

# Esercitazione SQL 4

# Esercizio 1

APPARTAMENTO(CodA, Superficie, Indirizzo, Citta)

CONTRATTO-AFFITTO(CodA, DataInizio, DataFine, NomePersona, RettaMensile)

- a) Trovare il nome delle persone che hanno stipulato più di due contratti di affitto per lo stesso appartamento (in tempi diversi).
- b) Trovare il codice e l'indirizzo degli appartamenti di Torino in cui la retta mensile è sempre stata superiore a 500 euro e per cui sono stati stipulati al più 5 contratti.

# Esercizio 1 soluzione query (a)

```
SELECT NomePersona  
FROM CONTRATTO-AFFITTO  
GROUP BY CodA, NomePersona  
HAVING COUNT(*) > 2
```

# Esercizio 1 soluzione query (b)

```
SELECT A.CodA, Indirizzo
FROM APPARTAMENTO A, CONTRATTO-AFFITTO CA
WHERE A.CodA=CA.CodA AND Citta='Torino'
AND A.CodA NOT IN (SELECT CodA
                   FROM CONTRATTO-AFFITTO
                   WHERE RettaMensile<=500)
GROUP BY A.CodA, Indirizzo
HAVING COUNT(*)<=5;
```

# Esercizio 1 soluzione query (b)

```
SELECT A.CodA,Indirizzo
FROM APPARTAMENTO A, CONTRATTO-AFFITTO CA
WHERE A.CodA=CA.CodA
AND Citta='Torino'
GROUP BY A.CodA, Indirizzo
HAVING COUNT(*)<=5 AND MIN(RettaMensile)>500;
```

# Esercizio 2

GARA(CodG, Luogo, Data, Disciplina)

ATLETA(CodA, Nome, Nazione, DataNascita)

PARTECIPAZIONE(CodG, CodA, PosizioneArrivo, Tempo)

- a) Trovare il nome e la data di nascita degli atleti italiani che non hanno partecipato a nessuna gara di discesa libera.
- b) Trovare le nazioni per cui concorrono almeno 5 atleti nati prima del 1980, ciascuno dei quali abbia partecipato ad almeno 10 gare di sci di fondo.

# Esercizio 2 soluzione query (a)

```
SELECT Nome,DataNascita
FROM ATLETA
WHERE Nazione='Italia' AND
CodA NOT IN (SELECT CodA FROM PARTECIPAZIONE P,GARA G
              WHERE P.CodG=G.CodG AND
              G.Disciplina='discesa libera');
```

# Esercizio 2 soluzione query (b)

```
SELECT Nazione
FROM ATLETA
WHERE DataNascita<'1/1/1980'
AND CodA IN (SELECT CodA FROM PARTECIPAZIONE P,GARA G
              WHERE P.CodG=G.CodG AND Disciplina='fondo'
              GROUP BY CodA HAVING COUNT(*)>=10)
GROUP BY Nazione
HAVING COUNT(*)>=5;
```

# Esercizio 3

QUIZ(CodQuiz, Argomento, Punteggio)

STUDENTE(Matricola, Nome, Indirizzo, Citta)

RISULTATO TEST(Matricola, CodQuiz, RispostaCorretta)

- a) Trovare il nome degli studenti che non hanno risposto correttamente a nessun quiz di matematica.
- b) Trovare il nome degli studenti di Torino che hanno conseguito il punteggio massimo possibile nei quiz di matematica.

# Esercizio 3 soluzione query (a)

```
SELECT Nome  
FROM STUDENTE S  
WHERE Matricola NOT IN
```

```
(SELECT Matricola  
FROM RISULTATO_TEST R,QUIZ Q  
WHERE R.CodQuiz=Q.CodQuiz  
AND RispostaCorretta='si' AND  
Argomento='matematica');
```

# Esercizio 3 soluzione query (b)

```
SELECT Nome
FROM STUDENTE S,RISULTATO_TEST R, QUIZ Q
WHERE S.Matricola=R.Matricola AND Q.CodQuiz=R.CodQuiz AND Citta='Torino'
AND RispostaCorretta= 'si' AND Argomento='matematica'
GROUP BY S.Matricola,Nome
HAVING SUM(Punteggio)=(
SELECT SUM(Punteggio) FROM QUIZ WHERE Argomento='matematica'
);
```

# Esercizio 4

AEREI(Mat, Modello, NumPosti)

ORARI (Sigla, ParteDa, Destinaz, OraPart, OraArr)

VOLI (Sigla, Matr, Data, PostiPren)

Trovare le tratte (città di partenza, città di arrivo) che non sono state mai effettuate con un aereo modello Boing-747.

# Esercizio 4 soluzione

```
SELECT ParteDa, Destinaz
FROM ORARIO O1
WHERE NOT EXISTS
(SELECT * FROM ORARIO O2, VOLI V, AEREI A
WHERE O2.Sigla=V.Sigla AND A.Matr=V.Matr AND
A.Modello='Boing-747' AND O2.ParteDa=O1.ParteDa
AND O2.Destinaz=O1.Destinaz);
```

# Esercitazione SQL 5

MECCANICO(MatrM, NomeM)

SA-RIPARARE(MatrM, TipoGuasto)

EFFETTUA-RIPARAZIONE(CodR, MatrM, Targa, Data, Durata, TipoGuasto)

- (a) Trovare il nome dei meccanici che hanno effettuato almeno una riparazione di un guasto che non sapevano riparare.
- (b) Trovare il meccanico che ha effettuato più riparazioni di guasti che non sapeva riparare (se ci fossero più meccanici a pari merito selezionare quello col primo nome in ordine alfabetico).
- (c) Trovare il nome del meccanico e il numero totale di riparazioni del meccanico che ha effettuato più riparazioni totali tra quelli che hanno effettuato almeno una riparazione di un guasto che non sapevano riparare (se ci fossero più meccanici a pari merito selezionare quello col primo nome in ordine alfabetico).

# Soluzione A

```
SELECT NomeM
FROM MECCANICO M, EFFETTUA-RIPARAZIONE ER
WHERE M.MatrM=ER.MatrM
AND (ER.MatrM, TipoGuasto) NOT IN
(SELECT MatrM, TipoGuasto
FROM SA-RIPARARE)
```

# Soluzione B

```
SELECT NomeM
FROM MECCANICO M, EFFETTUA-RIPARAZIONE ER
WHERE M.MatrM=ER.MatrM
AND (ER.MatrM, TipoGuasto) NOT IN
(SELECT MatrM, TipoGuasto
FROM SA-RIPARARE)
GROUP BY M.MatrM, NomeM
ORDER BY RiparazioniTotali DESC, NomeM
LIMIT 1
```

# Soluzione C

```
SELECT NomeM, COUNT(*) AS RiparazioniTotali
FROM MECCANICO M, EFFETTUA-RIPARAZIONE ER
WHERE M.MatrM=ER.MatrM
AND ER.MatrM IN(
    SELECT MatrM
    FROM EFFETTUA-RIPARAZIONE
    WHERE (MatrM, TipoGuasto) NOT IN (SELECT MatrM, TipoGuasto
                                     FROM SA-RIPARARE)
)
GROUP BY M.MatrM, NomeM
ORDER BY RiparazioniTotali DESC, NomeM
LIMIT 1
```

MECCANICO(MatrM, NomeM)

SA-RIPARARE(MatrM, TipoGuasto)

EFFETTUA-RIPARAZIONE(CodR, MatrM, Targa, Data, Durata, TipoGuasto)

- (d) Trovare i nomi e il numero totale di giorni di lavoro dei meccanici che hanno effettuato almeno una riparazione di un guasto che non sapevano riparare.
- (e) Per le autovetture per cui sono state necessarie riparazioni effettuate da almeno 3 meccanici diversi nello stesso giorno, visualizzare la targa dell'autovettura, la data delle riparazioni e i tipi di guasto che si sono verificati, ordinando il risultato in ordine crescente di targa e decrescente di data.

# Soluzione D

```
SELECT NomeM, COUNT(DISTINCT Data) AS GiorniDiLavoro
FROM MECCANICO M, EFFETTUA-RIPARAZIONE ER
WHERE M.MatrM=ER.MatrM
AND ER.MatrM IN(
    SELECT MatrM
    FROM EFFETTUA-RIPARAZIONE
    WHERE (MatrM, TipoGuasto) NOT IN (SELECT MatrM, TipoGuasto
                                      FROM SA-RIPARARE)
)
GROUP BY M.MatrM, NomeM
```

# Soluzione E

```
SELECT Targa, Data, TipoGuasto
FROM EFFETTUA-RIPARAZIONE
WHERE (Targa, Data) IN (SELECT Targa, Data
                        FROM EFFETTUA-RIPARAZIONE
                        GROUP BY Targa, Data
                        HAVING COUNT(DISTINCT MatrM)>=3)
ORDER BY Targa ASC, Data DESC;
```

# Esercitazione SQL 6

CAMPEGGIO (CodCampeggio, NomeC, Indirizzo, Comune, Provincia)

PIAZZOLA (NumeroPiazzola, CodCampeggio, Superficie, Tipologia, CostoGiornaliero)

PRENOTAZIONE (NumeroPiazzola, CodCampeggio, DataInizio, DataFine, NoneCliente )

- a) Visualizzare numero piazzola, codice campeggio e tipologia per le piazzole con superficie superiore a 20mq che sono state prenotate solo nel mese di agosto 2018.
- b) Per i campeggi nella provincia di Napoli che hanno ricevuto più di 1000 prenotazioni diverse per piazzole di tipologia fronte-mare, visualizzare codice del campeggio, indirizzo, comune, durata media delle prenotazioni e importo totale delle prenotazioni (calcolato come prodotto della durata della prenotazione per il costo giornaliero).
- c) Visualizzare il codice e il nome dei campeggi in cui tutte le piazzole con costo giornaliero superiore a 15 euro sono state prenotate almeno una volta.

# Esercizio 1 soluzione query (a)

```
SELECT NumeroPiazzola, CodCampeggio, Tipologia
FROM PIAZZOLA P, PRENOTAZIONE PR
P. NumeroPiazzola = PR.NumeroPiazzola
WHERE (NumeroPiazzola, CodCampeggio) NOT IN (
    SELECT NumeroPiazzola, CodCampeggio
    FROM PRENOTAZIONE
    WHERE DataInizio > '31/10/2018' OR DataFine < '1/10/2018'
)
AND Superficie > '20'
```

# Esercizio 1 soluzione query (b)

```
SELECT C.CodCampeggio, Indirizzo, Comune,  
AVG(DataFine-DataInizio), SUM( (DataFine-DataInizio)*CostoGiornaliero)  
FROM CAMPEGGIO C, PIAZZOLA P, PRENOTAZIONE PR  
WHERE C.CodCampeggio = P.CodCampeggio  
AND P. NumeroPiazzola = PR.NuneroPiazzola  
AND P.CodCampeggio = PR.CodCampeggio  
AND Provincia = 'Napoli'  
AND Tipologia = 'fronte-mare'  
GROUP BY C.CodCampeggio, Indirizzo, Comune  
HAVING COUNT(*) > 1000
```

# Esercizio 1 soluzione query (c)

```
SELECT C.CodCampeggio, NomeC
FROM CAMPEGGIO C, PIAZZOLA P, PRENOTAZIONE PR
WHERE C.CodCampeggio = P.CodCampeggio
AND P. NumeroPiazzola = PR.NumeroPiazzola
AND P.CodCampeggio = PR.CodCampeggio
AND CostoGiornaliero > 15
GROUP BY C.CodCampeggio, NomeC
HAVING COUNT(DISTINCT PR.NumeroPiazzola) =
(SELECT COUNT(*)
FROM PIAZZOLA P1
WHERE P1.CodCampeggio = P.CodCampeggio
AND CostoGiornallero > 15)
```

## Parte A - SQL

1. Sono date le relazioni seguenti (le chiavi primarie sono sottolineate):

```
FATTORINO(CodF, Nome, Cognome, AnnoNascita, Email)
AZIENDA(PartitaIVA, NomeAzienda, Indirizzo, Categoria, NumDipendenti)
CONSEGNA(PartitaIVA, Data, Ora, CodF, NumPacchi, Urgente)
```

Esprimere le seguenti interrogazioni

- (a) Scrivere la seguente query in linguaggio SQL (6 punti):

Per i fattorini che hanno effettuato consegne ad aziende con più di 30 dipendenti in almeno 12 giorni differenti nel periodo di Ottobre 2019, visualizzare codice, anno di nascita, email e la data dell'ultima consegna effettuata dal fattorino (in un qualunque periodo, non solo a Ottobre 2019).

```
SELECT CodF, AnnoNascita, Email, DataUltimaConsegna
FROM CONSEGNA C, FATTORINO F, AZIENDA A,
     (SELECT CodF, MAX(Data) AS DataUltimaConsegna
      FROM CONSEGNA C2
      GROUP BY C2.CodF) ULT
WHERE C.CodF = F.CodF
AND C.PartitaIVA = A.PartitaIVA
AND F.CodF = ULT.CodF
AND A.NumDipendenti > 30
AND (Data >= 1/10/2019 AND Data <= 31/10/2019)
GROUP BY CodF, AnnoNascita, Email, DataUltimaConsegna
HAVING COUNT(DISTINCT Data) >= 12;
```

Soluzione alternativa:

```
SELECT CodF, AnnoNascita, Email, Max(Data)
FROM CONSEGNA C, FATTORINO F
WHERE C.CodF = F.CodF
AND CodF IN (SELECT CodF
             FROM CONSEGNA C2, AZIENDA A
             WHERE C2.PartitaIVA = A.PartitaIVA
             AND A.NumDipendenti > 30
             AND (Data >= 1/10/2019 AND Data <= 31/10/2019)
             GROUP BY CodF
             HAVING COUNT (DISTINCT (Data)) >= 12)
GROUP BY CodF, AnnoNascita, Email;
```