

# Soluzioni Esercizi SQL

## Esercizio 1

Data la tabella: Clienti (Cognome, Nome, Citta, Salario, Eta)

Bianchi	Mario	Rimini	1000	20
Bianchi	Ettore	Milano	0	15
Casadei	Mario	Rimini	3000	35
Rossi	Mario	Bologna	1500	50
Rossi	Fabio	Firenze	8000	40
Bianchi	Ettore	Rimini	4500	25
Neri	Fabio	Arezzo	3500	35

Scrivere il codice SQL che permetta di:

### 1. Visualizzare il salario massimo dei clienti che abitano a Rimini.

```
SELECT MAX(Salario) AS Salario_massimo
FROM Clienti
WHERE Citta = 'Rimini';
```

### 2. Per ogni città, visualizzare il nome della città, il numero di abitanti e la loro età media.

```
SELECT Citta, COUNT(*) AS Nabitanti, AVG(Eta) AS Etamedia
FROM Clienti
GROUP BY Citta;
```

### 3. Visualizzare il nome della città e l'età media, delle città con età media maggiore di 30 anni.

```
SELECT Citta, AVG(Eta) AS Etamedia
FROM Clienti
GROUP BY Citta
HAVING AVG(Eta) > 30;
```

## Esercizio 2

Dato il seguente schema relazionale: INSEGNANTE(Matricola, Nome, Cognome, Città, Telefono, Stipendio)

Scrivere le interrogazioni SQL che restituiscono le seguenti informazioni:

### 1. Elencare l'insegnante o gli insegnanti con lo stipendio più alto tra tutti.

```
SELECT Nome, Cognome, Matricola
FROM INSEGNANTE
WHERE Stipendio = (SELECT MAX(Stipendio) FROM INSEGNANTE);
```

### 2. Elencare il nome e il cognome degli insegnanti che vivono a Milano o a Verona e guadagnano più di 2000 euro.

```
SELECT Nome, Cognome
FROM INSEGNANTE
WHERE (Città = 'Milano' OR Città = 'Verona') AND Stipendio > 2000;
```

### 3. Trovare la somma degli stipendi dei professori che guadagnano più di 2000 euro.

```
SELECT SUM(Stipendio) AS Somma_stipendi
FROM INSEGNANTE
WHERE Stipendio > 2000;
```

## Esercizio 3

Dato il seguente schema relazionale: Dischi (CodDisco, Titolo, CodAutore\*, dataPubblicazione, genere, ncopieVendute) Autori (CodAutore, Nome)

Scrivere le seguenti query:

### 1. Visualizzare il nome dell'autore con il numero massimo di dischi venduti.

```
SELECT Autori.Nome, SUM(Dischi.ncopieVendute) AS Somma_copie_vendute
FROM Autori INNER JOIN Dischi ON Autori.CodAutore = Dischi.CodAutore
GROUP BY Autori.Nome
HAVING SUM(Dischi.ncopieVendute) = (
    SELECT MAX(TotaleVendite)
    FROM (
        SELECT SUM(ncopieVendute) AS TotaleVendite
        FROM Dischi
```

```
        GROUP BY CodAutore
    ) AS VenditePerAutore
);
```

## 2. Visualizzare il numero di copie vendute per tutti i dischi (o il disco) pubblicato il 06/05/2024.

```
SELECT Titolo, ncopieVendute
FROM Dischi
WHERE dataPubblicazione = '2024-05-06';
```

## Esercizio 4

Dato il seguente schema relazionale (dal diagramma mostrato nell'immagine), scrivi la definizione SQL per la creazione della base di dati:

```
CREATE TABLE Attore (
    CodAttore INT AUTO_INCREMENT PRIMARY KEY,
    Cognome VARCHAR(50),
    Nome VARCHAR(50),
    AnnoNascita INT,
    Nazionalita VARCHAR(50)
);

CREATE TABLE Film (
    CodFilm INT AUTO_INCREMENT PRIMARY KEY,
    Titolo VARCHAR(100),
    AnnoProduzione INT,
    Nazionalita VARCHAR(50),
    Regista VARCHAR(50),
    Genere VARCHAR(50),
    Durata INT
);

CREATE TABLE Sala (
    CodSala INT AUTO_INCREMENT PRIMARY KEY,
    Posti INT,
    Nome VARCHAR(50),
    Citta VARCHAR(50)
);

CREATE TABLE Recita (
    CodAttore INT,
    CodFilm INT,
    Compenso DECIMAL(10,2),
    FOREIGN KEY (CodAttore) REFERENCES Attore(CodAttore),
    FOREIGN KEY (CodFilm) REFERENCES Film(CodFilm),
```

```

PRIMARY KEY (CodAttore, CodFilm)
);

CREATE TABLE Proiezione (
    CodProiezione INT AUTO_INCREMENT PRIMARY KEY,
    CodFilm INT,
    CodSala INT,
    DataProiezione DATE,
    Incasso DECIMAL(10,2),
    OraProiezione TIME,
    FOREIGN KEY (CodFilm) REFERENCES Film(CodFilm),
    FOREIGN KEY (CodSala) REFERENCES Sala(CodSala)
);

```

## Esercizio 5

Dell'esercizio e scrivere le seguenti interrogazioni:

### 1. Visualizzare gli attori con età superiore a 40 anni.

```

SELECT Cognome, Nome, AnnoNascita
FROM Attore
WHERE (2025 - AnnoNascita) > 40;

```

### 2. Visualizzare il titolo del o dei film che hanno totalizzato l'incasso maggiore.

```

-- Prima creiamo una vista che calcola l'incasso totale per ogni film
WITH IncassiTotali AS (
    SELECT Film.CodFilm, Film.Titolo, SUM(Proiezione.Incasso) AS
IncassoTotale
    FROM Film INNER JOIN Proiezione ON Film.CodFilm = Proiezione.CodFilm
    GROUP BY Film.CodFilm, Film.Titolo
)
-- Poi selezioniamo i film con l'incasso massimo
SELECT Titolo
FROM IncassiTotali
WHERE IncassoTotale = (SELECT MAX(IncassoTotale) FROM IncassiTotali);

```

### 3. Visualizzare l'incasso totale nella giornata del 01/04/2025.

```
SELECT SUM(Incasso) AS IncassoTotale
FROM Proiezione
WHERE DataProiezione = '2025-04-01';
```