

Basi di Dati
Proff. Stefano Ceri e Maristella Matera
2° RECUPERO - APPELLO DEL 28 GIUGNO 2004 - PARTE 1

A. SQL (P. 9)

Si considerino le tabelle:

DEPLIANT(ID, Titolo, Compositore, NumPag, Formato, Data, Durata)
CONTIENE(D-ID, P-ID, Foto, Descrizione, Sconto)
PRODOTTO(P-ID, Nome, Categoria, CostoListino)

Il depliant descrive un insieme di prodotti e ha un titolo, un compositore, un numero di pagine, un formato, la data di pubblicazione e una durata di validità. Lo sconto è espresso in percentuale. Esprimere in SQL le seguenti interrogazioni:

1. Estrarre il depliant valido (rispetto alla data corrente SYSDATE) che presenta lo sconto più alto relativo al prodotto MSOffice (P.3).

a) Se si interpreta 'MSOffice' come P-ID (Data di tipo *date* e Durata di tipo *interval*):

```
select ID, Titolo
from DEPLIANT join CONTIENE on ID=D-ID
where SYSDATE between Data and (Data + Durata) AND
      P-ID = 'MSOffice' AND
      Sconto = ( select max(Sconto)
                  from DEPLIANT join CONTIENE on ID=D-ID
                  where SYSDATE between Data and (Data + Durata) AND
                        P-ID = 'MSOffice' )
```

b) Se invece si interpreta MSOffice come Nome (e quindi, in teoria, più prodotti con tale nome possono partecipare a un depliant con sconti diversi), a rigore il confronto dovrebbe essere fatto tra i massimi sconti di ogni depliant (sempre rispetto ai prodotti della "famiglia" MSOffice)

```
select ID
from ( DEPLIANT join CONTIENE C on ID=D-ID )
      join PRODOTTO P on C.P-ID=P.P-ID
where SYSDATE between Data and (Data + Durata) AND Name = 'MSOffice'
group by ID
having max( sconto ) >= ALL
      ( select max( sconto )
        from ( DEPLIANT join CONTIENE C on ID=D-ID )
              join PRODOTTO P on C.P-ID=P.P-ID
              where SYSDATE between Data and (Data + Durata) AND Name = 'MSOffice'
        group by ID )
```

2. Estrarre i depliant che non contengono prodotti il cui costo scontato è inferiore a 1000 Euro e il cui numero di pagine supera il triplo dei prodotti presenti. (P.3)

Considerando lo Sconto un intero compreso tra 0 e 100:

```
select ID, Titolo
from DEPLIANT X
where ID NOT IN ( select D-ID
                  from CONTIENE C join PRODOTTO P on C.P-ID=P.P-ID
                  where 1000 > ( CostoListino * ( 100 – Sconto) / 100 ) )
AND NumPag > ( select 3 * count(*)
               from CONTIENE
               where D-ID = X.ID )
```

3. Estrarre il nome dei prodotti per i quali esistano depliant in cui la differenza fra lo sconto più alto e lo sconto più basso praticato sia superiore al 10 per cento e la cui categoria sia “PC” oppure “Mouse”. (P.3)

Se si considera lo scostamento tra gli sconti sullo stesso prodotto in depliant diversi

```
create view MAX-DELTA-1 ( Prod, Delta ) AS
select P-ID, max( Sconto ) – min( Sconto )
from CONTIENE
group by P-ID
```

```
select distinct Nome
from PRODOTTO
where Categoria IN ( ‘PC’, ‘Mouse’ ) AND
P-ID IN ( select Prod
          from MAX-DELTA-1
          where Delta > 10 )
```

Se si considera lo scostamento tra gli sconti su diversi prodotti in uno stesso depliant

```
create view MAX-DELTA-2 ( Dep, Delta ) AS
select D-ID, max( Sconto ) – min( Sconto )
from CONTIENE
group by D-ID
```

```
select distinct Nome
from PRODOTTO
where Categoria IN ( ‘PC’, ‘Mouse’ ) AND
P-ID IN ( select P-ID
          from MAX-DELTA-2 join CONTIENE on Dep = D-ID
          where Delta > 10 )
```

B. LINGUAGGI FORMALI DI INTERROGAZIONE (P. 5)

1. Esprimere in Calcolo e Datalog la prima interrogazione dell'esercizio precedente.

Nella interpretazione a)

```
{ t | ∃ tD ∈ DEPLIANT, ∃ tC ∈ CONTIENE
  ( t[ID] = tD[ID] ∧
    tC[P-ID] = 'MSOffice' ∧
    tD[ID] = tC[D-ID] ∧
    SYSDATE >= tD[Data] ∧
    SYSDATE <= ( tD[Data] + tD[Durata] ) ∧
    ¬ ( ∃ tC2 ∈ CONTIENE
      ( tC2[P-ID] = 'MSOffice' ∧
        tC2[Sconto] > tC[Sconto] ) )
  )
}
```

```
ScontoValidoProdotto(Dep, Prod, Sconto) :- DEPLIANT(Dep, _, _, _, Data, Durata),
                                             CONTIENE(Dep, Prod, _, _, Sconto),
                                             sysdate >= Data AND sysdate <= ( Data + Durata )
```

```
ScontoOfficeNonMax (Dep) :- ScontoValidoProdotto(Dep, 'MSOffice', Sconto),
                             ScontoValidoProdotto(_, 'MSOffice', Sconto1),
                             Sconto < Sconto1
```

```
ScontoOfficeMax (Dep) :- ScontoValidoProdotto (Dep, 'MSOffice', _),
                          ¬ ScontoOfficeNonMax (Dep),
```

? – ScontoOfficeMax (X)

C. DDL (P. 2)

Scrivere i comandi SQL per

1. Creare la tabella CONTIENE, effettuando opportune ragionevoli ipotesi su domini e vincoli e reazioni ai cambiamenti.
2. Eliminare dalla tabella CONTIENE l'attributo Foto.

```
create table CONTIENE (
  D-ID varchar(10) references DEPLIANT(ID) on update cascade on delete cascade,
  P-ID varchar(10) references PRODOTTO(P-ID) on update cascade on delete no action,
  Foto blob,
  Descrizione varchar(255),
  Sconto integer not null between 0 and 100,
  primary key (D-ID, P-ID)
)
```

Alla cancellazione di un depliant non ha senso conservare l'elenco dei prodotti che vi appartengono. Invece vietiamo la cancellazione di un prodotto finché restano nel DB depliant relativi a quel prodotto.

```
alter table CONTIENE drop column Foto
```