



SQL

SQL: esercizi

Esercizio #1

ALLOGGIO(CodA, Indirizzo,Citta,Superficie,CostoAttoMensile)

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

Trovare il codice, l'indirizzo e la città degli alloggi che hanno una superficie superiore alla superficie media degli alloggi delle città in cui si trovano.

```
SELECT CodA, Indirizzo, Citta
FROM ALLOGGIO A1
WHERE Superficie > (SELECT AVG(Superficie)
                    FROM ALLOGGIO A2
                    WHERE A2.Citta = A1.Citta);
```

Esercizio #2

MECCANICO(MatrM, NomeM)

SA-RIPARARE(MatrM, TipoGuasto)

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

Trovare il nome dei meccanici che hanno effettuato almeno una riparazione di un guasto che non sapevano riparare.

```
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);
```

Esercizio #3

MECCANICO(MatrM, NomeM)

SA-RIPARARE(MatrM, TipoGuasto)

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

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.

```
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;
```

Esercizio #4

SALA RIUNIONI(CodS, NumeroMaxPosti, Proiettore)

PRENOTAZIONE_SALA(CodS, Data, OraInizio, OraFine, CodDip)

DIPENDENTE(CodDip, Nome, Cognome, DataNascita, Città)

Visualizzare per ogni sala il codice della sala, il numero massimo di posti e il numero di prenotazioni considerando solo l'ultima data in cui la sala è stata prenotata.

```
SELECT S.CodS, NumeroMaxPosti, COUNT(*)
FROM PRENOTAZIONE_SALA PS1, SALA S
WHERE PS.Cods=S.CodS
AND DATA = (SELECT MAX(Data)
             FROM PRENOTAZIONE_SALA PS2
             WHERE PS2.Cods=PS1.Cods)
GROUP BY S.CodS, NumeroMaxPosti;
```

Esercizio #4

SALA RIUNIONI(CodS, NumeroMaxPosti, Proiettore)

PRENOTAZIONE_SALA(CodS, Data, OraInizio, OraFine, CodDip)

DIPENDENTE(CodDip, Nome, Cognome, DataNascita, Città)

Visualizzare per ogni sala il codice della sala, il numero massimo di posti e il numero di prenotazioni considerando solo l'ultima data in cui la sala è stata prenotata.

```
SELECT S.CodS, NumeroMaxPosti, COUNT(*)
FROM PRENOTAZIONE_SALA PS1, SALA S,
      (SELECT PS2.CodS, MAX(Data) AS MaxData
       FROM PRENOTAZIONE_SALA PS2
       GROUP BY PS2.Cods) AS LAST
WHERE PS.Cods=S.CodS
AND PS.CodS = LAST.CodS
AND DATA = LAST.MaxData
GROUP BY S.CodS, NumeroMaxPosti;
```

Esercizio #5

SALA RIUNIONI(CodS, NumeroMaxPosti, Proiettore)

PRENOTAZIONE_SALA(CodS, Data, OraInizio, OraFine, CodDip)

DIPENDENTE(CodDip, Nome, Cognome, DataNascita, Città)

Visualizzare il codice e il numero massimo di posti delle sale dotate di proiettore che sono state prenotate almeno 15 volte per riunioni che iniziano prima delle ore 15:00, ma non sono mai state prenotate per riunioni che cominciano dopo le ore 20:00.

Esercizio #5

```
SELECT S.CodS, NumeroMaxPosti
FROM PRENOTAZIONE_SALA PS1, SALA S
WHERE PS.Cods=S.CodS
AND Proiettore='si'
AND OraInizio<'15:00'
AND S.Cods NOT IN (Select CodS
                   FROM PRENOTAZIONE_SALA PS2
                   AND OraInizio>'20:00')

GROUP BY S.CodS, NumeroMaxPosti
HAVING COUNT(*)>=15;
```


Esercizio #6

GUIDA (CodGuida, Nome, Cognome, Nazionalità)

TIPO-VISITA (CodTipoVisita, Monumento, Durata, Città)

GRUPPO (CodGR, NumeroPartecipanti, Lingua)

VISITA-GUIDATA-EFFETTUATA (CodGR, Data, OraI, CodTipoVisita, CodGuida)

Tra i monumenti per cui sono state effettuate almeno 10 visite guidate, visualizzare il monumento che è stato visitato complessivamente dal maggior numero di persone.

Esercizio #6

```
SELECT Monumento
FROM TIPO-VISITA TP, VISITA-GUIDATA-EFFETTUATA VGE, GRUPPO G
WHERE TP.CodTipoVisita = VGE.CodTipoVisita AND G.CodGR = VGE.CodGR
GROUP BY Monumento
HAVING COUNT(*) >=10 AND
SUM(NumeroPartecipanti) = (SELECT MAX(TOTPart)
FROM (SELECT SUM(NumeroPartecipanti) AS TOTPart
FROM TIPO-VISITA TP1,
VISITA-GUIDATA-EFFETTUATA VGE1,
GRUPPO G1
WHERE TP1.CodTipoVisita = VGE1.CodTipoVisita
AND G1.CodGR = VGE1.CodGR
GROUP BY Monumento
HAVING COUNT(*) >= 10));
```

Esercizio #7

RAGAZZO(CodFiscale, Nome, Cognome, DataNascita, CittàResidenza)

ATTIVITA'(CodAttività, NomeA, Descrizione, Categoria)

CAMPO-ESTIVO(CodCampo, NomeCampo, Città)

ISCRIZIONE-PER-ATTIVITA'-IN-CAMPO-ESTIVO(CodFiscale, CodAttività,
CodCampo, DataIscrizione)

Visualizzare il nome e cognome del ragazzo che ha partecipato al maggior numero di campi estivi per l'attività della categoria «Tennis».

Esercizio #7

```
SELECT Nome, Cognome
FROM R, A, IPA
WHERE R.CodFiscale = IPA.CodFiscale
AND A.CodAttività = IPA.CodAttività
AND Categoria = 'Tennis'
GROUP BY R.CodFiscale, Nome, Cognome
HAVING COUNT(*) =
    (SELECT MAX(NUMCAMPPI)
     FROM (SELECT COUNT(*) AS NUMCAMPPI
           FROM IPA, A
           WHERE A.CodAttività = IPA.CodAttività
           AND Categoria = 'Tennis'
           GROUP BY CodFiscale) AS NUMCAMPPIPER RAGAZZO)
```