

## BASI DI DATI - PROFF. CERI, MATERA

### RECUPERO I PROVA CON SOLUZIONI - 13 FEBBRAIO 2004

PROPRIETARIOAUTO (IdProp, Targa, DataAcq, DataVendita, Nome, ComuneResid, ProvResid)  
AUTOIMM (Targa, ComImm, ProvImm, DataImm, Marca, Modello, Colore, PotenzaKW)  
DEMOLIZIONE (Targa, Data, Luogo, NomeDemolitore)  
PRA(Comune, Provincia, Regione)

Lo schema si riferisce alle immatricolazioni, ai passaggi di proprietà e alle demolizioni di auto in Italia. Si ipotizzi che ogni auto possa essere immatricolata una sola volta, che più persone possano essere comproprietarie di una auto (nel caso, le date di acquisto devono coincidere) e che l'attributo DataVendita valga **Null** fino all'avvenuta trascrizione degli atti di vendita.

#### A. DDL (2 punti)

Scrivere i comandi SQL per

- 1) Creare le tabelle AUTOIMM e PRA, effettuando opportune ragionevoli ipotesi su domini, vincoli e reazioni ai cambiamenti.
- 2) Aggiungere alla tabella PRA l'attributo TipoSede, che assuma il valore "Normale" in assenza di altre indicazioni.

```
create table AutoImm (  
    Targa          varchar(7) primary key,  
    ComImm         varchar(127),  
    ProvImm        varchar(2),  
    DataImm        date > 01-01-1900 default sysdate(),  
    Marca          varchar(30),  
    Modello        varchar(30),  
    Colore         varchar(30),  
    PotenzaKW      integer between 1 and 1000,  
    foreign key (ComImm, ProvImm) references PRA(ComImm, ProvImm)  
    on delete no action on update no action  
)  
  
create table PRA (  
    Comune        varchar(127),  
    Provincia      varchar(127),  
    Regione        varchar(25),  
    primary key (Comune, Provincia)  
)
```

#### B. LINGUAGGI FORMALI (7 punti)

1. Esprimere in Algebra Relazionale ottimizzata, Calcolo Relazionale e Datalog la seguente interrogazione (**4 punti**):  
*Trovare i proprietari che hanno acquistato sempre e solo auto nuove*

$\Pi_{IdProp, Nome} ( PRO - ( PRO \triangleright_{Targa=Targa \wedge DataAcq > DataAcq} PRO ) )$  Da ottimizzare...

**CALCOLO:** La soluzione seguente sembra giusta, ma considera una proprietà sola, tale per cui non esiste, *per quell'auto*, un proprietario precedente. Quindi trova tutti i proprietari di **almeno una** auto nuova, non di sole auto nuove. Per scriverla correttamente si può usare un per-ogni

oppure escludere la presenza di qualsiasi altra auto usata di proprietà dello stesso – come nella seconda qui sotto

~~$\{ t \mid \exists t_p \in PRO$   
 ~~$(t[IdProp, Nome] = t_p[IdProp, Nome] \wedge$   
 $\neg (\exists t_{p2} \in PRO$   
 $t_{p2}[Targa] = t_p[Targa] \wedge$   
 $t_{p2}[DataAcq] < t_p[DataAcq] )$   
 $\}$~~~~  **–ERRATA–**

$\{ t \mid \exists t_p \in PRO$  **–CORRETTA–**  
 $( t[IdProp, Nome] = t_p[IdProp, Nome] \wedge$   
 $\neg (\exists t_{p1} \in PRO, \exists t_{p2} \in PRO$   
 $t_{p1}[IdProp] = t_{p2}[IdProp] \wedge$   
 $t_{p1}[Targa] = t_{p2}[Targa] \wedge$   
 $t_{p1}[DataAcq] < t_{p2}[DataAcq] )$   
 $\}$

CompratoUsatoUnaVoltaAlmeno(P) :- PRO(P, Targa, Data1, \_, \_, \_),  
PRO(, Targa, Data2, \_, \_, \_), Data1 > Data2  
CompraSoloNuove(Id, Nome) :- PRO(Id, \_, \_, \_, Nome, \_),  
¬ CompratoUsatoUnaVoltaAlmeno (Id)  
? – CompraSoloNuove(X, Y)

2. Esprimere in algebra relazionale ottimizzata la seguente interrogazione (3 punti):  
Trovare i proprietari che hanno posseduto auto tutte dello stesso colore

$\Pi_{IdProp, Nome} PRO - \Pi_{IdProp, Nome} ( ( PRO \triangleright \triangleleft AUT ) \triangleright \triangleleft_{IdProp=IdProp \wedge Colore \triangleleft \triangleleft Colore} ( PRO \triangleright \triangleleft AUT ) )$

Da ottimizzare...

### C. Interrogazioni in SQL (8 PUNTI)

1. Calcolare, per ogni regione, la potenza media in KW delle auto immatricolate nel 2003 (1,5 punti)

```
SELECT Regione, avg(PotenzaKW)
FROM AUT A join PRA P on (ComImm=Comune AND ProvImm=Provincia)
WHERE DataImm between 01-01-03 and 31-12-03
GROUP BY Regione
```

2. Trovare le targhe delle auto demolite senza mai subire alcun passaggio di proprietà (2 punti)

~~SELECT Targa  
FROM DEM D  
WHERE 1 = ( SELECT count(\*)  
FROM PRO  
WHERE Targa = D.Targa )~~ **(ERRATA! Ci sono le multiproprietà...)**

```
SELECT Targa
FROM DEM D
WHERE 1 = ( SELECT count(distinct DataAcq) FROM PRO WHERE Targa = D.Targa )
```

Oppure anche, ad esempio:

```
( SELECT Targa
  FROM DEM D )
  except      Tutte le auto demolite meno quelle che sono state vendute almeno una volta
( SELECT Targa
  FROM PRO
  WHERE DataVendita IS NOT NULL )
```

o ancora:

```
SELECT Targa  Tutte le auto demolite che non hanno avuto due proprietari
FROM DEM
WHERE Targa NOT IN ( SELECT A.Targa
                     FROM PRO A join PRO B on A.Targa=B.Targa
                     WHERE A.DataAsq <> B.DataAcq )
```

3. *Determinare il modello di auto più “gradito” agli automobilisti della provincia di Savona (si conti come una “preferenza” ogni acquisto, sia del nuovo sia dell’usato) (2 punti)*

```
SELECT Modello
FROM AUT A join PRO P on A.Targa=P.Targa
WHERE ProvResid = “SV”
GROUP BY Modello
HAVING count(*) >= ALL ( SELECT count(*)
                        FROM AUT A join PRO P on A.Targa=P.Targa
                        WHERE ProvResid = “SV”
                        GROUP BY Modello )
```

Oppure

```
CREATE VIEW PREFERENZE-SAVONA(Mod, Num) AS
SELECT Modello, Count(*)
FROM AUT A join PRO P on A.Targa=P.Targa
WHERE ProvResid = “SV”
GROUP BY Modello
```

```
SELECT Mod
FROM PREFERENZE-SAVONA
WHERE Num = ( SELECT max(Num) FROM PREFERENZE-SAVONA )
```

4. *Costruire il registro delle auto in circolazione, nella forma (Targa, Modello, Proprietario, ComuneResPr, ProvinciaResPr), considerando solo l’ultimo proprietario di ogni auto non ancora demolita. (2,5 punti)*

```
SELECT Targa, Modello, IdProp as Proprietario, ComuneRes AS ComuneResPr, ProvRes as...
FROM AUT A join PRO P on A.Targa=P.Targa
WHERE A.Targa NOT IN ( SELECT Targa FROM DEM )
      AND NOT EXISTS ( SELECT *
                      FROM PRO
                      WHERE Targa = P.Targa AND DataAcq > P.DataAcq )
```

O, ancora più semplicemente:

```
SELECT A.Targa, Modello, IdProp as Proprietario, ComuneRes as ComuneResPr, ProvRes as...
FROM AUT A join PRO P on A.Targa=P.Targa
WHERE A.Targa NOT IN ( SELECT Targa FROM DEM ) AND DataAcq IS NULL
Null significa che è l’ultimo proprietario
```