BASI DI DATI – PROFF. S. CERI, M. MATERA, G. POZZI – A.A. 05/06 APPELLO DEL 4 LUGLIO 2006 – TEMPO: 2 h E 30 m

Scrivete le soluzioni delle due parti su DUE FOGLI DISTINTI, entrambi intestati

Parte 1: LINGUAGGI DI INTERROGAZIONE (su un foglio a parte rispetto alla Parte 2)

Il seguente schema rappresenta le informazioni di un insieme di orchestre in relazione ai loro concerti e alle sale in cui tali concerti vengono tenuti:

ORCHESTRA(CodOrchestra, NomeO, NomeDirettore, NumElementi, AnnoFondata)

CONCERTO(CodC, Data, CodO, CodS, PrezzoBiglietto)

SALA(CodSala, NomeS, Città, Capienza)

A. SQL (11 punti)

1. Trovare il codice e il nome delle orchestre con più di 30 elementi che hanno tenuto concerti sia a Genova, sia a Milano e non hanno mai tenuto concerti a Bologna. (3 p.)

SELECT CodOrchestra, NomeO

FROM ORCHESTRA

WHERE NumElementi > 30 AND

CodOrchestra IN (SELECT CodO

FROM Concerto JOIN Sala ON CodS=CodSala

WHERE Città='Milano') AND

CodOrchestra IN (SELECT CodO

FROM Concerto JOIN Sala ON CodS=CodSala

WHERE Città='Genova') AND

CodOrchestra NOT IN (SELECT CodO

FROM Concerto JOIN Sala ON CodS=CodSala

WHERE Città='Bologna')

2. Scrivere un comando SQL che aumenta del 10% il prezzo del biglietto dei concerti di quelle orchestre che sono state fondate prima del 2000, che hanno tenuto in totale nella loro vita più di 100 concerti e il cui prezzo medio dei biglietti sia inferiore ai 20 euro.(3 p.)

UPDATE Concerti

SET PrezzoBiglietto=PrezzoBiglietto*1.1

WHERE CodO IN (SELECT CodOrchestra

FROM Orchestra JOIN Concerti ON CodOrchestra=CodO

WHERE AnnoFondata<2000

GROUP BY CodOrchestra

HAVING COUNT(*)>100 AND AVG(PrezzoBiglietto)<20

)

3. Trovare il nome delle orchestre che hanno tenuto un concerto in tutte le sale (5 p.) Si può riformulare come: data un'orchestra non esiste una sala in cui quella orchestra non ha tenuto un concerto:

SELECT NomeO

FROM Orchestra O

WHERE NOT EXISTS (SELECT *

FROM SALA S

WHERE NOT EXISTS (SELECT *

FROM CONCERTO C WHERE C.CodS=S.CodSala AND

C.CodO=O.CodOrchestra))

B. Linguaggi Formali (6 punti)

4. Estrarre in Algebra Relazionale ottimizzata, Calcolo Relazionale e Datalog per ogni orchestra (codice e nome) il prezzo del concerto più caro (4 p.)

```
a) Algebra
```

```
\begin{split} &CONCCARO \!\!:= (\Pi_{CodO,\,PrezzoBiglietto}\,CONCERTO) - \\ &\left((\,\Pi_{\,CodO,\,PrezzoBiglietto}\,CONCERTO\,\,)\,\,\triangleright <_{CodO=CodO,\,PrezzoBiglietto}<\!(\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_{\,CodO,\,PrezzoBiglietto},\,\Pi_
```

 $\Pi_{CodOrchestra,\ NomeO, PrezzoBiglietto}\ (\Pi_{CodOrchestra,\ NomeO}\ ORCHESTRA) \triangleright \triangleleft _{CodOrchestra=CodO}\ CONCARO$

```
b) Calcolo
```

```
 \left\{ \begin{array}{l} t \mid \exists \ t_O \in ORCHESTRA, \ t_C \in CONCERTO \mid \\ (t[CodOrchestra] = t_O[CodOrchestra] \land \\ t[NomeO] = t_O[NomeO] \land \\ t[PrezzoBiglietto] = t_C[PrezzoBiglietto] \land \\ t_O [CodOrchestra] = t_C[CodO] \land \\ \neg \left(\exists \ t_{C1} \in CONCERTO \mid \\ (t_{C1}[CodO] = t_C[CodO] \land \\ t_{C1}[PrezzoBiglietto] > t_C[PrezzoBiglietto] \\ ) \\ ) \\ \right\} \end{array}
```

c) Datalog

```
? – ConcCaro (X,Y,Z)
```

5. Esprimere, in uno a scelta dei tre linguaggi formali (algebra ottimizzata), l'interrogazione che trova il codice e il nome delle orchestre il cui direttore non ha mai diretto più di una orchestra (2 p.)

Algebra

```
(\Pi_{CodOrchestra, NomeOrchestra}ORCHESTRA) - \\ (\Pi_{CodOrchestra, NomeOrchestra}((\Pi_{CodOrchestra, NomeOrchestra, NomeDirettore}ORCHESTRA))) \\ > < CodOrchestra! = CodOrchestra \land NomeDirettore = NomeDirettore (\Pi_{CodOrchestra, NomeDirettore}ORCHESTRA))))
```

C. DOMANDA (1 punti)

Si esprima, tramite un'opportuna asserzione SQL, il vincolo per cui un'orchestra non può avere due concerti diversi nella stessa data.

```
CREATE ASSERTION NonStessaData
CHECK ( NOT EXISTS ( SELECT *
FROM CONCERTO
GROUP BY CodOrchestra, Data
HAVING COUNT(*)>=2) )
```

Parte 2: PROGETTAZIONE CONCETTUALE (su un foglio a parte rispetto alla Parte 1)

D. PROGETTO CONCETTUALE (10 punti) E LOGICO (4 punti)

- Ricordare che il progetto è valutato per completezza, correttezza, leggibilità, minimalità e autoesplicatività, e che anche il progetto logico è un **grafo** i cui nodi e archi devono essere disposti coerentemente col progetto concettuale.
- Ricordare anche di specificare gli **identificatori** di tutte le entità e le **cardinalità** di tutte le associazioni, e di disporre i due grafi su due facciate affiancate, <u>in modo da poterli osservare simultaneamente</u>.