

Primo compito di Basi di Dati; Evoluzioni Recenti

15 Novembre 2004

soluzione

Esercizio 1

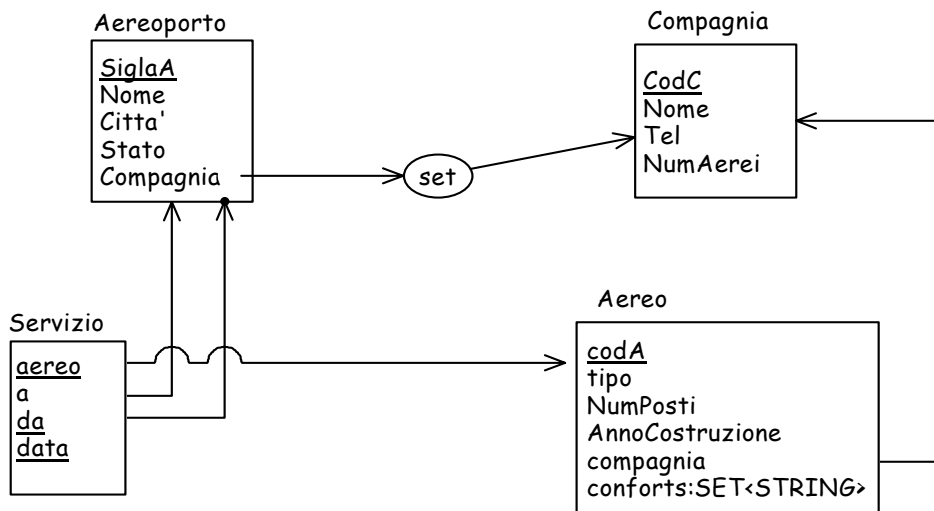
Si consideri il seguente schema relazionale, relativo ai servizi aerei forniti con aerei di varie compagnie su vari aeroporti internazionali:

AEROPORTO(SiglaA, Nome, Citta, Stato)
 COMPAGNIA (CodC, Nome, Tel, NumAerei)
 OSPITA(SiglaA, CodC)
 AEREO(CodA, Tipo, NumeroPosti, Anno_Costruzione, CodC)
 CONFORTS(CodA, Confort)
 SERVIZIO(Da, A, CodA, Data)

La tabella CONFORTS contiene per ogni aereo i conforti forniti (ad esempio: cinema, AC, cuccette, ecc.). Ogni confort e' una stringa. Nella tabella SERVIZI, gli attributi Da e A sono chiavi esterne su AEROPORTO. Tale tabella contiene i servizi aerei forniti da un aeroporto ad un altro in una certa data con un certo aereo.

Si richiede di:

1. Progettare uno schema concettuale ad oggetti equivalente allo schema relazionale precedente, motivando le scelte effettuate.



Si e' scelto di rappresentare la relazione Ospita con un traversal path da Aeroporto a Compagnia perche' questo migliora le 3 interrogazioni richieste sulla base di dati. I conforti sono rappresentati come set di stringhe (come richiesto nel testo).

2. Tradurre lo schema concettuale in uno schema logico relazionale ad oggetti, motivando le scelte effettuate.

Classe	Relazione	Tipo	Relazione Tipata	Attributi di tipo oggetto
Aeroporto		Taeroporto	Aeroporto	Compagnie setCompagnie
Compagnia		Tcompagnia	Compagnia	
Aereo		Taereo	Aereo	compagnia REF Tcompagnia SCOPE IS Compagnia conforts setConforts
Servizio	Servizio			da REF Taeroporto SCOPE IS Aeroporto a REF Taeroporto SCOPE IS Aeroporto aereo REF Taereo SCOPE IS Aereo
		SetConforts (TABLE OF string)		
		setCompagnie (TABLE OF REF Tcompagnia)		

Aeroporto, compagnia e aereo sono rappresentate nel modello logico come relazioni tipate perche' sono riferite da altre classi dello schema, mentre Servizio e' rappresentato con una relazione perche' non e' riferito da altre classi dello schema.

3. Presentare gli statement Oracle di creazione della parte di schema logico relativa ai servizi e agli aerei utilizzando il sistema dei tipi di Oracle.

```

create type Tservizio as OBJECT
(
aereo REF Taereo,
a REF Taeroporto,
da REF Taeroporto,
data date
);

create table Servizio OF Tservizio
( SCOPE FOR aereo IS Aereo,
SCOPE FOR a IS Aeroporto,
SCOPE FOR da IS Aeroporto
);

CREATE TYPE setCompagnie as TABLE OF REF Tcompagnia;
CREATE TYPE setConforts as TABLE OF varchar(30);

create type Taereo as OBJECT
(
codC varchar(5),
tipo varchar(10),
numPosti integer,
annoCostruzione integer,
compagnia REF Tcompagnia,
conforts setConforts
);

create table Aereo OF Taereo
( PRIMARY KEY (CodA),
SCOPE FOR compagnia IS Compagnia
) NESTED TABLE conforti STORE AS tab_conforts;

```

Si noti che non e' possibile definire un constraint di primary key nella tabella Servizi perche' alcuni attributi della chiave sono di tipo REF. Per garantire il vincolo, occorre definire un trigger.

4. Supponendo che le tabelle contengano un certo numero di tuple, presentare gli statement per inserire nella base di dati un nuovo servizio dal Cristoforo Colombo di Genova a Linate il 15/11/2004 con l'aereo B273.

```

insert into Servizio values(NULL,NULL, NULL, '15/11/2004');
update Servizio
set da = (select ref(p1) from aeroporto p1 where p1.nome='Cristoforo Colombo')
a = (select ref(p2) from aeroporto p2 where p2.nome='Linate')
aereo= (select ref(p3) from aereo p3 where p3.tipo='B273')
where data='15/11/2004' AND da IS NULL AND a IS NULL AND aereo IS NULL;

```

Soluzione alternativa (e anche migliore)

```

DECLARE
da REF Taeroporto;
a REF Taeroporto;
aereo REF Taereo;
BEGIN
da := select ref(p1) from aeroporto p1 where p1.nome='Cristoforo Colombo';
a := select ref(p2) from aeroporto p2 where p2.nome='Linate';
aereo := select ref(p3) from aereo p3 where p3.tipo='B273';

INSERT INTO Servizi values (da,a,aereo, '15/11/2004');
END

```

5. Presentare gli statement SQL per le seguenti interrogazioni:
- o Trovare i servizi aerei forniti con aerei dell'Alitalia con almeno 10 posti dall'aeroporto Cristoforo Colombo.

```

SELECT *
FROM Servizio s
WHERE s.aereo.Nposti >= 10 AND s.da.nome = 'Cristoforo Colombo'
AND s.aereo.compagnia.nome = 'Alitalia';

```

- o Determinare, per gli aerei che effettuano servizi da Milano, il loro nome e il nome della compagnia a cui appartengono.

```
SELECT s.aereo.nome, s.aereo.compagnia.nome
FROM Servizio s
WHERE s.da.citta='Milano';
```

- o Trovare gli aeroporti italiani che ospitano piu' compagnie aeree fra tutti gli aeroporti italiani.

```
SELECT a.siglaA, COUNT(*)
FROM Aeroporto a, TABLE(a.compagnie)
WHERE a.stato='ITALIA'
GROUP BY a.siglaA
HAVING COUNT(*) >= (SELECT COUNT(*)
                     FROM Aeroporto b, TABLE(b.compagnie)
                     WHERE b.stato='ITALIA'
                     GROUP BY b.siglaA)
```

6. (Opzionale) Si supponga che gli aerei vengano distinti in aerei di linea e aerei privati. Si supponga inoltre di volere mantenere le informazioni sui comforts solo per gli aerei privati. Si discuta come si dovrebbe cambiare lo schema logico per tenere traccia di questa situazione in Oracle.

Dal momento che in Oracle e' valido il concetto di sostituibilita' e' sufficiente:

- o eliminare dal tipo Taereo l'attributo comforts
- o dichiarare Taereo NOT FINAL
- o dichiarare il tipo TaereoPrivato come sottotipo di Taereo, con l'attributo comforts (di tipo setComforts)

Non e' necessario creare altre relazioni (per il concetto di sostituibilita')

In oracle:

```
create type Taereo as OBJECT
(
  codC varchar(5),
  tipo varchar(10),
  numPosti integer,
  annoCostruzione integer,
  compagnia REF Tcompagnia,
) NOT FINAL;

create type TaereoPrivato UNDER Taereo
(
  comforts setComforts
);

create table Aereo OF Taereo
( PRIMARY KEY (CodA),
  SCOPE FOR compagnia IS Compagnia
) NESTED TABLE comforts STORE AS tab_comforts;
```

Esercizio 2

In riferimento allo schema relazionale proposto nell'esercizio 1, si progettino i trigger in SQL-99 per ottenere i seguenti comportamenti:

1. Mantenere aggiornato il campo numAerei della tabella COMPAGNIA, a fronte di modifiche della tabella AEREO.

Uso trigger di tipo FOREACH ROW. Ad ogni modo lo riesco a risolvere anche con trigger FOR EACH STATEMENT. Ci sono tre eventi che bisogna considerare:

1. Inserimento di un nuovo aereo
2. Rimozione di un aereo
3. Cambio di compagnia di un aereo (un aereo passa dalla compagnia A alla compagnia B)

Dal momento che sviluppo trigger in SQL-99, creo 3 trigger distinti per ognuno degli eventi (in oracle avrei creato un solo trigger che si attiva al verificarsi di 3 eventi).

```
create trigger numAereiIns
after insert on Aereo
referencing new row as NUOVA
for each row
  update compagnia C
  set numaerei = numaerei + 1
  where c.codC = NUOVA.CodC;
```

```

create trigger numAereiDel
after delete on Aereo
referencing new row as VECCHIA
for each row
    update compagnia C
    set numaerei = numaerei - 1
    where c.codC = VECCHIA.CodC;

create trigger numAereiUpdate
after update of CodC on Aereo
referencing new row as NUOVA, old row as VECCHIA
for each row
    update compagnia C
    set numaerei = numaerei + 1
    where c.codC = NUOVA.CodC;

    update compagnia C
    set numaerei = numaerei - 1
    where c.codC = VECCHIA.CodC;

```

2. Si supponga di avere una tabella QUANTITASERVIZI(NumServizi) che contiene il numero di servizi aerei forniti da tutti gli aeroporti. Si vuole tenere aggiornata automaticamente questa tabella.

```

create trigger contaServiziIns
after insert on Servizi
referencing new table as NUOVA
for each statement
    update quantitaServizi
    set numServizi = numServizi + (select COUNT(*) FROM NUOVA);

create trigger contaServiziDel
after delete on Servizi
referencing old table as VECCHIA
for each statement
    update quantitaServizi
    set numServizi = numServizi - (select COUNT(*) FROM VECCHIA);

```

Si noti che l'uso del trigger for each statement in questo caso rende l'operazione piu' efficiente

3. Si vuole garantire il seguente vincolo di integrita': Non si vuole inserire un nuovo servizio da un aeroporto se la quantita' di servizi offerti da quell'aeroporto con aerei della stessa compagnia e' maggiore di 100 (si eviti di effettuare un rollback della transazione).

Questa interrogazione si risolveva facilmente con il modello relazionale ad oggetti, mentre era un po' piu' complicata con il modello relazionale puro. Infatti, utilizzando il modello relazionale ad oggetti la soluzione e':

```

create trigger checkServizi
after insert on Servizi
referencing new row as NUOVA
for each row
when (100 < (select count(*)
            from Servizi s
            where s.da.siglaA=NUOVA.da.siglaA and
                  s.aereo.compagnia.nome=NUOVA.aereo.compagnia.nome))
delete from Servizi s
where s.da.siglaA= NUOVA.da.siglaA and s.aereo.codA=NUOVA.aereo.codA and
      s.data = NUOVA.data;

```

Utilizzando il modello relazionale, la soluzione diventa:

```

create trigger checkServizi
after insert on Servizi
referencing new row as NUOVA
for each row
when (100 < (select count(*)
            from (Servizi NATURAL JOIN aerei) s
            where s.da=NUOVA.da and
                  s.compagnia = select compagnia
                                from aerei
                                where codA=NUOVA.codA))
delete from Servizi s
where s.da=NUOVA.da and s.aereo=NUOVA.aereo and
      s.data = NUOVA.data;

```