Basi di Dati e Programmazione Web
Prova di verifica - 9 Aprile 2010
Risoluzione

Tempo a disposizione: 2 ore

Es. 1) SQL [10 punti]: I seguenti esercizi hanno lo scopo di verificare la conoscenza del linguaggio SQL. Il DB di riferimento è formato dalle seguenti relazioni:

```
PROGETTI(CodProg,Budget,AnnoInizio,AnnoFine);
-- AnnoFine è NULL per progetti ancora in corso

SPESE(IdSpesa,Descrizione,Data,Importo,CodProg,CodRic),
-- CodProg è una FK che referenzia PROGETTI;
-- ogni spesa grava sul budget di un progetto
-- ed e' imputata a un ricercatore
```

Si descriva a parole cosa restituiscono le seguenti query SQL:

a) [1 p.]

```
SELECT P.CodProg
FROM PROGETTI P
WHERE P.Annofine IS NULL
AND   P.AnnoInizio <= 2007
AND   P.Budget >= 500000
```

Il codice dei progetti tuttora in corso iniziati prima del 2008 e con un budget di almeno 500000 (Euro)

b) [1 p.]

```
SELECT DISTINCT P.CodProg
FROM PROGETTI P JOIN SPESE S ON (P.CodProg = S.CodProg)
WHERE S.Importo > 0.01*P.Budget
```

Il codice dei progetti che hanno avuto almeno una spesa superiore all'1% del proprio budget

c) [2 p.]

```
SELECT P.CodProg
FROM PROGETTI P LEFT JOIN SPESE S
   ON (P.CodProg = S.CodProg) AND (S.Importo > 0.01*P.Budget)
WHERE S.IdSPesa IS NULL
AND P.Annofine IS NOT NULL
```

Il codice dei progetti conclusisi che non hanno avuto alcuna spesa superiore all'1%del proprio budget

d) [2 p.]

```
SELECT P.CodProg, P.Budget – SUM(S.Importo) AS XXX
FROM PROGETTI P JOIN SPESE S ON (P.CodProg = S.CodProg)
WHERE P.Annofine IS NOT NULL
GROUP BY P.CodProg, P.Budget
HAVING SUM(S.Importo) <> P.Budget
```

Il codice dei progetti, e il budget residuo degli stessi, conclusisi e che hanno avuto un residuo di budget non nullo

e) **[2 p.]**

```
SELECT S.CodRic, SUM(S.Importo)
FROM SPESE S
WHERE S.CodRic NOT IN     (SELECT S1.CodRic
                           FROM SPESE S1
                           GROUP BY S1.CodRic
                           HAVING MAX(S1.Importo) > 100000)
GROUP BY S.CodRic
```

Il codice dei ricercatori, e l'importo totale delle spese imputate agli stessi, che non hanno eseguito nessuna spesa superiore a 100000 (Euro)

f) **[1 p.]** Con riferimento alla query d), si proponga un nome significativo per il secondo campo in uscita e si spieghi perché è necessario raggruppare anche su P.Budget

Budget_residuo
Perché altrimenti non sarebbe possibile usare tale attributo né nella clausola HAVING né nella clausola SELECT

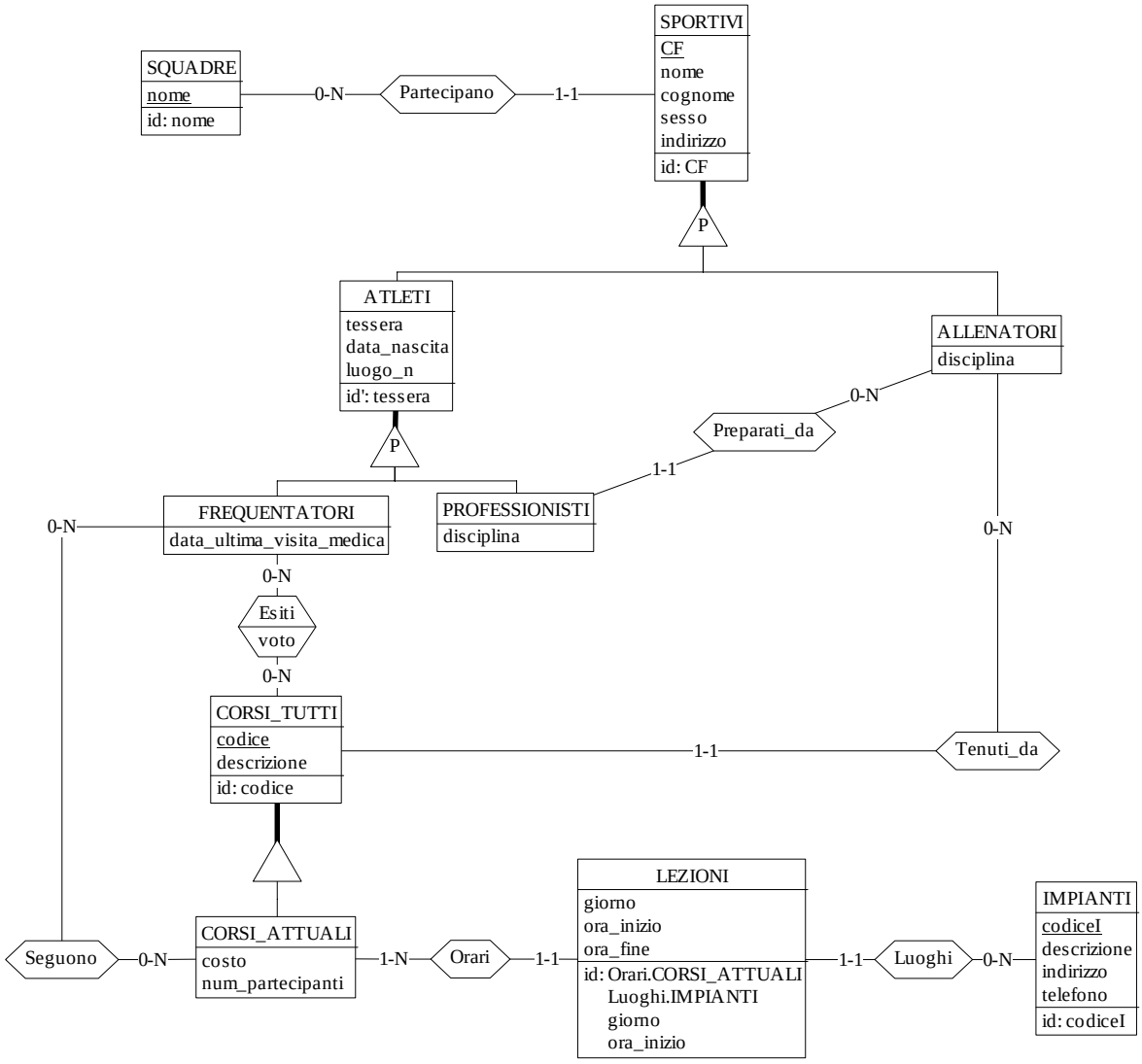
g) **[1 p.]** Si modifichi la query e) in modo da ottenere una query equivalente in cui la subquery non faccia uso di GROUP BY

```
SELECT S.CodRic, SUM(S.Importo)
FROM SPESE S
WHERE S.CodRic NOT IN     (SELECT S1.CodRic
                           FROM SPESE S1
                           WHERE S1.Importo > 100000)
GROUP BY S.CodRic
```

Basi di Dati e Programmazione Web
Prova di verifica - 9 Aprile 2010
Risoluzione

Es. 2) Modello E/R [5 punti]

Si descriva a parole il seguente schema E/R, relativo a una società sportiva (per brevità, si tralasci di descrivere gli attributi di entità e associazioni)



Ogni sportivo è iscritto a una squadra. Gli sportivi, identificati dal proprio Codice Fiscale, si dividono in allenatori e atleti, questi ultimi possono essere o frequentatori o professionisti. Ogni professionista è preparato da uno e un solo allenatore (viceversa, un allenatore può preparare diversi proefssionisti, ma anche nessuno). Per ogni corso (passato), viene registrato il voto ottenuto dai frequentatori che l'hanno seguito. Ogni corso è tenuto da uno e un solo allenatore (un allenatore può tenere diversi corsi, ma anche non tenerne nessuno). Dei corsi attuali si registra quali sono i frequentatori che li seguono e il calendario delle lezioni. Ogn lezione si svolge presso un determinato impianto. In un impianto si possono tenere diverse lezioni, ma anche nessuna.

Basi di Dati e Programmazione Web
Prova di verifica - 9 Aprile 2010
Risoluzione

Es. 3) Progettazione logica [5 punti]
Si fornisca una rappresentazione relazionale dello schema E/R dell'esercizio 2), indicando primary keys e foreign keys. Entrambe le gerarchie vanno tradotte collassandole verso l'alto e, quando possibile, le associazioni non vanno tradotte separatamente

Nota: per brevità si usa il punto interrogativo (?) per indicare gli attributi *selettori* e l'asterisco (*) per indicare la possibile presenza di valori nulli

SQUADRE(Nome)
SPORTIVI(CF,Nome,Cognome,Sesso,Indirizzo,NomeSquadra,
ATL_ALL?,Tessera*,DataNascita*,LuogoNascita*,DisciplinaAll*,
FREQ_PRO?*,DataUltimaVisitaMedica*,DisciplinaProf*,CFPreparatore*)
FK: NomeSquadra REFERENCES SQUADRE
FK: CFPreparatore REFERENCES SPORTIVI
CORSI(Codice,Descrizione,CFAllenatore,ATTUALE?,Costo*,NumPartecipanti*)
FK: CFAllenatore REFERENCES SPORTIVI
ESITI(CFFrequentatore,CodiceCorso,Voto)
FK: CFFrequentatore REFERENCES SPORTIVI
FK: CodiceCorso REFERENCES CORSI
FREQUENZE(CFFrequentatore,CodiceCorso)
FK: CFFrequentatore REFERENCES SPORTIVI
FK: CodiceCorso REFERENCES CORSI
IMPIANTI(CodiceImpianto,Descrizione,Indirizzo,Telefono)
LEZIONI(CodiceCorso,CodiceImpianto,Giorno,OraInizio,OraFine)
FK: CodiceCorso REFERENCES CORSI
FK: CodiceImpianto REFERENCES IMPIANTI

Es. 4) Vincoli [3 punti]
Si descrivano a parole i vincoli che non vengono preservati nella traduzione relazionale e si dica come è possibile farli rispettare

Nella traduzione alcuni **vincoli di foreign key** diventano **imprecisi**, in quanto l'entità referenziata è stata tradotta accorpandola in un'entità più generica. E' questo il caso per:

In SPORTIVI: FK: CFPreparatore REFERENCES SPORTIVI
In CORSI FK: CFAllenatore REFERENCES SPORTIVI
In ESITI e FREQUENZE: FK: CFFrequentatore REFERENCES SPORTIVI
In FREQUENZE: FK: CodiceCorso REFERENCES CORSI
In LEZIONI FK: CodiceCorso REFERENCES CORSI

Per ognuno di questi casi si può predisporre un trigger (di tipo BEFORE INSERT) che verifichi la correttezza del valore di foreign key. Ad esempio, per il primo vincolo la condizione da verificare sarebbe del tipo (:CFPreparatore è il valore fornito in input):

```
EXISTS(SELECT * FROM SPORTIVI S WHERE S.CF = :CFPreparatore
                                    AND ATL_ALL? = 'Allenatore')
```

Altri vincoli, che possono essere fati rispettare mediante dei CHECK, riguardano la **sincronizzazione dei valori nulli e dei selettori**. Ad esempio in CORSI:

```
CONSTRAINT CorsiAttuali CHECK
((ATTUALE? = 'Y' AND Costo IS NOT NULL AND NumPartecipanti IS NOT NULL) OR
 (ATTUALE? = 'N' AND Costo IS NULL AND NumPartecipanti IS NULL))
```

Infine, poiché ogni corso attuale deve avere almeno una lezione, l'inserimento di un nuovo corso deve avvenire congiuntamente a quello di una o più lezioni all'interno di una stessa transazione.

Basi di Dati e Programmazione Web
Prova di verifica - 9 Aprile 2010
Risoluzione

Es. 5) Esecuzione di interrogazioni [3 punti]
Considerando lo schema generato nell'es. 3), si descriva una possibile modalità di esecuzione per la seguente query:

Trovare gli orari delle lezioni dei corsi tenuti dagli istruttori di pallacanestro

La query si può formulare in SQL come:

```
SELECT L.*
FROM   SPORTIVI S, CORSI C, LEZIONI L
WHERE S.CF = C.CFAllenatore
AND    C.Codice = L.CodiceCorso
AND    S.DisciplinaAll = 'pallacanestro'
```

Una possibile modalità di esecuzione consiste nell'accedere prima a SPORTIVI (unica relazione su cui è presente un predicato per filtrare le tuple), quindi eseguire un join con CORSI e poi un secondo join con LEZIONI. Poiché i join sono entrambi del tipo FK-PK, è ragionevole pensare di poter contare su degli indici: un primo indice su C.CFAllenatore permette di reperire i corsi tenuti da un certo allenatore, un secondo indice su L.CodiceCorso le lezioni di ognuno di tali corsi.

Es. 6) Transazioni [4 punti]
Si consideri un DBMS che gestisce le transazioni secondo la modalità STEAL/FORCE. Si descriva come viene usato il Log in caso di transaction e system failure.

La modalità STEAL di gestione del buffer comporta che si debbano "disfare" le eventuali modifiche operate sul DB da transazioni che abortiscono. Il Log viene pertanto scandito a ritroso ripristando le *before images* delle pagine modificate. Ciò vale per entrambi i tipi di failure.
Viceversa, la modalità FORCE di gestione del Commit non richiede che si debbano "rifare" transazioni che hanno eseguito Commit. Pertanto nel Log non è necessario memorizzare le *after images* delle pagine modificate.