



Progettazione e sviluppo di una base di dati relazionale per un sistema di gestione di librerie musicali di un gruppo di utenti

Francesco Orlando Alfredo Valerij Laino

N86003618 N86003776

11 gennaio 2022

Indice

1	Descrizione del progetto	4
1.1	Descrizione e analisi del progetto	4
1.2	Requisiti identificati	4
2	Progettazione concettuale	6
2.1	Introduzione	6
2.2	Class Diagram	7
2.3	Analisi della ristrutturazione del Class Diagram	7
2.3.1	Analisi delle ridondanze	7
2.3.2	Analisi degli identificativi	8
2.3.3	Rimozione degli attributi multipli	8
2.3.4	Rimozione degli attributi composti	8
2.3.5	Partizione/Accorpamento delle associazioni	8
2.3.6	Rimozione delle gerarchie	9
2.4	Class Diagram ristrutturato	10
2.5	Dizionario delle classi	11
2.6	Dizionario delle associazioni	13
2.7	Dizionario dei vincoli	14
3	Schema logico	16
4	Progettazione fisica	18
4.1	Definizione dei domini	18
4.1.1	TYPE.FORMATO	18
4.1.2	TYPE.QUALITA	18
4.1.3	TYPE.EMAIL	18
4.1.4	TYPE.SESSO	18
4.1.5	TYPE.FASCIA	19
4.1.6	TYPE.VOTO	19
4.2	Definizione delle tabelle	19
4.2.1	ALBUM	19
4.2.2	TRACCIA	19
4.2.3	UTENTE	20
4.2.4	PLAYLIST	20

4.2.5	ASCOLTA	20
4.2.6	VOTA	21
4.2.7	CONTIENE	21
4.2.8	ARTISTA	21
4.2.9	INCIDE	22
4.2.10	PRODUCE	22
4.3	Funzioni e trigger per le automazioni	22
4.3.1	Edit_Voto_Traccia_INS	22
4.3.2	Edit_Voto_Traccia_DEL	23
4.3.3	Edit_Voto_Album_UPD	23
4.3.4	Ntracce_Durata_Voto_Album_DEL	24
4.3.5	Ntracce_Durata_Album_INS	24
4.3.6	Voto_Artista_UPD	24
4.3.7	Voto_Artista_DEL	25
4.3.8	Playlist_Default	26
4.3.9	Ntracce_durata_Playlist_INS	26
4.3.10	Ntracce_durata_Playlist_DEL	27
4.3.11	DEL_Artista_Pro	27
4.3.12	DEL_Artista_INC	28
4.4	Funzioni e trigger per i vincoli	29
4.4.1	AscoltiNoPremium	29
4.4.2	VotiNoPremium	29
4.4.3	TracciaCover	30
4.4.4	FormatoAnnoRemastered	31
4.4.5	AlbumAnnoCover	32
4.4.6	ArtistiUgualiRemastered	33
5	Manuale d'uso	35
5.1	Creazione utenti	35
5.1.1	UtenteBase	35
5.1.2	UtentePremium	35
5.1.3	UtenteAdmin	36
5.2	Popolamento	36
5.2.1	ARTISTI	36
5.2.2	ALBUM	37
5.2.3	TRACCE	38
5.2.4	UTENTI	48
5.2.5	INCISIONI	48
5.2.6	PRODUZIONI	49
5.2.7	ASCOLTI	53
5.2.8	VOTI	54
5.2.9	CONTIENE	55

Capitolo 1

Descrizione del progetto

1.1 Descrizione e analisi del progetto

In questo progetto si sviluppa una base di dati utile per la gestione delle librerie musicali per un gruppo di utenti. Ogni utente potrà ascoltare diverse canzoni, creare delle playlist ,pubbliche o private, ed aggiungere delle canzoni alla playlist brani preferiti. Viene data anche la possibilità di creare un profilo premium per poter accedere ad una qualità delle tracce migliori e poter dare un voto da 1 a 10 ad ogni traccia, quindi in base alla media dei voti ogni album ed ogni artista avrà un indice di gradimento, non verranno invece gestiti i prezzi degli account premium. Viene poi data la possibilità ad un admin di monitorare gli ascolti delle tracce in determinate fasce orarie e di aggiungere o eliminare tracce, artisti ecc. Ogni traccia può avere la versione rimasterizzata o la cover creata da un altro artista.

1.2 Requisiti identificati

Leggendo attentamente la traccia, abbiamo individuato le seguenti entità con le loro relative associazioni:

- Per gestire gli utenti abbiamo un'entità **Utente** che è l'entità padre di:
 - **UtentePremium:** un tipo di utente che ha una relazione con **Traccia** che gli permette di dare un voto, in più ha la possibilità di ascoltare le tracce ad una qualità migliore.
 - **Admin:** un utente che può monitorare gli ascolti delle tracce ordinate per fasce orarie e gestire la base di dati.

L'utente è in relazione con:

- **Playlist:** ogni utente può creare una playlist con i brani che vuole e decidere se rendere la playlist pubblica o privata. Viene fornita di

default una playlist “Brani preferiti” dove l’utente potrà inserire le tracce che preferisce.

- **Traccia:** in questa relazione vengono specificate quali tracce ha ascoltato l’utente e in quali fasce orarie.

- Abbiamo un’entità **Traccia** che è padre di:

- **TracciaRemastered:** è la remastered di una traccia ed avrà un formato dell’audio migliore rispetto all’originale.
- **TracciaCover:** un artista può fare una cover di una traccia di un altro artista.

Entrambe le specializzazioni saranno in relazione con Traccia, questo serve per poter sapere qual è la traccia originale.

L’entità **Traccia** oltre ad essere in relazione con **Utente** ha anche le seguenti associazioni con:

- **Playlist:** per poter specificare in quale playlist è contenuta una traccia.
- **Album:** una traccia può appartenere ad un album oppure può essere un singolo.
- **Artista:** una traccia può essere prodotta da uno o più artisti.

- Abbiamo poi bisogno di un’entità **Album** che oltre ad avere una relazione con **Traccia** è in relazione con:

- **Artista:** per sapere da chi è stato inciso un album, in questo caso un album può essere inciso da più artisti.

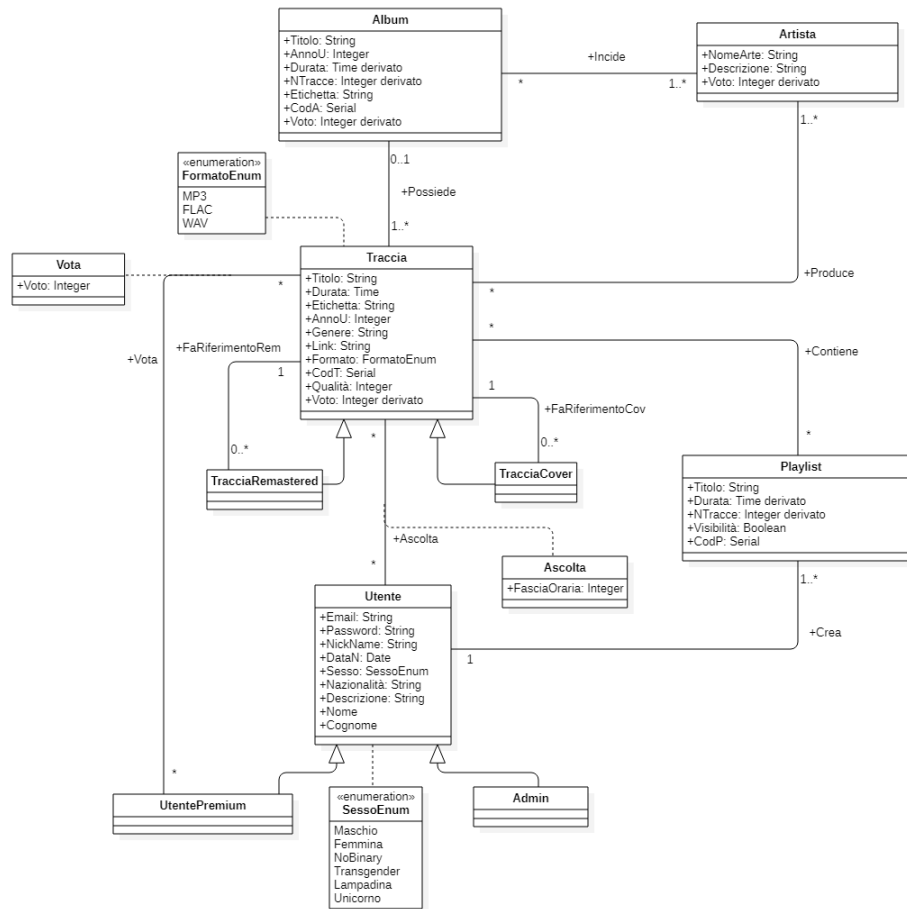
Capitolo 2

Progettazione concettuale

2.1 Introduzione

In questo capitolo viene mostrata la progettazione della base di dati. Grazie ai requisiti che abbiamo identificato definiamo un class diagram per lo schema concettuale completamente indipendente dalle strutture dati che useremo e dall'implementazione fisica.

2.2 Class Diagram



2.3 Analisi della ristrutturazione del Class Diagram

Dopo la fase di progettazione concettuale segue la fase di progettazione logica, per rendere più semplice questo passaggio andiamo a svolgere diverse analisi per ristrutturare il class diagram e rendere più efficiente l'implementazione.

2.3.1 Analisi delle ridondanze

Dato che le ridondanze che abbiamo individuato sono dei dati che verranno richiesti spesso dall'utente e non vogliamo appesantire il carico della macchina

che ospiterà la base di dati, abbiamo deciso di non eliminare le ridondanze e di memorizzarle come attributi derivati. Gli attributi in questione sono:

- **Album:**
 - **NTracce:** il numero di tracce dell'album.
 - **Durata:** la durata dell'intero album.
 - **Voto:** la media dei voti delle tracce che contiene l'album.
- **Artista:**
 - **Voto:** la media dei voti delle tracce che ha prodotto l'artista.
- **Traccia:**
 - **Voto:** la media dei voti che hanno dato gli utenti premium alla traccia.
- **Playlist:**
 - **Durata:** la durata delle tracce contenute nella playlist.
 - **NTracce:** il numero di tracce contenute nella playlist.

2.3.2 Analisi degli identificativi

Tutti gli attributi che iniziano con **Cod** sono chiave primaria dell'entità a cui appartengono. Per quanto riguarda le entità che non hanno questo tipo di attributo, abbiamo scelto i seguenti identificativi:

- Per **Artista** abbiamo scelto l'attributo **NomeArte**.
- Per **Utente** abbiamo scelto l'attributo **NickName**.

2.3.3 Rimozione degli attributi multipli

L'unico attributo che può essere multiplo è **Genere** in **Traccia**, infatti una canzone può avere come genere rap/rock, ma già dal primo class diagram l'abbiamo considerato come un unico attributo di tipo String.

2.3.4 Rimozione degli attributi composti

Non sono presenti attributi composti.

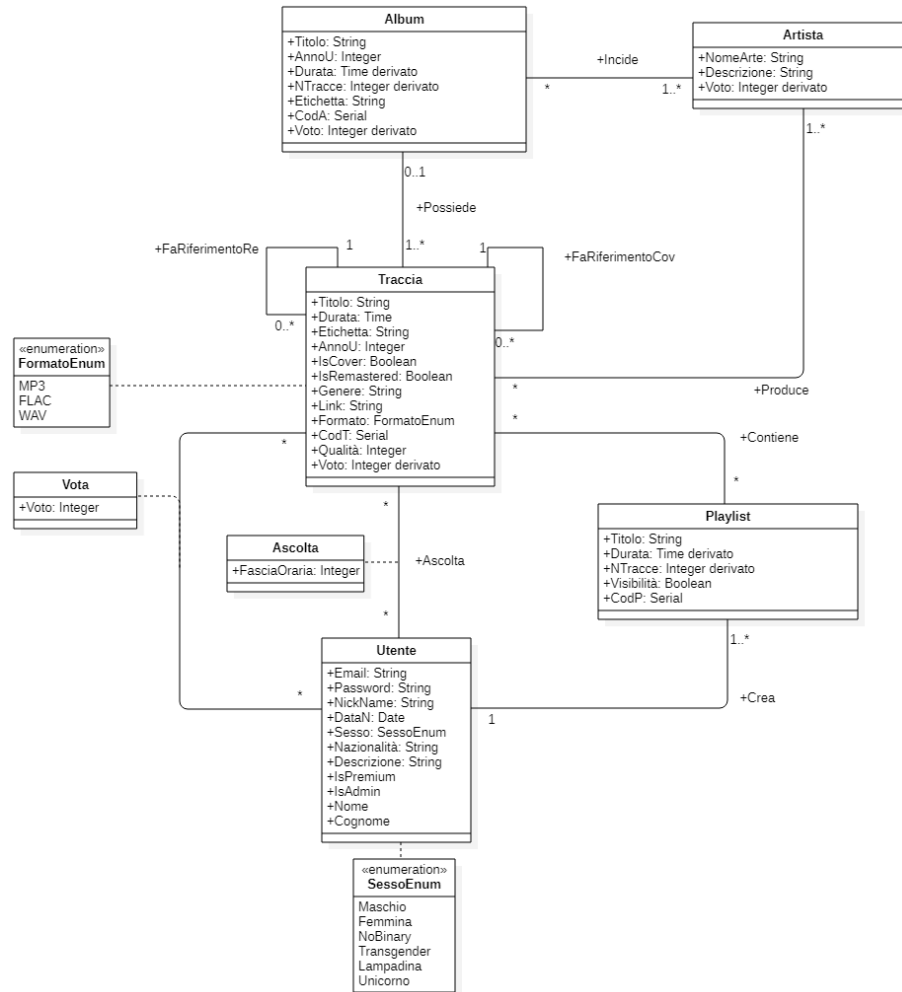
2.3.5 Partizione/Accorpamento delle associazioni

Non sono presenti associazioni da partizionare o accorpare.

2.3.6 Rimozione delle gerarchie

Le entità **UtentePremium** e **Admin** ereditano dall'entità **Utente**. Per rimuoverle abbiamo inserito due attributi **IsPremium** e **IsAdmin** nell'entità **Utente** e abbiamo aggiunto l'associazione **Vota** ad **Utente**. Abbiamo applicato la stessa strategia anche per **TracciaRemastered** e **TracciaCover**, quindi **Traccia** avrà un attributo **IsRemastered** e **IsCover**, e avrà due relazioni ricorsive per indicare qual è la traccia originale di una remastered e di una cover.

2.4 Class Diagram ristrutturato



2.5 Dizionario delle classi

Iniziamo a descrivere in maniera più dettagliata le nostre classi, andando anche a specificare gli attributi delle suddette classi.

DIZIONARIO DELLE CLASSI		
CLASSE	DESCRIZIONE	ATTRIBUTI
Album	L'album è una collezione di più tracce di uno o più artisti	CodA (<i>Serial</i>):Identificativo di un album Titolo (<i>String</i>):Titolo dell'album AnnoU (<i>Integer</i>):L'anno uscita dell'album Durata (<i>Time</i>):Durata totale dell'album Ntracce (<i>Integer</i>):Numero delle tracce in un album Etichetta (<i>String</i>):Etichetta che ha prodotto l'album Voto (<i>Integer</i>):Voto dell'album dato dalla media dei voti delle tracce
Traccia	La traccia è la canzone che l'utente potrà ascoltare, ed è composta da:	CodT (<i>Serial</i>):Codice identificativo per una traccia Durata (<i>Time</i>):Durata di una traccia Etichetta (<i>String</i>):L'etichetta che ha prodotto la traccia, che può essere anche autoprodotta AnnoU (<i>Integer</i>):L'anno di uscita della traccia IsCover (<i>Boolean</i>):Indica se la traccia è una cover o meno IsRemastered (<i>Boolean</i>):Indica se la traccia è una remastered o meno Genere (<i>String</i>):Indica i generi della traccia Link (<i>String</i>):Mostra il link di Spotify o di youtube alla traccia Formato (<i>Formato_enum</i>):Mostra il formato della traccia (tra MP3 , WAV, Flac) Voto (<i>Integer</i>):Rappresenta la media dei voti dati dagli utenti (Premium), espressa in decimali Qualità (<i>Integer</i>): Mostra la qualità della traccia (tra 128, 256, 512)
Utente	Un utente è un cliente di "Pinoxy" di cui si conservano le seguenti informazioni:	Nickname (<i>String</i>):Indica il Nickname(univoco) dell'utente Nome (<i>String</i>):Indica il nome dell'utente Cognome (<i>String</i>):Indica il cognome dell'utente Email (<i>String</i>):Indica l'indirizzo e-mail dell'utente Password (<i>String</i>):Indica la Password dell'utente DataN (<i>Date</i>):Indica la data di Nascita dell'utente Sesso (<i>Sesso_enum</i>):Indica il sesso dell'utente

CLASSE	DESCRIZIONE	ATTRIBUTI
		Nazionalità (<i>String</i>):Indica la nazionalità dell'utente Descrizione (<i>String</i>):Mostra l'eventuale descrizione dell'utente IsPremium (<i>Boolean</i>):Indica se l'utente è Premium IsAdmin (<i>Boolean</i>):Indica se l'utente è Admin
Playlist	Una Playlist è una raccolta di tracce presenti nel database, creata da un utente	CodP (<i>Serial</i>):Rappresenta il codice identificativo di una Playlist Titolo (<i>String</i>):Mostra il titolo della Playlist Durata (<i>Time</i>):Mostra la durata complessiva della Playlist (somma delle durate delle tracce che contiene) Visibilità (<i>Boolean</i>):Indica la visibilità della Playlist (True : pubblica , False : privata)
Artista	L'Artista è il creatore di una o più tracce.	NomeArte (<i>String</i>):Nome d'Arte del Artista (anche identificativo). Descrizione (<i>String</i>):Descrizione dell'artista (compilata e quindi approvata dal Artista stesso). Voto (<i>Integer</i>):Voto dell'artista (derivato dalla media dei voti delle tracce da lui create).

Questo conclude la descrizione delle nostre classi. Passiamo ora alla descrizione delle associazioni e dei vincoli.

2.6 Dizionario delle associazioni

Andiamo a descrivere, in maniera analoga anche le associazioni che ci sono tra le classi:

DIZIONARIO DELLE ASSOCIAZIONI		
ASSOCIAZIONE	DESCRIZIONE	CLASSI COINVOLTE
Possiede	Esprime quali tracce possiede un album.	Traccia [0..*] Indica le tracce che possiede un album Album [0..1] Indica a quale album appartiene una traccia
Produce	Esprime quali artisti hanno prodotto una traccia.	Traccia [0..*] Indica quali tracce ha prodotto un artista Artista [1..*] Indica quali artisti hanno prodotto una traccia
Incide	Esprime da quali artisti è stato inciso un album.	Album [0..*] Indica quali album ha inciso un artista Artista [1..*] Indica da quali artisti è stato inciso un album
Ascolta	Esprime da quali utenti è stata ascoltata una traccia.	Traccia [0..*] Indica quale traccia ha ascoltato un utente Utente [0..*] Indica da quale utente è stata ascoltata una traccia
Vota	Esprime i voti dati dagli utenti premium alle tracce.	Traccia [0..*] Indica la traccia che è stata votata da un utente Utente [0..*] Indica quale utente ha votato una traccia
FaRiferimentoRem	Esprime qual'è la traccia originale di una remastered	Traccia [0..*] Indica le remastered di una traccia originale Traccia [1] Indica la traccia originale di una remastered
FaRiferimentoCov	Esprime qual'è la traccia originale di una cover	Traccia [0..*] Indica quali sono le cover di una traccia originale Traccia [1] Indica la traccia originale di una cover
Crea	Esprime la creazione di una playlist da parte di un utente	Utente [1] Indica l'utente che ha creato una playlist. Playlist [1..*] Indica le playlist create dall'utente.
Contiene	Esprime quali tracce contiene una playlist	Traccia [0..*] Indica le tracce contenute in una playlist. Playlist [0..*] Indica le playlist che contengono la traccia.

Questo conclude anche il dizionario delle Associazioni, passiamo quindi all'ultimo dizionario, ovvero, quello dei vincoli.

2.7 Dizionario dei vincoli

DIZIONARIO DEI VINCOLI	
NOME VINCOLO	DESCRIZIONE
TYPE FORMATO	Una traccia può essere salvata in uno dei seguenti formati: MP3, WAV, FLAC
TYPE QUALITA	Una traccia può essere registrata solo in 128 / 256 / 512 kbps
TYPE EMAIL	Un utente, nel campo mail, deve inserire un indirizzo mail valido (quindi contenente una stringa, seguita dal carattere '@' seguita da un dominio)
TYPE SESSO	Un utente deve selezionare un sesso a scelta da quelli forniti dal servizio, se desidera specificarlo (tra Uomo, Donna, Transgender, Unicornio, Lamapdina)
TYPE FASCIA	Una traccia può essere ascoltata da un utente, esclusivamente in una delle 6 fasce orarie in cui la giornata è stata divisa. Di conseguenza una fascia può essere rappresentata esclusivamente da un naturale compreso tra 1 e 6
TYPE VOTO	Un voto di un utente per una traccia può essere espresso esclusivamente sotto forma di un naturale compreso tra 0 e 10
TRACCIA UNIQUE	Al fine di garantire l'unicità di una traccia bisogna imporre che il titolo, il CodA (Codice Album), l'anno di uscita ed il formato di una traccia siano attributi unici.
UTENTE UNIQUE	Per ogni e-mail possiamo aver registrato un solo utente, per questo motivo, l'attributo Email deve essere unico per ogni utente.
VOTA UNIQUE	Visto che vogliamo contare al più un voto da ogni Utente per ogni Traccia (il più recente) la coppia Nickname, CodT (Identificativo della Traccia) devono essere univoci.
CONTIENE UNIQUE	Al fine di garantire l'univocità della presenza di una traccia all'interno di una Playlist, anche la coppia CodT, CodP deve essere unica.
INCIDE UNIQUE	Al fine di garantire l'univocità della presenza di un album distinto tra quelli incisi da uno o più Artisti, anche la coppia NomeArte, CodA deve essere unica.
PRODUCE UNIQUE	Al fine di garantire l'univocità di una traccia prodotta da uno o più Artisti, anche la coppia NomeArte, CodT deve essere unica.
AscoltiNoPremium	Solo gli utenti di tipo premium (e superiore) possono ascoltare tracce di qualità maggiore a 128 kbps.
VotiNoPremium	Solo gli utenti di tipo premium (e superiore) possono votare una traccia.

NOME VINCOLO	DESCRIZIONE
TracciaCover	Se una traccia è una cover, allora gli artisti che l'hanno prodotta devono essere diversi da quelli che hanno prodotto l'originale.
FormatoAnnoRemastered	Se una traccia è remastered, allora il suo formato deve essere migliore dell'originale, inoltre l'anno di uscita deve essere maggiore di quello della traccia originale.
AlbumAnnoCover	Se una traccia è cover, allora l'anno di uscita deve essere maggiore o uguale a quello della traccia originale, inoltre l'album di appartenenza deve esser diverso.
ArtistiUgualiRemastered	Se una traccia è remastered, allora gli artisti che l'hanno prodotta devono essere gli stessi che hanno prodotto l'originale.

Capitolo 3

Schema logico

In questa fase avviene la traduzione dello schema concettuale ristrutturato nel capitolo precedente in uno schema logico. Indichiamo con una sottolineatura e il grassetto le **chiavi primarie**, mentre con due sottolineature indichiamo le chiavi esterne.

Album(**CodA**, Titolo, AnnoU, Durata, NTracce, Etichetta, Voto)

Traccia(**CodT**, Titolo, Durata, Etichetta, AnnoU, IsCover, IsRemastered, Genere, Link, Formato, Voto, Qualità, CodA, CodTR, CodTC)

CodA → Album.CodA

CodTR → Traccia.CodT

CodTC → Traccia.CodT

Utente(**NickName**, Nome, Cognome, Email, Password, DataN, Sesso, Nazionalità, Descrizione, IsPremium, IsAdmin)

Playlist(**CodP**, Titolo, Durata, NTracce, Visibilità, **NickName**)

NickName → Utente.NickName

Ascolta(**NickName**, **CodT**, FasciaOraria)

NickName → Utente.NickName

CodT → Traccia.CodT

Vota(NickName, CodT, Voto)
NickName \rightarrow Utente.NickName
CodT \rightarrow Traccia.Cod

Contiene(CodP, CodT)
CodP \rightarrow Playlist.CodP
CodT \rightarrow Traccia.CodT

Artista(NomeArte, Descrizione, Voto)

Incide(NomeArte, CodA)
NomeArte \rightarrow Artista.NomeArte
CodA \rightarrow Album.CodA

Produce(NomeArte, CodT)
NomeArte \rightarrow Artista.NomeArte
CodT \rightarrow Traccia.Cod

Capitolo 4

Progettazione fisica

Adesso mostriamo l'implementazione del database, nessun dettaglio è stato modificato ai fini dell'implementazione per rimanere coerenti con la progettazione che abbiamo fatto. Come DBMS useremo Postgres.

4.1 Definizione dei domini

Vediamo la definizione dei domini usati per alcuni tipi di dati.

4.1.1 TYPE_FORMATO

```
1 --DEFINIZIONE DEI DOMINII
2 --Dominio per il formato della traccia
3 CREATE DOMAIN Type_Formato as varchar(4)
4 CHECK (value = 'MP3' or value = 'WAV' or value = 'FLAC');
```

4.1.2 TYPE_QUALITA

```
1 --Dominio per la qualit della traccia
2 CREATE DOMAIN Type_Qualita as integer
3 CHECK (value = 128 or value = 256 or value = 512);
```

4.1.3 TYPE_EMAIL

```
1 --Dominio per l'email dell'utente
2 CREATE DOMAIN Type_Email as varchar(100)
3 CHECK (value LIKE '%@%.%.%');
```

4.1.4 TYPE_SESSO

```
1 --Dominio per il sesso dell'utente
2 CREATE DOMAIN Type_Sesso as varchar(15)
3 CHECK (value = 'Uomo' or value = 'Donna' or value = 'Altro' or
value = 'Transgender' or value = 'Lampadina' or value = '
Unicorno');
```

4.1.5 TYPE_FASCIA

```
1 --Dominio per la fascia oraria dell'ascolto
2 CREATE DOMAIN Type_Fascia as integer
3 CHECK (value >= 1 and value <= 6);
```

4.1.6 TYPE_VOTO

```
1 --Dominio per il voto dato dall'utente
2 CREATE DOMAIN Type_Voto integer
3 CHECK (value >= 0 and value <= 10);
```

4.2 Definizione delle tabelle

Adesso vediamo come sono state definite le tabelle e anche i vincoli di unicità di alcuni attributi.

4.2.1 ALBUM

```
1 --DEFINIZIONE DELLE TABELLE
2 --ALBUM
3 CREATE TABLE ALBUM (
4     CodA SERIAL,
5     Titolo VARCHAR(50) NOT NULL,
6     AnnoU INTEGER NOT NULL,
7     Durata TIME DEFAULT '00:00:00',
8     Ntracce INTEGER DEFAULT 0,
9     Etichetta VARCHAR(20) DEFAULT NULL,
10    Voto INTEGER DEFAULT 0,
11
12    PRIMARY KEY (CodA)
13 );
```

4.2.2 TRACCIA

```
1 --TRACCIA
2 CREATE TABLE TRACCIA (
3     CodT SERIAL,
4     Titolo VARCHAR(50) NOT NULL,
5     Durata TIME NOT NULL,
6     Etichetta VARCHAR(20) DEFAULT NULL,
7     AnnoU INTEGER,
8     IsCover BOOLEAN DEFAULT FALSE,
9     IsRemastered BOOLEAN DEFAULT FALSE,
10    Genere VARCHAR(30) NOT NULL,
11    Link VARCHAR(300),
12    Formato Type_Formato DEFAULT 'MP3',
13    Voto INTEGER DEFAULT 0,
14    Qualita Type_Qualita DEFAULT 128,
15    CodA INTEGER DEFAULT 0,
16    CodTR INTEGER DEFAULT 0,
17    CodTC INTEGER DEFAULT 0,
18
19    UNIQUE(Titolo, AnnoU, CodA, Formato),
```

```

20     PRIMARY KEY (CodT),
21     FOREIGN KEY (CodA) REFERENCES Album(CodA)
22         ON DELETE CASCADE
23         ON UPDATE CASCADE,
24     FOREIGN KEY (CodTR) REFERENCES Traccia(CodT)
25         ON DELETE CASCADE
26         ON UPDATE CASCADE,
27     FOREIGN KEY (CodTC) REFERENCES Traccia(CodT)
28         ON DELETE CASCADE
29         ON UPDATE CASCADE
30 );

```

4.2.3 UTENTE

```

1  --UTENTE
2  CREATE TABLE UTENTE (
3      NickName VARCHAR(20),
4      Nome VARCHAR(30) NOT NULL,
5      Cognome VARCHAR(30) NOT NULL,
6      Email Type_Email NOT NULL,
7      Password VARCHAR(20) NOT NULL,
8      DataN DATE NOT NULL,
9      Sesso Type_Sesso NOT NULL,
10     Nazionalita VARCHAR(15) DEFAULT NULL,
11     Descrizione VARCHAR(100) DEFAULT NULL,
12     IsPremium BOOLEAN DEFAULT FALSE,
13     IsAdmin BOOLEAN DEFAULT FALSE,
14
15     PRIMARY KEY (NickName),
16
17     UNIQUE(Email)
18 );

```

4.2.4 PLAYLIST

```

1  --PLAYLIST
2  CREATE TABLE PLAYLIST(
3      CodP SERIAL,
4      Titolo VARCHAR(20) NOT NULL,
5      Durata TIME DEFAULT '00:00:00',
6      NTracce INTEGER DEFAULT 0,
7      Visibilita BOOLEAN DEFAULT TRUE,
8      NickName VARCHAR(20),
9
10     PRIMARY KEY (CodP),
11     FOREIGN KEY (NickName) REFERENCES UTENTE(NickName)
12         ON DELETE CASCADE
13         ON UPDATE CASCADE
14 );

```

4.2.5 ASCOLTA

```

1  --ASCOLTA
2  CREATE TABLE ASCOLTA (
3      Nickname VARCHAR(20),
4      CodT INTEGER,
5      FasciaOraria Type_Fascia NOT NULL,

```

```

6
7 FOREIGN KEY (NickName) REFERENCES UTENTE(NickName)
8 ON DELETE CASCADE
9 ON UPDATE CASCADE,
10 FOREIGN KEY (CodT) REFERENCES TRACCIA(CodT)
11 ON DELETE CASCADE
12 ON UPDATE CASCADE
13 );

```

4.2.6 VOTA

```

1 --VOTA
2 CREATE TABLE VOTA (
3 NickName VARCHAR(20),
4 CodT INTEGER,
5 Voto Type_Voto NOT NULL,
6
7 UNIQUE(NickName, CodT),
8
9 FOREIGN KEY (NickName) REFERENCES UTENTE(NickName)
10 ON DELETE CASCADE
11 ON UPDATE CASCADE,
12 FOREIGN KEY (CodT) REFERENCES TRACCIA(CodT)
13 ON DELETE CASCADE
14 ON UPDATE CASCADE
15 );

```

4.2.7 CONTIENE

```

1 --CONTIENE
2 CREATE TABLE CONTIENE (
3 CodP INTEGER,
4 CodT INTEGER,
5
6 UNIQUE(CodP, CodT),
7
8 FOREIGN KEY (CodT) REFERENCES TRACCIA(CodT)
9 ON DELETE CASCADE
10 ON UPDATE CASCADE,
11 FOREIGN KEY (CodP) REFERENCES PLAYLIST(CodP)
12 ON DELETE CASCADE
13 ON UPDATE CASCADE
14 );

```

4.2.8 ARTISTA

```

1 --ARTISTA
2 CREATE TABLE ARTISTA (
3 NomeArte VARCHAR(30),
4 Descrizione VARCHAR(300) DEFAULT NULL,
5 Voto INTEGER DEFAULT 0,
6
7 PRIMARY KEY(NomeArte)
8 );

```

4.2.9 INCIDE

```
1 --INCIDE
2 CREATE TABLE INCIDE (
3     NomeArte VARCHAR(30),
4     CodA INTEGER,
5
6     UNIQUE(NomeArte, CodA),
7     FOREIGN KEY (NomeArte) REFERENCES ARTISTA(NomeArte)
8         ON DELETE CASCADE
9         ON UPDATE CASCADE,
10    FOREIGN KEY (CodA) REFERENCES ALBUM(CodA)
11        ON DELETE CASCADE
12        ON UPDATE CASCADE
13 );
```

4.2.10 PRODUCE

```
1 --PRODUCE
2 CREATE TABLE PRODUCE (
3     NomeArte VARCHAR(30),
4     CodT INTEGER,
5
6     UNIQUE(NomeArte, CodT),
7     FOREIGN KEY (NomeArte) REFERENCES ARTISTA(NomeArte)
8         ON DELETE CASCADE
9         ON UPDATE CASCADE,
10    FOREIGN KEY (CodT) REFERENCES TRACCIA(CodT)
11        ON DELETE CASCADE
12        ON UPDATE CASCADE
13 );
```

4.3 Funzioni e trigger per le automazioni

Seguono le funzioni più i relativi trigger per implementare le automazioni.

4.3.1 Edit_Voto_Traccia_INS

```
1 --PROCEDURE E TRIGGER PER LE AUTOMAZIONI
2 --PROCEDURA 1: All'aggiunta di un voto viene fatta la media in
   Traccia
3 CREATE OR REPLACE FUNCTION Edit_Voto_Traccia_INS() RETURNS TRIGGER
4 AS $Voto_Traccia_INS$
5 BEGIN
6     UPDATE TRACCIA
7     SET VOTO=(SELECT AVG(V.Voto)
8               FROM VOTA AS V
9               WHERE V.CodT=NEW.CodT
10              )
11     WHERE CodT=NEW.CodT;
12     RETURN Null;
13 END; $Voto_Traccia_INS$ LANGUAGE PLPGSQL;
14 --TRIGGER 1: aggiorna la media di una Traccia all'inserimento di un
   voto
```

```

15 CREATE OR REPLACE TRIGGER Voto_Traccia_INS
16 AFTER INSERT ON VOTA
17 FOR EACH ROW
18 EXECUTE PROCEDURE Edit_Voto_Traccia_INS();
19
20
21 -- TRIGGER 1.2: aggiorna il voto di una traccia dopo la modifica di
    un voto
22 CREATE OR REPLACE TRIGGER Voto_Traccia_UPD
23 AFTER UPDATE OF Voto ON VOTA
24 FOR EACH ROW
25 EXECUTE PROCEDURE Edit_Voto_Traccia_INS();

```

4.3.2 Edit_Voto_Traccia_DEL

```

1 --PROCEDURA 2: Dopo l'eliminazione di un voto viene aggiornata la
    media in Traccia
2 CREATE OR REPLACE FUNCTION Edit_Voto_Traccia_DEL() RETURNS TRIGGER
    AS $Voto_Traccia_DEL$
3 BEGIN
4     UPDATE TRACCIA
5     SET VOTO=(SELECT AVG(V.Voto)
6               FROM VOTA AS V
7               WHERE V.CodT=OLD.CodT
8               )
9     WHERE CodT=OLD.CodT;
10    RETURN Null;
11 END; $Voto_Traccia_DEL$ LANGUAGE PLPGSQL;
12
13 --TRIGGER 2: aggiorna la media di una Traccia all'eliminazione di
    un voto
14 CREATE OR REPLACE TRIGGER Voto_Traccia_DEL
15 AFTER DELETE ON VOTA
16 FOR EACH ROW
17 EXECUTE PROCEDURE Edit_Voto_Traccia_DEL();

```

4.3.3 Edit_Voto_Album_UPD

```

1 --PROCEDURA 3: Dopo aver aggiornato il voto di una traccia bisogna
    aggiornare i voti dell'album a cui appartiene
2 CREATE OR REPLACE FUNCTION Edit_Voto_Album_UPD() RETURNS TRIGGER AS
    $Voto_Album_UPD$
3 BEGIN
4     UPDATE ALBUM
5     SET Voto=(SELECT AVG(T.Voto)
6               FROM TRACCIA as T
7               WHERE T.codA=NEW.codA and T.Voto > 0
8               )
9     WHERE CodA=NEW.CodA;
10    RETURN Null;
11 END; $Voto_Album_UPD$ LANGUAGE PLPGSQL;
12
13 --TRIGGER 3: modifica il Voto del Album, a fronte della modifica del
    voto della traccia
14 CREATE OR REPLACE TRIGGER Voto_Album_UPD
15 AFTER UPDATE OF Voto ON TRACCIA
16 FOR EACH ROW

```

```
17 EXECUTE PROCEDURE Edit_Voto_Album_UPD();
```

4.3.4 Ntracce_Durata_Voto_Album_DEL

```
1 -- PROCEDURA 4: Dopo l'eliminazione di una traccia bisogna
  modificare il numero di tracce, la durata e la media dei voti
  dell'album a cui apparteneva
2 CREATE OR REPLACE FUNCTION Ntracce_Durata_Voto_Album_DEL() RETURNS
  TRIGGER AS $TRIGGER_Ntracce_Durata_Voto_Album_DEL$
3 BEGIN
4   UPDATE ALBUM
5   SET Ntracce=Ntracce-1,
6       Durata=(SELECT SUM(Durata)
7               FROM TRACCIA AS T
8               WHERE T.CodA = OLD.CodA),
9       Voto=(SELECT AVG(T.Voto)
10            FROM TRACCIA AS T
11            WHERE T.CodA=OLD.CodA and T.Voto > 0)
12 WHERE CodA=OLD.CodA;
13 RETURN NULL;
14 END;$TRIGGER_Ntracce_Durata_Voto_Album_DEL$ LANGUAGE PLPGSQL;
15
16 --TRIGGER 4: modifica il numero delle tracce, il tempo e il voto
  dell'album a cui apparteneva una traccia eliminata
17 CREATE OR REPLACE TRIGGER TRIGGER_Ntracce_Durata_Voto_Album_DEL
18 AFTER DELETE ON Traccia
19 FOR EACH ROW
20 EXECUTE PROCEDURE Ntracce_Durata_Voto_Album_DEL();
```

4.3.5 Ntracce_Durata_Album_INS

```
1 --PROCEDURA 5: quando viene aggiunta una traccia bisogna aggiornare
  il numero di tracce e la durata dell'album a cui appartiene
2 CREATE OR REPLACE FUNCTION Ntracce_Durata_Album_INS() RETURNS
  TRIGGER AS $TRIGGER_Ntracce_Durata_Album_INS$
3 BEGIN
4   UPDATE ALBUM
5   SET Ntracce=Ntracce + 1,
6       Durata=(SELECT sum(Durata)
7               FROM TRACCIA AS T
8               WHERE T.CodA=NEW.CodA)
9   WHERE CodA=NEW.CodA;
10  RETURN NULL;
11 END;$TRIGGER_Ntracce_Durata_Album_INS$ LANGUAGE PLPGSQL;
12
13 --TRIGGER 5: modifica numero tracce e durata all'aggiunta di una
  traccia
14 CREATE OR REPLACE TRIGGER TRIGGER_Ntracce_Durata_Album_INS
15 AFTER INSERT ON TRACCIA
16 FOR EACH ROW
17 EXECUTE PROCEDURE Ntracce_Durata_Album_INS();
```

4.3.6 Voto_Artista_UPD

```
1 --PROCEDURA 6: Dopo aver aggiornato il voto di una traccia bisogna
  aggiornare il voto degli artisti che l'hanno prodotta
```



```

2 CREATE OR REPLACE FUNCTION Voto_Artista_UPD() RETURNS TRIGGER AS
   $TRIGGER_Voto_Artista_UPD$
3 DECLARE
4   CursorArtistaVoto CURSOR for
5   select NomeArte, avg(voto) AS votoMedio
6   from Traccia as T, Produce as P
7   where T.CodT = p.CodT and voto > 0
8   group by Nomearte;
9
10  ArtistaVoto record;
11
12 BEGIN
13   open CursorArtistaVoto;
14   loop
15     fetch CursorArtistaVoto into ArtistaVoto;
16     exit when not found;
17
18     update Artista
19     set Voto = ArtistaVoto.votoMedio
20     where NomeArte = ArtistaVoto.NomeArte;
21
22   end loop;
23
24   close CursorArtistaVoto;
25
26   return null;
27 END;$TRIGGER_Voto_Artista_UPD$ LANGUAGE PLPGSQL;
28
29 --TRIGGER 6: modifica il Voto del Artista,a fronte della modifica
   del voto della traccia
30 CREATE OR REPLACE TRIGGER TRIGGER_Voto_Artista_UPD
31 AFTER UPDATE OF Voto ON TRACCIA
32 FOR EACH ROW
33 EXECUTE PROCEDURE Voto_Artista_UPD();

```

4.3.7 Voto_Artista_DEL

```

1 --PROCEDURA 7: Dopo aver eliminato il voto di una traccia bisogna
   aggiornare il voto degli artisti che l'hanno prodotta
2 CREATE OR REPLACE FUNCTION Voto_Artista_DEL() RETURNS TRIGGER AS
   $TRIGGER_Voto_Artista_DEL$
3 DECLARE
4   CursorArtistaVoto CURSOR for
5   select NomeArte, avg(voto) AS votoMedio
6   from Traccia as T, Produce as P
7   where T.CodT = p.CodT and voto > 0
8   group by Nomearte;
9
10  CursorArtistaNoProduce CURSOR for
11  select Artista.NomeArte
12  from Artista
13  where Artista.NomeArte not in (Select NomeArte From Produce);
14
15  ArtistaNoProduce record;
16  ArtistaVoto record;
17
18 BEGIN

```

```

19  open CursorArtistaVoto;
20  loop
21      fetch CursorArtistaVoto into ArtistaVoto;
22      exit when not found;
23
24      update Artista
25      set Voto = ArtistaVoto.votoMedio
26      where NomeArte = ArtistaVoto.NomeArte;
27
28  end loop;
29  close CursorArtistaVoto;
30
31  open CursorArtistaNoProduce;
32  loop
33      fetch CursorArtistaNoProduce into ArtistaNoProduce;
34      exit when not found;
35
36      update Artista
37      set Voto = 0
38      where NomeArte = ArtistaNoProduce.NomeArte;
39  end loop;
40  close CursorArtistaNoProduce;
41
42  return null;
43  END;$TRIGGER_Voto_Artista_DEL$ LANGUAGE PLPGSQL;
44
45  --TRIGGER 7: modifica il Voto del Artista,a fronte della
      eliminazione di una traccia
46  CREATE OR REPLACE TRIGGER Trigger_Voto_Artista_DEL
47  AFTER DELETE ON Produce
48  FOR EACH ROW
49  EXECUTE PROCEDURE Voto_Artista_DEL();

```

4.3.8 Playlist_Default

```

1  --PROCEDURA 8: Creazione della playlist brani preferiti quando
      viene aggiunto un nuovo utente
2  CREATE OR REPLACE FUNCTION Playlist_Default() RETURNS TRIGGER AS
      $Def_Playlist$
3  BEGIN
4      insert into Playlist(Titolo, NickName)
5      values('Brani preferiti', New.NickName);
6      RETURN NULL;
7  END; $Def_Playlist$ LANGUAGE PLPGSQL;
8
9  --TRIGGER 8: creazione playlist default
10 CREATE OR REPLACE TRIGGER Def_Playlist
11 AFTER INSERT ON UTENTE
12 FOR EACH ROW
13 EXECUTE PROCEDURE Playlist_Default();

```

4.3.9 Ntracce_durata_Playlist_INS

```

1  --PROCEDURA 9: aggiornamento del numero delle tracce e della durata
      di una playlist quando gli viene aggiunta una traccia
2  CREATE OR REPLACE FUNCTION Ntracce_durata_Playlist_INS() RETURNS
      TRIGGER AS $TRIGGER_Ntracce_durata_Playlist_INS$

```

```

3 BEGIN
4     UPDATE Playlist
5     SET NTracce=NTracce+1,
6         Durata=(SELECT SUM(Durata)
7                 FROM TRACCIA AS T, Contiene As C
8                 WHERE T.CodT=C.CodT and C.CodP = NEW.CodP)
9
10    WHERE CodP=NEW.CodP;
11    RETURN NULL;
12 END;$TRIGGER_Ntracce_durata_Playlist_INS$ LANGUAGE PLPGSQL;
13
14 --TRIGGER 9: aggiunta di una Traccia in una Playlist
15 CREATE OR REPLACE TRIGGER TRIGGER_Ntracce_durata_Playlist_INS
16 AFTER INSERT ON CONTIENE
17 FOR EACH ROW
18 EXECUTE PROCEDURE Ntracce_durata_Playlist_INS();

```

4.3.10 Ntracce_durata_Playlist_DEL

```

1 --PROCEDURA 10: aggiornamento numero tracce e durata alla
   eliminazione di una traccia che contenuta in playlist
2 CREATE OR REPLACE FUNCTION Ntracce_durata_Playlist_DEL() RETURNS
   TRIGGER AS $TRIGGER_Ntracce_durata_Playlist_DEL$
3 BEGIN
4     UPDATE Playlist
5     SET NTracce=NTracce-1,
6         Durata=(SELECT SUM(durata)
7                 FROM CONTIENE AS C, Traccia as T
8                 WHERE C.CodP=OLD.CodP and C.CodT = T.CodT)
9
10    WHERE CodP=OLD.CodP;
11    RETURN NULL;
12 END;$TRIGGER_Ntracce_durata_Playlist_DEL$ LANGUAGE PLPGSQL;
13
14 --TRIGGER 10: aggiornamento numero tracce e durata di una playlist
   dopo che stata eliminata una traccia che conteneva
15 CREATE OR REPLACE TRIGGER TRIGGER_Ntracce_durata_Playlist_DEL
16 AFTER DELETE ON CONTIENE
17 FOR EACH ROW
18 EXECUTE PROCEDURE Ntracce_durata_Playlist_DEL();

```

4.3.11 DEL_Artista_Pro

```

1 --PROCEDURA 11: quando viene eliminato un'artista vengono eliminate
   le tracce che ha prodotto
2 CREATE OR REPLACE FUNCTION DEL_Artista_Pro() RETURNS TRIGGER AS
   $TRIGGER_DEL_Artista_Pro$
3 DECLARE
4     CursorTracceNoArt CURSOR for
5     select CodT
6     from Traccia
7     where CodT not in (Select CodT from Produce) and codT <> 0;
8
9     TracceNoArt record;
10 BEGIN
11     open CursorTracceNoArt;
12     loop

```

```

13     fetch CursorTracceNoArt into TracceNoArt;
14     exit when not found;
15
16     delete from traccia where CodT = TracceNoArt.CodT;
17 end loop;
18 close CursorTracceNoArt;
19
20 RETURN NULL;
21 END;$TRIGGER_DEL_Artista_Pro$ LANGUAGE PLPGSQL;
22
23 --TRIGGER 11: eliminazione delle tracce quando un'artista viene
    eliminato
24 CREATE OR REPLACE TRIGGER TRIGGER_DEL_Artista_PRO
25 AFTER DELETE ON Produce
26 FOR EACH ROW
27 EXECUTE PROCEDURE DEL_Artista_PRO();

```

4.3.12 DEL_Artista_INC

```

1 --PROCEDURA 12: quando viene eliminato un'artista vengono eliminati
    gli album che ha inciso
2 CREATE OR REPLACE FUNCTION DEL_Artista_INC() RETURNS TRIGGER AS
    $TRIGGER_DEL_Artista_INC$
3 DECLARE
4     CursorAlbumNoArt CURSOR for
5     select CodA
6     from Album
7     where CodA not in (Select CodA from Incide) and codA <> 0;
8
9     AlbumNoArt record;
10 BEGIN
11     open CursorAlbumNoArt;
12     loop
13         fetch CursorAlbumNoArt into AlbumNoArt;
14         exit when not found;
15
16         delete from Album where CodA = AlbumNoArt.CodA;
17     end loop;
18     close CursorAlbumNoArt;
19
20     RETURN NULL;
21 END;$TRIGGER_DEL_Artista_INC$ LANGUAGE PLPGSQL;
22
23 --TRIGGER 12: eliminazione degli album incisi da un artista
    eliminato
24 CREATE OR REPLACE TRIGGER TRIGGER_DEL_Artista_INC
25 AFTER DELETE ON Incide
26 FOR EACH ROW
27 EXECUTE PROCEDURE DEL_Artista_INC();

```

4.4 Funzioni e trigger per i vincoli

Vediamo infine le funzioni e i trigger che chiameranno per i vincoli più complessi.

4.4.1 AscoltiNoPremium

```
1 --IMPLEMENTAZIONE VINCOLI PIU' COMPLESSI
2 --primo vincolo: un utente non premium non pu ascoltare tracce di
   qualit maggiore al 128
3 --CODICE ERRORE: QLNPR (qualita non premium)
4 Create or replace Function UtentePremium() returns Trigger as
   $AscoltiNoPremium$
5 DECLARE
6   Qualita1 Traccia.Qualita%TYPE;
7   IsPremium1 Utente.IsPremium%TYPE;
8 BEGIN
9   Select Qualita into Qualita1
10  From Traccia
11  Where New.CodT = Traccia.CodT;
12
13  Select IsPremium into IsPremium1
14  From Utente
15  Where New.NickName = Utente.NickName;
16
17  if(Qualita1 > 128 and not IsPremium1) then
18    raise exception using errcode = 'QLNPR';
19  end if;
20  Return null;
21
22 EXCEPTION
23   when SQLSTATE 'QLNPR' then
24     raise notice 'ERRORE: qualita della traccia non supportata dal
       tipo di account';
25     Delete From Ascolta Where (Ascolta.CodT = new.CodT and Ascolta.
       NickName = NEW.NickName);
26     RETURN NULL;
27 END;$AscoltiNoPremium$ Language plpgsql;
28
29 --Trigger per il primo vincolo
30 Create or replace Trigger AscoltiNoPremium
31 After Insert on Ascolta
32 FOR EACH ROW
33 Execute procedure UtentePremium();
```

4.4.2 VotiNoPremium

```
1 --secondo vincolo: i voti devono essere di utenti premium
2 --CODICE ERROE: VTNPR (voto no premium)
3 Create or replace Function VotoUtentePremium() returns Trigger as
   $VotiNoPremium$
4 DECLARE
5   IsPremium1 Utente.IsPremium%TYPE;
6 BEGIN
7   Select IsPremium into IsPremium1
8   From Utente
9   Where New.NickName = Utente.NickName;
```

```

10
11     if(not IsPremium1) then
12         raise exception using errcode = 'VTNPR';
13     end if;
14     Return null;
15 EXCEPTION
16     when SQLSTATE 'VTNPR' then
17         raise notice 'ERRORE: voto dato da un account non premium';
18         Delete From Vota Where (Vota.CodT = new.CodT and Vota.NickName
19             = NEW.NickName);
20         RETURN NULL;
21 END;$VotiNoPremium$ Language plpgsql;
22
23 --Trigger per il secondo vincolo
24 Create or replace Trigger VotiNoPremium
25 After Insert on Vota
26 FOR EACH ROW
27 Execute procedure VotoUtentePremium();

```

4.4.3 TracciaCover

```

1 --Terzo vincolo: se una traccia cover allora deve avere artista
2 --CODICE ERROE: ARTOU (artista originale uguale)
3 Create or replace Function TracciaCover() returns Trigger as
4 $TracciaCoverTrigger$
5 DECLARE
6     TracciaOr Traccia.CodTC%Type;
7     Cover Traccia.IsCover%Type;
8     Cursor_ArtistiO refcursor;
9     ArtistiO record;
10 BEGIN
11     SELECT Traccia.IsCover into Cover
12     FROM Traccia
13     where Traccia.CodT = new.Codt;
14
15     if(Cover) then
16         SELECT Traccia.CodTC into TracciaOr
17         FROM Produce, Traccia
18         Where Produce.CodT = New.CodT and Traccia.CodT = New.CodT;
19
20         open Cursor_ArtistiO for
21         SELECT Artista.NomeArte
22         FROM Artista
23         Where Artista.NomeArte in (Select NomeArte from Produce where
24             codt = TracciaOr);
25
26         loop
27             fetch Cursor_ArtistiO into ArtistiO;
28             exit when not found;
29
30             if(ArtistiO.NomeArte = New.NomeArte) then
31                 raise exception using errcode = 'ARTOU';
32             end if;
33         end loop;
34
35     close Cursor_ArtistiO;

```

```

34     end if;
35     Return null;
36
37 EXCEPTION
38     when SQLSTATE 'ARTOU' then
39         raise notice 'ERRORE: artista originale uguale';
40         delete from Traccia where (CodT = New.CodT);
41         RETURN NULL;
42 END;$TracciaCoverTrigger$ Language plpgsql;
43
44 --Trigger per il terzo vincolo
45 Create or replace Trigger TracciaCoverTrigger
46 After Insert on Produce
47 FOR EACH ROW
48 Execute procedure TracciaCover();

```

4.4.4 FormatoAnnoRemastered

```

1 --Quarto vincolo: il formato della remastered deve essere maggiore
  di quella originale e l'anno deve essere maggiore dell'
    originale
2 --CODICE ERRORE: FATRE (Formato anno remastered)
3 Create or replace Function FormatoAnnoRemastered() returns Trigger
  as $FormatoAnnoRemasteredTrigger$
4 DECLARE
5     Formato0 Traccia.Formato%Type;
6     Anno0 Traccia.AnnoU%Type;
7 BEGIN
8     if(New.IsRemastered) then
9         SELECT Traccia.Formato into Formato0
10        FROM Traccia
11        Where Traccia.CodT = New.CodTR;
12
13        SELECT Traccia.AnnoU into Anno0
14        FROM Traccia
15        Where Traccia.CodT = New.CodTR;
16
17        if(Formato0 = 'MP3' and new.Formato = 'MP3') then
18            raise exception using errcode = 'FATRE';
19        elseif(Formato0 = 'WAV' and (new.Formato = 'WAV' or new.Formato
20            = 'MP3')) then
21            raise exception using errcode = 'FATRE';
22        elseif (Formato0 = 'FLAC' and new.Formato <> 'FLAC') then
23            raise exception using errcode = 'FATRE';
24        end if;
25
26        if(New.AnnoU <= Anno0) then
27            raise exception using errcode = 'FATRE';
28        end if;
29    end if;
30    Return null;
31 EXCEPTION
32     when SQLSTATE 'FATRE' then
33         raise notice 'ERRORE: formato o anno non consistenti';
34         delete from Traccia where (CodT = New.CodT);
35         RETURN NULL;
36 END; $FormatoAnnoRemasteredTrigger$ Language plpgsql;

```

```

36
37 --Trigger per il quarto vincolo
38 Create or replace Trigger FormatoAnnoRemasteredTrigger
39 After Insert on Traccia
40 FOR EACH ROW
41 Execute procedure FormatoAnnoRemastered();

```

4.4.5 AlbumAnnoCover

```

1 --Quinto vincolo: L'anno della cover deve essere maggiore uguale
  dell'originale e l'album di appartenenza deve essere diverso
2 --CODICE ERRORE: AANCO (anno cover minore dell'originale)
3 Create or replace Function AlbumAnnoCover() returns Trigger as
  $AlbumAnnoCoverTrigger$
4 DECLARE
5   CodAO Traccia.CodA%Type;
6   AnnoO Traccia.AnnoU%Type;
7
8 BEGIN
9   if(New.IsCover) then
10     SELECT Traccia.CodA into CodAO
11     FROM Traccia
12     Where Traccia.CodT = New.CodTC;
13
14     SELECT Traccia.AnnoU into AnnoO
15     FROM Traccia
16     Where Traccia.CodT = New.CodTC;
17
18     if((CodAO = new.CodA and CodAO <> 0) or new.AnnoU < AnnoO) then
19       raise exception using errcode = 'AANCO';
20     end if;
21   end if;
22   Return null;
23
24 EXCEPTION
25   when SQLSTATE 'AANCO' then
26     raise notice 'ERRORE: Album uguale all originale o anno minore'
27     ;
28     delete from Traccia where (CodT = New.CodT);
29     RETURN NULL;
30 END; $AlbumAnnoCoverTrigger$ Language plpgsql;
31
32 --Trigger per il quinto vincolo
33 Create or replace Trigger AlbumAnnoCoverTrigger
34 After Insert on Traccia
35 FOR EACH ROW
36 Execute procedure AlbumAnnoCover();

```


4.4.6 ArtistiUgualiRemastered

```
1 --Sesto vincolo: una traccia remastered deve essere prodotta dagli
   stessi artisti
2 --CODICE ERRORE: ATRDO (artisti traccia remastered diversi dall'
   originale)
3 Create or replace Function ArtistiUgualiRemastered() returns
   Trigger as $ArtistiUgualiRemasteredTrigger$
4 DECLARE
5   TracciaOr Traccia.CodTC%Type;
6   Remastered Traccia.IsRemastered%Type;
7   Cursor_ArtistiO refcursor;
8   NArtO float4 := 0;
9   NArtODiversi float4 := 0;
10  ArtistiO record;
11 BEGIN
12   SELECT Traccia.IsRemastered into Remastered
13   FROM Traccia
14   where Traccia.CodT = new.CodT;
15
16   if(Remastered) then
17     SELECT Traccia.CodTR into TracciaOr
18     FROM Produce, Traccia
19     Where Produce.CodT = New.CodT and Traccia.CodT = New.CodT;
20
21     open Cursor_ArtistiO for
22     SELECT Artista.NomeArte
23     FROM Artista
24     Where Artista.NomeArte in (Select NomeArte from Produce where
25     codt = TracciaOr);
26
27     loop
28       fetch Cursor_ArtistiO into ArtistiO;
29       exit when not found;
30
31       NArtO := NArtO + 1;
32       NArtODiversi := NArtODiversi + 1;
33
34       if(ArtistiO.NomeArte = New.NomeArte) then
35         NArtODiversi := NArtODiversi - 1;
36       end if;
37     end loop;
38     close Cursor_ArtistiO;
39
40     if(NArtODiversi = NArtO) then
41       raise exception using errcode = 'ATRDO';
42     end if;
43   end if;
44   Return null;
45
46 EXCEPTION
47   when SQLSTATE 'ATRDO' then
48     raise notice 'ERRORE: artista originale diverso';
49     delete from Traccia where (CodT = New.CodT);
50     RETURN NULL;
51 END;$ArtistiUgualiRemasteredTrigger$ Language plpgsql;
```

```
52
53 --Trigger per il sesto vincolo
54 Create or replace Trigger ArtistiUgualiRemasteredTrigger
55 After Insert on Produce
56 FOR EACH ROW
57 Execute procedure ArtistiUgualiRemastered();
```

Capitolo 5

Manuale d'uso

Da [questo link](#) è possibile aprire la repository di GitHub dove è possibile vedere l'intero codice e altre informazioni.

5.1 Creazione utenti

Vediamo la creazione dei tre tipi di utenti con i relativi permessi

5.1.1 UtenteBase

```
1 --primo utente: utente base, pu solo ascoltare le tracce,  
    visitare le tracce, gli album, gli artisti, gli altri utenti e  
    le playlist.  
2 CREATE USER UtenteBase PASSWORD 'base';  
3 GRANT SELECT on Traccia to UtenteBase;  
4 GRANT SELECT on Album to UtenteBase;  
5 GRANT SELECT on Artista to UtenteBase;  
6 GRANT SELECT on Playlist to UtenteBase;  
7 GRANT SELECT on Contiene to UtenteBase;  
8 GRANT SELECT on Utente to UtenteBase;  
9 GRANT SELECT on Produce to UtenteBase;  
10 GRANT SELECT on Incide to UtenteBase;  
11 GRANT INSERT on Contiene to UtenteBase;  
12 GRANT INSERT on Playlist to UtenteBase;  
13 GRANT INSERT on Ascolta to UtenteBase;  
14 GRANT UPDATE on Playlist to UtenteBase;  
15 GRANT DELETE on Playlist to UtenteBase;  
16 GRANT UPDATE on Contiene to UtenteBase;  
17 GRANT DELETE on Contiene to UtenteBase;
```

5.1.2 UtentePremium

```
1 --secondo utente: oltre ad avere i privilegi dell'utente normale  
    pu anche votare delle tracce.  
2 CREATE USER UtentePremium PASSWORD 'premium';  
3 GRANT SELECT on Traccia to UtentePremium;
```

```

4 GRANT SELECT on Album to UtentePremium;
5 GRANT SELECT on Artista to UtentePremium;
6 GRANT SELECT on Playlist to UtentePremium;
7 GRANT SELECT on Contiene to UtentePremium;
8 GRANT SELECT on Utente to UtentePremium;
9 GRANT SELECT on Produce to UtentePremium;
10 GRANT SELECT on Incide to UtentePremium;
11 GRANT INSERT on Contiene to UtentePremium;
12 GRANT INSERT on Playlist to UtentePremium;
13 GRANT INSERT on Ascolta to UtentePremium;
14 GRANT UPDATE on Playlist to UtentePremium;
15 GRANT DELETE on Playlist to UtentePremium;
16 GRANT UPDATE on Contiene to UtentePremium;
17 GRANT DELETE on Contiene to UtentePremium;
18 GRANT INSERT on Vota to UtentePremium;
19 GRANT DELETE on Vota to UtentePremium;
20 GRANT UPDATE on Vota to UtentePremium;

```

5.1.3 UtenteAdmin

```

1 --terzo utente: utente admin, ha tutti i privilegi
2 CREATE USER UtenteAdmin PASSWORD 'admin';
3 GRANT ALL on Traccia to UtenteAdmin;
4 GRANT ALL on Album to UtenteAdmin;
5 GRANT ALL on Artista to UtenteAdmin;
6 GRANT ALL on Playlist to UtenteAdmin;
7 GRANT ALL on Contiene to UtenteAdmin;
8 GRANT ALL on Utente to UtenteAdmin;
9 GRANT ALL on Produce to UtenteAdmin;
10 GRANT ALL on Incide to UtenteAdmin;
11 GRANT ALL on Vota to UtenteAdmin;
12 GRANT ALL on Ascolta to UtenteAdmin;

```

5.2 Popolamento

In quest'ultima parte vediamo il popolamento del database

5.2.1 ARTISTI

```

1 --POPOLAMENTO
2 --ARTISTI
3 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
4 VALUES ('ColdPlay','Si divertono a creare generi, non aiutandoci a
   fare questo progetto...',DEFAULT);
5
6 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
7 VALUES ('Silvio Barra','Se vi interessa sapere di piu ce un sito di
   cui non ricordo il nome,cercate li',DEFAULT);
8
9 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
10 VALUES ('USA for Africa','stanno carichi niente da dire',DEFAULT);
11
12 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
13 VALUES ('Imagine Dragons','La leggenda narra che un giorno un
   prescelto sara in grado di determinare il loro genere, ma non
   oggi',DEFAULT);

```

```

14
15 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
16 VALUES ('Boney M.',DEFAULT,DEFAULT);
17
18 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
19 VALUES ('Squallor',DEFAULT,DEFAULT);
20
21 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
22 VALUES ('Daft Punk',DEFAULT,DEFAULT);
23
24 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
25 VALUES ('Molchat Doma','Neanche loro sanno che ci fanno qui',
        DEFAULT);
26
27 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
28 VALUES ('Paul Kalkbrenner','F',DEFAULT);
29
30 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
31 VALUES ('Yasuha',DEFAULT,DEFAULT);
32
33 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
34 VALUES ('1-800 GIRLS',DEFAULT,DEFAULT);
35
36 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
37 VALUES ('Lo stragen',DEFAULT,DEFAULT);
38
39 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
40 VALUES ('Randy Newman',DEFAULT,DEFAULT);
41
42 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
43 VALUES ('Camille',DEFAULT,DEFAULT);
44
45 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
46 VALUES ('OEL',DEFAULT,DEFAULT);
47
48 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
49 VALUES ('Cristina Davena',DEFAULT,DEFAULT);
50
51 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
52 VALUES ('Giorgio Vanni',DEFAULT,DEFAULT);
53
54 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
55 VALUES ('Maneskin','E senno chi le fa le cover...',DEFAULT);
56
57 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
58 VALUES ('Odeon Boys',DEFAULT,DEFAULT);
59
60 INSERT INTO ARTISTA(NomeArte,Descrizione,Voto)
61 VALUES ('Artisti Vari','Non mi pagano abbastanza per continuare',
        DEFAULT);

```

5.2.2 ALBUM

```

1 --ALBUM
2 INSERT INTO ALBUM (CodA, Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto
    )
3 VALUES (0,'Fittizio',2008,DEFAULT,DEFAULT,DEFAULT,DEFAULT);

```

```

4
5 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
6 VALUES('Viva la vida or Death and all his friends',2008,DEFAULT,
    DEFAULT,DEFAULT,DEFAULT);
7
8 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
9 VALUES('We Are The World',1985,DEFAULT,DEFAULT,DEFAULT,DEFAULT);
10
11 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
12 VALUES('Best of',2022,DEFAULT,DEFAULT,DEFAULT,DEFAULT);
13
14 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
15 VALUES('Curnutone',1981,DEFAULT,DEFAULT,DEFAULT,DEFAULT);
16
17 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
18 VALUES('Discovery',2001,DEFAULT,DEFAULT,DEFAULT,DEFAULT);
19
20 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
21 VALUES(' ',2018,DEFAULT,DEFAULT,DEFAULT,DEFAULT);
22
23 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
24 VALUES('Berling Calling',2013,DEFAULT,DEFAULT,DEFAULT,DEFAULT);
25
26 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
27 VALUES('Disney&Pixar',2018,DEFAULT,DEFAULT,DEFAULT,DEFAULT);
28
29 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
30 VALUES('Italia 1',2020,DEFAULT,DEFAULT,DEFAULT,DEFAULT);
31
32 INSERT INTO ALBUM (Titolo,AnnoU,Durata,Ntracce,Etichetta,Voto)
33 VALUES('Fantasy',2023,DEFAULT,DEFAULT,DEFAULT,DEFAULT);

```

5.2.3 TRACCE

```

1 --TRACCE
2 INSERT INTO TRACCIA(CodT,Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
3 VALUES(0, 'Fittizio', '00:02:29', DEFAULT, 2008, FALSE, FALSE, 'Arte',
    default, 'MP3', DEFAULT, 128, 0);
4
5 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
6 VALUES('Life in Technicolors', '00:02:29', DEFAULT, 2008, FALSE, FALSE, '
    Arte', 'https://open.spotify.com/track/7MT5mNCNNCoW6XP265UkdS?si
    =0250a49e39c7482a', 'MP3', DEFAULT, 128, 1);
7
8 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
9 VALUES('Life in Technicolors', '00:02:29', DEFAULT, 2008, FALSE, FALSE, '
    Arte', 'https://open.spotify.com/track/7MT5mNCNNCoW6XP265UkdS?si
    =0250a49e39c7482a', 'WAV', DEFAULT, 256, 1);
10
11 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
12 VALUES('Life in Technicolors', '00:02:29', DEFAULT, 2009, FALSE, TRUE, '
    Arte', 'https://open.spotify.com/track/7MT5mNCNNCoW6XP265UkdS?si
    =0250a49e39c7482a', 'FLAC', DEFAULT, 512, 1, 1);

```

```

13
14
15 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
16 VALUES('Cemeteries of London','00:03:21',DEFAULT,2008,FALSE,FALSE,'
    Sbang','https://open.spotify.com/track/7vIY0IXV5FDefLv3RZUq7Q?
    si=cf766f8b367042e2','MP3',DEFAULT,128,1);
17
18
19 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
20 VALUES('Cemeteries of London','00:03:21',DEFAULT,2008,FALSE,FALSE,'
    Sbang','https://open.spotify.com/track/7vIY0IXV5FDefLv3RZUq7Q?
    si=cf766f8b367042e2','WAV',DEFAULT,256,1);
21
22 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
23 VALUES('Cemeteries of London','00:03:21',DEFAULT,2010,FALSE,TRUE,'
    Sbang','https://open.spotify.com/track/7vIY0IXV5FDefLv3RZUq7Q?
    si=cf766f8b367042e2','FLAC',DEFAULT,512,1,4);
24
25
26 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
27 VALUES('Lost!','00:03:56',DEFAULT,2008,FALSE,FALSE,'Gang','https://
    open.spotify.com/track/1PdWRsAEovbR8JlpjqjRxm?si=8
    caf2d8b359d46d8','MP3',DEFAULT,128,1);
28
29
30 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
31 VALUES('Lost!','00:03:56',DEFAULT,2008,FALSE,FALSE,'Gang','https://
    open.spotify.com/track/1PdWRsAEovbR8JlpjqjRxm?si=8
    caf2d8b359d46d8','WAV',DEFAULT,256,1);
32
33
34 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
35 VALUES('Lost!','00:03:56',DEFAULT,2009,FALSE,TRUE,'Gang','https://
    open.spotify.com/track/1PdWRsAEovbR8JlpjqjRxm?si=8
    caf2d8b359d46d8','FLAC',DEFAULT,512,1,7);
36
37
38
39 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
40 VALUES('42','00:03:56',DEFAULT,2008,FALSE,FALSE,'Punk','https://
    open.spotify.com/track/3IL3zHHAFzqu9JkGzMVqsE?si=
    aa0849e1281944d4','MP3',DEFAULT,128,1);
41
42 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
43 VALUES('42','00:03:56',DEFAULT,2008,FALSE,FALSE,'Punk','https://
    open.spotify.com/track/3IL3zHHAFzqu9JkGzMVqsE?si=
    aa0849e1281944d4','WAV',DEFAULT,256,1);
44
45 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,

```

```

IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
46 VALUES('42','00:03:56',DEFAULT,2012,FALSE,TRUE,'Punk','https://open
.spotify.com/track/3IL3zHHAFzqu9JkGzMVqsE?si=aa0849e1281944d4',
'FLAC',DEFAULT,512,1,10);
47
48
49
50 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
51 VALUES('Lovers In Japan','00:06:51',DEFAULT,2008,FALSE,FALSE,'Dance
','https://open.spotify.com/track/2GkCxLKP04KfDpUYLTmTN1?si=2
c683b312db1479f','MP3',DEFAULT,128,1);
52
53 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
54 VALUES('Lovers In Japan','00:06:51',DEFAULT,2008,FALSE,FALSE,'Dance
','https://open.spotify.com/track/2GkCxLKP04KfDpUYLTmTN1?si=2
c683b312db1479f','WAV',DEFAULT,256,1);
55
56 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
57 VALUES('Lovers In Japan','00:06:51',DEFAULT,2012,FALSE,TRUE,'Dance
','https://open.spotify.com/track/2GkCxLKP04KfDpUYLTmTN1?si=2
c683b312db1479f','FLAC',DEFAULT,512,1,13);
58
59
60 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
61 VALUES('Yes','00:07:06',DEFAULT,2008,FALSE,FALSE,'Pimp','https://
open.spotify.com/track/6fw0Ih9M1GQBmi9yqAt1dP?si=
d414b661342240ce','MP3',DEFAULT,128,1);
62
63 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
64 VALUES('Yes','00:07:06',DEFAULT,2008,FALSE,FALSE,'Pimp','https://
open.spotify.com/track/6fw0Ih9M1GQBmi9yqAt1dP?si=
d414b661342240ce','WAV',DEFAULT,256,1);
65
66 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
67 VALUES('Yes','00:07:06',DEFAULT,2009,FALSE,TRUE,'Pimp','https://
open.spotify.com/track/6fw0Ih9M1GQBmi9yqAt1dP?si=
d414b661342240ce','FLAC',DEFAULT,512,1,16);
68
69
70
71 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
72 VALUES('Viva La Vida','00:04:02',DEFAULT,2008,FALSE,FALSE,'Bamb','
https://open.spotify.com/track/3Fcfwhm8oRrBvBZ8KGhtea?si=8
f157cfb46b2437c','MP3',DEFAULT,128,1);
73
74
75 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
76 VALUES('Viva La Vida','00:04:02',DEFAULT,2008,FALSE,FALSE,'Bamb','
https://open.spotify.com/track/3Fcfwhm8oRrBvBZ8KGhtea?si=8

```



```

f157cfb46b2437c', 'WAV', DEFAULT, 256, 1);
77
78
79 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA, CodTR)
80 VALUES('Viva La Vida', '00:04:02', DEFAULT, 2012, FALSE, TRUE, 'Bamb',
    https://open.spotify.com/track/3Fcfwhm8oRrBvBZ8KGhtea?si=8
    f157cfb46b2437c', 'FLAC', DEFAULT, 512, 1, 19);
81
82
83 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
84 VALUES('Violet Hill', '00:03:42', DEFAULT, 2008, FALSE, FALSE, 'Bimb',
    https://open.spotify.com/track/5147DzKnamWyQdZwsYLHEJ?si=
    e60ef222e1e54eb5', 'MP3', DEFAULT, 128, 1);
85
86 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
87 VALUES('Violet Hill', '00:03:42', DEFAULT, 2008, FALSE, FALSE, 'Bimb',
    https://open.spotify.com/track/5147DzKnamWyQdZwsYLHEJ?si=
    e60ef222e1e54eb5', 'WAV', DEFAULT, 256, 1);
88
89 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA, CodTR)
90 VALUES('Violet Hill', '00:03:42', DEFAULT, 2009, FALSE, TRUE, 'Bimb',
    https://open.spotify.com/track/5147DzKnamWyQdZwsYLHEJ?si=
    e60ef222e1e54eb5', 'FLAC', DEFAULT, 512, 1, 22);
91
92 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
93 VALUES('Strawberry Swing', '00:04:09', DEFAULT, 2008, FALSE, FALSE, 'Bumb
    ', 'https://open.spotify.com/track/4NmcfahJGawtwaMATGgPOL?si=8
    d98ed1ba6a24ab9', 'MP3', DEFAULT, 128, 1);
94
95 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
96 VALUES('Strawberry Swing', '00:04:09', DEFAULT, 2008, FALSE, FALSE, 'Bumb
    ', 'https://open.spotify.com/track/4NmcfahJGawtwaMATGgPOL?si=8
    d98ed1ba6a24ab9', 'WAV', DEFAULT, 256, 1);
97
98 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA, CodTR)
99 VALUES('Strawberry Swing', '00:04:09', DEFAULT, 2012, FALSE, TRUE, 'Bumb'
    , 'https://open.spotify.com/track/4NmcfahJGawtwaMATGgPOL?si=8
    d98ed1ba6a24ab9', 'FLAC', DEFAULT, 512, 1, 25);
100
101
102 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
103 VALUES('Death and Hills', '00:06:18', DEFAULT, 2008, FALSE, FALSE, 'Bomb'
    , 'https://open.spotify.com/track/36kG5UXshT0PqnZKsgU8c2?si
    =630215581c6c4476', 'MP3', DEFAULT, 128, 1);
104
105 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
106 VALUES('Death and Hills', '00:06:18', DEFAULT, 2008, FALSE, FALSE, 'Bomb'
    , 'https://open.spotify.com/track/36kG5UXshT0PqnZKsgU8c2?si

```

```

=630215581c6c4476', 'WAV', DEFAULT, 256, 1);
107
108 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA, CodTR)
109 VALUES('Death and Hills', '00:06:18', DEFAULT, 2009, FALSE, TRUE, 'Bomb',
    'https://open.spotify.com/track/36kG5UXshTOPqnZKsgU8c2?si
    =630215581c6c4476', 'FLAC', DEFAULT, 512, 1, 28);
110
111
112 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
113 VALUES('We Are the World', '00:07:07', DEFAULT, 1985, FALSE, FALSE, 'Pop',
    'https://open.spotify.com/track/3Z2tPWniIpg8UMMoowHIk?si=
    a35cf24db54f4fad', 'MP3', DEFAULT, 128, 2);
114
115 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
116 VALUES('Gotta go home', '00:03:45', DEFAULT, 1979, FALSE, FALSE, 'Funk', '
    https://open.spotify.com/track/4MvGHDenL4t9JW1RHB4rK2?si=3412
    cf0a657e4ae2', 'MP3', DEFAULT, 128, 3);
117
118 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
119 VALUES('Gotta go home', '00:03:45', DEFAULT, 1979, FALSE, FALSE, 'Funk', '
    https://open.spotify.com/track/4MvGHDenL4t9JW1RHB4rK2?si=3412
    cf0a657e4ae2', 'WAV', DEFAULT, 256, 3);
120
121 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA, CodTR)
122 VALUES('Gotta go home', '00:03:45', DEFAULT, 1981, FALSE, TRUE, 'Funk', '
    https://open.spotify.com/track/4MvGHDenL4t9JW1RHB4rK2?si=3412
    cf0a657e4ae2', 'FLAC', DEFAULT, 512, 3, 32);
123
124
125 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
126 VALUES('Ma baker', '00:04:36', DEFAULT, 1977, FALSE, FALSE, 'Funk', 'https
    ://open.spotify.com/track/1BqnZ0kYJbvYLOhN0qPJDm?si=714357878
    d6f4d77', 'MP3', DEFAULT, 128, 3);
127
128 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
129 VALUES('Ma baker', '00:04:36', DEFAULT, 1977, FALSE, FALSE, 'Funk', 'https
    ://open.spotify.com/track/1BqnZ0kYJbvYLOhN0qPJDm?si=714357878
    d6f4d77', 'WAV', DEFAULT, 256, 3);
130
131 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA, CodTR)
132 VALUES('Ma baker', '00:04:36', DEFAULT, 1979, FALSE, TRUE, 'Funk', 'https
    ://open.spotify.com/track/1BqnZ0kYJbvYLOhN0qPJDm?si=714357878
    d6f4d77', 'FLAC', DEFAULT, 512, 3, 35);
133
134
135
136 INSERT INTO TRACCIA(Titolo, Durata, Etichetta, AnnoU, IsCover,
    IsRemastered, Genere, Link, Formato, Voto, Qualita, CodA)
137 VALUES('Daddy Cool', '00:03:28', DEFAULT, 1976, FALSE, FALSE, 'Funk', '

```

```

https://open.spotify.com/track/3WMbD10yfKuwWDWMNbPQ4g?si=
c71ff2b628dc4cf7','MP3',DEFAULT,128,3);
138
139 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
140 VALUES('Daddy Cool','00:03:28',DEFAULT,1976,FALSE,FALSE,'Funk',
https://open.spotify.com/track/3WMbD10yfKuwWDWMNbPQ4g?si=
c71ff2b628dc4cf7','WAV',DEFAULT,256,3);
141
142 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
143 VALUES('Daddy Cool','00:03:28',DEFAULT,1979,FALSE,TRUE,'Funk',
https://open.spotify.com/track/3WMbD10yfKuwWDWMNbPQ4g?si=
c71ff2b628dc4cf7','FLAC',DEFAULT,512,3,38);
144
145
146 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
147 VALUES('Curnutone','00:04:12',DEFAULT,1981,FALSE,FALSE,'Folk-Nap',
https://open.spotify.com/track/5FK9QdVXh9kcxsy6bMABU?si=847
cf1ef6efb4f20','MP3',DEFAULT,128,4);
148
149 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
150 VALUES('O teimp s ne va','00:03:45',DEFAULT,1981,FALSE,FALSE,'Folk-
Nap',https://open.spotify.com/track/5YPXVzvnHSsVQptz0zLiE7?si
=2aaecb06bde1415f','MP3',DEFAULT,128,4);
151
152 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
153 VALUES('O riccutar Nnamurato','00:03:04',DEFAULT,1981,FALSE,FALSE,
Folk-Nap',https://open.spotify.com/track/6
AZELCmg1RwMrmf0EkqyFj?si=08c21217a8864a25','MP3',DEFAULT,128,4)
;
154
155 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
156 VALUES('Something About Us','00:03:52',DEFAULT,2001,FALSE,FALSE,
Elettronica',https://open.spotify.com/track/1
NeLwFETswx8Fzxl2AF191?si=49426af88d574c6b','MP3',DEFAULT,128,5)
;
157
158 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
159 VALUES('Something About Us','00:03:52',DEFAULT,2001,FALSE,FALSE,
Elettronica',https://open.spotify.com/track/1
NeLwFETswx8Fzxl2AF191?si=49426af88d574c6b','WAV',DEFAULT,256,5)
;
160
161 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
162 VALUES('Something About Us','00:03:52',DEFAULT,2003,FALSE,TRUE,
Elettronica',https://open.spotify.com/track/1
NeLwFETswx8Fzxl2AF191?si=49426af88d574c6b','FLAC',DEFAULT
,512,5,44);
163
164

```

```

165 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
166 VALUES('Giorgio by Moroder','00:09:04',DEFAULT,2001,FALSE,FALSE,'
    Elettronica','https://open.spotify.com/track/0
    oks4FnzhNp5QPTZtoet7c?si=3d577136eb7944a9','MP3',DEFAULT,128,5)
    ;
167
168 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
169 VALUES('Giorgio by Moroder','00:09:04',DEFAULT,2001,FALSE,FALSE,'
    Elettronica','https://open.spotify.com/track/0
    oks4FnzhNp5QPTZtoet7c?si=3d577136eb7944a9','WAV',DEFAULT,256,5)
    ;
170
171 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
172 VALUES('Giorgio by Moroder','00:09:04',DEFAULT,2006,FALSE,TRUE,'
    Elettronica','https://open.spotify.com/track/0
    oks4FnzhNp5QPTZtoet7c?si=3d577136eb7944a9','FLAC',DEFAULT
    ,512,5,47);
173
174
175 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
176 VALUES('Loose your self to dance','00:05:53',DEFAULT,2001,FALSE,
    FALSE,'Elettronica Funk','https://open.spotify.com/track/5
    CMjjywIOeZMixPeqNd75R?si=9c6255337e62447e','MP3',DEFAULT,128,5)
    ;
177
178 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
179 VALUES('Loose your self to dance','00:05:53',DEFAULT,2001,FALSE,
    FALSE,'Elettronica Funk','https://open.spotify.com/track/5
    CMjjywIOeZMixPeqNd75R?si=9c6255337e62447e','WAV',DEFAULT,256,5)
    ;
180
181 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
182 VALUES('Loose your self to dance','00:05:53',DEFAULT,2004,FALSE,
    TRUE,'Elettronica Funk','https://open.spotify.com/track/5
    CMjjywIOeZMixPeqNd75R?si=9c6255337e62447e','FLAC',DEFAULT
    ,512,5,50);
183
184
185 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
186 VALUES('Sudno','00:02:21',DEFAULT,2018,FALSE,FALSE,'Russian Post-
    Punk','https://open.spotify.com/track/1SHB1hp6267UK9bJQUxYv0?si
    =4ea5dfdc865b4531','MP3',DEFAULT,128,6);
187
188 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
    IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
189 VALUES('Sudno','00:02:21',DEFAULT,2019,FALSE,FALSE,'Russian Post-
    Punk','https://open.spotify.com/track/1SHB1hp6267UK9bJQUxYv0?si
    =4ea5dfdc865b4531','WAV',DEFAULT,256,6,53);
190
191 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,

```

```

IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
192 VALUES('Sudno','00:02:21',DEFAULT,2021,FALSE,TRUE,'Russian Post-
Punk','https://open.spotify.com/track/1SHB1hp6267UK9bJQUxYv0?si
=4ea5dfdc865b4531','FLAC',DEFAULT,512,6,53);
193
194
195 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
196 VALUES('Tanzevat','00:03:22',DEFAULT,2018,FALSE,FALSE,'Russian Post
-Punk','https://open.spotify.com/track/782VcXkRqyevFaJlcoIIEz?si
=5c4780b13540431f','MP3',DEFAULT,128,6);
197
198 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
199 VALUES('Tanzevat','00:03:22',DEFAULT,2020,FALSE,TRUE,'Russian Post-
Punk','https://open.spotify.com/track/782VcXkRqyevFaJlcoIIEz?si
=5c4780b13540431f','WAV',DEFAULT,256,6,56);
200
201 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTR)
202 VALUES('Tanzevat','00:03:22',DEFAULT,2022,FALSE,TRUE,'Russian Post-
Punk','https://open.spotify.com/track/782VcXkRqyevFaJlcoIIEz?si
=5c4780b13540431f','FLAC',DEFAULT,512,6,56);
203
204 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
205 VALUES('Sky and Sand','00:03:50',DEFAULT,2013,FALSE,FALSE,'Tech
House','https://open.spotify.com/track/4IsHMzDbRE8q5Z4ALsQj3o?
si=25b8b9fe5ae24350','MP3',DEFAULT,128,7);
206
207
208 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
209 VALUES('Revolte','00:02:39',DEFAULT,2013,FALSE,FALSE,'Tech House','
https://open.spotify.com/track/439Z0bTQ35Py7TMGNUJmlp?si=
d7c48a1fa15f4b9d','MP3',DEFAULT,128,7);
210
211 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
212 VALUES('Train','00:02:48',DEFAULT,2013,FALSE,FALSE,'Tech House','
https://open.spotify.com/track/0Tcm3XyhzXJOUoSVlNX7M?si=5
aa08fc0349a4f9e','MP3',DEFAULT,128,7);
213
214 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
215 VALUES('Monsters Inc','00:02:06',DEFAULT,2001,FALSE,FALSE,'Cartoon',
'https://open.spotify.com/track/5e007MjhNHq9G67qDFM8nR?si=
e9748639b7c34b45','MP3',DEFAULT,128,8);
216
217 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
218 VALUES('Le Festin','00:02:50',DEFAULT,2007,FALSE,FALSE,'Cartoon','
https://open.spotify.com/track/02JIdsrod3BYucThfUFDUX?si=6
b4c171e94754988','MP3',DEFAULT,128,8);
219
220 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)

```

```

221 VALUES('I cavalieri dello Zodiaco','00:02:45',DEFAULT,2009,FALSE,
FALSE,'Cartoon','https://open.spotify.com/track/05
A9bpJIvKqts4uswSjNla?si=09964b1ed424438c','MP3',DEFAULT,128,9);
222
223 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
224 VALUES('Siamo fatti cosi','00:02:50',DEFAULT,2019,FALSE,FALSE,'
Cartoon','https://open.spotify.com/track/5pVqiyo6FbpGP20w06DPip
?si=93fefb2c0abf4e85','MP3',DEFAULT,128,9);
225
226 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
227 VALUES('Pokemon','00:03:20',DEFAULT,2014,FALSE,FALSE,'Cartoon','
https://open.spotify.com/track/3S7Q50qgcPyEXQgyrHbvze?si=
da1b827e8fba459a','MP3',DEFAULT,128,9);
228
229 -- Popolati gli album, creiamo alcune tracce senza album (ed alcune
Cover)
230
231 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
232 VALUES('Easter egg','00:01:42',DEFAULT,2021,FALSE,FALSE,'Arte','
https://open.spotify.com/track/1dwTG4PVhiWzeuOfUfMMb?si=8
e6406c15b2d42d0','FLAC',DEFAULT,512,DEFAULT);
233
234 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
235 VALUES('Easter egg2','00:03:51',DEFAULT,2021,FALSE,FALSE,'Sport','
https://www.youtube.com/watch?v=aR5BsvCA21s','FLAC',DEFAULT
,512,DEFAULT);
236
237 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
238 VALUES('Napoli','00:03:20',DEFAULT,1998,FALSE,FALSE,'Sport','https
://www.youtube.com/watch?v=aR5BsvCA21s','FLAC',DEFAULT,512,
DEFAULT);
239
240 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
241 VALUES('Procida','00:04:34',DEFAULT,2019,FALSE,FALSE,'Blues','https
://www.youtube.com/watch?v=4CmXDBm6gZA','FLAC',DEFAULT,512,
DEFAULT);
242
243 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
244 VALUES('Demons','00:02:57',DEFAULT,2012,FALSE,FALSE,'BOH','https://
open.spotify.com/track/5qaEfEh1AtSdrdrByCP7qR?si=
a4e9be06da044402','MP3',DEFAULT,128,DEFAULT);
245
246 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
247 VALUES('Flyday','00:03:25',DEFAULT,2021,FALSE,FALSE,'Jap80','https
://open.spotify.com/track/5N61cz1UZJVax2gFjREz5c?si=2466
b2f95a834b13','MP3',DEFAULT,128,DEFAULT);
248
249 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)

```

```

250 VALUES('U me and madonna','00:04:10',DEFAULT,2017,FALSE,FALSE,'
      House','https://open.spotify.com/track/1tpTQDEv7I3rpTOW816meN?
      si=4e34ff5cd0d24f01','MP3',DEFAULT,128,DEFAULT);
251
252 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
      IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
253 VALUES('Ecole','00:06:24',DEFAULT,2020,FALSE,FALSE,'House','https
      ://open.spotify.com/track/6ZcN26OG9FY2inZxJwTX6g?si=3
      d1473829c5941e0','MP3',DEFAULT,256,DEFAULT);
254
255 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
      IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
256 VALUES('Focacine','00:02:20',DEFAULT,2016,FALSE,FALSE,'Trap','https
      ://open.spotify.com/track/7rYlGT75z1srGu5i8wuTwF?si=
      f230ed80f3ae442b','MP3',DEFAULT,128,DEFAULT);
257
258 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
      IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
259 VALUES('Dan Dan','00:03:30',DEFAULT,1995,FALSE,FALSE,'Cartoon','
      https://open.spotify.com/track/6vjNTnRMJuoktlhC7PFAJY?si=
      c6bbe8a6978446c0','MP3',DEFAULT,128,DEFAULT);
260
261 -- Passiamo alle tracce dei Maneskin...ops le cover--
262 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
      IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTC)
263 VALUES('Tanzavat','00:03:22',DEFAULT,2023,TRUE,FALSE,'American
      After-Punk','https://open.spotify.com/track/782
      VcXkRqyevFaJlcoIIez?si=5c4780b13540431f','MP3',DEFAULT,128,
      DEFAULT,56);
264
265 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
      IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTC)
266 VALUES('Loose your self to dance','00:05:53',DEFAULT,2023,TRUE,
      FALSE,'Neomelodico','https://open.spotify.com/track/5
      CMjjywIOeZMixPeqNd75R?si=9c6255337e62447e','WAV',DEFAULT,256,
      DEFAULT,50);
267
268 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
      IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTC)
269 VALUES('Curnutone','00:04:12',DEFAULT,2023,TRUE,FALSE,'Meeelano','
      https://open.spotify.com/track/5FK9QdVXh9kcxsy6bMABU?si=847
      cf1ef6efb4f20','MP3',DEFAULT,128,DEFAULT,41);
270
271 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
      IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA,CodTC)
272 VALUES('Viva La Vida','00:04:02',DEFAULT,2023,TRUE,FALSE,'Classica'
      , 'https://open.spotify.com/track/3Fcfxwm8oRrBvBZ8KGhtea?si=8
      f157cfb46b2437c','FLAC',DEFAULT,512,DEFAULT,19);
273
274 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
      IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)
275 VALUES('Begin','00:03:31',DEFAULT,2020,FALSE,FALSE,'Rock','https://
      open.spotify.com/track/3Wrjm47oTz2sjIgck11l5e?si=
      a69bd451b8df459f','FLAC',DEFAULT,512,10);
276
277 INSERT INTO TRACCIA(Titolo,Durata,Etichetta,AnnoU,IsCover,
      IsRemastered,Genere,Link,Formato,Voto,Qualita,CodA)

```

```

278 VALUES('Holly e benji','00:03:04',DEFAULT,2020,FALSE,FALSE,'Cartoon
', 'https://open.spotify.com/track/3Wrjm47oTz2sjlgck11l5e?si=
a69bd451b8df459f','WAV',DEFAULT,512,DEFAULT);

```

5.2.4 UTENTI

```

1 -- UTENTI
2 INSERT INTO UTENTE(Nome,Cognome,Nickname,Email>Password,DataN,Sesso
,Nazionalita,Descrizione,IsPremium,IsAdmin)
3 Values('Rami','MaleK','Rami_Malek','silvio.barra@unina.it','
I@MrR0bot','1991-01-01','Uomo',DEFAULT,'Google is the way',
TRUE,TRUE);
4
5 INSERT INTO UTENTE(Nome,Cognome,Nickname,Email>Password,DataN,Sesso
,Nazionalita,Descrizione,IsPremium,IsAdmin)
6 Values('Porfirio','Tramontana','King_of_Procida','porfirio.
tramontata@unina.it','ForzaNapoli','1990-01-01','Uomo',DEFAULT,
'Facciamo pausa dai',TRUE,TRUE);
7
8 INSERT INTO UTENTE(Nome,Cognome,Nickname,Email>Password,DataN,Sesso
,Nazionalita,Descrizione,IsPremium,IsAdmin)
9 Values('Alfredo','Laino','Lord_Pino','al.laino@studenti.unina.it'
,'1q2w3e4r5t','1995-08-21','Lampadina',DEFAULT,'Che bello
popolare le basi di dati...Quanto mi rilassa...',TRUE,FALSE);
10
11 INSERT INTO UTENTE(Nome,Cognome,Nickname,Email>Password,DataN,Sesso
,Nazionalita,Descrizione,IsPremium,IsAdmin)
12 Values('Francesco','Orlando','Effeo','francesco.orlando3@studenti
.unina.it','y6t5r4e3w2q1','2002-01-20','Unicorno',DEFAULT,'Sono
il terzo Francesco Orlando di UNINA,tutto apposto...',TRUE,
FALSE);
13
14 INSERT INTO UTENTE(Nome,Cognome,Nickname,Email>Password,DataN,Sesso
,Nazionalita,Descrizione,IsPremium,IsAdmin)
15 Values('Marco','Pastore','ElPator','loso@ahokok.com','Miscusi','
2002-01-01','Donna',DEFAULT,'Mi scusi Posso fare una Domanda',
FALSE,FALSE); --Cute--

```

5.2.5 INCISIONI

```

1 --INCISIONI
2 INSERT INTO INCIDE (NomeArte,CodA)
3 Values('ColdPlay',1);
4
5 INSERT INTO INCIDE (NomeArte,CodA)
6 Values('USA for Africa',2);
7
8 INSERT INTO INCIDE (NomeArte,CodA)
9 Values('Boney M.',3);
10
11 INSERT INTO INCIDE (NomeArte,CodA)
12 Values('Squallor',4);
13
14 INSERT INTO INCIDE (NomeArte,CodA)
15 Values('Daft Punk',5);
16
17 INSERT INTO INCIDE (NomeArte,CodA)

```



```

18 Values('Molchat Doma',6);
19
20 INSERT INTO INCIDE (NomeArte,CodA)
21 Values('Paul Kalkbrenner',7);
22
23 INSERT INTO INCIDE (NomeArte,CodA)
24 Values('Randy Newman',8);
25
26 INSERT INTO INCIDE (NomeArte,CodA)
27 Values('Camille',8);
28
29 INSERT INTO INCIDE (NomeArte,CodA)
30 Values('Cristina Davena',9);
31
32 INSERT INTO INCIDE (NomeArte,CodA)
33 Values('Giorgio Vanni',9);
34
35 INSERT INTO INCIDE (NomeArte,CodA)
36 Values('Odeon Boys',9);
37
38 INSERT INTO INCIDE (NomeArte,CodA)
39 Values('Maneskin',10);

```

5.2.6 PRODUZIONI

```

1 --PRODUZIONI
2 INSERT INTO PRODUCE(NomeArte,CodT)
3 VALUES('ColdPlay',1);
4
5 INSERT INTO PRODUCE(NomeArte,CodT)
6 VALUES('ColdPlay',2);
7
8 INSERT INTO PRODUCE(NomeArte,CodT)
9 VALUES('ColdPlay',3);
10
11 INSERT INTO PRODUCE(NomeArte,CodT)
12 VALUES('ColdPlay',4);
13
14 INSERT INTO PRODUCE(NomeArte,CodT)
15 VALUES('ColdPlay',5);
16
17 INSERT INTO PRODUCE(NomeArte,CodT)
18 VALUES('ColdPlay',6);
19
20 INSERT INTO PRODUCE(NomeArte,CodT)
21 VALUES('ColdPlay',7);
22
23 INSERT INTO PRODUCE(NomeArte,CodT)
24 VALUES('ColdPlay',8);
25
26 INSERT INTO PRODUCE(NomeArte,CodT)
27 VALUES('ColdPlay',9);
28
29 INSERT INTO PRODUCE(NomeArte,CodT)
30 VALUES('ColdPlay',10);
31
32 INSERT INTO PRODUCE(NomeArte,CodT)

```

```

33 VALUES ('ColdPlay',11);
34
35 INSERT INTO PRODUCE(NomeArte,CodT)
36 VALUES ('ColdPlay',12);
37
38 INSERT INTO PRODUCE(NomeArte,CodT)
39 VALUES ('ColdPlay',13);
40
41 INSERT INTO PRODUCE(NomeArte,CodT)
42 VALUES ('ColdPlay',14);
43
44 INSERT INTO PRODUCE(NomeArte,CodT)
45 VALUES ('ColdPlay',15);
46
47 INSERT INTO PRODUCE(NomeArte,CodT)
48 VALUES ('ColdPlay',16);
49
50 INSERT INTO PRODUCE(NomeArte,CodT)
51 VALUES ('ColdPlay',17);
52
53 INSERT INTO PRODUCE(NomeArte,CodT)
54 VALUES ('ColdPlay',18);
55
56 INSERT INTO PRODUCE(NomeArte,CodT)
57 VALUES ('ColdPlay',19);
58
59 INSERT INTO PRODUCE(NomeArte,CodT)
60 VALUES ('ColdPlay',20);
61
62 INSERT INTO PRODUCE(NomeArte,CodT)
63 VALUES ('ColdPlay',21);
64
65 INSERT INTO PRODUCE(NomeArte,CodT)
66 VALUES ('ColdPlay',22);
67
68 INSERT INTO PRODUCE(NomeArte,CodT)
69 VALUES ('ColdPlay',23);
70
71 INSERT INTO PRODUCE(NomeArte,CodT)
72 VALUES ('ColdPlay',24);
73
74 INSERT INTO PRODUCE(NomeArte,CodT)
75 VALUES ('ColdPlay',25);
76
77 INSERT INTO PRODUCE(NomeArte,CodT)
78 VALUES ('ColdPlay',26);
79
80 INSERT INTO PRODUCE(NomeArte,CodT)
81 VALUES ('ColdPlay',27);
82
83 INSERT INTO PRODUCE(NomeArte,CodT)
84 VALUES ('ColdPlay',28);
85
86 INSERT INTO PRODUCE(NomeArte,CodT)
87 VALUES ('ColdPlay',29);
88
89 INSERT INTO PRODUCE(NomeArte,CodT)

```

```

90 VALUES ('ColdPlay',30);
91
92 INSERT INTO PRODUCE(NomeArte,CodT)
93 VALUES ('USA for Africa',31);
94
95 INSERT INTO PRODUCE(NomeArte,CodT)
96 VALUES ('Boney M.',32);
97
98 INSERT INTO PRODUCE(NomeArte,CodT)
99 VALUES ('Boney M.',33);
100
101 INSERT INTO PRODUCE(NomeArte,CodT)
102 VALUES ('Boney M.',34);
103
104 INSERT INTO PRODUCE(NomeArte,CodT)
105 VALUES ('Boney M.',35);
106
107 INSERT INTO PRODUCE(NomeArte,CodT)
108 VALUES ('Boney M.',36);
109
110 INSERT INTO PRODUCE(NomeArte,CodT)
111 VALUES ('Boney M.',37);
112
113 INSERT INTO PRODUCE(NomeArte,CodT)
114 VALUES ('Boney M.',38);
115
116 INSERT INTO PRODUCE(NomeArte,CodT)
117 VALUES ('Boney M.',39);
118
119 INSERT INTO PRODUCE(NomeArte,CodT)
120 VALUES ('Boney M.',40);
121
122 INSERT INTO PRODUCE(NomeArte,CodT)
123 VALUES ('Squallor',41);
124
125 INSERT INTO PRODUCE(NomeArte,CodT)
126 VALUES ('Squallor',42);
127
128 INSERT INTO PRODUCE(NomeArte,CodT)
129 VALUES ('Squallor',43);
130
131 INSERT INTO PRODUCE(NomeArte,CodT)
132 VALUES ('Daft Punk',44);
133
134 INSERT INTO PRODUCE(NomeArte,CodT)
135 VALUES ('Daft Punk',45);
136
137 INSERT INTO PRODUCE(NomeArte,CodT)
138 VALUES ('Daft Punk',46);
139
140 INSERT INTO PRODUCE(NomeArte,CodT)
141 VALUES ('Daft Punk',47);
142
143 INSERT INTO PRODUCE(NomeArte,CodT)
144 VALUES ('Daft Punk',48);
145
146 INSERT INTO PRODUCE(NomeArte,CodT)

```

```

147 VALUES ('Daft Punk',49);
148
149 INSERT INTO PRODUCE(NomeArte,CodT)
150 VALUES ('Daft Punk',50);
151
152 INSERT INTO PRODUCE(NomeArte,CodT)
153 VALUES ('Daft Punk',51);
154
155 INSERT INTO PRODUCE(NomeArte,CodT)
156 VALUES ('Daft Punk',52);
157
158 INSERT INTO PRODUCE(NomeArte,CodT)
159 VALUES ('Molchat Doma',53);
160
161 INSERT INTO PRODUCE(NomeArte,CodT)
162 VALUES ('Molchat Doma',54);
163
164 INSERT INTO PRODUCE(NomeArte,CodT)
165 VALUES ('Molchat Doma',55);
166
167 INSERT INTO PRODUCE(NomeArte,CodT)
168 VALUES ('Molchat Doma',56);
169
170 INSERT INTO PRODUCE(NomeArte,CodT)
171 VALUES ('Molchat Doma',57);
172
173 INSERT INTO PRODUCE(NomeArte,CodT)
174 VALUES ('Molchat Doma',58);
175
176 INSERT INTO PRODUCE(NomeArte,CodT)
177 VALUES ('Paul Kalkbrenner',59);
178
179 INSERT INTO PRODUCE(NomeArte,CodT)
180 VALUES ('Paul Kalkbrenner',60);
181
182 INSERT INTO PRODUCE(NomeArte,CodT)
183 VALUES ('Paul Kalkbrenner',61);
184
185 INSERT INTO PRODUCE(NomeArte,CodT)
186 VALUES ('Randy Newman',62);
187
188 INSERT INTO PRODUCE(NomeArte,CodT)
189 VALUES ('Camille',63);
190
191 INSERT INTO PRODUCE(NomeArte,CodT)
192 VALUES ('Odeon Boys',64);
193
194 INSERT INTO PRODUCE(NomeArte,CodT)
195 VALUES ('Cristina Davena',65);
196
197 INSERT INTO PRODUCE(NomeArte,CodT)
198 VALUES ('Giorgio Vanni',66);
199
200 INSERT INTO PRODUCE(NomeArte,CodT)
201 VALUES ('Silvio Barra',67);
202
203 INSERT INTO PRODUCE(NomeArte,CodT)

```

```

204 VALUES('Silvio Barra',68);
205
206 INSERT INTO PRODUCE(NomeArte,CodT)
207 VALUES('Artisti Vari',69); --Nice --
208
209 INSERT INTO PRODUCE(NomeArte,CodT)
210 VALUES('Artisti Vari',70);
211
212 INSERT INTO PRODUCE(NomeArte,CodT)
213 VALUES('Imagine Dragons',71);
214
215 INSERT INTO PRODUCE(NomeArte,CodT)
216 VALUES('Yasuha',72);
217
218 INSERT INTO PRODUCE(NomeArte,CodT)
219 VALUES('1-800 GIRLS',73);
220
221 INSERT INTO PRODUCE(NomeArte,CodT)
222 VALUES('Lo straqen',74);
223
224 INSERT INTO PRODUCE(NomeArte,CodT)
225 VALUES('OEL',75);
226
227 INSERT INTO PRODUCE(NomeArte,CodT)
228 VALUES('Artisti Vari',76);
229
230 INSERT INTO PRODUCE(NomeArte,CodT)
231 VALUES('Maneskin',77);
232
233 INSERT INTO PRODUCE(NomeArte,CodT)
234 VALUES('Maneskin',78);
235
236 INSERT INTO PRODUCE(NomeArte,CodT)
237 VALUES('Maneskin',79);
238
239 INSERT INTO PRODUCE(NomeArte,CodT)
240 VALUES('Maneskin',80);
241
242 INSERT INTO PRODUCE(NomeArte,CodT)
243 VALUES('Maneskin',81);
244
245 INSERT INTO PRODUCE(NomeArte,CodT)
246 VALUES('Cristina Davena',82);
247
248 INSERT INTO PRODUCE(NomeArte,CodT)
249 VALUES('Giorgio Vanni',82);

```

5.2.7 ASCOLTI

```

1 --ASCOLTI
2 insert into Ascolta
3 values ('Rami_Malek', 1, 1);
4 insert into Ascolta
5 values ('Rami_Malek', 4, 2);
6 insert into Ascolta
7 values ('Rami_Malek', 3, 1);
8 insert into Ascolta

```

```

9 values ('Rami_Malek', 6, 4);
10 insert into Ascolta
11 values ('Rami_Malek', 5, 6);
12 insert into Ascolta
13 values ('Rami_Malek', 77, 1);
14 insert into Ascolta
15 values ('Rami_Malek', 66, 5);
16 insert into Ascolta
17 values ('Lord_Pino', 14, 1);
18 insert into Ascolta
19 values ('Lord_Pino', 44, 2);
20 insert into Ascolta
21 values ('Lord_Pino', 32, 1);
22 insert into Ascolta
23 values ('Lord_Pino', 64, 4);
24 insert into Ascolta
25 values ('Lord_Pino', 55, 6);
26 insert into Ascolta
27 values ('Lord_Pino', 76, 1);
28 insert into Ascolta
29 values ('Lord_Pino', 63, 5);

```

5.2.8 VOTI

```

1 --VOTI
2 insert into Vota
3 values ('Rami_Malek', 1, 5);
4 insert into Vota
5 values ('Rami_Malek', 4, 6);
6 insert into Vota
7 values ('Rami_Malek', 3, 7);
8 insert into Vota
9 values ('Rami_Malek', 6, 8);
10 insert into Vota
11 values ('Rami_Malek', 5, 9);
12 insert into Vota
13 values ('Rami_Malek', 77, 10);
14 insert into Vota
15 values ('Rami_Malek', 66, 2);
16 insert into Vota
17 values ('Lord_Pino', 1, 4);
18 insert into Vota
19 values ('Lord_Pino', 4, 3);
20 insert into Vota
21 values ('Lord_Pino', 3, 2);
22 insert into Vota
23 values ('Lord_Pino', 6, 5);
24 insert into Vota
25 values ('Lord_Pino', 5, 6);
26 insert into Vota
27 values ('Lord_Pino', 77, 5);
28 insert into Vota
29 values ('Lord_Pino', 66, 7);

```

5.2.9 CONTIENE

```
1  --CONTIENE
2  insert into Contiene
3  values (1, 1);
4  insert into Contiene
5  values (1, 4);
6  insert into Contiene
7  values ('1', 3);
8  insert into Contiene
9  values (1, 6);
10 insert into Contiene
11 values (1, 5);
12 insert into Contiene
13 values (1, 77);
14 insert into Contiene
15 values (1, 66);
16 insert into Contiene
17 values (2, 11);
18 insert into Contiene
19 values (2, 42);
20 insert into Contiene
21 values ('2', 33);
22 insert into Contiene
23 values (2, 64);
24 insert into Contiene
25 values (2, 65);
26 insert into Contiene
27 values (2, 80);
28 insert into Contiene
29 values (2, 61);
```