

**ESERCITAZIONE 5 – Giovedì 20 novembre 2014 (3 ore)**  
**Esercizi tratti da temi d'esame (SQL e linguaggi formali)**

**Tema d'esame del 28 febbraio 2012 – Esercizio B1**

Si consideri il seguente schema di base di dati che vuole tenere traccia delle multe effettuate nelle città italiane.

PERSONA (CF, Cognome, Nome, CittàResidenza, DataNascita)

VEICOLO (Targa, DataImmatricolazione, CFProprietario)

MULTA (CodiceMulta, Data, Ora, Targa, CFGuidatore, Importo, Tipologia, Descrizione, DataScadenzaPagamento)

PAGAMENTO\_MULTA (CodPagamento, CodiceMulta, DataPagamento, ImportoPagato)

Il campo Tipologia nella tabella MULTA rappresenta la tipologia dell'infrazione commessa ("divieto di sosta", "eccesso di velocità", ecc). Il campo DataScadenzaPagamento nella tabella MULTA contiene l'informazione sulla data entro la quale si deve effettuare il pagamento della multa onde evitare il pagamento di penali. Il campo CFGuidatore nella tabella MULTA rappresenta il codice fiscale di colui o colei che guida il veicolo al momento della multa, che non è necessariamente il proprietario.

Scrivere in SQL le seguenti interrogazioni:

- a) Trovare la targa dei veicoli immatricolati prima del 1980 che sono stati multati almeno 20 volte nella loro vita ma non hanno mai ricevuto più di una multa nel corso della stessa giornata.
  - b) Trovare codice fiscale, cognome e nome delle persone residenti a Torino per cui il numero di multe ricevute nel corso dell'anno 2009 è aumentato rispetto al numero di multe ricevute nel corso dell'anno 2008.
- 

**Tema d'esame del 11 settembre 2014**

Si consideri il seguente schema di base di dati che rappresenta l'orario dei voli di varie compagnie aeree, con le prenotazioni e poi gli effettivi check-in dei clienti.

VOLO (Codice, Compagnia, AeropPartenza, AeropArrivo, OraPart, OraArrivo, TipoAereo)

PRENOTAZIONE (IdPasseggero, CodVolo, DataVolo, Nome, Cognome, Email, DataNascita)

CHECKIN (IdPasseggero, CodVolo, DataVolo, Posto, OraEffettivaCheckIn, Priorità)

**Esercizio B**

Scrivere in SQL le seguenti interrogazioni:

- 1) Trovare id, nome e cognome dei passeggeri che hanno fatto il check-in di meno del 90% dei voli prenotati.
- 2) Trovare, per ogni compagnia, il codice del volo, l'aeroporto di arrivo e di partenza, del volo con il maggior numero di check-in con priorità.

**Esercizio A**

Esprimere in algebra relazionale, calcolo relazionale, Datalog ed **SQL (non richiesto nel tema d'esame)** la seguente query:

Trovare id, nome e cognome dei passeggeri che hanno prenotato almeno due voli ma nessuno della compagnia “Delta”.

---

## **ESERCIZI PER CASA**

### **Tema d’esame del 13 febbraio 2014 – Esercizio B1**

Si consideri il seguente schema di base di dati che vuole tenere traccia delle attività di un consorzio di dentisti.

DENTISTA (CF, Nome, Cognome, Specializzazione, Telefono)  
STUDIO (Codice, Nome, Indirizzo, Città, Titolare)  
PAZIENTE (CF, Nome, Cognome, Indirizzo, CittàResidenza, Età)  
LAVORO (Dentista, Studio, Paziente, Data, Tipo\_Intervento, Prezzo)

Il campo Tipo\_Intervento nella tabella LAVORO può assumere valori “Implantologia”, “Igiene orale”, “Ortodonzia”, etc.

Scrivere in SQL le seguenti interrogazioni:

- a) Trovare i dentisti specializzati in Ortodonzia che hanno lavorato in almeno 5 studi diversi e hanno fatto solo interventi di tipo “Ortodonzia” e solo su pazienti minorenni.
- b) Trovare, per ogni tipo di intervento, lo studio che ha guadagnato più soldi e il suo titolare.

### **Tema d’esame del 9 settembre 2005 – Esercizio B**

Si consideri il seguente schema di base di dati di una catena di palestre:

CLIENTE(NumTessera, CodFiscCli, Nome, Cognome, Sesso, Via, Città, IscrittoPiscina, DataIscrizione)  
ISTRUTTORE(CodFiscIstr, Nome, Cognome, Sesso, Via, Città, Data Assunzione)  
CORSO(NomeCorso, Turno, CittàClub, CodFiscIstr)  
ISCRIZIONICORSI(NumTessera, NomeCorso, Turno, DataInizio)

Dove l’attributo “Iscritto Piscina” è un flag che indica se la quota di iscrizione del cliente prevede che possa frequentare la piscina, e l’attributo “Turno” assume come valori stringhe del formato “12:00-13:00”.

Esprimere in SQL le seguenti interrogazioni:

- 1) Trovare il numero totale di iscritti di ogni corso che abbia più di 20 iscritti maschi, e il nome e il sesso dei relativi istruttori.
- 2) Trovare i turni dei corsi di aerobica che si tengono nel club di Milano e che hanno almeno due iscritti di Sesto San Giovanni ma nessun iscritto di Lodi.

## SOLUZIONI

### Tema d'esame del 28 febbraio 2012 – Esercizio B1

Si consideri il seguente schema di base di dati che vuole tenere traccia delle multe effettuate nelle città italiane.

PERSONA (CF, Cognome, Nome, CittàResidenza, DataNascita)

VEICOLO (Targa, DataImmatricolazione, CFProprietario)

MULTA (CodiceMulta, Data, Ora, Targa, CFGuidatore, Importo, Tipologia, Descrizione, DataScadenzaPagamento)

PAGAMENTO\_MULTA (CodPagamento, CodiceMult

Scrivere in SQL le seguenti interrogazioni:

a) Trovare la targa dei veicoli immatricolati prima del 1980 che sono stati multati almeno 20 volte nella loro vita ma non hanno mai ricevuto più di una multa nel corso della stessa giornata. (3 punti)

```
SELECT V.Targa
FROM Multa AS M, Veicolo AS V
WHERE M.Targa=V.Targa AND V.DataImmatricolazione<'01/01/1980'
GROUP BY V.Targa
HAVING COUNT(*)>=20
```

**EXCEPT**

```
SELECT Targa
FROM Multa
GROUP BY Targa, Data
HAVING COUNT(*)>1
```

Oppure, con NOT IN al posto di EXCEPT:

```
SELECT V.Targa
FROM Multa AS M, Veicolo AS V
WHERE M.Targa=V.Targa AND V.DataImmatricolazione<'01/01/1980' AND
      V.Targa NOT IN (
          SELECT Targa
          FROM Multa
          GROUP BY Targa, Data
          HAVING COUNT(*)>1
      )
GROUP BY V.Targa
HAVING COUNT(*)>=20
```

b) Trovare codice fiscale, cognome e nome delle persone residenti a Torino per cui il numero di multe ricevute nel corso dell'anno 2009 è aumentato rispetto al numero di multe ricevute nel corso dell'anno 2008.

```
SELECT P.CF, P.Cognome, P.Nome
FROM Multa AS M, Persona AS P
WHERE M.CFGuidatore=P.CF AND P.CittàResidenza='Torino' AND M.Data BETWEEN
    '01/01/2009' AND '31/12/2009'
GROUP BY P.CF, P.Cognome, P.Nome
HAVING COUNT(*) > ( SELECT COUNT(*)
    FROM Multa AS M2
    WHERE P.CF=M2.CFGuidatore AND M2.Data BETWEEN
        '01/01/2008' AND '31/12/2008'
    )
)
```

## Tema d'esame del 11 settembre 2014

VOLO (Codice, Compagnia, AeropPartenza, AeropArrivo, OraPart, OraArrivo, TipoAereo)  
PRENOTAZIONE (IdPasseggero, CodVolo, DataVolo, Nome, Cognome, Email, DataNascita)  
CHECKIN (IdPasseggero, CodVolo, DataVolo, Posto, OraEffettivaCheckIn, Priorità)

### Esercizio B

1) Trovare id, nome e cognome dei passeggeri che hanno fatto il check-in di meno del 90% dei voli prenotati.

```
SELECT P.IdPasseggero, P.Nome, P.Cognome
FROM Prenotazione AS P
GROUP BY P.IdPasseggero, P.Nome, P.Cognome
HAVING 0.9*COUNT(*) > (
    SELECT COUNT(*)
    FROM CheckIn AS C
    WHERE C.IdPasseggero=P.IdPasseggero
)
```

2) Trovare, per ogni compagnia, il codice del volo, l'aeroporto di arrivo e di partenza, del volo con il maggior numero di check-in con priorità.

```
CREATE VIEW CheckInPriorità (CodVolo, Compagnia, NumCheckInPriorità) AS (
    SELECT V.Codice, V.Compagnia, COUNT(*)
    FROM CheckIn AS C, Volo AS V
    WHERE C.CodVolo=V.Codice AND C.Priorità=true
    GROUP BY V.Codice, V.Compagnia
)
```

```
SELECT V.Codice, V.Compagnia, V.AeropPartenza, V.AeropArrivo
FROM Volo AS V, CheckInPriorità AS C
WHERE V.Codice=C.CodVolo AND C.NumCheckInPriorità = (
    SELECT MAX(C2.NumCheckInPriorità)
    FROM CheckInPriorità AS C2
    WHERE C2.Compagnia=V.Compagnia
)
```

Oppure, senza vista:

```

SELECT V.Codice, V.Compagnia, V.AeropPartenza, V.AeropArrivo
FROM CheckIn AS C, Volo AS V
WHERE C.CodVolo=V.Codice AND C.Priorità='true'
GROUP BY V.Codice, V.Compagnia, V.AeropPartenza, V.AeropArrivo
HAVING COUNT(*) >= ALL (

```

```

SELECT COUNT(*)
FROM CheckIn AS C2, Volo AS V2
WHERE C2.CodVolo=V2.Codice AND
      V2.Compagnia=V.Compagnia AND
      C2.Priorità=true
GROUP BY V2.Codice
)

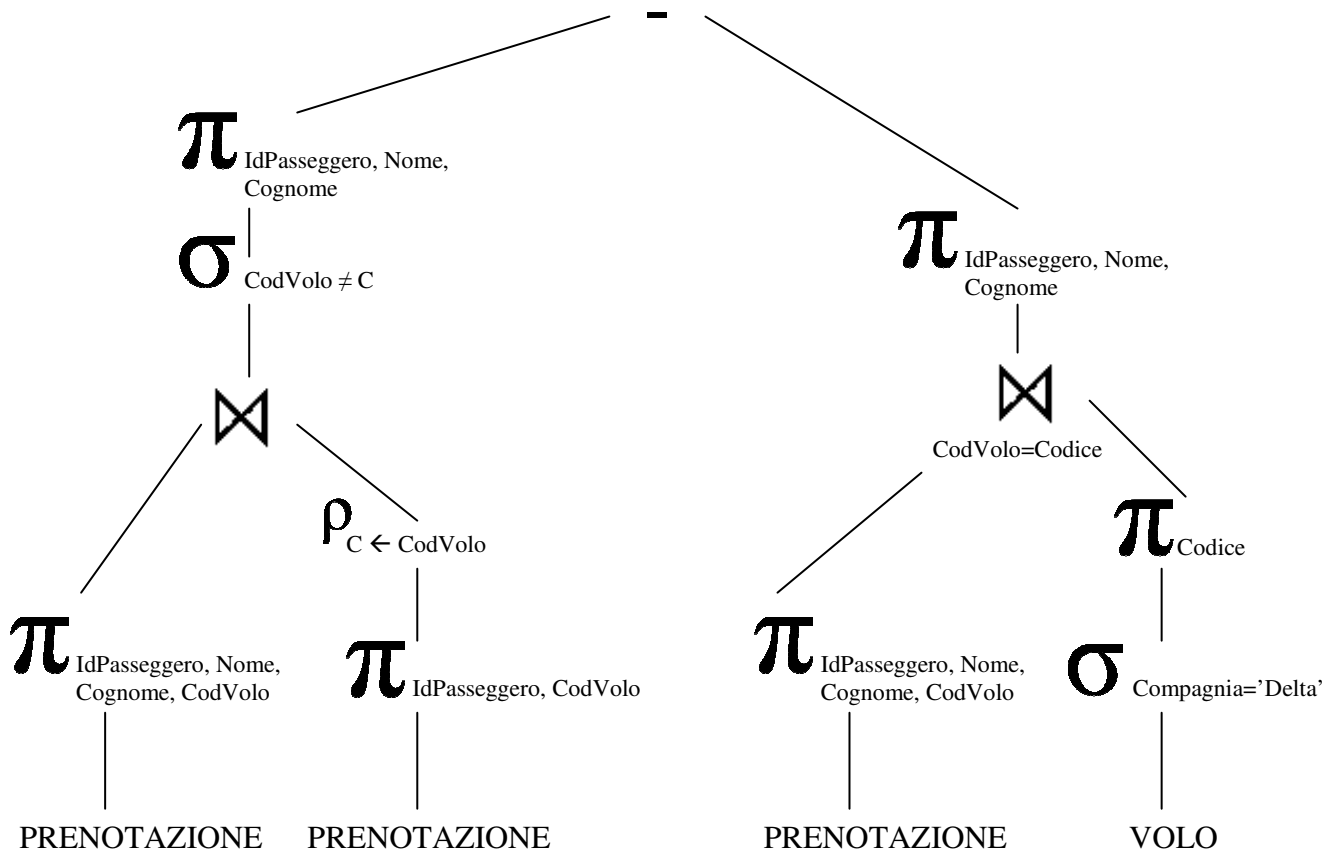
```

### Esercizio A

Esprimere in algebra relazionale, calcolo relazionale, Datalog ed SQL (non richiesto nel tema d'esame) la seguente query:

Trovare id, nome e cognome dei passeggeri che hanno prenotato almeno due voli ma nessuno della compagnia "Delta".

### Algebra relazionale



## Calcolo relazionale

$\{t \mid \exists t1 \in \text{PRENOTAZIONE}, \exists t2 \in \text{PRENOTAZIONE} (t[\text{IdPasseggero}, \text{Nome}, \text{Cognome}] = t1[\text{IdPasseggero}, \text{Nome}, \text{Cognome}] \wedge t1[\text{IdPasseggero}] = t2[\text{IdPasseggero}] \wedge t1[\text{CodVolo}] \neq t2[\text{CodVolo}] \wedge \neg (\exists t3 \in \text{PRENOTAZIONE}, \exists t4 \in \text{VOLO} (t4[\text{Compagnia}] = \text{'Delta'} \wedge t3[\text{CodVolo}] = t4[\text{Codice}] \wedge t3[\text{IdPasseggero}] = t1[\text{IdPasseggero}]))))\}$

## Datalog

PASSEGGERODELTA(IdPasseggero) :- PRENOTAZIONE(IdPasseggero, CodVolo, \_, \_, \_, \_, \_),  
VOLO(CodVolo, "Delta", \_, \_, \_, \_, \_)  
PASSEGGERODUEPREN(IdPasseggero, Nome, Cognome) :- PRENOTAZIONE(IdPasseggero,  
CodVolo, \_, Nome, Cognome, \_, \_), PRENOTAZIONE(IdPasseggero, CodVolo2, \_, \_, \_,  
\_, \_), CodVolo <> CodVolo2  
PASSEGGEROSCELTO(IdPasseggero, Nome, Cognome) :-  
PASSEGGERODUEPREN(IdPasseggero, Nome, Cognome),  
¬PASSEGGERODELTA(IdPasseggero)  
?- PASSEGGEROSCELTO (x, y, z)

## SQL

```
SELECT IdPasseggero, Nome, Cognome
FROM Prenotazione
WHERE IdPasseggero NOT IN ( SELECT P.IdPasseggero
                            FROM Prenotazione AS P JOIN Volo AS V ON
                                P.CodVolo=Volo.Codice
                            WHERE V.Compagnia='Delta'
                            )
GROUP BY IdPasseggero, Nome, Cognome
HAVING COUNT(DISTINCT CodVolo)>=2
```

## ESERCIZI PER CASA

### Tema d'esame del 13 febbraio 2014 – Esercizio B1

DENTISTA (CF, Nome, Cognome, Specializzazione, Telefono)  
STUDIO (Codice, Nome, Indirizzo, Città, Titolare)  
PAZIENTE (CF, Nome, Cognome, Indirizzo, CittàResidenza, Età)  
LAVORO (Dentista, Studio, Paziente, Data, TipoIntervento, Prezzo)

Scrivere in SQL le seguenti interrogazioni:

a) Trovare i dentisti specializzati in Ortodonzia che hanno lavorato in almeno 5 studi diversi e hanno fatto solo interventi di tipo "Ortodonzia" e solo su pazienti minorenni. (3.5 punti)

```
SELECT D.CF
FROM Dentista AS D, Lavoro AS L
WHERE D.Specializzazione='Ortodonzia' AND D.CF NOT IN
    (
        SELECT L1.Dentista
        FROM Lavoro AS L1, Paziente AS P
        WHERE L1.Paziente=P.CF AND (L1.TipoIntervento<>'Ortodonzia' OR
            P.Età >= 18)
    )
GROUP BY D.CF
HAVING COUNT(DISTINCT L.Studio) >= 5
```

b) Trovare, per ogni tipo di intervento, lo studio che ha guadagnato più soldi e il suo titolare.

```
CREATE VIEW TipoStudioSoldi AS (
    SELECT TipoIntervento, Studio, SUM(Prezzo) AS Guadagno
    FROM Lavoro
    GROUP BY TipoIntervento, Studio
)
SELECT T.TipoIntervento, S.Codice, S.Titolare
FROM Studio AS S, TipoStudioSoldi AS T
WHERE S.Codice=T.Studio AND T.Guadagno = (
    SELECT MAX(T2.Guadagno)
    FROM TipoStudioSoldi AS T2
    WHERE T.TipoIntervento = T2.TipoIntervento)
```

Oppure, senza vista:

```
SELECT L.TipoIntervento, S.Codice, S.Titolare
FROM Studio AS S, Lavoro AS L
WHERE S.Codice=L.Studio
GROUP BY L.TipoIntervento, S.Codice, S.Titolare
HAVING SUM(L.Prezzo) >= ALL (
    SELECT SUM(L2.Prezzo)
    FROM Lavoro AS L2
    WHERE L2.TipoIntervento=L.TipoIntervento
    GROUP BY L2.Studio
)
```



## Tema d'esame del 9 settembre 2005 – Esercizio B

CLIENTE(NumTessera, CodFiscCli, Nome, Cognome, Sesso, Via, Città, IscrittoPiscina,  
DataIscrizione)

ISTRUTTORE(CodFiscIstr, Nome, Cognome, Sesso, Via, Città, Data Assunzione)

CORSO(NomeCorso, Turno, CittàClub, CodFiscIstr)

ISCRIZIONICORSI(NumTessera, NomeCorso, Turno, DataInizio)

1) Trovare il numero totale di iscritti di ogni corso che abbia più di 20 iscritti maschi, e il nome e il sesso dei relativi istruttori.

```
SELECT C.NomeCorso, C.Turno, I.Nome, I.Sesso, COUNT(*)
FROM IscrizioniCorsi AS IC, Corso AS C, Istruttore AS I
WHERE IC.NomeCorso=C.NomeCorso AND IC.Turno=C.Turno AND
      C.CodFiscIstr=I.CodFiscIstr
GROUP BY C.NomeCorso, C.Turno, I.Nome, I.Sesso
HAVING 20< ( SELECT COUNT(*)
              FROM IscrizioniCorsi AS IC2, Cliente AS Cli
              WHERE IC2.NumTessera=Cli.NumTessera AND
                    IC2.NomeCorso=C.NomeCorso AND IC2.Turno=C.Turno AND
                    Cli.Sesso='M'
            )
```

2) Trovare i turni dei corsi di aerobica che si tengono nel club di Milano e che hanno almeno due iscritti di Sesto San Giovanni ma nessun iscritto di Lodi.

```
SELECT Corso.Turno
FROM IscrizioniCorsi, Corso, Cliente
WHERE IscrizioniCorsi.NomeCorso=Corso.NomeCorso AND IscrizioniCorsi.Turno=Corso.Turno
      AND IscrizioniCorsi.NumTessera=Cliente.NumTessera AND Corso.CittàClub='Milano'
      AND Cliente.Città='Sesto San Giovanni' AND Corso.NomeCorso='Aerobica'
GROUP BY Corso.Turno
HAVING COUNT(*)>=2
```

**EXCEPT**

```
SELECT IscrizioniCorsi.Turno
FROM IscrizioniCorsi, Cliente
WHERE IscrizioniCorsi.NumTessera=Cliente.NumTessera AND
      IscrizioniCorsi.NomeCorso='Aerobica' AND Cliente.Città='Lodi'
```