# Esercitazione finale SQL

## Esercizio

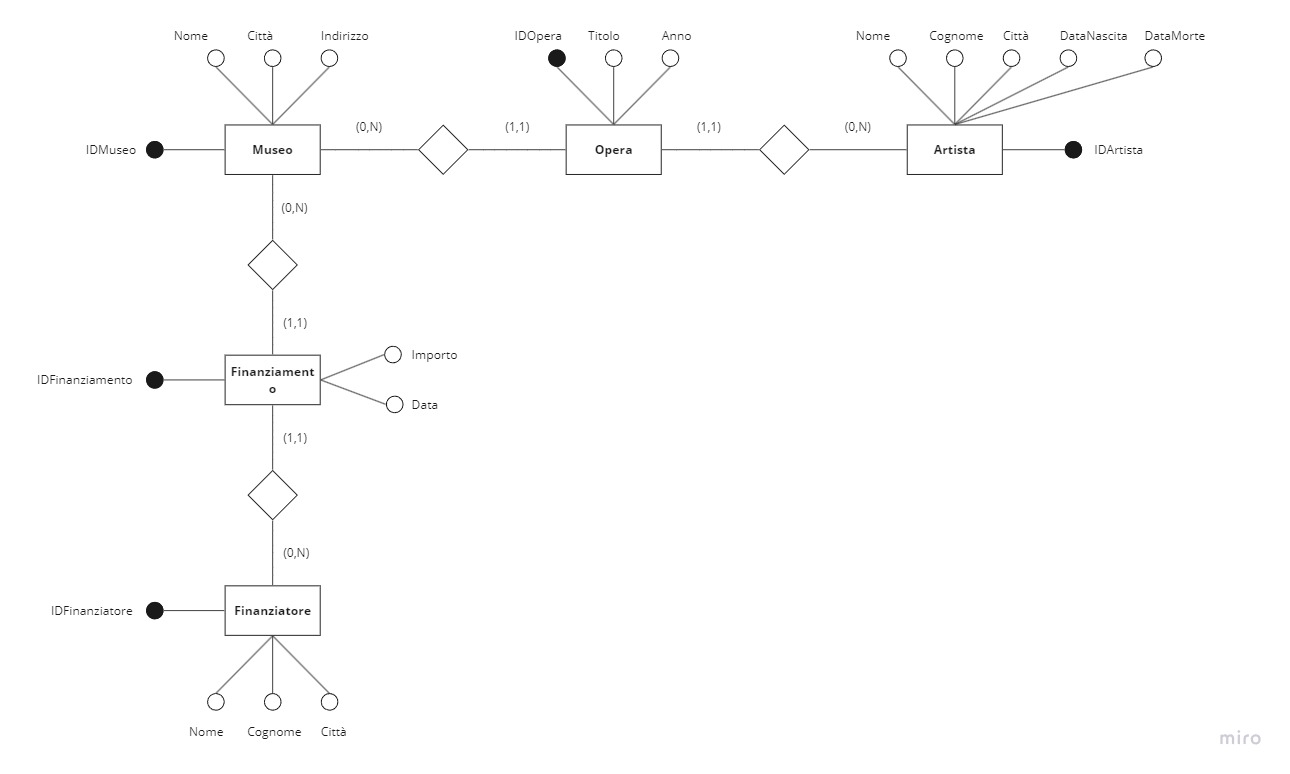
A partire da un caso di studio di suo interesse (es. ecommerce di prodotti cosmetici; dipendenti di una multinazionale farmaceutica, personale amministrativo, docente e studenti di un liceo; utenti, pagine, gruppi di un social network), si esegua la progettazione di una base dati (almeno 5 relazioni dello schema concettuale). La progettazione consiste di un documento con:

* Descrizione caso di studio scelto
* Modello ER
* Modello logico
* Progettazione fisica (codice DDL e DML per la creazione della base di dati, delle tabelle, delle chiavi primarie ed esterne, degli attributi), creare delle viste.
* Almeno 10 interrogazioni (tutte con almeno due tra raggruppamenti e ordinamenti, join e date)

## Descrizione caso di studio scelto

In questo caso studio viene progettato in maniera semplice un sistema di gestione per musei che desiderano tenere traccia delle opere d’arte, degli artisti, dei finanziamenti ricevuti e dei finanziatori coinvolti. Ogni museo ha una collezione di diverse opere e ogni opera è creata da un artista specifico.   
  
Ogni tabella contiene informazioni specifiche delle varie entità in modo tale da ottenere informazioni dettagliate sulle varie collezioni, artisti ecc.

## Modello ER



## Modello funzionale

## 

## Progettazione fisica

CREATE DATABASE SistemaGestioneMusei;

-- Creazione delle tabelle (codice DDL)

CREATE TABLE Museo (

IDMuseo INT PRIMARY KEY,

Nome VARCHAR(100),

Città VARCHAR(100),

Indirizzo VARCHAR(200)

);

CREATE TABLE Artista (

IDArtista INT PRIMARY KEY,

Nome VARCHAR(100),

Cognome VARCHAR(100),

Città VARCHAR(100),

DataNascita DATE,

DataMorte DATE

);

CREATE TABLE Opera (

IDOpera INT PRIMARY KEY,

Titolo VARCHAR(200),

Anno INT,

IDMuseo INT,

IDArtista INT,

FOREIGN KEY (IDMuseo) REFERENCES Museo(IDMuseo),

FOREIGN KEY (IDArtista) REFERENCES Artista(IDArtista)

);

CREATE TABLE Finanziatore (

IDFinanziatore INT PRIMARY KEY,

Nome VARCHAR(100),

Cognome VARCHAR(100),

Città VARCHAR(100)

);

CREATE TABLE Finanziamento (

IDFinanziamento INT PRIMARY KEY,

Importo DECIMAL(10, 2),

Data DATE,

IDMuseo INT,

IDFinanziatore INT,

FOREIGN KEY (IDMuseo) REFERENCES Museo(IDMuseo),

FOREIGN KEY (IDFinanziatore) REFERENCES Finanziatore(IDFinanziatore)

);

-- Inserimento dati in ogni tabella (Codice DML)

INSERT INTO Museo (IDMuseo, Nome, Città, Indirizzo)

VALUES

(1, 'Museo di Arte Moderna', 'Roma', 'Via Artistica, 123'),

(2, 'Galleria d''Arte Contemporanea', 'Milano', 'Corso Creativo, 456'),

-- ... altre righe ...

INSERT INTO Artista (IDArtista, Nome, Cognome, Città, DataNascita, DataMorte)

VALUES

(1, 'Leonardo', 'Da Vinci', 'Firenze', '1452-04-15', '1519-05-02'),

(2, 'Vincent', 'Van Gogh', 'Zundert', '1853-03-30', '1890-07-29'),

-- ... altre righe ...

INSERT INTO Opera (IDOpera, Titolo, Anno, IDMuseo, IDArtista)

VALUES

(1, 'Mona Lisa', 1503, 1, 1),

(2, 'Notte Stellata', 1889, 2, 2),

-- ... altre righe ...

INSERT INTO Finanziatore (IDFinanziatore, Nome, Cognome, Città)

VALUES

(1, 'John', 'Doe', 'New York'),

(2, 'Alice', 'Smith', 'Los Angeles'),

-- ... altre righe ...

INSERT INTO Finanziamento (IDFinanziamento, Importo, Data, IDMuseo, IDFinanziatore)

VALUES

(1, 10000.00, '2023-08-01', 1, 1),

(2, 5000.00, '2023-08-10', 2, 2),

## Creazione Viste

-- Vista: Elenco opere e informazioni dell'artista

CREATE VIEW OpereConArtista AS

SELECT Opera.IDOpera, Opera.Titolo, Artista.Nome AS NomeArtista, Artista.Cognome AS CognomeArtista

FROM Opera

JOIN Artista ON Opera.IDArtista = Artista.IDArtista;

-- Vista: Totale finanziamenti per ogni museo

CREATE VIEW TotaleFinanziamentiMusei AS

SELECT Museo.Nome, SUM(Finanziamento.Importo) AS TotaleFinanziamenti

FROM Museo

LEFT JOIN Finanziamento ON Museo.IDMuseo = Finanziamento.IDMuseo

GROUP BY Museo.Nome;

## Interrogazioni DB

1. Elenco delle opere di un museo specifico:

SELECT Opera.Titolo, Artista.Nome AS NomeArtista, Artista.Cognome AS CognomeArtista

FROM Opera

JOIN Artista ON Opera.IDArtista = Artista.IDArtista

WHERE Opera.IDMuseo = <ID\_Museo>;

1. Elenco degli artisti nati dopo una certa data:

SELECT Nome, Cognome, DataNascita

FROM Artista

WHERE DataNascita > 'YYYY-MM-DD';

1. Totale finanziamenti ricevuti da un finanziatore specifico:

SELECT Finanziatore.Nome, Finanziatore.Cognome, SUM(Finanziamento.Importo) AS TotaleFinanziamenti

FROM Finanziatore

JOIN Finanziamento ON Finanziatore.IDFinanziatore = Finanziamento.IDFinanziatore

WHERE Finanziatore.IDFinanziatore = <ID\_Finanziatore>;

1. Numero di finanziamenti per ogni finanziatore:

SELECT Finanziatore.Nome, Finanziatore.Cognome, COUNT(Finanziamento.IDFinanziamento) AS NumeroFinanziamenti

FROM Finanziatore

LEFT JOIN Finanziamento ON Finanziatore.IDFinanziatore = Finanziamento.IDFinanziatore

GROUP BY Finanziatore.Nome, Finanziatore.Cognome;

1. Elenco degli artisti con il numero di opere create, ordinati per numero di opere:

SELECT Artista.Nome, Artista.Cognome, COUNT(Opera.IDOpera) AS NumeroOpere

FROM Artista

LEFT JOIN Opera ON Artista.IDArtista = Opera.IDArtista

GROUP BY Artista.Nome, Artista.Cognome

ORDER BY NumeroOpere DESC;

1. Media dell'importo dei finanziamenti ricevuti da ciascun museo, con almeno un finanziamento:

SELECT Museo.Nome, AVG(Finanziamento.Importo) AS MediaFinanziamenti

FROM Museo

JOIN Finanziamento ON Museo.IDMuseo = Finanziamento.IDMuseo

GROUP BY Museo.Nome

HAVING COUNT(Finanziamento.IDFinanziamento) >= 1;

1. Elenco delle opere con il nome dell'artista, il nome del museo e l'importo totale dei finanziamenti ricevuti dal museo per ciascuna opera:

SELECT Opera.Titolo, Artista.Nome AS NomeArtista, Museo.Nome AS NomeMuseo,

COALESCE(SUM(Finanziamento.Importo), 0) AS TotaleFinanziamenti

FROM Opera

JOIN Artista ON Opera.IDArtista = Artista.IDArtista

JOIN Museo ON Opera.IDMuseo = Museo.IDMuseo

LEFT JOIN Finanziamento ON Museo.IDMuseo = Finanziamento.IDMuseo

GROUP BY Opera.Titolo, Artista.Nome, Museo.Nome;

1. Elenco delle opere con l'anno di creazione, il nome dell'artista e il nome del finanziatore (se esistente):

SELECT Opera.Titolo, Opera.Anno, Artista.Nome AS NomeArtista,

COALESCE(Finanziatore.Nome, 'Nessun Finanziatore') AS NomeFinanziatore

FROM Opera

JOIN Artista ON Opera.IDArtista = Artista.IDArtista

LEFT JOIN Finanziamento ON Opera.IDMuseo = Finanziamento.IDMuseo

LEFT JOIN Finanziatore ON Finanziamento.IDFinanziatore = Finanziatore.IDFinanziatore

ORDER BY Opera.Anno;

1. Elenco degli artisti con l'anno di nascita, il numero di opere create e il totale degli importi finanziati da ciascun finanziatore:

SELECT Artista.Nome, Artista.DataNascita, COUNT(Opera.IDOpera) AS NumeroOpere,

COALESCE(SUM(Finanziamento.Importo), 0) AS TotaleFinanziamenti

FROM Artista

LEFT JOIN Opera ON Artista.IDArtista = Opera.IDArtista

LEFT JOIN Finanziamento ON Opera.IDMuseo = Finanziamento.IDMuseo

GROUP BY Artista.Nome, Artista.DataNascita

ORDER BY TotaleFinanziamenti DESC;

1. Elenco dei finanziamenti con l'importo, la data, il nome del museo, il nome dell'artista e il nome del finanziatore, ordinati per importo decrescente:

SELECT Finanziamento.Importo, Finanziamento.Data, Museo.Nome AS NomeMuseo,

Artista.Nome AS NomeArtista, COALESCE(Finanziatore.Nome, 'Nessun Finanziatore') AS NomeFinanziatore

FROM Finanziamento

LEFT JOIN Museo ON Finanziamento.IDMuseo = Museo.IDMuseo

LEFT JOIN Opera ON Museo.IDMuseo = Opera.IDMuseo

LEFT JOIN Artista ON Opera.IDArtista = Artista.IDArtista

LEFT JOIN Finanziatore ON Finanziamento.IDFinanziatore = Finanziatore.IDFinanziatore

ORDER BY Finanziamento.Importo DESC;