

```
FILM(IDFilm, Titolo)
PROIEZIONE(IDFilm, Incasso)
```

Il film di maggior incasso tra quelli proiettati

```
/* In un primo pezzo, calcolo per tutti l'incasso*/
```

```
(SELECT Film.Titolo, Film.IdFilm, SUM (Proiezione.Incasso) AS Incasso_totale
FROM Film INNER JOIN Proiezione ON
Film.Idfilm = Proiezione.Idfilm
GROUP BY Film.Idfilm, Film.IdFilm) AS Q1;
```

```
/* In un secondo pezzo, che sia il massimo per effettivamente tutti*/
```

```
SELECT Q1.titolo
FROM Q1
WHERE incassotot=(SELECT MAX(incassotot) FROM Q1);
```

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

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

```
/* 1 - Numero dischi venduti */
```

```
(SELECT Nome, SUM(NCopieVendute) AS Numero_dischi_venduti
FROM Autore
JOIN Dischi AS Autore.CodAutore = Dischi.CodAutore) AS Q1;
```

```
/* 2 - Ma il numero di dischi è effettivamente il massimo? */
```

```
SELECT Q1.Nome
FROM Q1
WHERE Numero_dischi_venduti = (SELECT MAX(Numero_dischi_venduti) FROM Q1);
```

# Parole chiave SQL

- HAVING = Come WHERE ma per il COUNT

Contare gli album che hanno venduto più di 10 milioni di copie.

```
SELECT CodDisco, Titolo, COUNT(*) AS N_dischi
FROM Dischi
HAVING N_dischi >= "10.000.000";
```

- IN = Sottoquery = Query sulla stessa tabella
  - Prendi nella query solo le righe che rispettano quella specifica condizione

Gli album che devono ancora uscire -> Prendi solo quei dischi là.

- Logica: DataPubblicazione = NULL !

```
SELECT CodDisco, Titolo
FROM Dischi
WHERE CodDisco IN (SELECT CodDisco
                   FROM Dischi
                   WHERE DataPubblicazione IS NULL)
```

Appunto: Possiamo usare NOT IN (in questo caso, prendi solo gli album che hanno data nulla = Evita quelli che hanno una data non nulla)

```
SELECT CodDisco, Titolo
FROM Dischi
WHERE CodDisco NOT IN (SELECT CodDisco
                      FROM Dischi
                      WHERE DataPubblicazione IS NOT NULL)
```

- BETWEEN
  - Tra una data e l'altra

Gli album usciti tra il 2015 e il 2020 scritti da Bob Dylan.

```
SELECT Titolo, CodDisco
FROM Dischi
JOIN Autore AS Autore.CodAutore = Dischi.CodAutore
WHERE DataPubblicazione BETWEEN 2015 AND 2020
```

- ORDER BY
  - Ordinamento di solito crescente dei campi

Gli album usciti tra il 2015 e il 2020 scritti da Bob Dylan in ordine alfabetico.

```
SELECT Titolo, CodDisco
FROM Dischi
JOIN Autore AS Autore.CodAutore = Dischi.CodAutore
WHERE DataPubblicazione BETWEEN 2015 AND 2020
ORDER BY Titolo;
```

- LIKE

Si usa "%" per fare "match" con quella specifica combinazione di caratteri

I dischi che hanno "Thriller" nel nome

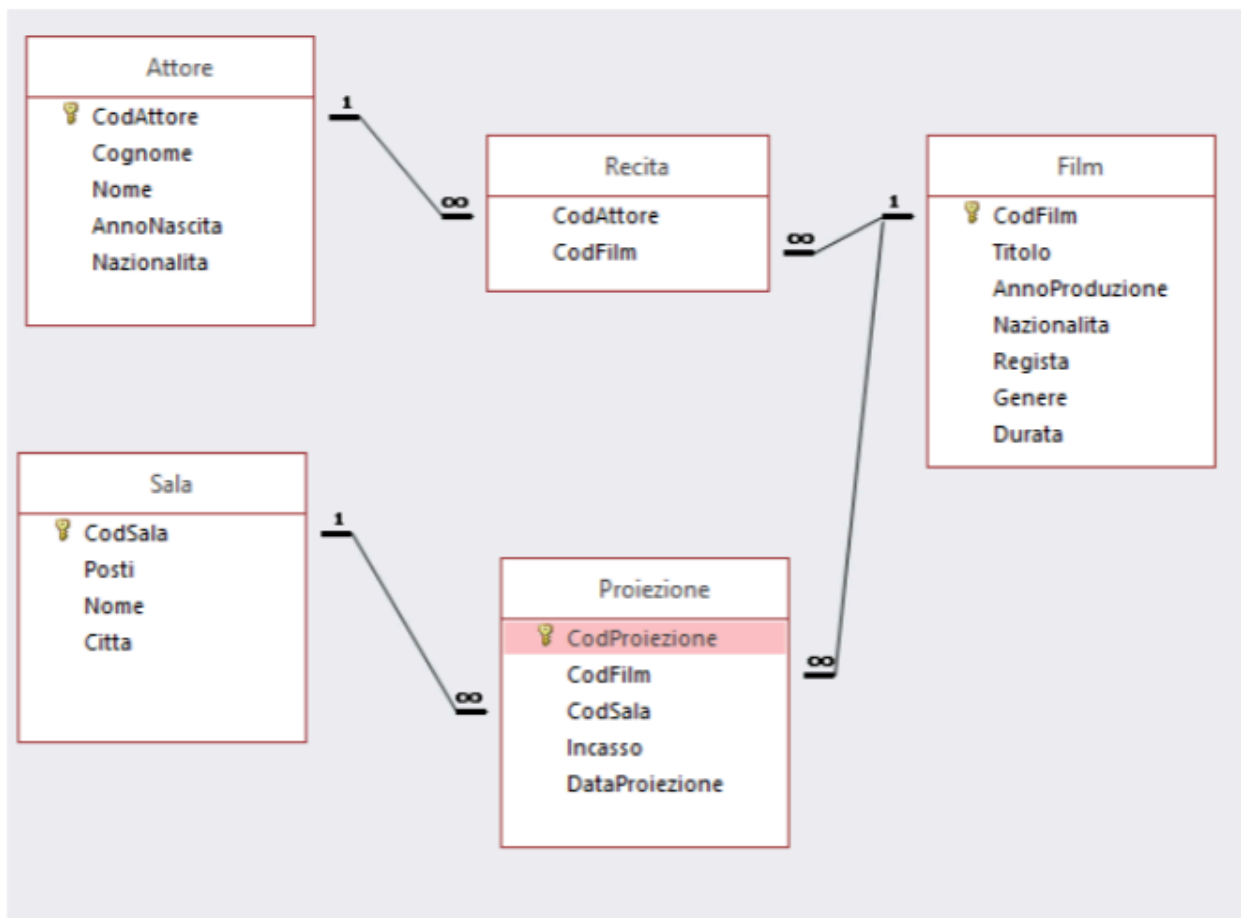
```
SELECT Titolo, CodDisco
FROM Dischi
WHERE TITOLO "Thriller%"
```

- AS

Alias dei campi o delle righe

```
... SELECT COUNT(*) AS N_righe
```

## Creazione tabelle



Esempio:

```
CREATE TABLE Cliente(  
    Codcliente INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50),  
    Cognome VARCHAR(50),  
    Citta VARCHAR(50),  
    Salario INT,  
    DataNascita DATE  
);  
  
CREATE TABLE Albergo(  
    Codalbergo INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50),  
    Citta VARCHAR(50)  
);  
  
CREATE TABLE Prenot(  
    Dataprenot DATE,  
    Acconto INT,  
    Codcliente INT,  
    Codalbergo INT,  
    FOREIGN KEY(codcliente) REFERENCES Cliente(codcliente),  
    FOREIGN KEY(codalbergo) REFERENCES Albergo(codalbergo)  
);
```