

# Basi di Dati

ANNO ACCADEMICO 2018/2019

Daniela Bonvini | mat. 284887

## Sommario

1.	Analisi dei requisiti .....	2
1.1	Descrizione dei requisiti .....	2
1.2	Glossario dei termini .....	3
1.3	tabella delle operazioni .....	3
	<i>Aggiornamenti:</i> .....	3
	<i>Interrogazioni:</i> .....	4
2.	Progettazione concettuale .....	5
2.1	Strategia di progetto .....	5
2.2	Sviluppo schema e/r .....	5
2.3	Integrazione viste parziali e Schema e/r finale .....	6
3.	Progettazione logica.....	8
3.1	Semplificazione .....	8
3.2	Traduzione .....	8
3.3	Normalizzazione .....	9
4.	Database MySQL.....	11
4.1	Tabelle.....	11
4.2	Operazioni .....	15
	<i>Aggiornamenti:</i> .....	15
	<i>Interrogazioni:</i> .....	16

# 1. Analisi dei requisiti

## 1.1 DESCRIZIONE DEI REQUISITI

*Creazione di un database per una piattaforma di streaming di contenuti multimediali quali film ed eventi sportivi.*

Un cliente (che può avere un solo account) è identificato con i consueti dati anagrafici oltre a user e pwd che gli consentono di accedere alla piattaforma.

Sia film che documentari hanno uno o più registi, anno di uscita, durata, descrizione, genere, lingua(e) disponibili (parlato e sottotitoli) e una durata di permanenza nella piattaforma (il listino viene periodicamente aggiornato con nuove uscite e la rimozione di vecchi contenuti).

I film hanno riferimenti agli attori, che possono interpretare più film e che possono essere protagonisti o meno.

Gli eventi sportivi hanno un nome, data, descrizione e sport.

I clienti sottoscrivono un piano che ha data di attivazione e di termine. I piani (3 in tutto) sono mensili e si differenziano tra loro in base al numero di dispositivi che è possibile utilizzare in contemporanea e al livello di definizione dello streaming.

Per tutti i piani è acquistabile anche l'opzione SPORT (sempre mensile) che consente di fruire anche degli eventi sportivi (live e in differita), altrimenti preclusi ai clienti. Alcuni eventi sportivi di un certo rilievo possono essere acquistati solo in modalità stand-alone, con un costo extra (es. € 5).

Ulteriori specifiche:

- Permettere agli utenti di lasciare un voto da 1 a 5 per ogni contenuto multimediale
- Mettere ai film un rating per limitare i contenuti in base all'età
- Offrire per ogni abbonamento un periodo di prova di un mese usufruibile una sola volta per cliente

## 1. FRASI DI CARATTERE GENERALE

Si vuole costruire una piattaforma di streaming di contenuti multimediali quali film ed eventi sportivi.

## 2. FRASI RELATIVE AL CLIENTE

Un cliente (che può avere un solo account) è identificato con i consueti dati anagrafici oltre a user e pwd che gli consentono di accedere alla piattaforma.

## 3. FRASI RELATIVE A FILM E DOCUMENTARI

Sia film che documentari hanno uno o più registi, anno di uscita, durata, descrizione, genere, lingua(e) disponibili (parlato e sottotitoli) e una durata di permanenza nella piattaforma (il listino viene periodicamente aggiornato con nuove uscite e la rimozione di vecchi contenuti).

I film hanno inoltre riferimenti agli attori, un rating del film relativo all'età consigliata per la visione e un feedback (positivo o negativo).

## 4. FRASI RELATIVE AGLI ATTORI

Gli attori possono interpretare più film e possono essere protagonisti o meno.

## 5. FRASI RELATIVE AI PIANI DI SOTTOSCRIZIONE

I clienti sottoscrivono un piano che ha data di attivazione e di termine. I piani (3 in tutto) sono mensili e si differenziano tra loro in base al numero di dispositivi che è possibile utilizzare in contemporanea e al livello di definizione dello streaming.

Per tutti i piani è acquistabile anche l'opzione SPORT (sempre mensile) che consente di fruire anche degli eventi sportivi (live e in differita), altrimenti preclusi ai clienti. Alcuni eventi sportivi di un certo rilievo possono essere acquistati solo in modalità stand-alone, con un costo extra (es. € 5).

#### 6. FRASI RELATIVE AGLI EVENTI SPORTIVI

Gli eventi sportivi hanno un nome, data, descrizione e sport.

#### 7. FRASI RELATIVE AGLI EXTRA

Alcuni eventi sportivi di un certo rilievo possono essere acquistati solo in modalità stand-alone, con un costo extra (es. € 5).

### 1.2 GLOSSARIO DEI TERMINI

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
CLIENTE	Colui che sottoscrive un piano di sottoscrizione	Utente	Piano di sottoscrizione
FILM	Contenuti disponibili per ogni piano di sottoscrizione		Attore, Piano di sottoscrizione
DOCUMENTARIO	Contenuti disponibili per ogni piano di sottoscrizione		Attore, Piano di sottoscrizione
ATTORE	Colui che interpreta i film e i documentari	Interprete	Film, Documentario
EVENTI SPORTIVI	Contenuti opzionali associabili ad ogni piano di sottoscrizione	Sport	Piano di sottoscrizione
EXTRA	Eventi sportivi importanti acquistabili in modalità stand-alone	Eventi sportivi speciali	Cliente
PIANO DI SOTTOSCRIZIONE	Piano sottoscrivibile dal cliente	Abbonamento	Cliente, Film, Documentario, Sport

### 1.3 TABELLA DELLE OPERAZIONI

Aggiornamenti:

1. Creazione nuovo utente
2. Sottoscrizione nuovo periodo di prova
3. Cancellazione abbonamento esistente
4. Rimozione contenuti multimediali in scadenza oggi
5. Acquisto opzione sport

6. Aggiunta voto su contenuto multimediale

Interrogazioni:

7. Visualizzazione di tutte le partite live oggi
8. Ricerca contenuti multimediali con un determinato attore come protagonista
9. Mostrare quanti film ci sono per ogni genere
10. Elenco della qualità degli abbonamenti dei clienti che hanno acquistato un determinato contenuto extra
11. Contare quanti film ci sono per la più bassa età minima consigliata
12. Mostrare gli attori che hanno sia diretto che recitato in un film

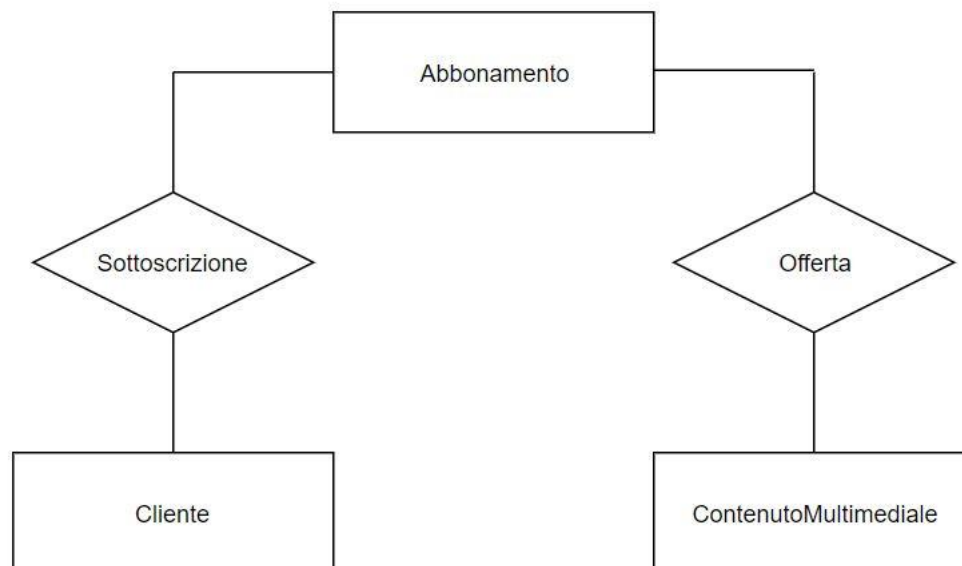
## 2. Progettazione concettuale

### 2.1 STRATEGIA DI PROGETTO

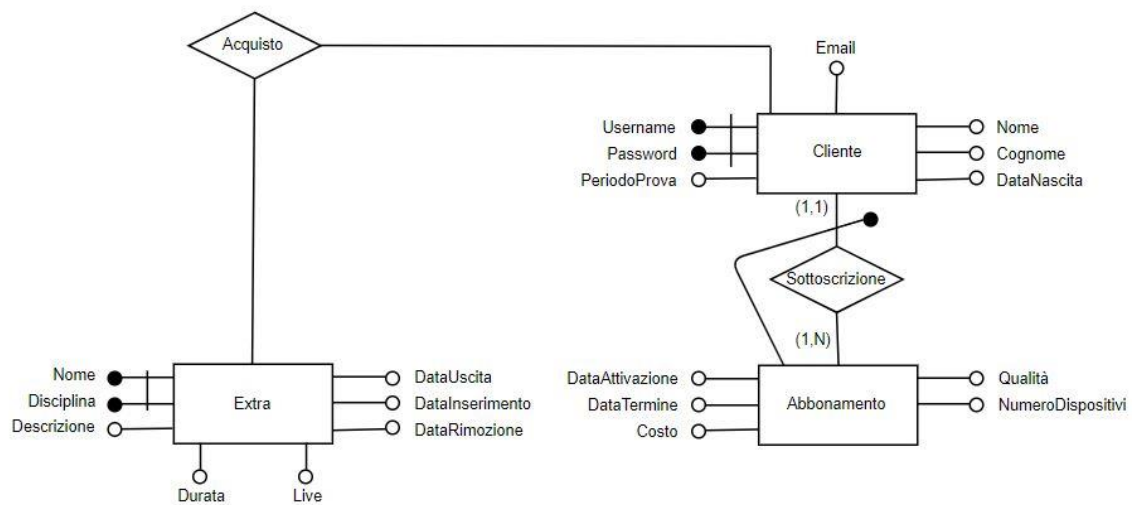
Si è deciso di adottare una strategia mista con integrazione, formata in questo modo: si parte da un passo base in cui si definisce uno schema scheletro contenente i concetti più importanti. Si scompongono poi i requisiti con riferimento ai concetti nello schema scheletro. In seguito, si effettua un ciclo di raffinamento dei concetti presenti sulla base delle loro specifiche e di aggiunta di concetti per descrivere specifiche non ancora esplicitate; questo passo si ripete per ogni sottoschema finché si ritiene necessario. Una volta soddisfatti del risultato del passo precedente si effettua l'integrazione, in cui si uniscono i vari sottoschemi in uno schema complessivo, facendo riferimento allo schema scheletro. Infine, c'è l'analisi di qualità, in cui si verificano le qualità dello schema e se necessario vi si apportano delle modifiche.

### 2.2 SVILUPPO SCHEMA E/R

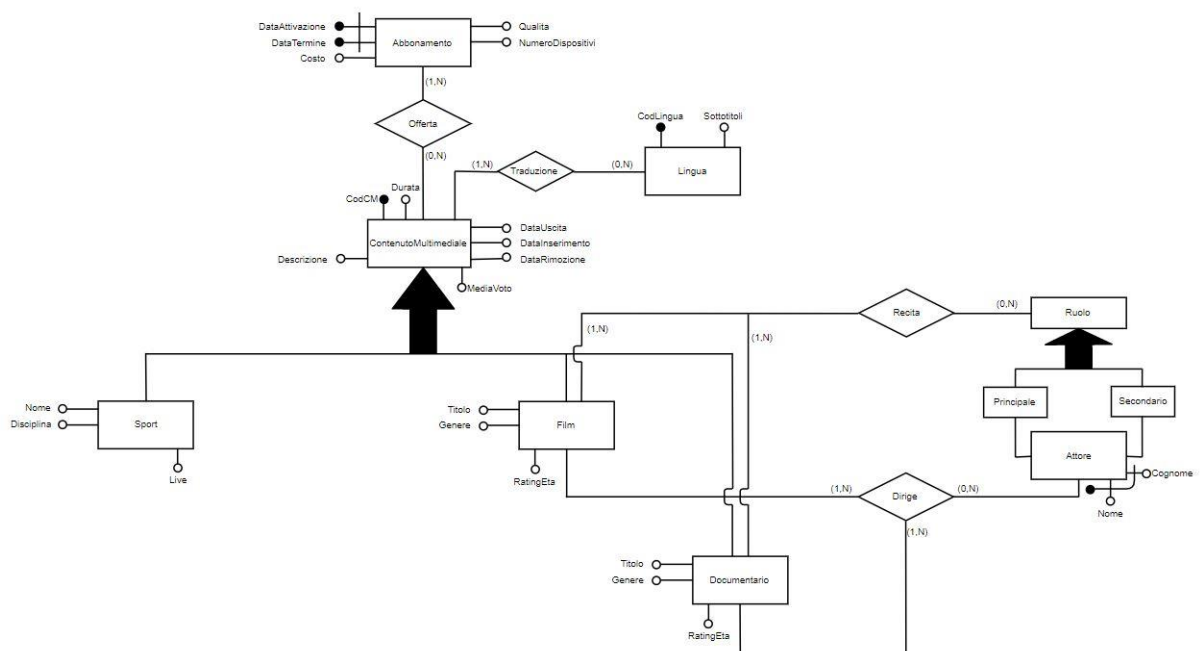
Schema scheletro base:



Schema scheletro 1:

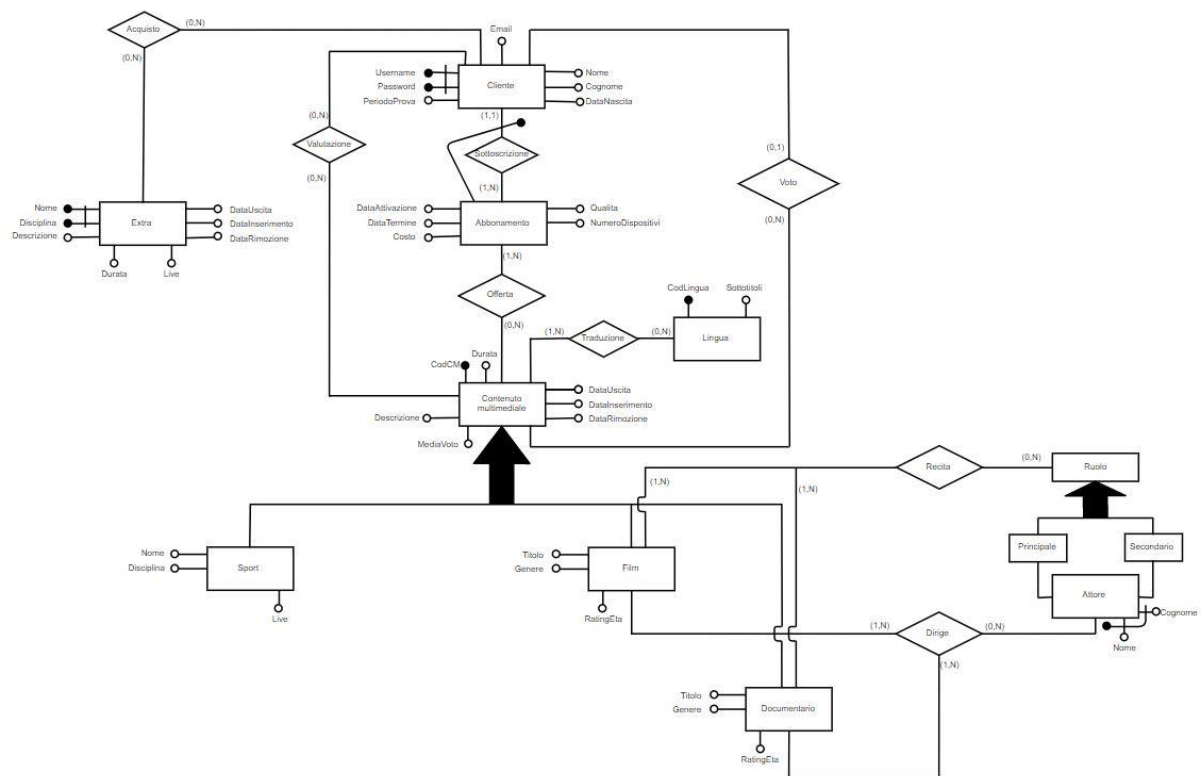


Schema scheletro 2:



## 2.3 INTEGRAZIONE VISTE PARZIALI E SCHEMA E/R FINALE

Conflitti strutturali incompatibili per l'identificatore dell'entità abbonamento. Si è scelto di mantenere l'identificatore dello schema scheletro 1:





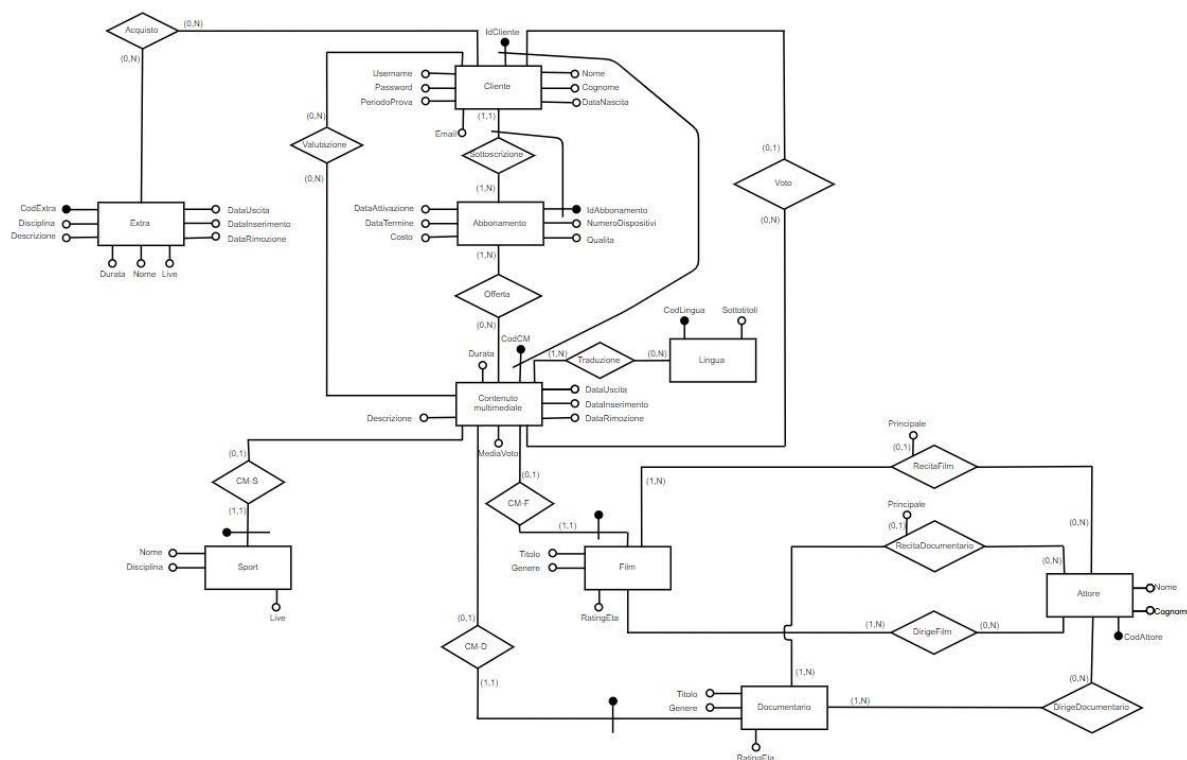
### 3. Progettazione logica

#### 3.1 SEMPLIFICAZIONE

In questa fase si è scelto di togliere le generalizzazioni per film, documentario e sport sostituendo la generalizzazione con delle relazioni; la generalizzazione si trasforma in tante relazioni uno a uno che legano l'entità padre con tutte le figlie.

La generalizzazione di "ruolo" è stata accorpata nella relazione "recita", decidendo di rendere il ruolo non più un'entità ma una proprietà della relazione e di mantenere solo "principale" come ruolo, poiché l'unico realmente importante. Si è deciso inoltre di dividere le relazioni "recita" e "dirige" in base all'entità a cui si riferiscono per semplificare le operazioni di ricerca che verranno analizzate più avanti.

Si è inoltre deciso di togliere gli identificatori multipli in Attore, Film, Documentario e Sport e al loro posto inserire un identificatore singolo denominato "cod+nome entità".



#### 3.2 TRADUZIONE

Legenda: chiavi primarie e *importate*.

**Cliente**(Idcliente, Username, Password, Nome, Cognome, Email, DataNascita, PeriodoProva)

**Abbonamento**(IdCliente, IdAbbonamento, DataAttivazione, DataTermine, Costo, Qualità, NumeroDispositivi, OpzioneSport)

**ClienteLasciaVoto**(IdCliente, CodCM, Voto)

**Extra**(CodExtra, Nome, Disciplina, Descrizione, Durata, Live, DataUscita, DataInserimento, DataRimozione)

**ContenutoMultimediale**(CodCM, MediaVoto, Descrizione, Durata, DataUscita, DataInserimento, DataRimozione)

**Lingua**(CodLingua, Sottotitoli)

**Sport**(CodCM, Nome, Disciplina, Live)

**Film**(CodCM, Titolo, Genere, RatingEta)

**Documentario**(CodCM, Titolo, Genere, RatingEta)

**Attore**(CodAttore, Nome, Cognome)

**AttoreRecitaFilm**(CodAttore, CodCMFilm, Principale)

**AttoreRecitaDocumentario**(CodAttore, CodCMDocumentario, Principale)

**AttoreDirigeFilm**(CodAttore, CodCMFilm)

**AttoreDirigeDocumentario**(CodAttore, CodCMDocumentario)

**ClienteAcquistaSport**(IdCliente, CodCMSport)

### 3.3 NORMALIZZAZIONE

#### 1. Prima forma normale (1NF)

Per rispettare la 1NF uno schema deve:

- avere ogni riga diversa dalle altre
- tutti i campi devono contenere valori atomici, riferibili allo stesso dominio

Per rispettare il primo punto è stato applicato ad ogni relazione un identificatore che distingue in maniera univoca tutte le righe. Per quanto riguarda il secondo punto, tutti i campi contengono informazioni elementari.

#### 2. Seconda forma normale (2NF)

Per rispettare la 2NF uno schema deve:

- essere in 1NF
- in presenza di una chiave composta da più campi, non devono esistere campi non-chiave che abbiano una dipendenza funzionale da un sottoinsieme dei campi chiave

La prima condizione è rispettata e riguardo alla seconda nello schema E-R sono presenti chiavi composte ma tutti i campi fanno parte della chiave:

**ClienteLasciaVoto**(IdCliente, CodCM, Voto)

**AttoreRecitaFilm**(CodAttore, CodCMFilm, Principale)

**AttoreRecitaDocumentario**(CodAttore, CodCMDocumentario, Principale)

Nello schema E-R sono presenti chiavi composte in cui le dipendenze funzionali dipendono dall'intera chiave:

**Abbonamento**(IdCliente, IdAbbonamento, DataAttivazione, DataTermine, Costo, Qualita, NumeroDispositivi, OpzioneSport)

Tutte le dipendenze funzionali dipendono dall'intera chiave primaria composta, in quanto sono relativi al tipo di abbonamento sottoscritto da quello specifico utente.

#### 3. Terza forma normale (3NF)

Per rispettare la 2NF uno schema deve:

- essere in 2NF
- non devono essere presenti campi non chiave che dipendono da altri campi non chiave

La 3NF è rispettata poiché non esistono nello schema campi non chiave dipendenti da altri campi non chiave.

#### 4. Forma normale di Boyce e Codd (BCNF)

Questa forma richiede che ogni tabella della base di dati:

- sia in prima forma normale
- ogni campo dal quale possono dipendere altri campi (determinanti) possa svolgere la funzione di chiave. Ovvero per ogni dipendenza funzionale  $A \rightarrow B$ , A deve essere determinante di B, cioè deve contenere una chiave

Nelle relazioni non esistono relazioni funzionali in cui A non sia una chiave primaria ed essendo già in 3NF, sono di conseguenza anche in BCNF.

**Lingua**(CodLingua, Sottotitoli)

In questo caso "Sottotitoli" potrebbe essere determinante ma dato che l'unico altro campo, "CodLingua", è chiave primaria, allora è anch'esso in BCNF.

**AttoreRecitaFilm**(CodAttore, CodCMFilm, Principale)

**AttoreRecitaDocumentario**(CodAttore, CodCMDocumentario, Principale)

In questi due casi "Principale" potrebbe essere determinante ma dato che gli altri due campi sono chiave primaria combinata, allora è anch'esso in BCNF.

Le altre relazioni, non avendo problemi di determinanti ed essendo già in 3NF, sono anche in BCNF.

## 4. Database MySQL

### 4.1 TABELLE

#### Database

```
CREATE DATABASE Netflix DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

**Cliente** (IdCliente, Username, Password, Email, Nome, Cognome, DataNascita, PeriodoProva)

```
CREATE TABLE Cliente(  
    IdCliente INT NOT NULL AUTO_INCREMENT,  
    Username VARCHAR(255) NOT NULL,  
    Password VARCHAR(40) NOT NULL,  
    Email VARCHAR(255) NOT NULL,  
    Nome VARCHAR(255) NOT NULL,  
    Cognome VARCHAR(255) NOT NULL,  
    DataNascita DATETIME NOT NULL,  
    PeriodoProva BIT(1) NOT NULL DEFAULT 0,  
    PRIMARY KEY(IdCliente)  
);
```

Note: PeriodoProva è mappato ad un bit per ridurre le dimensioni della tabella:  
0 = Non usufruito | 1 = Usufruito

**Abbonamento**(IdCliente, IdAbbonamento, DataAttivazione, DataTermine, Costo, Qualita, NumeroDispositivi, OpzioneSport)

```
CREATE TABLE Abbonamento(  
    IdCliente INT NOT NULL AUTO_INCREMENT,  
    IdAbbonamento INT NOT NULL,  
    DataAttivazione DATE NOT NULL,  
    DataTermine DATE NOT NULL,  
    Costo DECIMAL(5,2) NOT NULL,  
    Qualita INT NOT NULL,  
    NumeroDispositivi INT NOT NULL,  
    OpzioneSport BIT(1) NOT NULL DEFAULT 0,  
    PRIMARY KEY(IdCliente, IdAbbonamento),  
    FOREIGN KEY(IdCliente) REFERENCES Cliente(IdCliente)  
    ON UPDATE CASCADE ON DELETE CASCADE  
);
```

Note: Qualita è mappato ad un intero per ridurre le dimensioni della tabella:  
0 = Economy | 1 = Business | 2 = Premium  
OpzioneSport è mappato ad un bit per ridurre le dimensioni della tabella:  
0 = Non acquistato | 1 = Acquistato

**ClienteLasciaVoto**(IdCliente, CodCM, Voto)

```
CREATE TABLE ClienteLasciaVoto(
    IdCliente INT NOT NULL AUTO_INCREMENT,
    CodCM INT NOT NULL AUTO_INCREMENT,
    Voto INT NOT NULL DEFAULT 0,
    PRIMARY KEY(IdCliente, CodCM),
    FOREIGN KEY(IdCliente) REFERENCES Cliente(IdCliente)
);
```

Note: Voto è mappato ad un intero per ridurre le dimensioni della tabella:

0 = Non lasciato | 1 = Negativo | 2 = Abbastanza negativo | 3 = Neutro | 4 = Abbastanza positivo | 5 = Positivo

**Extra**(CodExtra, Nome, Disciplina, Descrizione, Durata, Live, DataUscita, DataInserimento, DataRimozione)

```
CREATE TABLE Extra(
    CodExtra VARCHAR(255) NOT NULL,
    Nome VARCHAR(255) NOT NULL,
    Disciplina VARCHAR(255) NOT NULL,
    Descrizione VARCHAR(2000) NOT NULL,
    Durata INT NOT NULL,
    Live DATETIME NOT NULL,
    DataUscita DATE NOT NULL,
    DataInserimento DATE NOT NULL,
    PRIMARY KEY(CodExtra)
);
```

**ContenutoMultimediale**(CodCM, MediaVoto, Descrizione, Durata, DataUscita, DataInserimento, DataRimozione)

```
CREATE TABLE ContenutoMultimediale(
    CodCM INT NOT NULL AUTO_INCREMENT,
    MediaVoto INT NOT NULL DEFAULT 0,
    Descrizione VARCHAR(2000) NOT NULL,
    Durata INT NOT NULL,
    DataUscita DATE NOT NULL,
    DataInserimento DATE NOT NULL,
    DataRimozione DATE NOT NULL,
    PRIMARY KEY(CodCM)
);
```

**Lingua**(CodLingua, Sottotitoli)

```
CREATE TABLE Lingua (
    CodLingua INT NOT NULL AUTO_INCREMENT,
```

```
Sottotitoli BIT(1) NOT NULL DEFAULT 0,  
PRIMARY KEY(CodLingua)  
);
```

Note: Sottotitoli è mappato ad un bit per ridurre le dimensioni della tabella:  
0 = Non presenti | 1 = Presenti

**Sport**(CodCM, Nome, Disciplina, Live)

```
CREATE TABLE Sport(  
    CodCM INT NOT NULL,  
    Nome VARCHAR(255) NOT NULL,  
    Disciplina VARCHAR(255) NOT NULL,  
    Live DATE NOT NULL,  
    PRIMARY KEY(CodCM),  
    FOREIGN KEY(CodCM) REFERENCES ContenutoMultimediale(CodCM)  
    ON DELETE CASCADE  
);
```

**Film**(CodCM, Titolo, Genere, RatingEta)

```
CREATE TABLE Film(  
    CodCM INT NOT NULL,  
    Titolo VARCHAR(255) NOT NULL,  
    Genere VARCHAR(255) NOT NULL,  
    RatingEta INT NOT NULL,  
    PRIMARY KEY(CodCM),  
    FOREIGN KEY(CodCM) REFERENCES ContenutoMultimediale(CodCM)  
    ON DELETE CASCADE  
);
```

**Documentario**(CodCM, Titolo, Genere, RatingEta)

```
CREATE TABLE Documentario(  
    CodCM INT NOT NULL,  
    Titolo VARCHAR(255) NOT NULL,  
    Genere VARCHAR(255) NOT NULL,  
    RatingEta INT NOT NULL,  
    PRIMARY KEY(CodCM),  
    FOREIGN KEY(CodCM) REFERENCES ContenutoMultimediale(CodCM)  
    ON DELETE CASCADE  
);
```

**Attore**(CodAttore, Nome, Cognome)

```
CREATE TABLE Attore(
    CodAttore INT NOT NULL AUTO_INCREMENT,
    Nome VARCHAR(255) NOT NULL,
    Cognome VARCHAR(255) NOT NULL,
    PRIMARY KEY(CodAttore)
);
```

**AttoreRecitaFilm**(CodAttore, CodCMFilm, Principale)

```
CREATE TABLE AttoreRecitaFilm(
    CodAttore INT NOT NULL,
    CodCM INT NOT NULL,
    Principale BIT(1) NOT NULL DEFAULT 0,
    PRIMARY KEY(CodAttore, CodCM),
    FOREIGN KEY(CodAttore) REFERENCES Attore(CodAttore)
    ON DELETE CASCADE,
    FOREIGN KEY(CodCM) REFERENCES Film(CodCM)
    ON DELETE CASCADE
);
```

Note: Principale è mappato ad un bit per ridurre le dimensioni della tabella:  
0 = Non protagonista 1 = Protagonista

**AttoreRecitaDocumentario**(CodAttore, CodCMDocumentario, Principale)

```
CREATE TABLE AttoreRecitaDocumentario(
    CodAttore INT NOT NULL,
    CodCM INT NOT NULL,
    Principale BIT(1) NOT NULL DEFAULT 0,
    PRIMARY KEY(CodAttore, CodCM),
    FOREIGN KEY(CodAttore) REFERENCES Attore(CodAttore)
    ON DELETE CASCADE,
    FOREIGN KEY(CodCM) REFERENCES Documentario(CodCM)
    ON DELETE CASCADE
);
```

Note: Principale è mappato ad un bit per ridurre le dimensioni della tabella:  
0 = Non protagonista 1 = Protagonista

**AttoreDirigeFilm**(CodAttore, CodCMFilm)

```
CREATE TABLE AttoreDirigeFilm(
    CodAttore INT NOT NULL,
    CodCM INT NOT NULL,
    PRIMARY KEY(CodAttore, CodCM),
    FOREIGN KEY(CodAttore) REFERENCES Attore(CodAttore)
    ON DELETE CASCADE,
    FOREIGN KEY(CodCM) REFERENCES Film(CodCM)
```

```
ON DELETE CASCADE  
);
```

#### **AttoreDirigeDocumentario(CodAttore, CodCMDocumentario)**

```
CREATE TABLE AttoreDirigeDocumentario(  
    CodAttore INT NOT NULL,  
    CodCM INT NOT NULL,  
    PRIMARY KEY(CodAttore, CodCM),  
    FOREIGN KEY(CodAttore)  
    REFERENCES Attore(CodAttore)  
    ON DELETE CASCADE,  
    FOREIGN KEY(CodCM) REFERENCES Documentario(CodCM)  
    ON DELETE CASCADE  
);
```

#### **ClienteAcquistaExtra(IdCliente, CodExtra)**

```
CREATE TABLE ClienteAcquistaExtra(  
    IdCliente INT NOT NULL AUTO_INCREMENT,  
    CodExtra INT NOT NULL,  
    PRIMARY KEY(IdCliente, CodExtra),  
    FOREIGN KEY(IdCliente) REFERENCES Cliente(IdCliente)  
);
```

## 4.2 OPERAZIONI

### Aggiornamenti:

1. Creazione nuovo utente  

```
INSERT INTO Cliente(Username, Password, Email, Nome, Cognome,  
DataNascita)  
VALUES ('Pippo', 'Pluto', 'pipppo87@gmail.com', 'Mario', 'Rossi',  
'19871021');
```
2. Sottoscrizione nuovo periodo di prova  

```
UPDATE Cliente  
SET PeriodoProva = 1  
WHERE IdCliente = '3';
```
3. Cancellazione abbonamento esistente  

```
DELETE FROM Abbonamento  
WHERE IdCliente = '3'  
AND IdAbbonamento = '7';
```
4. Rimozione contenuti multimediali in scadenza oggi  

```
DELETE * FROM ContenutoMultimediale  
WHERE DataRimozione = CURDATE();
```



5. Acquisto opzione sport

```
UPDATE ClienteAcquistaSport  
INNER JOIN Abbonamento  
ON ClienteAcquistaSport.IdCliente = Abbonamento.IdCliente  
SET OpzioneSport = '1';
```

6. Aggiornamento media voto su contenuto multimediale

```
UPDATE ContenutoMultimediale  
SET MediaVoto = (SELECT AVG(Voto)  
FROM ClienteLasciaVoto  
WHERE CodCM = '384')  
WHERE CodCM = '384';
```

Interrogazioni:

7. Visualizzazione di tutte le partite live oggi

```
SELECT * FROM Sport  
WHERE Disciplina = 'Calcio'  
AND Live = CURDATE();
```

8. Ricerca contenuti multimediali con un determinato attore come protagonista

```
SELECT Titolo FROM AttoreRecitaDocumentario  
INNER JOIN Documentario  
ON AttoreRecitaDocumentario.CodAttore = Documentario.CodAttore  
INNER JOIN Attore  
ON AttoreRecitaDocumentario.CodAttore = CodAttore  
WHERE Nome = 'Bruno' AND Cognome = 'Verdi' AND Principale = '1';
```

9. Mostrare quanti film ci sono per ogni genere

```
SELECT Genere,  
COUNT (*) FROM Film  
GROUP BY Genere;
```

10. Elenco della qualità degli abbonamenti dei clienti che hanno acquistato un determinato contenuto extra

```
SELECT Qualita FROM ClienteAcquistaExtra  
INNER JOIN Abbonamento  
ON ClienteAcquistaExtra.IdCliente = Abbonamento.IdCliente  
WHERE CodExtra = 'xyz';
```

11. Contare quanti film ci sono per la più bassa età minima consigliata

```
SELECT COUNT(*) FROM Film  
WHERE RatingEta = (SELECT MIN(RatingEta) FROM Film);
```

12. Mostrare gli attori che hanno sia diretto che recitato in un film

```
SELECT Nome, Cognome FROM Attore  
INNER JOIN AttoreDirigeFilm  
ON Attore.CodAttore = AttoreDirigeFilm.CodAttore  
INNER JOIN AttoreRecitaFilm  
ON Attore.CodAttore = AttoreRecitaFilm.CodAttore;
```