

## **BASI DI DATI - Registro per studiare uno strumento**

# **1 PROGETTAZIONE CONCETTUALE**

## **1.1 Analisi dei requisiti**

Si vuole creare una base di dati per aiutare dei musicisti nel loro percorso di studio di un brano. I vari brani che verranno inseriti nel database possederanno un nome, un autore e avranno la possibilità di essere associati ad informazioni quali l'editore specifico dello spartito, l'anno di pubblicazione e un link per trovarne la copia online. Presenteranno anche, se possibile, dei commenti per facilitarne l'approccio stilistico al momento dello studio e una serie di link che rimandino a dei video su youtube per poter ascoltare altri interpreti del brano.

Gli utenti della base di dati potranno, una volta cominciato lo studio, segnarsi, sulle battute che stanno approcciando, la velocità di esecuzione e eventuali commenti sulla sezione.

L'utente dovrà essere in grado, se desidera di ricavare informazioni sull'autore per poter al meglio interpretare il brano.

Il compositore del pezzo infatti, oltre al nome, potrà avere informazioni quali la data di nascita, la eventuale data di morte e la corrente artistica cui fa parte. Se necessario sarà anche possibile avere una sezione commenti per poter anche qui descrivere informazioni stilistiche tipiche del dato autore.

Anche il movimento artistico, cui fa parte l'autore, presenterà alcune info quali il nome, le caratteristiche tecniche-stilistiche che la rendono una corrente a se stante, e se necessario il range di anni su cui si sviluppò maggiormente.

Al movimento aggiungiamo anche notizie storiche, sotto forma di eventi decisivi per lo sviluppo della corrente, che potranno essere registrati tramite nome e anno di appartenenza.

In ultimo richiediamo di aggiungere una lista di strumenti musicali che saranno di tipi diversi e di case produttrici differenti dei quali l'utente ne possederà almeno uno. Questo, al momento dell'iscrizione, dovrà fornire dati utili quali il nome, la data di nascita e l'anno di inizio del suo percorso musicale.

## 1.2 Glossario dei termini

<i>Termine</i>	<i>Descrizione</i>	<i>Sinonimi</i>	<i>Collegamenti</i>
Autore	Colui che ha prodotto un'opera artistica.	Compositore	Brano, Movimento
Brano	Parte più o meno estesa di una composizione musicale.	Spartito, Pezzo	Autore, Utente, Url, Battuta
Battuta	Unità di tempo rappresentata sulla partitura tramite uno spazio compreso tra due stanghette.	Sezione	Utente, Brano
Movimento	Insieme di regole stilistiche e tecniche che riguardano un periodo artistico	Corrente	Autore, Eventi

## 1.3 Suddivisione del testo in frasi omogenee

### 1.3.1 Frasi di carattere generale

Si vuole creare una base di dati per aiutare dei musicisti nel loro percorso di studio di un brano.

### 1.3.2 Frasi relative al brano

I vari brani che verranno inseriti nel database possederanno un nome, un autore e avranno la possibilità di essere associati ad informazioni quali l'editore specifico dello spartito, l'anno di pubblicazione e un link per trovarne la copia online. Presenteranno anche, se possibile, dei commenti per facilitarne l'approccio stilistico al momento dello studio e una serie di link che rimandino a dei video su youtube per poter ascoltare altri interpreti del brano.

L'utente dovrà essere in grado, se desidera di ricavare informazioni sull'autore per poter al meglio interpretare il brano.

### 1.3.3 Frasi relative alle battute

Gli utenti della base di dati potranno, una volta cominciato lo studio, segnarsi, sulle battute che stanno approcciando, la velocità di esecuzione e eventuali commenti sulla sezione.

### 1.3.4 Frasi relative all'autore

L'utente dovrà essere in grado, se desidera di ricavare informazioni sull'autore per poter al meglio interpretare il brano.

Il compositore del pezzo infatti, oltre al nome, potrà avere informazioni quali la data di nascita, la eventuale data di morte e la corrente artistica cui fa parte. Se necessario sarà anche possibile avere una sezione commenti per poter anche qui descrivere informazioni stilistiche tipiche del dato autore.

Anche il movimento artistico, cui fa parte l'autore [...]

#### **1.3.5 Frasi relative al movimento**

Anche il movimento artistico, cui fa parte l'autore, presenterà alcune info quali il nome, le caratteristiche tecniche-stilistiche che la rendono una corrente a se stante, e se necessario il range di anni su cui si sviluppò maggiormente.

Al movimento aggiungiamo anche notizie storiche[...]

#### **1.3.6 Frasi relative agli eventi**

Al movimento aggiungiamo anche notizie storiche, sotto forma di eventi decisivi per lo sviluppo della corrente, che potranno essere registrati tramite nome e anno di appartenenza.

#### **1.3.7 Frasi relative agli strumenti**

In ultimo richiediamo di aggiungere una lista di strumenti musicali che saranno di tipi diversi e di case produttrici differenti dei quali l'utente ne possederà almeno uno.

#### **1.3.8 Frasi relative all'utente**

[...]dei quali l'utente ne possederà almeno uno. Questo, al momento dell'iscrizione, dovrà fornire dati utili quali il nome, la data di nascita e l'anno di inizio del suo percorso musicale.

## 1.4 Dizionario delle entità

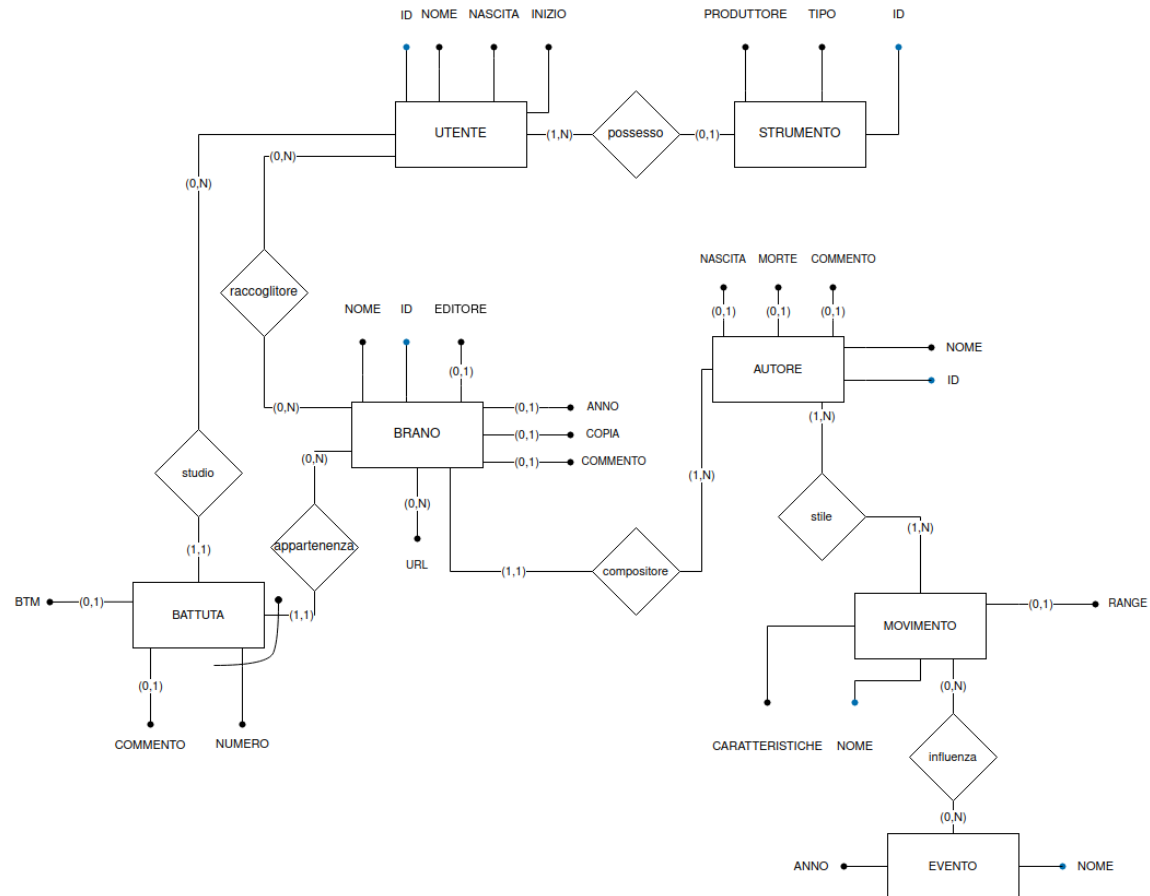
<i>Entità</i>	<i>Descrizione</i>	<i>Attributi</i>	<i>Identificatore</i>
Strumento	Oggetto in grado di emettere suoni. Viene utilizzato per fare musica.	Produttore, Tipo, ID	ID
Utente	Il musicista che usufruisce della base di dati.	ID, Nome, Nascita, Inizio	ID
Brano	Parte più o meno estesa di una composizione musicale.	Nome, Editore, Anno, Copia, Commento, ID, URL	ID
Battuta	Unità di tempo rappresentata sulla partitura tramite uno spazio compreso tra due stanghette	Numero, Btm, Commento	Numero, Appartenenza
Autore	Colui che ha prodotto un'opera artistica	Nascita, Morte, Commento, Nome, ID	ID
Movimento	Corrente artistica	Range, Caratteristiche, Nome	Nome
Evento	Evento che merita di essere tramandato per la sua importanza	Anno, Nome	Nome

## 1.5 Dizionario delle relazioni

<i>Relazioni</i>	<i>Descrizione</i>	<i>Componenti</i>	<i>Attributi</i>
Possesso	Lo strumento è proprietà di un utente	Strumento, Utente	
Raccoglitore	Insieme di brani che un utente può o ha già studiato	Utente, Brano	
Studio	L'utente elabora in modo dettagliato e personale un brano al fine di riuscire a suonarlo	Battuta, Utente	
Appartenenza	Una battuta fa parte di un insieme di battute, il cosiddetto brano	Brano, Battuta	
Compositore	Artista che produce un'opera musicale	Brano, Autore	
Stile	Caratteristiche tecnico artistiche riguardanti l'interpretazione storicamente accurata	Autore, Movimento	
Influenza	Un movimento artistico viene definito dagli eventi storici accaduti durante il suo tempo	Movimento, Evento	

## 1.6 Modello ER

Coerentemente con quanto appena detto, si giunge alla formulazione del seguente modello entità-relazione:



## 1.7 Vincoli non esprimibili

- L'anno di nascita dell'autore non può essere maggiore dell'anno attuale
- L'anno di nascita dell'utente non può essere maggiore dell'anno attuale
- L'utente non può avere più di 120 anni
- Gli eventi devono essersi svolti nel passato

## 2 PROGETTAZIONE LOGICA

### 2.1 Tavola dei volumi

<i>Concetto</i>	<i>Tipo</i>	<i>Volume</i>
Strumento	Entità	150
Possesso	Relazione	150
Utente	Entità	100
Raccoglitore	Relazione	500
Brano	Entità	500
Studio	Relazione	250000
Battuta	Entità	250000
Compositore	Relazione	500
Autore	Entità	30
Stile	Relazione	30
Movimento	Entità	10
Influenza	Relazione	200
Evento	Entità	200
Appartenenza	Relazione	250000

### 2.2 Valutazione del costo

<i>Operazione</i>	<i>Tipo</i>	<i>Frequenza</i>
Aggiunta utente	Interattiva	10/anno
Aggiunta brano	Interattiva	10/mese
Aggiunta battuta	Batch	200/giorno
Aggiornamento battuta	Interattiva	800/giorno
Prelevare informazioni tecnico-stilistiche sul brano	Interattiva	50/mese
Prelevare informazioni sul brano e sull'autore	Interattiva	10/mese
Prelevare informazioni sulle battute di un brano di un utente	Interattiva	10/giorno
Preleva i brani di ogni utente ordinati in base al maggior numero di battute studiate	Batch	200/giorno

## 2.3 Analisi delle rindondanze

Si osserva che è presente una rindondanza dovuta ad un ciclo che coinvolge le entità Utente, Brano, Battuta e le relazioni Studio, Appartenenza, Raccoglitore.

Si decide di andare a vedere se eliminare la relazione Raccoglitore possa influire in maniera positiva sul numero di accessi. Si vanno quindi a studiare le operazioni che coinvolgono tale relazione.

### 2.3.1 Prelevare informazioni sulle battute di un brano di un utente

Presenza di rindondanza			
<i>Concetto</i>	<i>Costrutto</i>	<i>Accessi</i>	<i>Tipo</i>
Battuta	E	500	L
Brano	E	1	L
Utente	E	1	L
Studio	R	500	L
Appartenenza	R	500	L
Raccoglitore	R	1	L
<i>Accessi totali:</i>			1503

Assenza di rindondanza			
<i>Concetto</i>	<i>Costrutto</i>	<i>Accessi</i>	<i>Tipo</i>
Battuta	E	500	L
Brano	E	1	L
Utente	E	1	L
Studio	R	500	L
Appartenenza	R	500	L
<i>Accessi totali:</i>			1502



**2.3.2 Preleva i brani di ogni utente ordinati in base al maggior numero di battute studiate**

Presenza di rindondanza			
<i>Concetto</i>	<i>Costrutto</i>	<i>Accessi</i>	<i>Tipo</i>
Battuta	E	2500	L
Brano	E	5	L
Utente	E	1	L
Studio	R	2500	L
Appartenenza	R	2500	L
Raccoglitore	R	5	L
<i>Accessi totali:</i>			7511

Assenza di rindondanza			
<i>Concetto</i>	<i>Costrutto</i>	<i>Accessi</i>	<i>Tipo</i>
Battuta	E	2500	L
Brano	E	5	L
Utente	E	1	L
Studio	R	2500	L
Appartenenza	R	2500	L
<i>Accessi totali:</i>			7506

**2.3.3 Aggiunta battuta**

Presenza di rindondanza			
<i>Concetto</i>	<i>Costrutto</i>	<i>Accessi</i>	<i>Tipo</i>
Battuta	E	1	S
Brano	E	1	L
Utente	E	1	L
Studio	R	1	S
Appartenenza	R	1	S
Raccoglitore	R	1	L
<i>Accessi totali:</i>			9

Assenza di rindondanza			
<i>Concetto</i>	<i>Costrutto</i>	<i>Accessi</i>	<i>Tipo</i>
Battuta	E	1	S
Brano	E	1	L
Utente	E	1	L
Studio	R	1	S
Appartenenza	R	1	S
<i>Accessi totali:</i>			8

#### 2.3.4 Analisi convenienza

Tenendo conto del numero di accessi al giorno si ottiene un miglioramento di 1210 accessi totali. Anche se non risulta essere una differenza elevata si decide di eliminare la rindondanza, ipotizzando un futuro accrescimento del database che aumenterebbe la dimensione del sopracitato miglioramento.

## 2.4 Eliminazione delle generalizzazioni

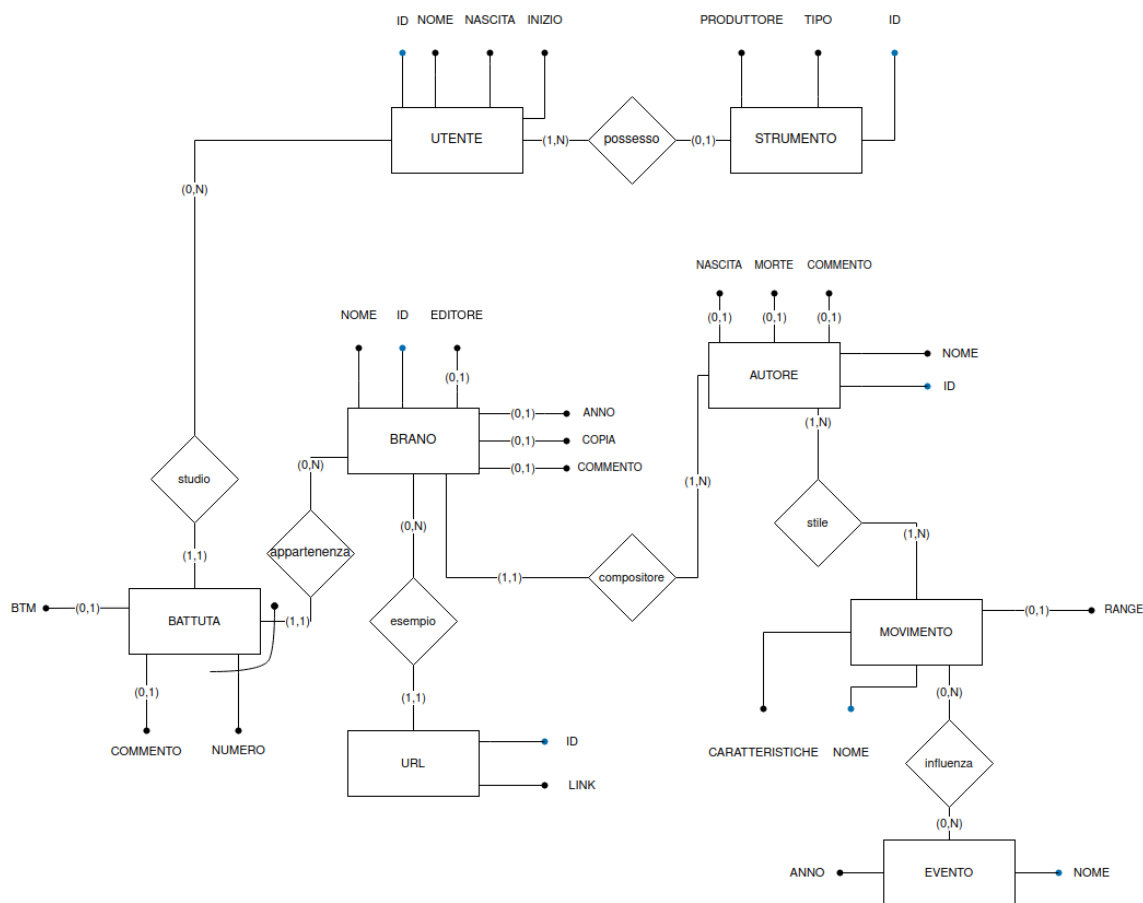
Non sono presenti generalizzazioni da poter eliminare.

## 2.5 Analisi degli attributi multivalore

Si può notare la presenza di un attributo multivalore nell'entità Brano, andiamo quindi a eliminarlo aggiungendo l'entità URL.

## 2.6 Presentazione modello E-R ristrutturato

A seguito delle considerazioni sopra elencate, si è rielaborato un nuovo modello entità-relazioni di seguito mostrato:



## 2.7 Scelta degli identificatori primari

Come identificatori primari sono stati scelti:

<i>Chiave primaria</i>	<i>Attributo</i>
Strumento	ID
Utente	ID
Battuta	Numero e Brano (identificatore esterno)
Brano	ID
URL	ID
Autore	ID
Movimento	Nome
Evento	Nome

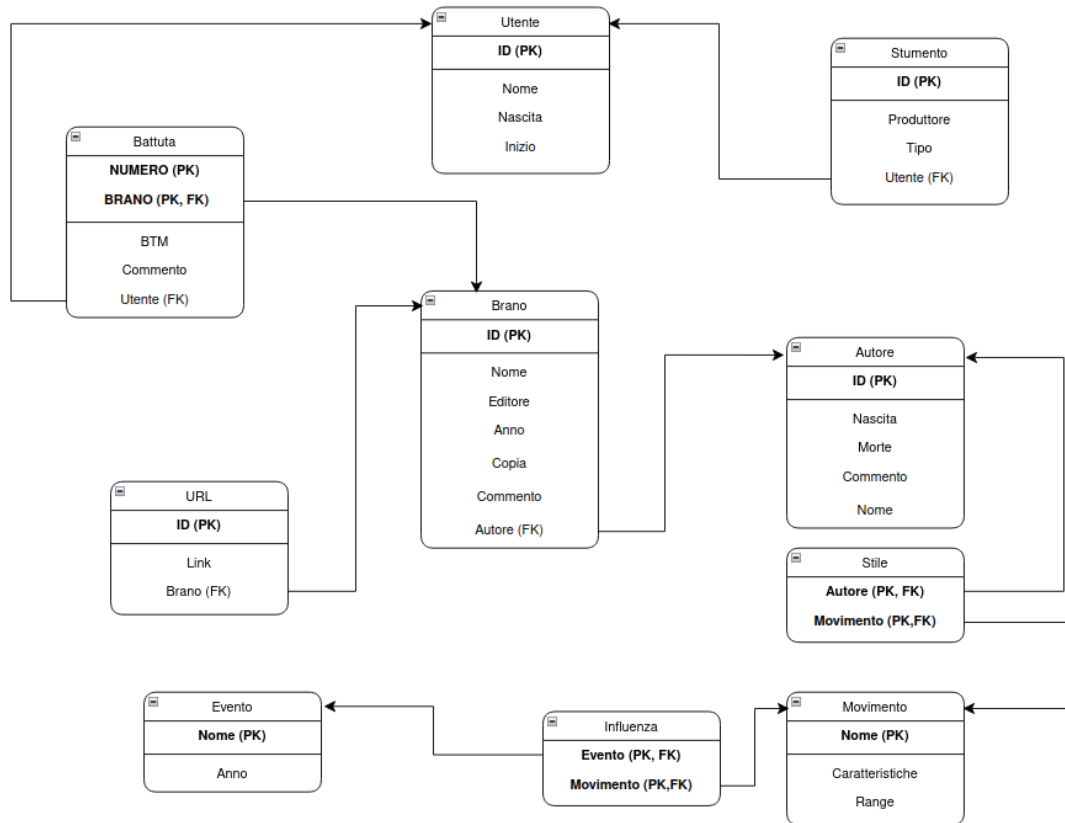
## 2.8 Passaggio al modello relazionale

Si individuano le seguenti relazioni, identificatori e attributi per lo schema logico con il modello relazionale:

- Strumento(ID, Produttore, Tipo, Utente);
- Utente(ID, Nome, Nascita, Inizio);
- Battuta(Numero, Brano, BTM, Commento, Utente);
- Brano(ID, Nome, Editore, Anno, Copia, Commento, Autore);
- URL(ID, Link, Brano);
- Autore(ID, Nascita, Morte, Commento, Nome);
- Stile(Autore, Movimento);
- Movimento(Nome, Caratteristiche, Range);
- Influenza(Movimento, Evento);
- Evento(Nome, Anno).

## 2.9 Modello logico

In accordo con tutte le considerazioni espresse, il modello logico formulato è il seguente:



## 2.10 Normalizzazione

Il database proposto:

- É in prima forma normale: tutte le colonne sono atomiche, non sono presenti unità ripetitive
- É in seconda forma normale: ogni tabella memorizza solamente dati relativi alla entità descritta dalla primary key. Non occorre attuare un procedimento di decomposizione.
- **Non è in terza forma normale:** Si può discutere sulla dipendenza della colonna **Copia** dalle colonne **Nome**, **Editore** e **Anno**.

Si sceglie di non completare la normalizzazione e di lasciare stare la terza forma normale. Tentare di eliminare la denormalizzazione risulterebbe un lavoro inutile, perchè per definizione del database non risulterà mai necessario modificare le colonne **Nome**, **Editore** e **Anno**, poichè lo studio di un brano risulterà collegato strettamente all'edizione utilizzata.

### 3 Realizzazione operazioni

#### 3.1 Create da eliminare

```
CREATE DATABASE musica;

USE musica;

CREATE TABLE Utente (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Nome VARCHAR(50) NOT NULL,
    Nascita DATE NOT NULL ,
    Inizio DATE NOT NULL
);

CREATE TABLE Autore (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Nome VARCHAR(50) NOT NULL,
    Nascita DATE,
    Morte DATE,
    Commento TEXT
);

CREATE TABLE Brano (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Nome VARCHAR(100) NOT NULL,
    Editore VARCHAR(100),
    Anno INT,
    Copia INT,
    Commento TEXT,
    Autore INT,
    FOREIGN KEY (Autore) REFERENCES Autore(ID)
);

CREATE TABLE Strumento (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Produttore VARCHAR(50) NOT NULL ,
    Tipo VARCHAR(50) NOT NULL ,
    Utente INT,
    FOREIGN KEY (Utente) REFERENCES Utente(ID)
);

CREATE TABLE Battuta (
    Numero INT,
    Brano INT,
    BTM INT,
```

```

        Commento TEXT,
        Utente INT,
        PRIMARY KEY (Numero, Brano),
        FOREIGN KEY (Brano) REFERENCES Brano(ID),
        FOREIGN KEY (Utente) REFERENCES Utente(ID)
    );

CREATE TABLE URL (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Link TEXT NOT NULL ,
    Brano INT,
    FOREIGN KEY (Brano) REFERENCES Brano(ID)
);

CREATE TABLE Movimento (
    Nome VARCHAR(50) PRIMARY KEY,
    Caratteristiche TEXT NOT NULL,
    'Range' VARCHAR(50)
);

CREATE TABLE Stile (
    Autore INT,
    Movimento VARCHAR(50),
    PRIMARY KEY (Autore, Movimento),
    FOREIGN KEY (Autore) REFERENCES Autore(ID),
    FOREIGN KEY (Movimento) REFERENCES Movimento(Nome)
);

CREATE TABLE Evento (
    Nome VARCHAR(100) PRIMARY KEY,
    Anno INT NOT NULL
);

CREATE TABLE Influenza (
    Movimento VARCHAR(50),
    Evento VARCHAR(100),
    PRIMARY KEY (Movimento, Evento),
    FOREIGN KEY (Movimento) REFERENCES Movimento(Nome),
    FOREIGN KEY (Evento) REFERENCES Evento(Nome)
);

```

### 3.2 Aggiunta utente

```
DELIMITER $$

CREATE PROCEDURE addUtente(
    IN newNome VARCHAR(50),
    IN newNascita DATE,
    IN newInizio DATE,
    IN StrumentoID INT
)
BEGIN
    -- Variabile per memorizzare l'ID del nuovo utente
    DECLARE UtenteID INT;

    -- Inserisce il nuovo utente nella tabella Utente
    INSERT INTO Utente (Nome, Nascita, Inizio)
    VALUES (newNome, newNascita, newInizio);

    -- Ottiene l'ID dell'utente appena inserito
    SET UtenteID = LAST_INSERT_ID();

    -- Aggiorna la tabella Strumento per associare lo strumento all'utente
    UPDATE Strumento
    SET Utente = UtenteID
    WHERE ID = StrumentoID;

END $$

DELIMITER ;
```

### 3.3 Aggiunta brano

```
DELIMITER $$

CREATE PROCEDURE addBrano(
    IN newNome VARCHAR(100),
    IN newEditore VARCHAR(100),
    IN newAnno INT,
    IN newCopia INT,
    IN newCommento TEXT,
    IN AutoreID INT
)
BEGIN
    -- Inserisce il nuovo brano nella tabella Brano
    INSERT INTO Brano (Nome, Editore, Anno, Copia, Commento, Autore)
    VALUES (newNome, newEditore, newAnno, newCopia, newCommento, AutoreID);
```



```
END $$
```

```
DELIMITER ;
```

### 3.4 Aggiunta battuta

```
DELIMITER $$
```

```
CREATE PROCEDURE addBattuta(  
    IN newNumero INT,  
    IN BranoID INT,  
    IN newBTM INT,  
    IN newCommento TEXT,  
    IN UtenteID INT  
)  
BEGIN  
    -- Inserisce una nuova battuta nella tabella Battuta  
    INSERT INTO Battuta (Numero, Brano, BTM, Commento, Utente)  
    VALUES (newNumero, BranoID, newBTM, newCommento, UtenteID);  
END $$  
  
DELIMITER ;
```

### 3.5 Aggiornamento battuta

```
DELIMITER $$
```

```
CREATE PROCEDURE updateBattuta(  
    IN selectedNumero INT,  
    IN BranoID INT,  
    IN newBTM INT,  
    IN newCommento TEXT  
)  
BEGIN  
    -- Aggiorna il campo BTM se il nuovo valore non è NULL  
    IF newBTM IS NOT NULL THEN  
        UPDATE Battuta  
        SET BTM = newBTM  
        WHERE Numero = selectedNumero AND Brano = BranoID;  
    END IF;  
  
    -- Aggiorna il campo Commento se il nuovo valore non è NULL  
    IF newCommento IS NOT NULL THEN  
        UPDATE Battuta
```

```

        SET Commento = newCommento
        WHERE Numero = selectedNumero AND Brano = BranoID;
    END IF;
END $$

DELIMITER ;

```

### 3.6 Prelevare informazioni tecnico- stilistiche sul brano

```

DELIMITER $$

CREATE PROCEDURE getInformazioniStilistiche(
    IN BranoID INT
)
BEGIN

    -- Dichiaro una variabile AutoreID e ci assegna l'ID dell'autore associato al brano
    DECLARE AutoreID INT;

    SELECT Autore.ID
    INTO AutoreID
    FROM Brano
    INNER JOIN Autore
    ON Brano.Autore = Autore.ID
    WHERE Brano.ID = BranoID;

    -- Ricavo gli URL associati al brano
    SELECT URL.Link
    FROM URL
    WHERE URL.Brano = BranoID;

    -- Ricavo il commento del brano
    SELECT Brano.Commento AS BranoCommento
    FROM Brano
    WHERE Brano.ID = BranoID;

    -- Ricavo il commento dell'autore del brano
    SELECT Autore.Commento AS AutoreCommento
    FROM Autore
    WHERE Autore.ID = AutoreID;

    -- Ricavo le informazioni relative ai movimenti associati all'autore
    SELECT Movimento.Nome, Movimento.Caratteristiche, Movimento.'Range'
    FROM Autore
    INNER JOIN Stile

```

```

ON Autore.ID = Stile.Autore
INNER JOIN Movimento
ON Movimento.Nome = Stile.Movimento
WHERE Autore.ID = AutoreID;

-- Ricava le informazioni relative agli eventi associati ai movimenti
SELECT Evento.Nome AS EventoNome, Evento.Anno AS EventoAnno
FROM Autore
INNER JOIN Stile
ON Autore.ID = Stile.Autore
INNER JOIN Movimento
ON Movimento.Nome = Stile.Movimento
INNER JOIN Influenza
ON Movimento.Nome = Influenza.Movimento
INNER JOIN Evento
ON Evento.Nome = Influenza.Evento
WHERE Autore.ID = AutoreID;
END $$

DELIMITER ;

```

### 3.7 Prelevare informazioni sul brano e sull'autore

```

DELIMITER $$

CREATE PROCEDURE getInformazioniEditoriali(
    IN BranoID INT
)
BEGIN
    -- Ricava il nome, l'editore, l'anno e la copia del brano
    SELECT
        Brano.Nome AS NomeBrano,
        Brano.Editore,
        Brano.Anno,
        Brano.Copia
    FROM Brano
    WHERE Brano.ID = BranoID;

    -- Ricava il nome dell'autore, la sua data di nascita e di morte associato al brano
    SELECT
        Autore.Nome AS NomeAutore,
        Autore.Nascita,
        Autore.Morte
    FROM Brano
    INNER JOIN Autore ON Brano.Autore = Autore.ID

```

```

        WHERE Brano.ID = BranoID;
    END $$

    DELIMITER ;

```

### 3.8 Prelevare informazioni sulle battute di un brano di un utente

```

    DELIMITER $$

    CREATE PROCEDURE getInformazioniBattute(
        IN BranoID INT,
        IN UtenteID INT
    )
    BEGIN
        -- Ricava tutte le informazioni delle battute associate
        SELECT
            Battuta.Numero,
            Battuta.BTM,
            Battuta.Commento
        FROM Battuta
        WHERE Battuta.Brano = BranoID AND Battuta.Utente = UtenteID;
    END $$

    DELIMITER ;

```

### 3.9 Preleva i brani di ogni utente ordinati in base al maggior numero di battute studiate

```

    -- Da ripetere per ogni utente se si vuole averne informazioni su più di uno

    DELIMITER $$

    CREATE PROCEDURE getBrani(
        IN UtenteID INT
    )
    BEGIN
        -- Selezione dei brani dell'utente ordinati
        SELECT
            Brano.ID,
            Brano.Nome,
            COUNT(Battuta.ID) AS NumBattute
        FROM Brano
        INNER JOIN Battuta ON Brano.ID = Battuta.Brano
        WHERE Battuta.Utente = UtenteID
    END

```

```

        GROUP BY Brano.ID
        ORDER BY NumBattute DESC;
END $$

```

```

DELIMITER ;

```

## 4 Realizzazione Trigger

### 4.1 L'anno di nascita dell'autore non può essere maggiore dell'anno attuale

```

DELIMITER $$

```

```

CREATE TRIGGER VerificaAnnoNascitaAutore
BEFORE INSERT ON Autore
FOR EACH ROW
BEGIN
    IF NEW.Nascita > CURDATE() THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'L\'anno di nascita non può essere maggiore dell\'anno attuale.';
    END IF;
END $$

```

```

DELIMITER ;

```

### 4.2 L'anno di nascita dell'utente non può essere maggiore dell'anno attuale

```

DELIMITER $$

```

```

CREATE TRIGGER VerificaAnnoNascitaUtente
BEFORE INSERT ON Utente
FOR EACH ROW
BEGIN
    IF NEW.Nascita > CURDATE() THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'L\'anno di nascita non può essere maggiore dell\'anno attuale.';
    END IF;
END $$

```

```

DELIMITER ;

```

### 4.3 L'utente non può avere più di 120 anni

```
DELIMITER $$

CREATE TRIGGER VerificaEtaUtente
BEFORE INSERT ON Utente
FOR EACH ROW
BEGIN
    DECLARE eta INT;
    SET eta = YEAR(CURDATE()) - YEAR(NEW.Nascita);

    IF eta > 120 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'L\'utente non può avere più di 120 anni.';
    END IF;
END $$

DELIMITER ;
```

### 4.4 Gli eventi devono essersi svolti nel passato

```
DELIMITER $$

CREATE TRIGGER VerificaDataEvento
BEFORE INSERT ON Evento
FOR EACH ROW
BEGIN
    IF NEW.Anno > YEAR(CURDATE()) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'L\'evento non può avere una data nel futuro.';
    END IF;
END $$

DELIMITER ;
```