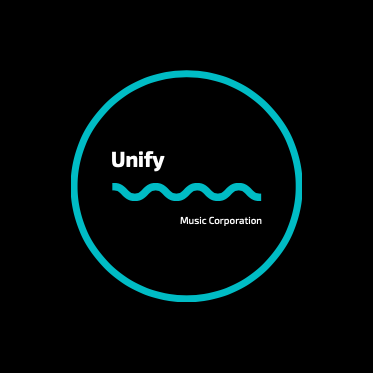
***Unify***

PROGETTO BASI DI DATI – GRUPPO 29



Alberto Stefani [134297]

Elia Tolin [136515]

Danil Lugli [128504]

Sommario

[Progetto Unify 2](#_Toc64579906)

[Glossario 3](#_Toc64579907)

[Schema Scheletro 4](#_Toc64579908)

[Progettazione Concettuale 5](#_Toc64579909)

[Auto Associazione 8](#_Toc64579910)

[Gerarchia 8](#_Toc64579911)

[Schema ER - Completo 9](#_Toc64579912)

[Progettazione Logica 10](#_Toc64579913)

[Eliminazione delle Gerarchie ISA 10](#_Toc64579914)

[Selezione delle chiavi primarie ed eliminazione delle identificazioni esterne 11](#_Toc64579915)

[Trasformazione degli attributi opposti 13](#_Toc64579916)

[Traduzione di entità e associazioni in schemi di relazioni 14](#_Toc64579917)

[Verifica di Normalizzazione 17](#_Toc64579918)

[Creazione Tabelle 17](#_Toc64579919)

[Inserimento Valori 20](#_Toc64579920)

[Query 25](#_Toc64579921)

[Query di Modifica 25](#_Toc64579922)

[Query di Eliminazione 25](#_Toc64579923)

[Query di Interrogazione 25](#_Toc64579924)

# Progetto Unify

Si vuole gestire il database per una piattaforma di musica in streaming, usando le seguenti informazioni:

Gli utenti sono identificati tramite un codice univoco che viene assegnato al momento dell’iscrizione, sono descritti da nome, cognome, indirizzo e-mail, data di nascita e paese di provenienza.

Gli artisti sono identificati nello stesso modo degli utenti, in più possiedono una breve descrizione e la possibilità di informare gli utenti di concerti che terranno.

Un concerto è identificato univocamente, e al contempo caratterizzato, da data, ora e luogo, un artista può tenere zero o più concerti

I brani sono identificati tramite *idBrano* univoco, nel caso di featuring abbiamo un’entità diversa per ogni artista che collabora, sono descritti da durata, data di caricamento, casa discografica e genere.

Gli utenti possono riprodurre i brani, per ogni riproduzione deve essere salvato un record identificato univocamente da utente che ha eseguito l’ascolto e momento dell’ascolto, un utente può ascoltare più volte lo stesso brano, ma ovviamente non può ascoltare due brani diversi nello stesso momento.

Un artista può caricare più brani, ma lo stesso brano può avere più featuring.

Gli album sono identificati da un codice univoco, un album è descritto dal numero di brani, dal titolo dell’album e la durata complessiva, un album può essere composto anche da più di un artista, ad esempio un album scritto in collaborazione.

Un album può contenere uno o più brani ma un brano può anche non far parte di un album.

Le playlist sono identificate dal codice dell’utente che le ha create e dal titolo della playlist, sono descritte da una piccola descrizione fatta dall’utente.

Un utente può creare più playlist o nessuna, una playlist è creata da un solo utente.

Una playlist può contenere più brani e uno stesso brano può essere in nessuna o più playlist.

Un utente può seguire ed essere seguito da altri utenti e artisti, a sua volta anche un artista può essere seguito e seguire altri utenti.

Le pubblicità vengono visualizzate o ascoltate dagli utenti e lo spazio pubblicitario può essere acquistato da aziende esterne, una pubblicità è identificata dal nome dell’azienda che la produce e da un codice univoco assegnato dalla piattaforma ed è descritta da durata e dalla tipologia, che può essere banner visivo, video o solo audio.

L’inserzionista è identificato da un id univoco e descritto dai dati di fatturazione, come partita iva, sede legale e numero di telefono.

Il record delle visualizzazioni delle pubblicità è salvato in modo analogo a quello dei brani.

# Glossario

|  |  |  |  |
| --- | --- | --- | --- |
| TERMINE | DESCRIZIONE | SINONIMI | LEGAME |
| Brano | Titolo, durata, idArtista, dataUscita, **idBrano** | Canzone, musica | Utente, Artista, Playlist, Album, Riproduzione |
| Utente | **idUtente,** nome, cognome, dataIscrizione, email, paese | Ascoltatore | Brano, Pubblicità, Playlist, Riproduzione |
| Artista | **idArtista**, nome, cognome, descrizione |  | Utente, Brano |
| Album | **Titolo, idArtista,** numeroBrani, durata, dataUscita |  | Artista, Brano |
| Playlist | **idPlaylist**, titolo, numeroBrani, durata, username |  | Utente, Brano |
| Riproduzione | **oraAscolto**, **idUtente,** idBrano**,** | Ascolto | Brano, Utente |
| Pubblicità | **idPubblicità,** nome, durata, tipo, inserzionista | Inserzione | Inserzionista, Utente |
| Inserzionista | **pIva,** nome, email, numeroTelefono | Azienda | Pubblicità |
| Concerto | **Data, Luogo, Ora** |  | Artista |
| Visualizzazione | **Ora\_visualizzazione,  idUtente,** idPubblicità |  | Utente, Pubblicità |

# Schema Scheletro

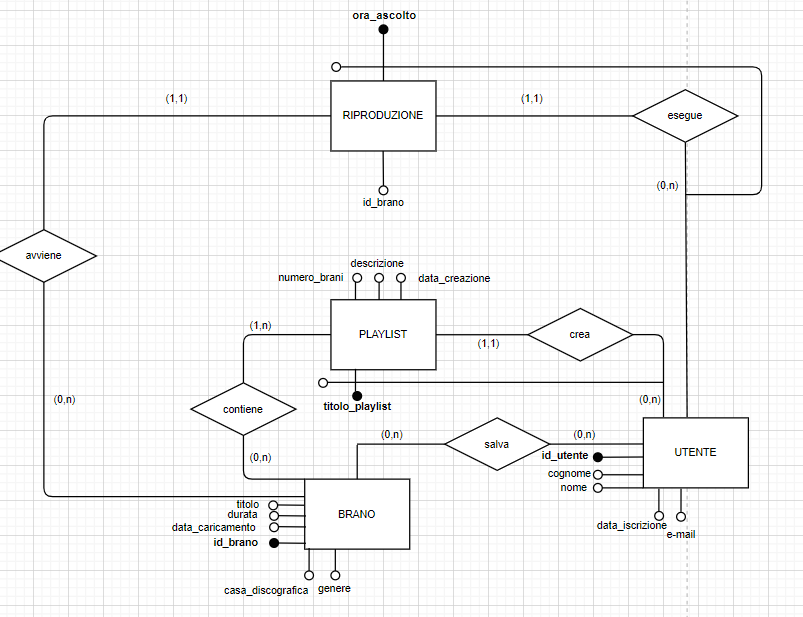
**Immagine che contiene testo, mappa

Descrizione generata automaticamente**

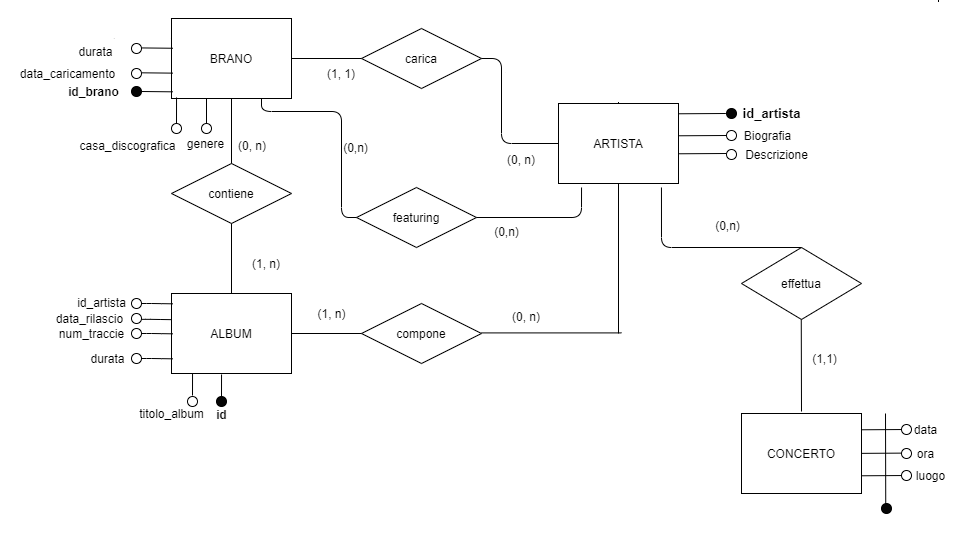
Con lo schema scheletro vogliamo fare un’analisi di massima dello schema senza inoltrarci sul tipo delle entità e delle associazioni, descrivendo in generale i collegamenti tra le varie entità, non sono ancora definiti concetti più avanzati come auto associazioni e gerarchie, quest’ultime verranno definite in seguito e implementate nello schema ER finale.

# Progettazione Concettuale

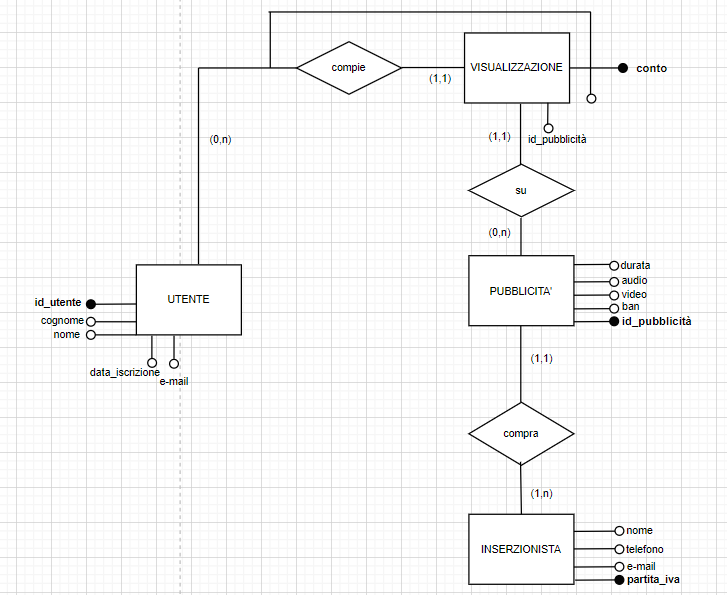
Iniziamo definendo lo schema ER per la parte riguardante l’utente, i brani, le playlist e le riproduzioni, è importante salvare il record per ogni riproduzione, un utente può infatti ascoltare una stessa canzone più volte ed ogni brano avrà più ascolti, è necessario quindi usare un identificatore esterno in quanto riproduzione è un’entità debole rispetto a utente, ogni riproduzione è quindi identificata dall’Id dell’utente e dall’ora di ascolto. Usiamo un’identificazione esterna anche per la playlist, siccome ogni playlist dipende dall’utente che l’ha creata, uno stesso utente non può creare più playlist con lo stesso nome ma due utenti distinti possono creare una playlist con nome uguale.



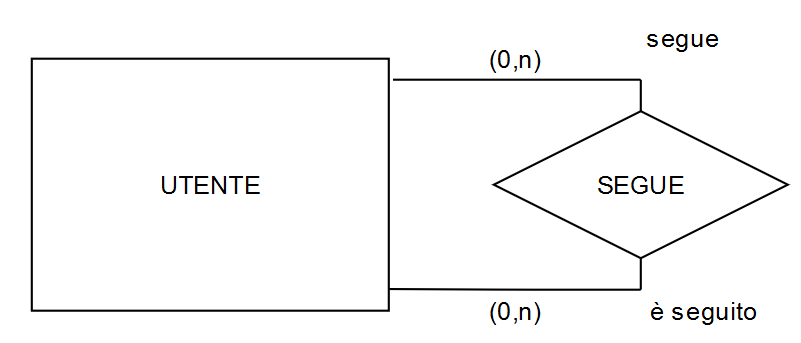
Continuiamo definendo la parte di progetto concettuale che comprende artista, brano, album e concerti, artista è una sottoclasse di utente, infatti può compiere tutte le azioni di un utente normale e in aggiunta ha la possibilità di caricare brani e album, informare su eventuali concerti e partecipare al brano di un altro artista, in modo che sia possibile visualizzare tutti i brani in cui è presente un certo artista, e non solo quelli caricati direttamente da lui. Brano è un’entità indipendente dall’artista ed è definito da un proprio ID.



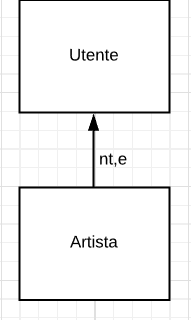
Infine è necessario specificare la parte di schema ER che riguarda la gestione delle pubblicità e la visualizzazione di esse da parte degli utenti, l’entità di pubblicità dipende dall’inserzionista, ogni pubblicità può essere di tre tipi, banner visivo, audio o video, per permettere ciò usiamo tre proprietà opzionali con cardinalità (0,1), un utente durante l’utilizzo della piattaforma può visualizzare zero o più pubblicità, dobbiamo salvare un record per ogni volta che un’utente guarda una certa pubblicità, creiamo quindi un entita ‘visualizzazione’ in modo analogo alla riproduzione dei brani, ogni visualizzazione è identificata da id utente e un data che memorizza giorno e ora, con precisione al secondo.



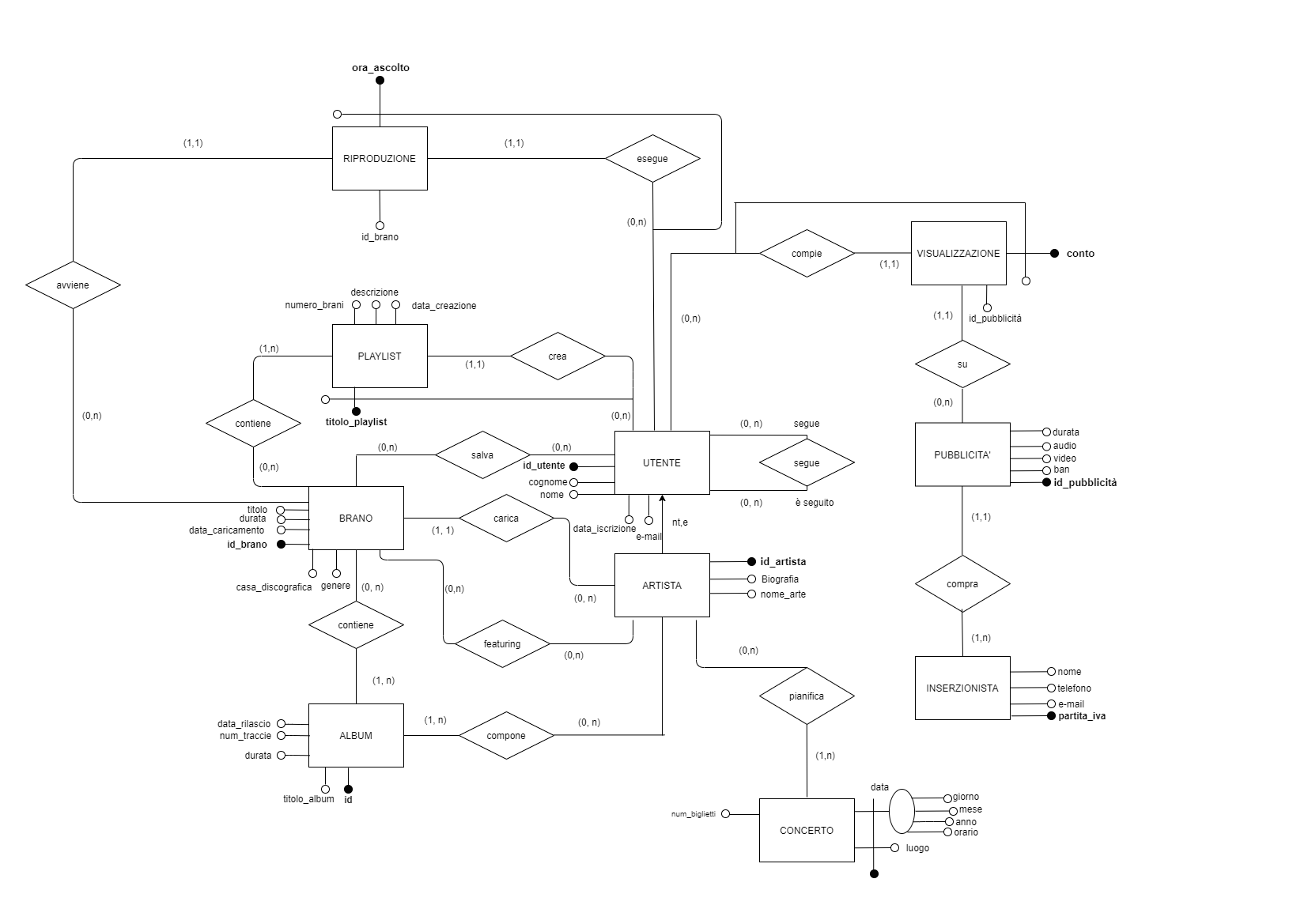
# Auto Associazione

Il concetto di auto associazione è necessario se vogliamo implementare una funzione “social network” alla piattaforma, un utente può seguire altri utenti o artisti vedendo così cosa sta ascoltando in un preciso momento o accedere alle sue playlist, l’associazione è bidirezionale e ha cardinalità (n, m) perché un utente può seguire ed essere seguito da più utenti.

# Gerarchia

Implementare una gerarchia ci viene in aiuto quando dobbiamo fare una distinzione tra i possibili utilizzatori della piattaforma, un artista infatti è una sottoclasse di utente, in quanto ha tutti gli attributi di utente e può eseguire tutte le sue azioni, in aggiunta un artista ha la possibilità di caricare album e canzoni, la gerarchia è di tipo non totale in quanto un utente può non essere un artista ed è esclusiva siccome è presente solo un’entità figlio

# Schema ER - Completo



# Progettazione Logica

A questo punto del progetto è necessario trasformare lo schema ER in uno schema logico relazionale attraverso le seguenti fasi:

1. Eliminazione delle gerarchie ISA.
2. Selezione delle chiavi primarie e eliminazione delle identificazioni esterne
3. Trasformazione degli attributi composti e/o multipli
4. Traduzione di entità e associazioni in schemi di relazioni
5. Verifica della normalizzazione

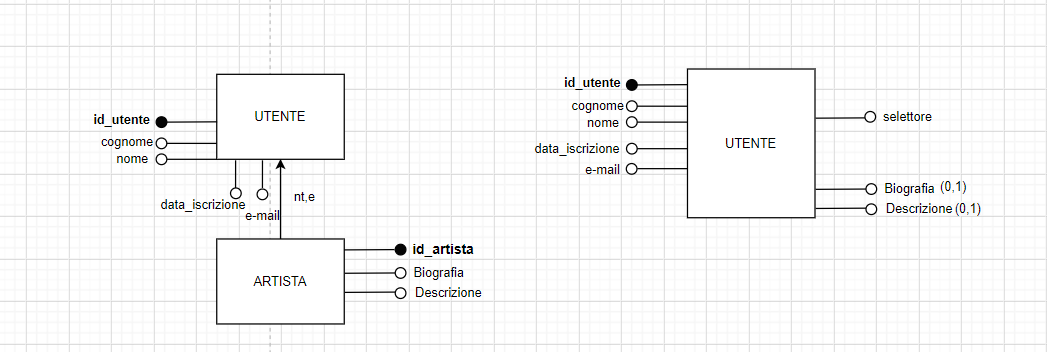
## Eliminazione delle Gerarchie ISA

Abbiamo una sola gerarchia ISA, quella che definisce l’artista come un utente che può effettuare altre operazioni ed avere più attributi, la gerarchia è non totale ed esclusiva, quindi non possiamo applicare un collasso verso il basso, possiamo scegliere tra mantenere le due entità separate collegate da una associazione oppure applicare un collasso verso l’alto, quest’ultima risulta conveniente anche perché tutte le associazioni che partono da artista hanno cardinalità minima 0.

Abbiamo quindi un’entità utente con un selettore, valore 0 per utente semplice e 1 per artista, abbiamo due attributi opzionali, ovvero biografia e descrizione, che sono attributi unici dell’artista.

L’identificatore diventa id\_utente, e distingue in modo univoco un qualsiasi utilizzatore della piattaforma

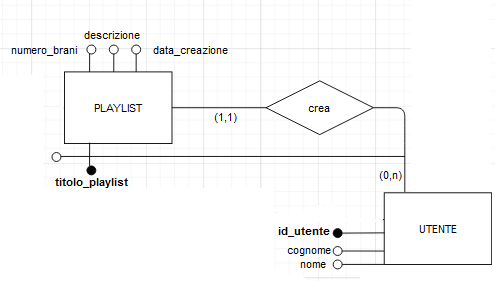
Le associazioni che erano solo di Artista hanno cardinalità minima e avranno valori Null nel caso di account utente. La pecca del collasso verso l’alto è una quantità abbastanza grande di valori null, in quanto il numero di utenti supera quello degli artisti.



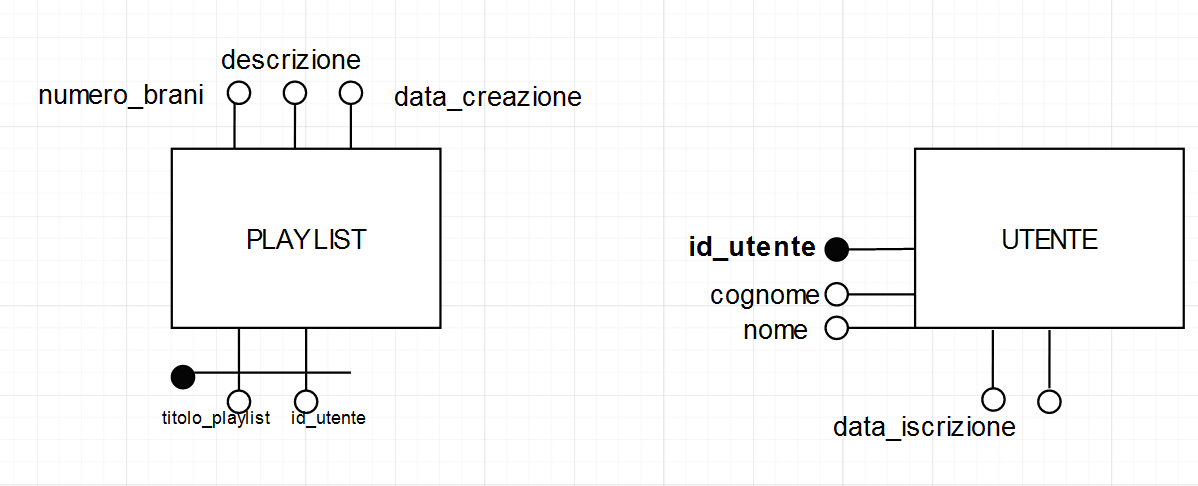
## Selezione delle chiavi primarie ed eliminazione delle identificazioni esterne

Occorre designare una chiave primaria, solitamente si sceglie la chiave che è usata più di frequente per accedere all’entità, e in secondo luogo si preferiscono chiavi semplici ed interne.

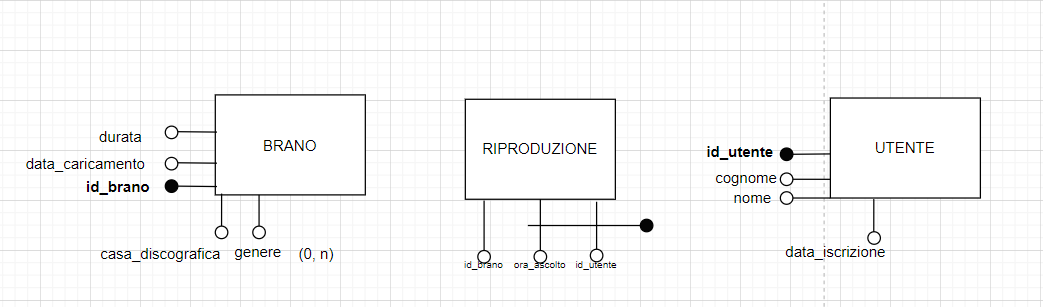
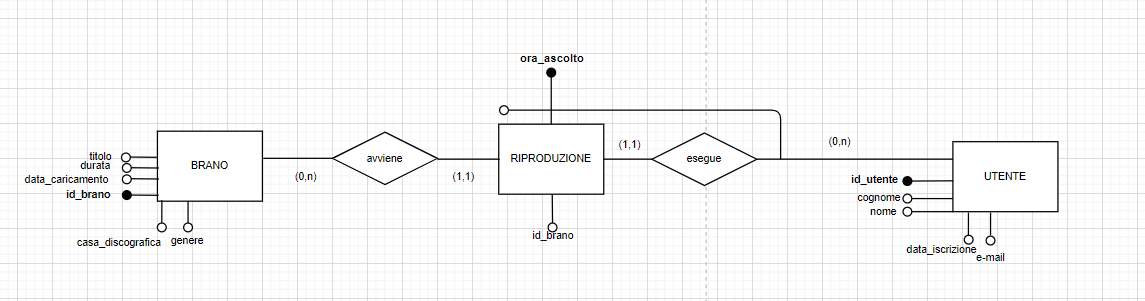
Gli identificatori esterni vanno eliminati, questo comporta il trasporto della chiave primaria dell’entità forte sull’entità debole, la chiave trasportata diventa chiave esterna, in questo modo l’associazione è rappresentata attraverso la chiave e può essere eliminata.



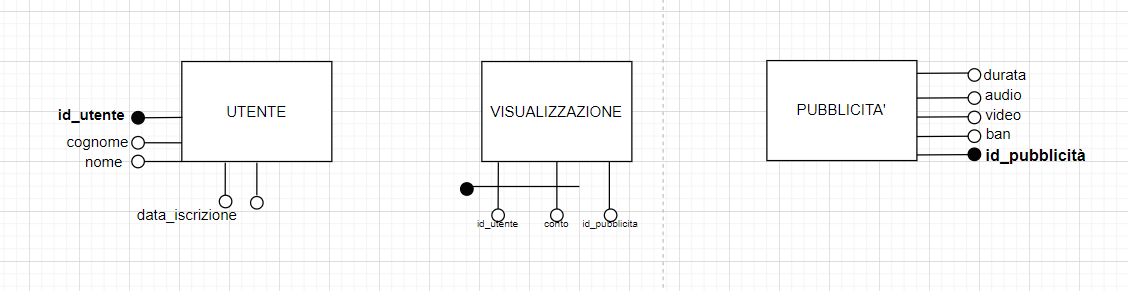
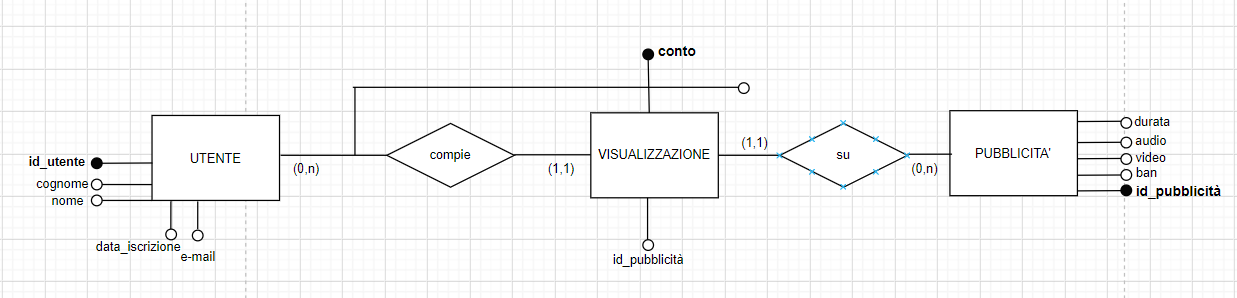
Partiamo da utente e playlist, una playlist è identificata dal titolo della playlist e dall’utente che l’ha creata, quindi la chiave di playlist diventerà (**titolo\_playlist, id\_utente**), in questo modo possiamo eliminare l’associazione.



Proseguiamo con l’entità riproduzione che è un’entità debole rispetto a brano e rispetto ad utente, possiamo eliminare le due associazioni scegliendo come chiave di riproduzione (**ora\_ascolto, id\_utente**):

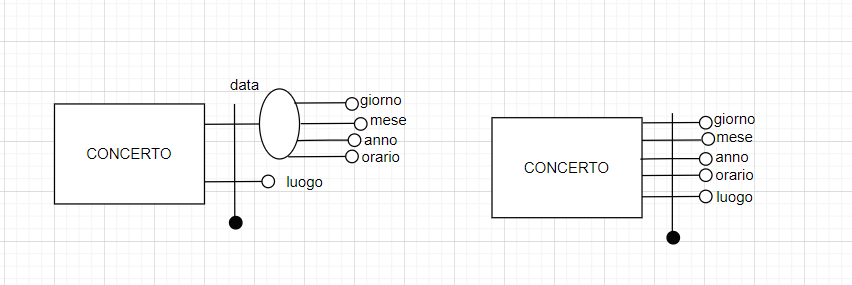


Infine rimane l’entità visualizzazione che verrà sviluppata in modo simile all’entità riproduzione, identificata univocamente tramite (**id\_utente, ora\_visualizzazione**):



## Trasformazione degli attributi composti

Nell’entità Concerto abbiamo fatto uso di un attributo composto per indicare informazioni su un dato concerto, come data e ora. Siccome i dati sono essenziali, in quanto distinguono univocamente un concerto, dobbiamo mantenere tutti gli attributi, per fare ciò eliminiamo l’attributo composto



## Traduzione di entità e associazioni in schemi di relazioni

Le entità Brano e Utente sono legate dall’associazione Carica (N:1), possiamo unire l’associazione a Brano,

**Brano (id\_brano, data\_caricamento, titolo, casa\_discografica, genere, durata, id\_utente)**

**FK: id\_utente references utente**

Anche Inserzionista e Pubblicità sono legate dall’associazione Compra(N:1), quindi procediamo in modo analogo:

**Pubblicità(id\_pubblicità, partita\_iva, durata, banner, audio, video)**

**FK: partita\_iva references Inserzionista**

Possiamo quindi tradurre l’entità Inserzionista:

**Inserzionista(partita\_iva, telefono, email, nome)**

**AK: email**

Come visto in precedenza, nelle entità che fanno uso di identificazione esterna possiamo accorpare l’associazione nell’entità debole, quindi per Riproduzione, Visualizzazione e Playlist avremo le seguenti relazioni:

**Utente (id\_utente, cognome, nome, email, data\_iscrizione, biografia, descrizione)**

**AK: email**

**Album (id\_album, titolo, durata, num\_traccie, data\_rilascio)**

**Composizione(id\_album, id\_utente)**

**FK: id\_album references Album**

**FK: id\_artista references Utente**

**Concerto (giorno, mese, anno, orario\_inzio, luogo, biglietti\_disponibili)**

**Pianificazione (id\_utente, giorno, mese, anno, orario\_inizio, luogo)**

**FK: id\_utente references Utente**

**FK: (giorno, mese, anno, orario\_inizio, luogo) references Concerto**

**Riproduzione (id\_utente, ora\_ascolto, id\_brano)**

**FK: id\_utente references Utente**

**FK: id\_brano references Brano**

**Visualizzazione (id\_pubblicità, ora\_visualizzazione, id\_utente)**

**FK: id\_pubblicità references Pubblicità**

**FK: id\_utente references Utente**

**Playlist (titolo\_playlist, id\_utente, numero\_brani, descrizione, data\_creazione)**

**FK: id\_utente References Utente**

Le altre entità legate ad Utente sono tutte associazioni (N:M) possiamo quindi procedere con una traduzione standard

Traduciamo l’associazione Featuring tra Brano e Artista

**Featuring (id\_utente, id\_brano)**

**FK: id\_utente references Utente**

**FK: id\_brano references Brano**

Mentre l’associazione Salva(N:M) che lega un Utente e i vari Brani che può salvare, prenderà il nome di Libreria, per questioni di comodità e pragmaticità:

**Libreria(id\_utente, id\_brano)**

**FK: id\_utente references Utente**

**FK: id\_brano references Brano**

Traduciamo l’associazione Contiene(N:M) tra Brano e Album in modo standard, prenderà il nome di ContieneAlbum per distinguerla dal legame tra brano e playlist

**ContieneAlbum (id\_album, id\_brano)**

**FK: id\_album references Album**

**FK: id\_brano references Brano**

In modo analogo per la relazione che lega un brano ad una playlist:

**ContienePlaylist (titolo\_playlist, id\_utente, id\_brano)**

**FK: id\_brano references Brano**

**FK: (titolo\_playlist, id\_utente) references Playlist**

Rimane da tradurre l’auto associazione N:M Segue, l’associazione si traduce con una relazione che conterrà due volte la chiave dell’entità Utente, follower indica l’utente che “segue”, mentre following l’utente che si sta “seguendo”

**Segue (follower, following)**

**FK: follower references Utente**

**FK: following references Utente**

Ricapitolando:

***Utente*** (id\_utente, cognome, nome, email, data\_iscrizione, biografia,nome\_arte)

AK: email

***Brano*** (id\_brano, data\_caricamento, titolo, casa\_discografica, genere, durata, id\_utente)

FK: id\_utente references utente

***Inserzionista***(partita\_iva, telefono, email, nome)

AK: email

***Pubblicità***(id\_pubblicità, partita\_iva, durata, banner, audio, video)

FK: partita\_iva references Inserzionista

***Album*** (id\_album, titolo, durata, num\_traccie, data\_rilascio)

***Composizione*** (id\_album, id\_utente)

FK: id\_album references Album

FK: id\_artista references Utente

***Concerto*** (giorno, mese, anno, orario\_inzio, luogo, biglietti\_disponibili)

***Pianificazione*** (id\_utente, giorno, mese, anno, orario\_inizio, luogo)

FK: id\_utente references Utente

FK: (giorno, mese, anno, orario\_inizio, luogo) references Concert**o**

***Riproduzione*** (id\_utente, ora\_ascolto, id\_brano)

FK: id\_utente references Utente

FK: id\_brano references Brano

***Visualizzazione*** (id\_pubblicità, conto, id\_utente)

FK: id\_pubblicità references Pubblicità

FK: id\_utente references Utente

***Playlist*** (titolo\_playlist, id\_utente, numero\_brani, descrizione, data\_creazione)

FK: id\_utente References Utente

***Featuring*** (id\_utente, id\_brano)

FK: id\_utente references Utente

FK: id\_brano references Brano

***Libreria***(id\_utente, id\_brano)

FK: id\_utente references Utente

FK: id\_brano references Brano

***ContieneAlbum*** (id\_album, id\_brano)

FK: id\_album references Album

FK: id\_brano references Brano

***ContienePlaylist*** (titolo\_playlist, id\_utente, id\_brano)

FK: id\_brano references Brano

FK: (titolo\_playlist, id\_utente) references Playlist

***Segue*** (follower, following)

FK: follower references Utente

FK: following references Utente

## Verifica di Normalizzazione

Dopo aver effettuato la verifica della normalizzazione non abbiamo ritenuto necessaria alcuna ulteriore modifica.

# Creazione Tabelle

CREATE TABLE Utente

(

id\_utente char(16) PRIMARY KEY,

nome varchar(25) NOT NULL,

cognome varchar(25) NOT NULL,

nome\_arte varchar(50),

email varchar(80) NOT NULL UNIQUE,

data\_iscrizione date NOT NULL,

biografia varchar(100),

CHECK(LENGTH(id\_utente)=16 AND id\_utente LIKE 'U%')

);

CREATE TABLE Brano

(

id\_brano char(16) PRIMARY KEY,

id\_utente char(16) NOT NULL,

titolo varchar(25) NOT NULL,

casa\_discografica varchar(25),

genere char(16),

durata time NOT NULL,

FOREIGN KEY (id\_utente) REFERENCES Utente(id\_utente)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_utente)=16 and id\_utente LIKE 'U%'),

CHECK(LENGTH(id\_brano)=16 AND id\_brano LIKE 'B%')

);

CREATE TABLE Inserzionista

(

partita\_iva char(11) PRIMARY KEY,

telefono varchar(22) NOT NULL,

email varchar(80) NOT NULL UNIQUE,

nome varchar(25) NOT NULL,

CHECK(LENGTH(partita\_iva)=11)

);

CREATE TABLE Pubblicità

(

id\_pubblicità char(16) PRIMARY KEY,

partita\_iva varchar(11) NOT NULL,

durata time NOT NULL,

banner boolean,

audio boolean,

video boolean,

FOREIGN KEY (partita\_iva) REFERENCES Inserzionista(partita\_iva)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_pubblicità)=16 AND id\_pubblicità LIKE 'P%')

);

CREATE TABLE Album

(

id\_album char(16) PRIMARY KEY,

titolo varchar(50) NOT NULL,

data\_rilascio date NOT NULL

CHECK(LENGTH(id\_album)=16 AND id\_album LIKE 'A%')

);

CREATE TABLE Composizione

(

id\_album char(16) NOT NULL,

id\_utente char(16) NOT NULL,

PRIMARY KEY(id\_album,id\_utente),

FOREIGN KEY (id\_album) REFERENCES Album(id\_album)

ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (id\_utente) REFERENCES Utente(id\_utente)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_album)=16 AND id\_album LIKE 'A%'),

CHECK(LENGTH(id\_utente)=16 AND id\_utente LIKE 'U%')

);

CREATE TABLE Concerto

(

giorno\_ora timestamp NOT NULL,

luogo char(16) NOT NULL,

biglietti\_disponibili integer,

PRIMARY KEY(giorno\_ora,luogo)

);

CREATE TABLE Pianificazione

(

id\_utente char(16) NOT NULL,

giorno\_ora timestamp NOT NULL,

luogo char(16) NOT NULL,

PRIMARY KEY(id\_utente,giorno\_ora,luogo),

FOREIGN KEY(id\_utente) references Utente(id\_utente),

FOREIGN KEY(giorno\_ora,luogo) references Concerto(giorno\_ora,luogo),

CHECK(LENGTH(id\_utente)=16 AND id\_utente LIKE 'U%')

);

CREATE TABLE Riproduzione

(

id\_utente CHAR(16) NOT NULL,

ora\_ascolto timestamp NOT NULL,

id\_brano VARCHAR(16) NOT NULL,

PRIMARY KEY(id\_utente,ora\_ascolto),

FOREIGN KEY(id\_utente) references Utente(id\_utente)

ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(id\_brano) references Brano(id\_brano)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_utente)=16 AND id\_utente LIKE 'U%'),

CHECK(LENGTH(id\_brano)=16 AND id\_brano LIKE 'B%')

);

CREATE TABLE Visualizzazione

(

id\_pubblicità char(16) NOT NULL,

ora\_visualizzazione timestamp NOT NULL,

id\_utente char(16) NOT NULL,

PRIMARY KEY(ora\_visualizzazione,id\_utente),

FOREIGN KEY(id\_utente) references Utente(id\_utente)

ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(id\_pubblicità) references Pubblicità (id\_pubblicità)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_utente)=16 AND id\_utente LIKE 'U%'),

CHECK(LENGTH(id\_pubblicità)=16 AND id\_pubblicità LIKE 'P%')

);

CREATE TABLE Playlist

(

titolo\_playlist varchar(16) NOT NULL,

id\_utente char(16) NOT NULL,

descrizione char(25),

data\_creazione date NOT NULL,

PRIMARY KEY(titolo\_playlist, id\_utente),

FOREIGN KEY(id\_utente) references Utente(id\_utente)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_utente)=16 AND id\_utente LIKE 'U%')

);

CREATE TABLE Featuring

(

id\_utente char(16) NOT NULL,

id\_brano varchar(16) NOT NULL,

PRIMARY KEY(id\_brano, id\_utente),

FOREIGN KEY(id\_utente) references Utente(id\_utente)

ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(id\_brano) references Brano(id\_brano)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_utente)=16 AND id\_utente LIKE 'U%'),

CHECK(LENGTH(id\_brano)=16 AND id\_brano LIKE 'B%')

);

CREATE TABLE ContieneAlbum

(

id\_album char(16) NOT NULL,

id\_brano varchar(16) NOT NULL,

PRIMARY KEY(id\_album, id\_brano),

FOREIGN KEY(id\_album) references album(id\_album)

ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(id\_brano) references Brano(id\_brano)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_brano)=16 AND id\_brano LIKE 'B%'),

CHECK(LENGTH(id\_album)=16 AND id\_album LIKE 'A%')

);

CREATE TABLE ContienePlaylist

(

titolo\_playlist char(16) NOT NULL,

id\_utente char(16) NOT NULL,

id\_brano varchar(16) NOT NULL,

PRIMARY KEY(titolo\_playlist, id\_brano, id\_utente),

FOREIGN KEY(id\_utente,titolo\_playlist) references Playlist(id\_utente,titolo\_playlist)

ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(id\_brano) references Brano(id\_brano)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_brano)=16 AND id\_brano LIKE 'B%'),

CHECK(LENGTH(id\_utente)=16 AND id\_utente LIKE 'U%')

);

CREATE TABLE Segue

(

id\_follower char(16) NOT NULL,

id\_following char(16) NOT NULL,

PRIMARY KEY(id\_follower,id\_following),

FOREIGN KEY(id\_follower) REFERENCES Utente(id\_utente)

ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(id\_following) REFERENCES Utente(id\_utente)

ON UPDATE CASCADE ON DELETE CASCADE,

CHECK(LENGTH(id\_follower)=16 AND id\_follower LIKE 'U%'),

CHECK(LENGTH(id\_following)=16 AND id\_following LIKE 'U%')

);

# Inserimento Valori

INSERT INTO Utente VALUES('U000000000000001','Alberto','Stefani',NULL,'albiemail@gmail.com','2012-08-12',NULL);

INSERT INTO Utente VALUES('U000000000000002','Danil','Lugli',NULL,'danillugli@gmail.com','2011-03-24',NULL);

INSERT INTO Utente VALUES('U000000000000004',Elia,'Tolin',NULL,'mail@eliatolin.it,'2019-01-09',NULL);

INSERT INTO Utente VALUES('U000000000000005','Giovanni','Verdi',NULL,'giovanniemail@gmail.com','2019-02-09',NULL);

INSERT INTO Utente VALUES('U000000000000007','Marco','Rossi',NULL,'marcorossi@hotmail.it','2020-07-18',NULL);

INSERT INTO Utente VALUES('U000000000000009','Luca','Bianchi',NULL,'bianchiluca@gmail.com','2016-05-26',NULL);

INSERT INTO Utente VALUES('U000000000000008','Fabrizio','De André','Fabrizio De André','FDeandre@libero.it','2011-02-18','Considerato da molti critici musicali uno dei più importanti e influenti cantautori italiani');

INSERT INTO Utente VALUES('U000000000000006','Bob','Dylan','Bob Dylan','BobbyD@gmail.com','2012-12-24','Bob Dylan è un cantautore, compositore, musicista e poeta statunitense');

INSERT INTO Utente VALUES('U000000000000003','Johnny','Cash','Johnny Cash','johnnycash@gmail.com','2013-02-16','Johnny Cash è stato un cantautore, chitarrista e attore statunitense');

INSERT INTO Utente VALUES('U000000000000014','Abel Makkonen','Tesfaye','The Weeknd','abeltesfaye@gmail.com','2010-3-1','The Weeknd è un artista e produttore discografico canadese.');

INSERT INTO Utente VALUES('U000000000000015','Giorgio','Ferrari',NULL,'giorgioferrari@gmail.com','2017-09-13',NULL);

INSERT INTO Utente VALUES('U000000000000016','Andrea','Fontana',NULL,'andreafontana@gmail.com','2016-08-12',NULL);

INSERT INTO Utente VALUES('U000000000000011','Luciano','Ligabue','Luciano Ligabue','Lucianoligabue@gmail.com','2010-1-1','Luciano Ligabue è un cantautore italiano.');

INSERT INTO Utente VALUES('U000000000000012','Tommaso','Paradiso','Tommaso Paradiso','Tommasoparadiso@gmail.com','2006-2-5','Tommaso Paradiso è un cantautore, compositore, musicista italiano.');

INSERT INTO Utente VALUES('U000000000000013', 'Aubrey', 'Grey', 'Drake', 'Aubrey@gmail.com','2013-6-1','Drake è un rapper, attore e produttore discografico americano.');

INSERT INTO Utente VALUES('U000000000000010','Vasco','Rossi', 'Vasco Rossi', 'Vascorossi@gmail.com','2009-10-1','Vasco Rossi è un cantautore italiano.');

INSERT INTO Utente VALUES('U000000000000017','Daft Punk','Daft Punk', 'Daft Punk', 'daftpunk@gmail.com','2009-10-1','Daft Punk è un gruppo musicale francese di musica elettronica');

INSERT INTO Brano VALUES('B000000000000001','U000000000000008','Una storia sbagliata','Sony','Musica d''autore','0:05:27');

INSERT INTO Brano VALUES('B000000000000002','U000000000000008','Ho visto Nina volare','Sony','Musica d''autore','0:03:59');

INSERT INTO Brano VALUES('B000000000000003','U000000000000008','Hotel Supramonte','Sony','Musica d''autore','0:04:34');

INSERT INTO Brano VALUES('B000000000000004','U000000000000003','I Walk the Line',NULL,'Country','0:02:56');

INSERT INTO Brano VALUES('B000000000000005','U000000000000003','Folsom Prison Blue',NULL,'Country','0:02:35');

INSERT INTO Brano VALUES('B000000000000007','U000000000000014','Blinding Lights','Republic Records','Pop','0:03:20');

INSERT INTO Brano VALUES('B000000000000008','U000000000000014','Faith','Republic Records','Pop','0:04:43');

INSERT INTO Brano VALUES('B000000000000009','U000000000000011','Happy Hour','Warner Music Italia','Rock','0:04:10');

INSERT INTO Brano VALUES('B000000000000010','U000000000000011','Tu Sei Lei','Warner Music Italia','Pop Rock','0:4:35');

INSERT INTO Brano VALUES('B000000000000011','U000000000000011','Per Sempre','Warner Music Italia','Rock','0:4:35');

INSERT INTO Brano VALUES('B000000000000012','U000000000000013','Feel No Ways','Young Money Entertainment','Rap','0:04:21');

INSERT INTO Brano VALUES('B000000000000013','U000000000000013','Hype','Young Money Entertainment','Rap','0:4:33');

INSERT INTO Brano VALUES('B000000000000014','U000000000000013','God''s Plan','Young Money Entertainment','Rap','0:3:51');

INSERT INTO Brano VALUES('B000000000000006','U000000000000006','Blowin'' in the Wind',NULL,'Folk','0:02:45');

INSERT INTO Brano VALUES('B000000000000015','U000000000000006','Stormy Weather','Columbia Records','Pop Rock','0:4:33');

INSERT INTO Brano VALUES('B000000000000016','U000000000000006','Stardust','Columbia Records','Rock','0:5:12');

INSERT INTO Brano VALUES('B000000000000017','U000000000000003','Tennessee','Sony','Rock','0:4:21');

INSERT INTO Brano VALUES('B000000000000018','U000000000000003','Call Your Mother','Sony','Rock','0:4:21');

INSERT INTO Brano VALUES('B000000000000019','U000000000000012','Ricordami','Sold Out Music','Pop','0:4:20');

INSERT INTO Brano VALUES('B000000000000020','U000000000000012','Ma lo vuoi capire?','Sold Out Music','Pop','0:4:31');

INSERT INTO Brano VALUES('B000000000000021','U000000000000012','Non Avere Paura','Sold Out Music','Pop','0:4:40');

INSERT INTO Brano VALUES('B000000000000022','U000000000000010','Vivere','Carosello Records','Rock','00:04:02');

INSERT INTO Brano VALUES('B000000000000023','U000000000000010','Senza Parole','Carosello Records','Rock','00:04:17');

INSERT INTO Brano VALUES('B000000000000024','U000000000000010','Albachiara','Carosello Records','Rock','00:04:12');

INSERT INTO Brano VALUES('B000000000000025','U000000000000008','Il testamento','Butterfly Music','Canzone D''autore','0:4:33');

INSERT INTO Brano VALUES('B000000000000026','U000000000000008','La Città Vecchia','Butterfly Music','Canzone D''autore','0:4:50');

INSERT INTO Brano VALUES('B000000000000027','U000000000000008','Nuvole Barocche','Butterfly Music','Canzone D''autore','0:4:13');

INSERT INTO Brano VALUES('B000000000000028','U000000000000014','Starboy','Republic Records','Pop','0:4:33');

INSERT INTO Riproduzione VALUES('U000000000000001','2020-08-20 02:34:20','B000000000000019');

INSERT INTO Riproduzione VALUES('U000000000000001','2020-08-21 12:30:12','B000000000000012');

INSERT INTO Riproduzione VALUES('U000000000000001','2020-10-03 18:34:20','B000000000000002');

INSERT INTO Riproduzione VALUES('U000000000000001','2020-08-20 14:34:20','B000000000000003');

INSERT INTO Riproduzione VALUES('U000000000000001','2019-10-01 15:04:15','B000000000000001');

INSERT INTO Riproduzione VALUES('U000000000000001','2019-10-01 15:04:15','B000000000000004');

INSERT INTO Riproduzione VALUES('U000000000000001','2019-10-01 16:00:56','B000000000000006');

INSERT INTO Riproduzione VALUES('U000000000000002','2019-10-03 15:10:56','B000000000000021');

INSERT INTO Riproduzione VALUES('U000000000000002','2020-10-09 11:10:56','B000000000000019');

INSERT INTO Riproduzione VALUES('U000000000000002','2021-01-12 09:00:00','B000000000000004');

INSERT INTO Riproduzione VALUES('U000000000000002','2020-12-23 15:10:56','B000000000000010');

INSERT INTO Riproduzione VALUES('U000000000000002','2020-10-19 13:41:33','B000000000000014');

INSERT INTO Riproduzione VALUES('U000000000000002','2020-07-05 16:33:36','B000000000000017');

INSERT INTO Riproduzione VALUES('U000000000000002','2020-08-01 22:17:22','B000000000000007');

INSERT INTO Riproduzione VALUES('U000000000000005','2020-01-08 15:10:56','B000000000000003');

INSERT INTO Riproduzione VALUES('U000000000000005','2020-05-19 14:10:56','B000000000000015');

INSERT INTO Riproduzione VALUES('U000000000000005','2020-11-16 11:00:00','B000000000000028');

INSERT INTO Riproduzione VALUES('U000000000000005','2020-12-22 17:14:56','B000000000000011');

INSERT INTO Riproduzione VALUES('U000000000000005','2020-08-23 19:31:33','B000000000000013');

INSERT INTO Riproduzione VALUES('U000000000000005','2020-04-11 23:37:36','B000000000000019');

INSERT INTO Riproduzione VALUES('U000000000000005','2020-05-01 20:57:22','B000000000000022');

INSERT INTO Riproduzione VALUES('U000000000000004','2020-03-12 17:20:56','B000000000000001');

INSERT INTO Riproduzione VALUES('U000000000000004','2020-06-23 11:40:56','B000000000000016');

INSERT INTO Riproduzione VALUES('U000000000000004','2020-12-13 10:30:00','B000000000000023');

INSERT INTO Riproduzione VALUES('U000000000000004','2020-09-12 17:34:56','B000000000000019');

INSERT INTO Riproduzione VALUES('U000000000000004','2020-06-06 14:21:33','B000000000000012');

INSERT INTO Riproduzione VALUES('U000000000000004','2020-02-14 22:53:36','B000000000000018');

INSERT INTO Riproduzione VALUES('U000000000000004','2020-07-11 19:43:22','B000000000000026');

INSERT INTO Riproduzione VALUES('U000000000000007','2020-04-15 13:21:46','B000000000000003');

INSERT INTO Riproduzione VALUES('U000000000000007','2020-06-23 11:44:33','B000000000000011');

INSERT INTO Riproduzione VALUES('U000000000000007','2020-10-19 13:22:10','B000000000000015');

INSERT INTO Riproduzione VALUES('U000000000000007','2020-06-13 18:54:26','B000000000000016');

INSERT INTO Riproduzione VALUES('U000000000000007','2020-02-16 15:21:13','B000000000000022');

INSERT INTO Riproduzione VALUES('U000000000000007','2020-01-20 12:43:36','B000000000000025');

INSERT INTO Riproduzione VALUES('U000000000000007','2020-01-29 14:13:22','B000000000000027');

INSERT INTO Riproduzione VALUES('U000000000000009','2020-01-04 11:31:46','B000000000000013');

INSERT INTO Riproduzione VALUES('U000000000000009','2020-03-25 15:24:33','B000000000000001');

INSERT INTO Riproduzione VALUES('U000000000000009','2020-05-29 13:42:10','B000000000000015');

INSERT INTO Riproduzione VALUES('U000000000000009','2020-09-11 18:32:26','B000000000000026');

INSERT INTO Riproduzione VALUES('U000000000000009','2020-08-24 10:21:13','B000000000000012');

INSERT INTO Riproduzione VALUES('U000000000000009','2020-11-21 08:43:36','B000000000000005');

INSERT INTO Riproduzione VALUES('U000000000000009','2020-07-29 07:53:22','B000000000000027');

INSERT INTO Riproduzione VALUES('U000000000000015','2020-06-12 01:11:46','B000000000000019');

INSERT INTO Riproduzione VALUES('U000000000000015','2020-05-21 08:24:33','B000000000000015');

INSERT INTO Riproduzione VALUES('U000000000000015','2020-12-30 11:22:59','B000000000000002');

INSERT INTO Riproduzione VALUES('U000000000000015','2020-03-15 17:38:36','B000000000000005');

INSERT INTO Riproduzione VALUES('U000000000000015','2020-02-20 12:25:23','B000000000000002');

INSERT INTO Riproduzione VALUES('U000000000000015','2020-11-30 11:53:16','B000000000000024');

INSERT INTO Riproduzione VALUES('U000000000000015','2020-01-01 08:23:20','B000000000000011');

INSERT INTO Riproduzione VALUES('U000000000000016','2020-05-12 12:11:46','B000000000000015');

INSERT INTO Riproduzione VALUES('U000000000000016','2020-01-21 06:24:33','B000000000000009');

INSERT INTO Riproduzione VALUES('U000000000000016','2020-09-03 15:22:59','B000000000000007');

INSERT INTO Riproduzione VALUES('U000000000000016','2020-02-15 12:35:36','B000000000000004');

INSERT INTO Riproduzione VALUES('U000000000000016','2020-07-15 17:27:23','B000000000000028');

INSERT INTO Riproduzione VALUES('U000000000000016','2020-06-22 11:13:16','B000000000000027');

INSERT INTO Riproduzione VALUES('U000000000000016','2020-12-01 08:03:20','B000000000000013');

INSERT INTO Album VALUES('A000000000000001','Gli Originali','2017-11-24');

INSERT INTO Album VALUES('A000000000000002','Nome e Cognome','2005-06-01');

INSERT INTO Album VALUES('A000000000000003','After Hour','2020-03-01');

INSERT INTO Album VALUES('A000000000000004','Views','2016-04-29');

INSERT INTO Album VALUES('A000000000000005','Scorpion','2018-06-29');

INSERT INTO Album VALUES('A000000000000006','Triplicate','2017-03-31');

INSERT INTO Album VALUES('A000000000000007','Out Amoung The Stars','2014-03-25');

INSERT INTO Album VALUES('A000000000000008','Ma lo vuoi capire?','2020-07-20');

INSERT INTO Album VALUES('A000000000000009','Ricordami','2020-02-18');

INSERT INTO Album VALUES('A000000000000010','Non Avere Paura','2020-06-22');

INSERT INTO Album VALUES('A000000000000011','Gli Spari Sopra','1993-02-06');

INSERT INTO Album VALUES('A000000000000012','Albachiara','1979-05-19');

INSERT INTO Album VALUES('A000000000000013','Starboy','2016-11-25');

INSERT INTO Album VALUES('A000000000000014','Mondovisione','2013-09-20');

INSERT INTO Album VALUES('A000000000000015','Tu che m''ascolti insegnami','2017-08-28');

INSERT INTO ContieneAlbum VALUES('A000000000000015','B000000000000001');

INSERT INTO ContieneAlbum VALUES('A000000000000015','B000000000000002');

INSERT INTO ContieneAlbum VALUES('A000000000000015','B000000000000003');

INSERT INTO ContieneAlbum VALUES('A000000000000002','B000000000000009');

INSERT INTO ContieneAlbum VALUES('A000000000000014','B000000000000010');

INSERT INTO ContieneAlbum VALUES('A000000000000002','B000000000000011');

INSERT INTO ContieneAlbum VALUES('A000000000000003','B000000000000007');

INSERT INTO ContieneAlbum VALUES('A000000000000003','B000000000000008');

INSERT INTO ContieneAlbum VALUES('A000000000000004','B000000000000012');

INSERT INTO ContieneAlbum VALUES('A000000000000004','B000000000000013');

INSERT INTO ContieneAlbum VALUES('A000000000000005','B000000000000014');

INSERT INTO ContieneAlbum VALUES('A000000000000006','B000000000000006');

INSERT INTO ContieneAlbum VALUES('A000000000000006','B000000000000015');

INSERT INTO ContieneAlbum VALUES('A000000000000006','B000000000000016');

INSERT INTO ContieneAlbum VALUES('A000000000000007','B000000000000004');

INSERT INTO ContieneAlbum VALUES('A000000000000007','B000000000000005');

INSERT INTO ContieneAlbum VALUES('A000000000000008','B000000000000020');

INSERT INTO ContieneAlbum VALUES('A000000000000009','B000000000000019');

INSERT INTO ContieneAlbum VALUES('A000000000000010','B000000000000021');

INSERT INTO ContieneAlbum VALUES('A000000000000011','B000000000000022');

INSERT INTO ContieneAlbum VALUES('A000000000000011','B000000000000023');

INSERT INTO ContieneAlbum VALUES('A000000000000012','B000000000000024');

INSERT INTO ContieneAlbum VALUES('A000000000000013','B000000000000028');

INSERT INTO ContieneAlbum VALUES('A000000000000001','B000000000000025');

INSERT INTO ContieneAlbum VALUES('A000000000000001','B000000000000026');

INSERT INTO ContieneAlbum VALUES('A000000000000001','B000000000000027');

INSERT INTO Composizione VALUES('A000000000000001','U000000000000008');

INSERT INTO Composizione VALUES('A000000000000002','U000000000000011');

INSERT INTO Composizione VALUES('A000000000000003','U000000000000014');

INSERT INTO Composizione VALUES('A000000000000004','U000000000000013');

INSERT INTO Composizione VALUES('A000000000000005','U000000000000013');

INSERT INTO Composizione VALUES('A000000000000006','U000000000000006');

INSERT INTO Composizione VALUES('A000000000000007','U000000000000003');

INSERT INTO Composizione VALUES('A000000000000008','U000000000000012');

INSERT INTO Composizione VALUES('A000000000000009','U000000000000012');

INSERT INTO Composizione VALUES('A000000000000010','U000000000000012');

INSERT INTO Composizione VALUES('A000000000000011','U000000000000010');

INSERT INTO Composizione VALUES('A000000000000015','U000000000000008');

INSERT INTO Composizione VALUES('A000000000000013','U000000000000014');

INSERT INTO Composizione VALUES('A000000000000012','U000000000000010');

INSERT INTO Composizione VALUES('A000000000000014','U000000000000011');

INSERT INTO Playlist VALUES('italian','U000000000000002','Canzoni italiane','2020-11-06');

INSERT INTO Playlist VALUES('straniere','U000000000000005','Canzoni straniere','2020-01-20');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000019');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000020');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000021');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000001');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000002');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000003');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000009');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000010');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000011');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000022');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000023');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000024');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000025');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000026');

INSERT INTO ContienePlaylist VALUES ('italian','U000000000000002','B000000000000027');

INSERT INTO ContienePlaylist VALUES ('straniere','U000000000000005','B000000000000007');

INSERT INTO ContienePlaylist VALUES ('straniere','U000000000000005','B000000000000008');

INSERT INTO ContienePlaylist VALUES ('straniere','U000000000000005','B000000000000012');

INSERT INTO ContienePlaylist VALUES ('straniere','U000000000000005','B000000000000013');

INSERT INTO ContienePlaylist VALUES ('straniere','U000000000000005','B000000000000014');

INSERT INTO Segue VALUES('U000000000000001','U000000000000008');

INSERT INTO Segue VALUES('U000000000000002','U000000000000008');

INSERT INTO Segue VALUES('U000000000000004','U000000000000008');

INSERT INTO Segue VALUES('U000000000000005','U000000000000008');

INSERT INTO Segue VALUES('U000000000000009','U000000000000008');

INSERT INTO Segue VALUES('U000000000000001','U000000000000017');

INSERT INTO Segue VALUES('U000000000000015','U000000000000017');

INSERT INTO Segue VALUES('U000000000000016','U000000000000017');

INSERT INTO Segue VALUES('U000000000000005','U000000000000017');

INSERT INTO Segue VALUES('U000000000000009','U000000000000017');

INSERT INTO Segue VALUES('U000000000000003','U000000000000014');

INSERT INTO Segue VALUES('U000000000000015','U000000000000014');

INSERT INTO Segue VALUES('U000000000000016','U000000000000014');

INSERT INTO Segue VALUES('U000000000000007','U000000000000014');

INSERT INTO Segue VALUES('U000000000000002','U000000000000014');

INSERT INTO Segue VALUES('U000000000000003','U000000000000002');

INSERT INTO Segue VALUES('U000000000000007','U000000000000001');

INSERT INTO Segue VALUES('U000000000000016','U000000000000001');

INSERT INTO Segue VALUES('U000000000000002','U000000000000003');

INSERT INTO Segue VALUES('U000000000000001','U000000000000002');

INSERT INTO inserzionista VALUES('01234567891','2465764321','volkswagen@commercial.it','Volkswagen');

INSERT INTO inserzionista VALUES('12345678910','3465765331','cocacola@commercial.it','Coca-Cola');

INSERT INTO pubblicità VALUES('P000000000000001','01234567891','00:00:30',false,true,false);

INSERT INTO pubblicità VALUES('P000000000000002','01234567891','00:01:00',false,false,true);

INSERT INTO pubblicità VALUES('P000000000000003','12345678910','00:02:30',true,false,false);

INSERT INTO visualizzazione VALUES('P000000000000001','2020-11-26 20:30:23','U000000000000001');

INSERT INTO visualizzazione VALUES('P000000000000002','2020-02-12 20:14:20','U000000000000002');

INSERT INTO visualizzazione VALUES('P000000000000001','2020-01-13 18:30:01','U000000000000003');

INSERT INTO visualizzazione VALUES('P000000000000003','2020-01-26 16:30:23','U000000000000005');

INSERT INTO visualizzazione VALUES('P000000000000002','2020-12-23 06:14:20','U000000000000001');

INSERT INTO visualizzazione VALUES('P000000000000003','2020-10-06 17:30:01','U000000000000003');

INSERT INTO Featuring VALUES('U000000000000017','B000000000000028');

INSERT INTO Concerto VALUES('2021-09-12 16:00:00','Bologna',30000);

INSERT INTO Concerto VALUES('2021-11-30 21:00:00','Roma',60000);

INSERT INTO Concerto VALUES('2021-12-02 20:00:00','Bologna',25000);

INSERT INTO Pianificazione VALUES('U000000000000010','2021-11-30 21:00:00','Roma');

INSERT INTO Pianificazione VALUES('U000000000000017','2021-09-12 16:00:00','Bologna');

INSERT INTO Pianificazione VALUES('U000000000000011','2021-12-02 20:00:00','Bologna');

# Query

## Query di Modifica

Nel caso un utente volesse cambiare nome:

update utente

set nome='Giorgio', cognome='Fiori', email='gFiori@gmail.com'

where id\_utente='U000000000000007';

## Query di Eliminazione

Nel caso un utente si volesse eliminare verranno eliminati anche tutti i riferiementi nelle altre tabelle:

DELETE FROM utente

WHERE id\_utente='U000000000000009'

## Query di Interrogazione

Visualizzare tutti i concerti pianificati in un determinato luogo:

select concerto.giorno\_ora as Giorno, nome\_arte as Artista

from concerto, pianificazione, utente

where concerto.giorno\_ora = pianificazione.giorno\_ora

and pianificazione.id\_utente = utente.id\_utente

and concerto.luogo = 'Bologna'

Visualizzare tutti i dettagli di ogni album presente:

select album.titolo, nome\_arte as artista, count(brano.id\_brano) as NumeroBrani, SUM(brano.durata) as DurataTotale

from album, contieneAlbum, brano, composizione, utente

where album.id\_album = contieneAlbum.id\_album

and contieneAlbum.id\_brano = brano.id\_brano

and album.id\_album = composizione.id\_album

and composizione.id\_utente = utente.id\_utente

group by album.titolo, artista

Mostrare tutti i follower di un’artista:

select ut.nome, ut.cognome

from utente as ut, segue, utente as art

where ut.id\_utente = segue.id\_follower

and segue.id\_following = art.id\_utente

and art.nome\_arte = 'Fabrizio De André'

Mostrare gli ascolti complessivi di tutti gli artisti:

select nome\_arte, count(ora\_ascolto)

from riproduzione, brano, utente

where riproduzione.id\_brano = brano.id\_brano

and brano.id\_utente = utente.id\_utente

group by nome\_arte