dati, indicando in modo testuale gli eventuali vincoli non catturati graficamente



- b) Si traduca lo schema concettuale in uno schema relazionale grafico e testuale, usando la notazione $R(A, B^*, \dots)$

Indicheremo gli attributi direttamente all'interno dello schema grafico.



2) Si consideri il seguente schema relazionale che tiene traccia di libri, lettori e autori:

Lettori(IdLe, NomeLet, NazioneLet, Età)

LettoriLibri(IdLi*, IdLe*)

Libri(IdLi, Titolo, Lingua, Anno, NPag)

AutoriLibri(IdLi*, IdA*)

Autori(IdA, NomeAut, NazioneAut)

Edizioni(IdLi*, Editore, Anno)

Scrivere le interrogazioni SQL che restituiscono, senza ripetizioni, le seguenti informazioni

- a) Il nome e l'IdLe di tutti i lettori che hanno letto almeno un libro più lungo di 100 pagine in Italiano

```
SELECT DISTINCT le.NomeLet, le.IdLe
FROM Lettori le, LettoriLibri ll, Libri li
WHERE le.IdLe = ll.IdLe AND ll.IdLi=l.IdLi AND l.NPag > 100 AND l.Lingua = 'Italiano'
```

- b) Il nome e l'IdLe di tutti i lettori che hanno letto solo libri in Italiano

```
SELECT DISTINCT le.NomeLet, le.IdLe
FROM Lettori le,
WHERE NOT EXISTS
(SELECT *
FROM LettoriLibri ll, Libri li
WHERE le.IdLe = ll.IdLe AND ll.IdLi=l.IdLi
AND l.Pagine > 100 and NOT (l.Lingua = 'Italiano'))
```

- c) Gli autori tali che tutti i loro libri hanno avuto almeno un lettore di Età minore di 18 anni

Diamo prima una soluzione non minimale – in realtà la parte sottolineata non serve

```
SELECT a.NomeAut, a.IdA
FROM Autori a
WHERE NOT EXISTS
```

```

(SELECT *
FROM AutoriLibri al, Libri li
WHERE a.IdA = al.IdA AND
      al.IdLi = li.IdLi AND
NOT EXISTS
      (SELECT *
FROM LettoriLibri ll, Lettori le
WHERE li.IdLi = ll.IdLi AND ll.IdLe = le.IdLe
AND l.Età <18 ))

```

Soluzione più semplice:

```

SELECT a.NomeAut, a.IdA
FROM Autori a
WHERE NOT EXISTS
      (SELECT *
FROM AutoriLibri al
WHERE a.IdA = al.IdA AND
NOT EXISTS
      (SELECT *
FROM LettoriLibri ll, Lettori le
WHERE al.IdLi = ll.IdLi AND ll.IdLe = le.IdLe
AND l.Età <18 ))

```

- d) Per ogni NazioneLet, la nazione e il numero totale di libri letti da lettori di quella nazione (se un libro è stato letto da più lettori della stessa nazione, deve essere contato più volte)

```

SELECT le.NazioneLet, count(*)
FROM Lettori le, LettoriLibri ll, Libri li
WHERE le.IdLe = ll.IdLe AND ll.IdLi=l.IdLi
GROUP BY le.NazioneLet

```

- e) Per ogni libro, IdLi, Titolo, e numero di lettori.

```

SELECT li.IdLi, li.Titolo, count(*)
FROM Lettori le, LettoriLibri ll, Libri li
WHERE le.IdLe = ll.IdLe AND ll.IdLi=l.IdLi
GROUP BY li.IdLi, li.Titolo

```

- f) Per ogni lettore ed ogni lingua in cui quel lettore abbia letto almeno due libri: IdLe, Nome lettore, Lingua, e numero di libri scritti in quella lingua letti dal lettore.

```

SELECT le.IdLe, le.NomeLet, li.Lingua, count(*)
FROM Lettori le, LettoriLibri ll, Libri li
WHERE le.IdLe = ll.IdLe AND ll.IdLi=l.IdLi
GROUP BY le.IdLe, le.NomeLet, li.Lingua
HAVING count(*) >= 2

```

- g) Id e Nome degli autori che, dal 2000 in poi, hanno scritto solo libri con più di 100 pagine (ovvero, quelli scritti prima del 2000 li trascuriamo).

```

SELECT a.NomeAut, a.IdA
FROM Autori a
WHERE NOT EXISTS

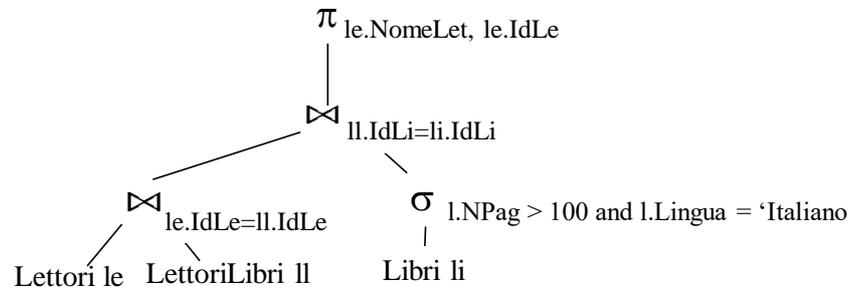
```

```

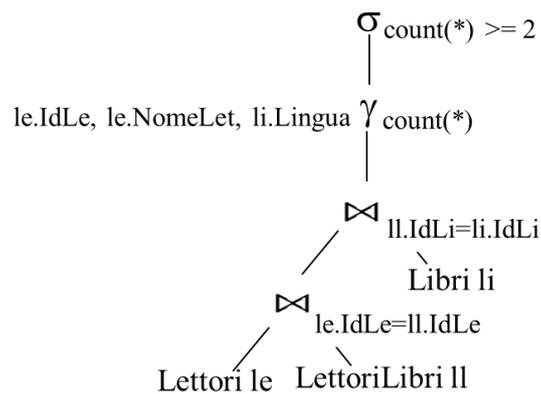
(SELECT *
FROM AutoriLibri al , Libri l
WHERE a.IdA = al.IdA AND
      al.IdL = L.IdL AND
      li.Anno > 2000 AND NOT (li.NPag > 100 ))

```

h) Disegnare un albero algebrico per la query a



i) Disegnare un albero algebrico per la query f

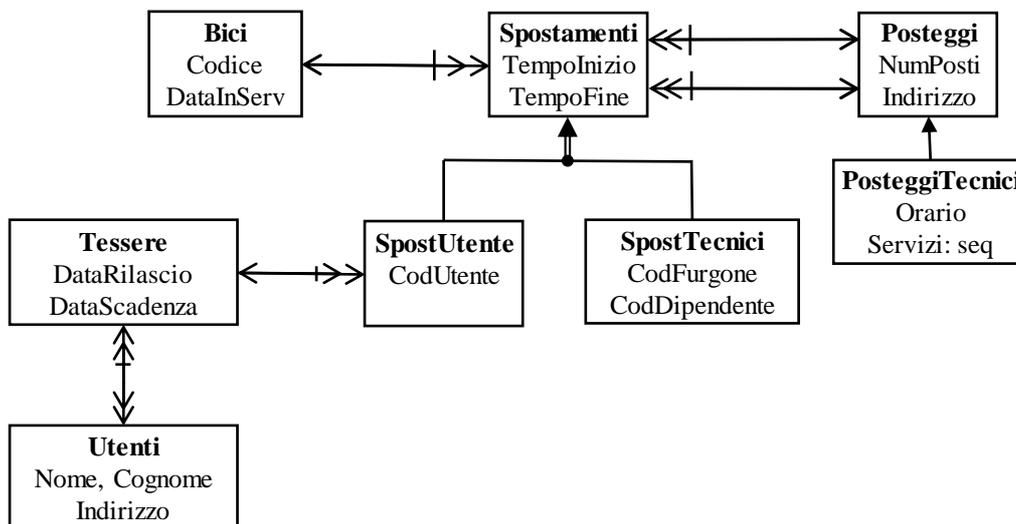


Primo compito di Basi di Dati - 9/4/2018 – testo 2

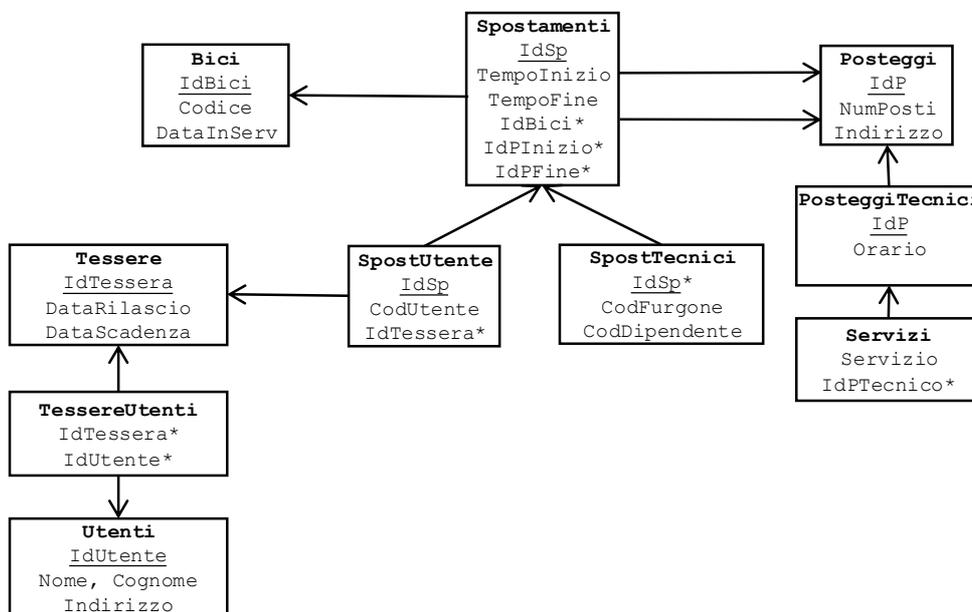
1. Una città ha un servizio di bici pubbliche basato su posteggi: un utente arriva a un posteggio, si identifica tramite tessera, ritira una bici, e la lascia ad un posteggio diverso. Quando una bici viene messa in servizio o quando la bici si guasta, viene portata, a cura della società, da un posteggio a un posteggio tecnico, o vice versa. La società che gestisce le bici vuole tenere traccia di tutti i loro spostamenti, da utente o tecnici.

Di ogni bici interessano un codice e la data di inizio servizio. Di ogni posteggio interessa conoscere il numero di posti, l'indirizzo, gli spostamenti che sono iniziati dal posteggio, e gli spostamenti terminati sul posteggio. Se si tratta di un posteggio tecnico interessano le stesse informazioni e anche l'orario di servizio ed un elenco dei servizi a disposizione in quel luogo (si tratta di un elenco arbitrario di termini). Per ogni spostamento interessa conoscere la bici coinvolta, a che ora e da quale posteggio è iniziato, e a che ora e in quale posteggio si è concluso. Se era uno spostamento da utente, interessano la tessera utente e la bicicletta spostata. Se era uno spostamento tecnico, interessa il codice che identifica il furgoncino che lo ha effettuato e il codice del dipendente che lo guidava. Di ogni tessera utente interessano la data di rilascio, la data di scadenza, e gli utenti che sono associati alla tessera – che possono essere più d'uno. Per ogni utente interessano nome, cognome, indirizzo, e l'insieme delle tessere associate.

- a) Si disegni lo schema concettuale della base di dati, indicando in modo testuale gli eventuali vincoli non catturati graficamente.



- b) Si traduca lo schema concettuale in uno schema relazionale grafico e testuale, usando la notazione $R(A, B^*, \dots)$



2) Si consideri il seguente schema relazionale che tiene traccia di libri, lettori e autori:

Lettori(IdLe, NomeLet, NazioneLet, Età)
LettoriLibri(IdLi*, IdLe*)
Libri(IdLi, Titolo, Lingua, Anno, Pagine, NPag)
AutoriLibri(IdLi*, IdA*)
Autori(IdA, NomeAut, NazioneAut)
Edizioni(IdLi*, Editore, Anno)

Scrivere le interrogazioni SQL che restituiscono, senza ripetizioni, le seguenti informazioni

- a) Il nome e l'IdA di tutti gli autori che hanno scritto almeno un libro in Inglese dopo il 2000
- b) Il nome e l'IdA di tutti gli autori che hanno scritto solo libri in Inglese
- c) Gli autori tali che tutti i loro libri hanno avuto almeno un lettore di nazionalità italiana
- d) Per ogni NazioneLet, la nazione e il numero totale di libri letti da lettori di quella nazione (se un libro è stato letto da più lettori della stessa nazione, deve essere contato più volte)
- e) Per ogni libro, IdLi, Titolo, e numero di autori.
- f) Per ogni autore ed ogni lingua in cui quell'autore abbia scritto almeno due libri: IdA, Nome Autore, Lingua, e numero di libri scritti in quella lingua dall'autore.
- g) Id e Nome dei lettori che hanno letto libri con più di 100 pagine solo in Italiano (ovvero, quelli con meno di 100 pagine li trascuriamo).
- h) Disegnare un albero algebrico per la query a
- i) Disegnare un albero algebrico per la query f

Per le soluzioni, si veda la query analoga nel testo 1