



UNIVERSITÀ DEGLI STUDI  
DI NAPOLI FEDERICO II

## **Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione**

Corso di Laurea in Informatica

### ***PROGETTO DI BASI DI DATI:***

Progettazione e sviluppo di un sistema informativo per la gestione  
delle librerie musicali di un gruppo di utenti.

Anno Accademico 2021/2022

### **Studenti**

TORCICOLLO ADOLFO      N86003538

PUGLIESE FRANCESCA      N86003507

# **INDICE**

## **1. Descrizione del progetto**

- ◆ Descrizione Sintetica

## **2. Progettazione concettuale**

- ◆ Class Diagram
- ◆ Ristrutturazione del Class Diagram
  - I. Analisi delle ridondanze
  - II. Rimozione degli attributi multipli
  - III. Rimozione delle gerarchie di specializzazione
  - IV. Rimozione delle molteplicità delle associazioni
- ◆ Class Diagram ristrutturato
- ◆ Dizionario delle classi
- ◆ Dizionario delle associazioni
- ◆ Dizionario dei vincoli

## **3. Progettazione logica**

- ◆ Schema logico

## **4. Definizioni SQL**

- ◆ Definizione delle tabelle
  - I. Tabella ALBUM
  - II. Tabella TRACCIA
  - III. Tabella UTENTE
  - IV. Tabella ORARIO
  - V. Tabella FOLLOWING
- ◆ Procedure e funzioni
- ◆ Trigger

## **5. Popolamento del Database**

## **1. DESCRIZIONE DEL PROGETTO**

Si vuole sviluppare un sistema informativo basato su un database relazionale al fine di gestire delle librerie musicali di un gruppo di utenti.

Il sistema garantisce la gestione di tracce, ascolti e le liste dei brani preferiti dei singoli utenti.

### Obiettivo:

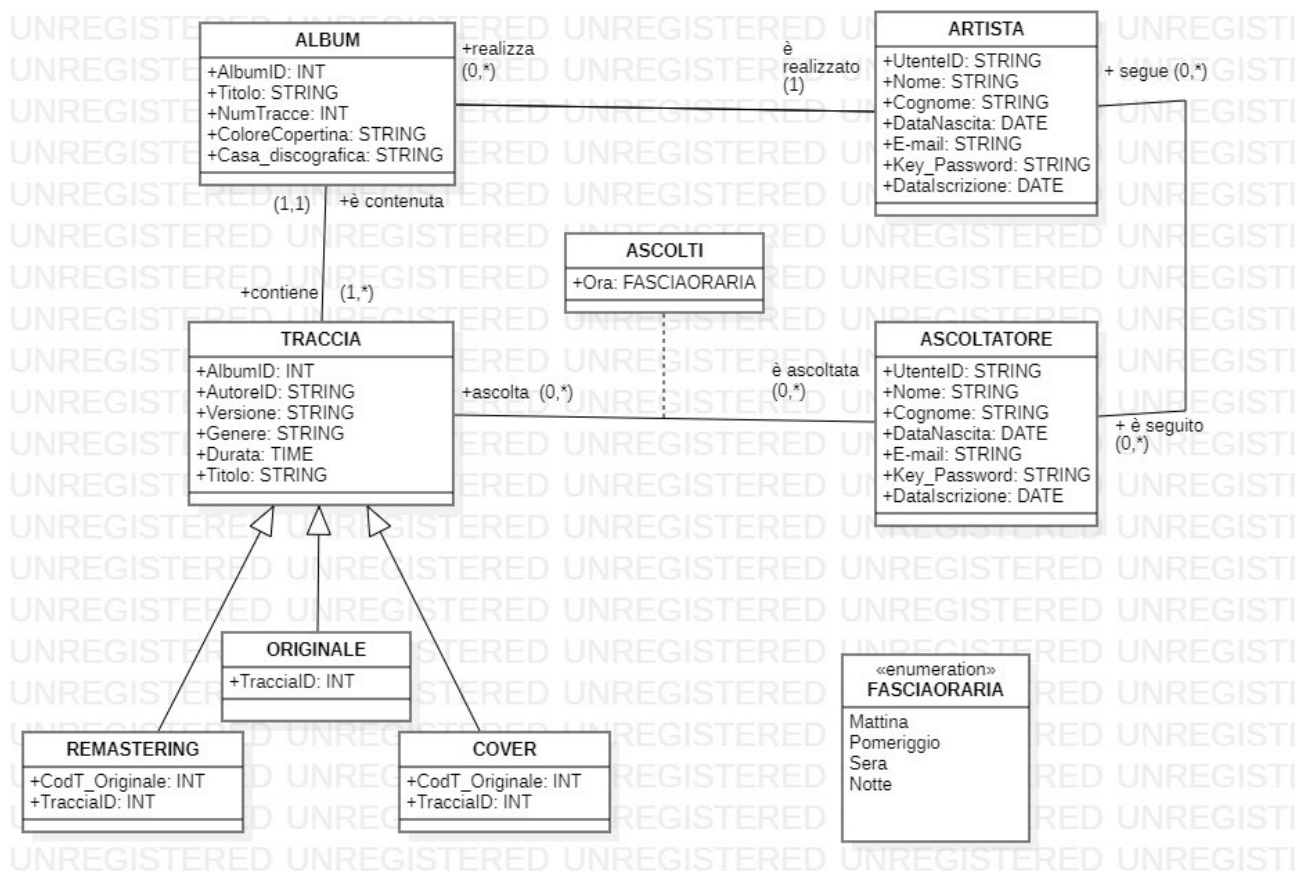
Il sistema deve permettere ad un admin di recuperare una lista di utenti che ha effettuato più ascolti di una specifica traccia, differenziando per versione.

Inoltre, il sistema deve permettere di identificare la fascia oraria in cui un determinato utente ha effettuato più ascolti.

## 2. Progettazione Concettuale.

In questo capitolo verrà riportato il CLASS DIAGRAM realizzato in fase di progettazione.

### CLASS DIAGRAM



## RISTRUTTURAZIONE DEL CLASS DIAGRAM.

La ristrutturazione di un Class Diagram è necessaria per poter procedere alla progettazione logica.

Quattro sono i punti su cui ci soffermiamo:

### I. ANALISI DEGLI ATTRIBUTI DERIVATI E DELLE RITONDANZE:

- a) Nel Class Diagram appaiono attributi che possono essere “calcolati” tramite una query; procediamo all’eliminazione di *NumTracce* nella classe ALBUM.
- b) Le classi ARTISTA e ASCOLTATORE risultano essere costituite da attributi pressoché simili; fondiamo le due classi nella classe UTENTE e definiamo l’associazione following come ricorsiva.  
Per distinguere il caso di ARTISTA e ASCOLTATORE abbiamo introdotto un nuovo attributo discriminante: TIPOUTENTE.

### II. RIMOZIONE DEGLI ATTRIBUTI MULTIPLI

Non sono presenti attributi multipli.

### III. RIMOZIONE DELLE GERARCHIE DELLE SPECIALIZZAZIONI

Una traccia può essere un Remastering, una cover, o un’originale.

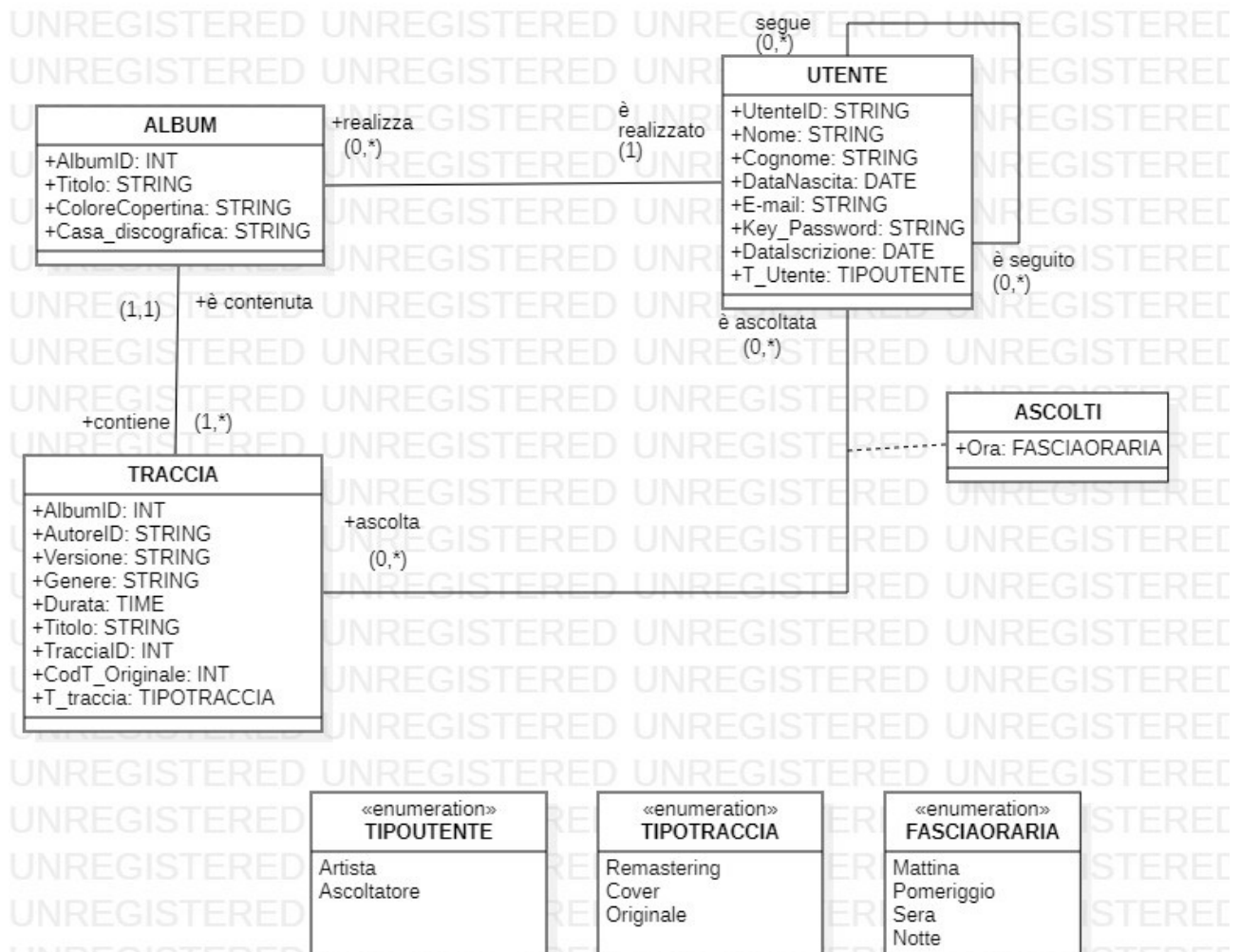
Eliminiamo la specializzazione e inseriamo all’interno della classe TRACCIA un attributo discriminante, ossia *TIPOTRACCIA* che potrà avere valore “Remastering”, “Cover” oppure “Originale”.

Differenziamo il caso in cui TIPOTRACCIA avesse valore uguale a “Cover” oppure “Remastering”, introducendo l’attributo parziale “CodT\_Originale” per poter risalire alla traccia originale.

### IV. ANALISI DEGLI ATTRIBUTI STRUTTURATI.

Non sono presenti attributi strutturati.

## CLASS DIAGRAM RISTRUTTURATO



# DIZIONARIO DELLE CLASSI

CLASSE	DESCRIZIONE	ATTRIBUTI
<b>ALBUM</b>	Descrittore del supporto discografico, contenente una raccolta di tracce musicali	<b>AlbumID</b> ( <i>Integer</i> ): identificativo dell'album <b>Titolo</b> ( <i>String</i> ): nome dell'album <b>ColoreCopertina</b> ( <i>String</i> ): colore scelto per la copertina dell'album <b>Casa_discografica</b> ( <i>String</i> ): Marchio commerciale creato dalla casa discografica che ha lavorato sull'album
<b>TRACCIA</b>	Unità che compone un album discografico. E' un brano musicale.	<b>AlbumID</b> ( <i>Integer</i> ):identificativo dell'album di appartenenza. <b>AutoreID</b> ( <i>Integer</i> ):L'autore della traccia <b>Versione</b> ( <i>String</i> ): anno in cui è stata realizzata la traccia. <b>Genere</b> ( <i>String</i> ): genere musicale a cui appartiene la traccia. <b>Durata</b> ( <i>Time</i> ): tempo necessario per l'ascolto completo della traccia. <b>Titolo</b> ( <i>String</i> ): nome/titolo associato alla traccia che consente di scegliere cosa ascoltare. <b>TracciaID</b> ( <i>Integer</i> ): Identificativo traccia. <b>CodT_Originale</b> ( <i>Integer</i> ): ci da il riferimento alla traccia originale <b>T_traccia</b> ( <i>TIPOTRACCIA</i> ): ci da informazioni sulla tipologia
<b>UTENTE</b>	Identità virtuale che usufruisce della piattaforma musicale	<b>UtenteID</b> ( <i>String</i> ): Identificativo univoco scelto dall'utente che comparirà al pubblico sulla piattaforma <b>Nome</b> ( <i>String</i> ): Nome ufficiale dell'utente <b>Cognome</b> ( <i>String</i> ): Cognome ufficiale dell'utente <b>DataNascita</b> ( <i>Date</i> ): Informazione sulla data di nascita utente. <b>Email</b> ( <i>String</i> ): email personale utilizzata dall'utente per accedere alla piattaforma <b>Key_Password</b> ( <i>String</i> ): password dell'account <b>DataIscrizione</b> ( <i>Date</i> ): Data dell'iscrizione alla piattaforma <b>T_Utente</b> ( <i>TIPOUTENTE</i> ) : Ci dice se l'utente è un autore o un ascoltatore
<b>ASCOLTI</b>	Ci fornisce un riferimento temporale in cui un utente ascolta una traccia.	<b>Ora</b> ( <i>FASCIADORARIA</i> ): indica un momento della giornata.

# DIZIONARIO DELLE ASSOCIAZIONI

NOME	DESCRIZIONE	CLASSI COINVOLTE
<b>REALIZZAZIONE</b>	Esprime la relazione tra chi ha progettato/realizzato l'album e l'album stesso	<b>UTENTE</b> [0...*] ruolo " <i>realizza</i> ": indica che l'utente può realizzare nessuno o molti album. <b>ALBUM</b> [1] ruolo " <i>è realizzato</i> ": indica che un album può essere creato da un solo utente
<b>FOLLOWING</b>	Descrive la connessione tra gli utenti come autori e utenti come ascoltatori	<b>UTENTE</b> [0...*] ruolo " <i>è seguito</i> ": indica che l'autore può essere seguito da nessuno o da più utenti <b>UTENTE</b> [0..*] ruolo " <i>segue</i> ": indica che l'utente può seguire nessuno/più utenti
<b>ASCOLTARE</b>	Ci da informazioni su una traccia e da quale utente è ascoltata; tale associazione è caratterizzata dalla classe ORARIO.	<b>TRACCIA</b> [0...*] ruolo " <i>è ascoltata</i> ": indica che una traccia può essere riprodotta o meno da un utente. <b>UTENTE</b> [0...*] ruolo " <i>ascolta</i> ": indica che un utente può ascoltare o meno una traccia.
<b>CONTENERE</b>	Descrive l'appartenenza di tracce agli album	<b>Album</b> [1...*] ruolo " <i>contiene</i> ": indica che un album contiene almeno una traccia. <b>Traccia</b> [1] ruolo " <i>è contenuta</i> ": indica che una traccia deve essere contenuta in un solo album.



## DIZIONARIO DEI VINCOLI

NOME VINCOLO	DESCRIZIONE
CHECK_EMAILLEGALFORM	Gli indirizzi e-mail degli utenti devono essere di forma legittima ovvero della forma stringa@stringa.stringa
CHECK_DATA	In UTENTE, DataIscrizione deve essere successiva a DataNascita
CHECK_BIRTH	In UTENTE, DataNascita deve essere precedente a CurrentDate
CHECK_AUTHOR	L'associazione <i>realizzazione</i> è valida solo se T_Utente è Artista.
CHECK_TIME	Durata deve avere un valore compreso tra 00:01:00.00000000 e 00:30:00.00000000
CHECK_CODT_OR	L'attributo CodT_Originale ha valore associato solo se T_Traccia è Remastering o Cover.
CHECK_LEGALFORMATVERSIONE	Versione deve essere di forma legittima ovvero deve avere la seguente forma gg-mm-yyy oppure gg/mm/yyyy.

### 3. PROGETTAZIONE LOGICA

#### Schema logico

Di seguito, la rappresentazione della progettazione logica derivata dal Class Diagram ristrutturato e che va a sostituire la struttura a tabelle del database relazionale.

 PRIMARY KEY

 FOREIGN KEY

ALBUM (**ALBUMID**, **ARTISTAID**, TITOLO, COLORECOPERTINA, CASA\_DISCOGRAFICA)

UTENTE (**UTENTEID**, NOME, COGNOME, DATANASCITA, EMAIL, KEY\_PASSWORD, DATAISCRIZIONE, T\_UTENTE)

TRACCIA (**TRACCAID**, **VERSIONE**, **ARTISTAID**, **ALBUMID**, CODT\_ORIGINALE, GENERE, DURATA, TITOLO, T\_TRACCIA)

FOLLOWERS (**UTENTE1**, **UTENTE2**)

ASCOLTI (ORA, **ASCOLTATOREID**, **TRACCAID**, **VERSIONE**, **ARTISTAID**, **ALBUMID**)

## 4. DEFINIZIONI SQL

**TABELLE.** In questo paragrafo sono riportate le definizioni delle tabelle come riportate all'interno dello schema logico.

```
CREATE TYPE FasciaOraria AS ENUM  
( 'Mattina', 'Pomeriggio', 'Sera', 'Notte' );
```

```
CREATE TYPE TipoUtente AS ENUM  
( 'Ascoltatore', 'Artista' );
```

```
CREATE TYPE TipoTraccia AS ENUM ( 'Originale', 'Remastering', 'Cover' );
```

```
CREATE EXTENSION PGCRYPTO;
```

### DEFINIZIONE DELLA TABELLA ALBUM

```
CREATE TABLE ALBUM (  
    AlbumID SERIAL not null,  
    Artistaid varchar(50) not null,  
    Titolo VARCHAR(50) NOT NULL,  
    ColoreCopertina VARCHAR(50) ,  
    Casa_discografica VARCHAR(50) ,
```

```
CONSTRAINT albumid_pk primary key ( albumid ),
```

```
CONSTRAINT check_PermissionUtente CHECK (check_UserIsArtist( artistaid )  
) ,
```

```
CONSTRAINT artistaid_fk FOREIGN KEY (artistaid) REFERENCES utente (  
utenteid ) match simple
```

```
ON UPDATE CASCADE
```

```
ON DELETE CASCADE );
```

## DEFINIZIONE DELLA TABELLA TRACCIA

```
CREATE TABLE TRACCIA (  
    TracciaID SERIAL UNIQUE NOT NULL,  
    ArtistaID VARCHAR (50) NOT NULL,  
    AlbumID INTEGER NOT NULL,  
    Versione VARCHAR(50) NOT NULL,  
    Titolo VARCHAR(50) NOT NULL,  
    Genere VARCHAR(50),  
    Durata TIME NOT NULL,  
    CodT_Originale integer default 0,  
    T_Traccia TIPOTRACCIA NOT NULL ,  
  
    CONSTRAINT pk2composta_traccia PRIMARYKEY ( TracciaID , versione ,  
albumid ,artistaId ),  
    CONSTRAINT check_legalForm_Ttraccia CHECK ( T_Traccia in  
('Originale','Remastering','Cover') ),  
    CONSTRAINT check_PermissionUtente check ( check_UserIsArtist( artistaId )  
),  
    CONSTRAINT album_tracciafk FOREIGN KEY (AlbumID) REFERENCES Album (  
Albumid ) match simple  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    CONSTRAINT artista_tracciafk FOREIGN KEY (artistaId) REFERENCES UTENTE (  
UTENTEID ) match simple  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    CONSTRAINT check_LegalFormatVersione CHECK (  
Versione ~* '^(0[1-9]|1[0-9]|2[0-9]|3[0-1])(\\|\\-)(0[1-9]|1[0-2])(\\|\\-  
)([0-9]{4})$' ) ,  
    CONSTRAINT check_durataTraccia CHECK (  
Durata < '01:00:00' AND Durata > '00:01:00' )  
);
```

## DEFINIZIONE DELLA TABELLA UTENTE

```
CREATE TABLE UTENTE(  
    UtenteID VARCHAR(50) not null ,  
    Key_Password VARCHAR(1000) NOT NULL,  
    Nome VARCHAR(50) ,  
    Cognome VARCHAR(50) ,  
    DataNascita DATE ,  
    Email VARCHAR(100) UNIQUE NOT NULL,  
    DataIscrizione DATE NULL,                /* da verificare */  
    T_Utente TIPOUTENTE DEFAULT 'Ascoltatore',  
  
    CONSTRAINT pk_utente primary key ( utenteid ),  
  
    CONSTRAINT check_emailLegalForm CHECK (  
        Email ~* '^[A-Za-z0-9._+%~]+@[A-Za-z0-9.-]+[.][A-Za-z]+$'),  
  
    CONSTRAINT check_correctBirthUtente CHECK (  
        DataNascita < current_date )  
);
```

## DEFINIZIONE DELLA TABELLA ASCOLTI

```
CREATE TABLE ASCOLTI(  
  Ora FASCIAORARIA NOT NULL,  
  AscoltatoreID VARCHAR(50) NOT NULL,  
  ArtistaID VARCHAR (50) NOT NULL,  
  TracciaID INTEGER NOT NULL,  
  Versione VARCHAR (50) NOT NULL,  
  AlbumID INTEGER NOT NULL,  
  
  CONSTRAINT check_LegalFormatVersion CHECK (  
    Versione ~* '^(0[1-9]|1[0-9]|2[0-9]|3[0-1])(\|/|-)(0[1-9]|1[0-2])(\|/|-)([0-9]{4})$' ) ,  
  
  CONSTRAINT utente_FK FOREIGN KEY (AscoltatoreID) REFERENCES  
    UTENTE (UtenteID) match simple  
    ON UPDATE CASCADE  
    ON DELETE CASCADE,  
  
  CONSTRAINT traccia__FK FOREIGN KEY (TracciaID , Versione,  
    ArtistaID, AlbumID ) REFERENCES TRACCIA (TracciaID ,  
    Versione, ArtistaID, AlbumID ) match simple  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
);
```

## DEFINIZIONE DELLA TABELLA FOLLOWERS

```
CREATE TABLE FOLLOWERS(  
    utente1 VARCHAR(255) NOT NULL,  
    utente2 VARCHAR(255) NOT NULL,  
  
    CONSTRAINT utente1_fk FOREIGN KEY (utente1 ) REFERENCES  
    Utente (UtenteID) match simple  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
  
    CONSTRAINT Utente2_fk FOREIGN KEY (utente2 )REFERENCES  
    Utente (UtenteID) match simple  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
  
    CONSTRAINT check_distinctFollowers CHECK(  
    utente1 <> utente2 )  
);
```

## FUNZIONI

**/\* restituisce un gruppo di utenti che ha ascoltato una determinata traccia differenziando per versione\*/**

```
Create or replace function who_listened (track varchar(120))  
Returns TABLE( ascoltatore varchar , titoloTraccia varchar,  
versioneTraccia varchar , numeroascolti bigint ) AS  
$who_listened$
```

```
    BEGIN  
        return query  
        SELECT *  
        FROM traccePiuAscoltate ( )  
        where traccePiuAscoltate.titolotraccia = track;  
    END
```

```
$who_listened$ language plpgsql;
```

**/\* funzione specifica che mi ritorna gli autori piu ascoltati differenziando per utenti\*/**

```
create or replace function artistiPiuAscoltati ( )  
returns table (ascoltatore varchar, artista varchar,  
numeroascolti bigint ) as  
$$  
begin  
    return query  
    SELECT ascoltatoreid , artistaid, count( * ) as nAscolti  
    FROM ascolti  
    GROUP BY ascoltatoreid, artistaid  
    order by nAscolti desc;  
end  
$$ language plpgsql;
```



**/\* funzione specifica che mi ritorna le tracce piu ascoltate differenziando per versione e utenti\*/**

```
create or replace function traccePiuAscoltate ( )  
returns table ( ascoltatore varchar , titoloTraccia varchar ,  
versioneTraccia varchar , numeroascolti bigint)as  
$$  
  
begin  
return query  
SELECT ascolti.ascoltatoreid, tracciaid1.titolo,  
ascolti.versione, COUNT(*) as nAscolti  
FROM ascolti inner join (  
    select traccia.titolo, traccia.tracciaid,  
    traccia.versione  
    from traccia )  
    as tracciaid1  
    on ascolti.tracciaid = tracciaid1.tracciaid and  
    ascolti.versione = tracciaid1.versione  
    GROUP BY ascolti.ascoltatoreid, tracciaid1.titolo,  
    ascolti.versione  
    ORDER BY nAscolti desc;  
  
end  
$$ language plpgsql;
```

**/\* funzione specifica che mi ritorna gli album piu ascoltati differenziando per utenti\*/**

```
create or replace function albumPiuAscoltati ( )
returns table ( ascoltatore varchar, titoloAlbum varchar ,
album int , numeroAscolti bigint ) as
$$

Begin

    return query
SELECT ascolti.ascoltatoreid, album1.titolo, ascolti.albumid,
COUNT(*) as nAscolti
FROM ascolti inner join (
                        select titolo, albumid
                        from album ) as album1
on ascolti.albumid = album1.albumid
GROUP BY ascolti.ascoltatoreid, album1.titolo, ascolti.albumid
ORDER by nAscolti desc;

End

$$ language plpgsql;
```

**/\* restituisce l'orario con piu ascolti dato un nome utente \*/**

```
CREATE OR REPLACE FUNCTION orarioConpiuAscolti ( utenteName
varchar ( 50 ))
RETURNS varchar (1000) AS
$timemostlistened$
```

```
BEGIN
```

```
return max( orario1.ora ) from (
SELECT ORA, count(*) as nascolti
FROM ascolti
WHERE ascoltatoreid = utentename
GROUP BY (ORA) as orario1;
```

```
END
```

```
$timemostlistened$ LANGUAGE plpgsql;
```

**/\* ritorna il numero dei followers di un singolo utente \*/**

```
CREATE OR REPLACE FUNCTION totalfollowers ( utenteid varchar (
50 ))
RETURNS integer AS
$totalfollowers$
```

```
BEGIN
```

```
return ( SELECT count(*) FROM followers
        where utente1 = utenteid ) ;
```

```
END;
```

```
$totalfollowers$ LANGUAGE plpgsql;
```

**/\* ritorna il numero dei following di un singolo utente \*/**

```
CREATE OR REPLACE FUNCTION totalfollowing ( utenteid varchar (
50 ) )
RETURNS integer AS
$totalfollowing$

BEGIN

return (
SELECT count(*)
FROM followers
WHERE utente2 = utenteid ) ;

END;

$totalfollowing$ LANGUAGE plpgsql;
```

**/\*decrittografia la l'utente esaminato\*/**

```
CREATE OR REPLACE FUNCTION decrittografia( utentemail
VARCHAR(1000) , password varchar (50))
RETURNS VARCHAR(1000) AS
$decripto$
declare
decripted varchar (1000);

BEGIN
if password = 'admin' then
    decripted = ( select key_password
FROM utente
WHERE Email = utentemail);

return (
select convert_from (decrypt(decripted::bytea, 'salty',
'aes'), 'SQL_ASCII') ) ;
else
raise EXCEPTION using message = 'Non hai diritto, per vedere
password altrui';
end if;

END
$decripto$ LANGUAGE plpgsql;
```

**/\*funzione utilizzata nel vincolo check\_permission aiuta il check a capire se l'utente che ha inserito una traccia o album è un artista\*/**

```
create or replace FUNCTION check_UserIsArtist( utenteNome
varchar (50))
returns boolean as
$checkPermissionAlbum$
    declare

    tipoutente1 tipoutente = (
SELECT t_utente
FROM utente
WHERE utenteNome = utente.utenteid) ;
    BEGIN

        if ( tipoutente1 = 'artista')
        then
            return true;
        end if;

        RAISE NOTICE 'Utente % non ha i permessi per creare un
album o traccia', utentenome;
        return false;

    END

$checkPermissionAlbum$ LANGUAGE plpgsql;
```

## TRIGGER

```
/* procedura per il trigger insertPassword, prima  
dell'inserimento di un utente cripta la password scelta  
dall'utente */
```

```
CREATE OR REPLACE FUNCTION Crittografia()  
RETURNS trigger AS  
$cripto$  
  
    BEGIN  
  
new.Key_Password = encrypt(new.Key_Password::bytea, 'salty',  
'aes');  
  
return new;  
  
END  
  
$cripto$ LANGUAGE plpgsql;
```

```
/*trigger che viene attivato prima dell'inserimento di un  
utente*/
```

```
CREATE TRIGGER insertPassword before INSERT ON utente  
  
FOR EACH ROW  
  
EXECUTE PROCEDURE Crittografia();
```

```
/* procedura per il trigger insertRemasterSong, dopo  
l'inserimento di un utente modifica codT_originale in traccia  
*/
```

```
create or replace FUNCTION aggiornaCodT_originaleTraccia()  
returns trigger as  
$updateTraccia$
```

```
BEGIN
```

```
    if new.t_traccia <> 'Originale'  
    then
```

```
        if new.t_traccia = 'Remastering'  
        then
```

```
            UPDATE traccia SET CodT_Originale = (  
select
```

```
tracciaid
```

```
from
```

```
traccia as t  
where t.titolo = new.titolo AND t.t_traccia = 'Originale' AND  
t.albumid <> new.albumid AND t.artistaid = new.artistaid)
```

```
WHERE
```

```
tracciaid = new.tracciaid;  
    else UPDATE traccia SET CodT_Originale = (  
select
```

```
tracciaid
```

```
from
```

```
traccia as t  
where t.titolo LIKE new.titolo AND t.t_traccia
```

```
= 'Originale' AND
```

```
t.albumid <> new.albumid )
```

```
WHERE
```

```
tracciaid = new.tracciaid;  
    end if;
```

```
else
```

```
update
```

```
traccia SET CodT_Originale = new.tracciaid  
WHERE
```

```
tracciaid = new.tracciaid;  
end if;
```

```
return null;
```

```
END
```

```
$updateTraccia$ LANGUAGE plpgsql;
```

```
/*trigger che viene attivato dopo l'inserimento di una  
traccia*/  
CREATE TRIGGER insertRemasterSong AFTER INSERT ON traccia  
  
FOR EACH ROW  
  
EXECUTE PROCEDURE aggiornaCodT_originaleTraccia();
```



## 5. POPOLAMENTO DEL DATABASE

```
INSERT INTO UTENTE ( UtenteID, Key_Password, nome, Cognome, DataNascita,
DataIscrizione, Email, T_Utente )
VALUES
('Squallor','Squallor',' ',' ','01/01/1969','20/04/2022','Squallor@gmail.com','artista'),
('THESCOTTS','THESCOTTS',' ',' ','24/04/2020','20/04/2022','TheSCOOTs@gmail.com','artista'),
('Travis Scott','Travis Scott', 'Jack', 'Cactus', '30/04/1991', '10/07/2010',
'Travis.Scott@outlook.com','artista'),
('Kid Cudi','Kid Cudi','Scott','Ramon','30/01/1984','11/07/2010','Kid.Cudi@live.it','artista'),
('Bob Dylan','Bob Dylan',
'Robert','Zimmerman','24/05/1941','15/09/2017','RobertZimmer@virgilio.com','artista'),
('Francesco De Gregorio','Francesco De Gregorio','Francesco','De Gregorio',
'04/04/1951','12/12/2020','DeGrego@hotmail.com','artista'),
('Guns N Roses','Guns N
Roses',' ',' ','01/01/1985','30/12/2012','RockNeverDie@hotmail.com','artista'),
('Johnny Cash','Johnny
Cash','J.R.','Cash','26/02/1932','30/10/2019','RngOFFire@virgilio.com','artista'),
('Depeche Mode','Depeche
Mode',' ',' ','01/01/1980','01/01/2022','RockModeon@live.it','artista'),
('Francesco Guccini','Francesco
Guccini','Francesco','Guccini','14/06/1940','03/12/2022','Avvelenato@live.it','artista'),
('Nomadi','Nomadi',' ',' ','01/01/1963','03/12/2021','vagabondi@gmail.com','artista'),
('Gino Paoli','Gino Paoli','Gino','Paoli','23/08/1934','01/01/2015','SaleGrosso@live.it','artista'),
('Mina','Mina','Anna
Maria','Mazzini','25/03/1940','26/08/2002','DrinDrin@outlook.com','artista'),
('MinaCelentano','MinaCelentano',' ',' ','14/05/1998','26/08/2002','CoooolTogheter@live.it','artista'),
('Adriano Celentano','Adriano','Adriano','Celentano','06/01/1938','20/04/2022',
'Gluck17@hotmail.com','artista'),
('Francesca_pgl','Francesca_pgl','Francesca','Pugliese','01/01/1996',
'20/04/2022','F.pugliese@studenti.unina.it','Ascoltatore'),
('Torci','Torci','Adolfo','Torcicollo','13/03/1999','20/04/2022','a.torcicollo@studenti.unina.it',
'Ascoltatore');
```

**/\*popolamento tabella album \*/**

**INSERT INTO ALBUM** (albumid, Titolo , ColoreCopertina, Casa\_discografica, artistaid )  
**VALUES**

('1','Mutando','Grigia','EastWest Italy','Squallor'),  
('2','THE SCOTTS','Verde','Sony Music','THESCOTTS'),  
('3','ASTROWORLD','Nero','Sony Music','Travis Scott'),  
('4','Man of The Moon','Arancione','Columbia Records','Kid Cudi'),  
('5','The Essential Bob Dylan','Bianco','Columbia Records','Bob Dylan'),  
('6','De gregorio Canta Bob Dylan - Amore e Odio','Rosso','Sony Music','Francesco  
De Gregorio'),  
('7','Use Your illusion II','Blue','Geffen Records','Guns N Roses'),  
('8','American IV: The Man Comes Around','Nero','American Recordings','Guns N  
Roses'),  
('9','Violator(Deluxe)','Nero', 'Akuma Records','Depeche Mode'),  
('10','Due anni Dopo','Bianco','Columbia Records','Francesco Guccini'),  
('11','The Platinum Collection','Grigio','EMI','Francesco Guccini'),  
('12','Via Paolo Fabbri 43','Marrone','EMI','Francesco Guccini'),  
('13','Ma noi no','Giallo','EastWest Italy','Nomadi'),  
('14','I grandi successi originali','Rosso','RCA Records','Gino Paoli'),  
('15','Il Cielo In Una Stanza','Blue','Italdisc','Mina'),  
('16','I Primi Anni Vol.II','Bianco','Italdisc','Mina'),  
('17','Mina Celentano','Giallo','PDU','MinaCelentano'),  
('18','Unicamentecelentano','Rosa','Clan Celentano','Adriano Celentano'),  
('19','Basta chiudere gli occhi', 'Verde','RCA Records','Gino Paoli');

**/\*popolamento tabella traccia \*/**

**INSERT INTO traccia** (tracciaid, titolo, durata, genere, t\_traccia, albumid, versione, artistaid)

**VALUES**

('1','Squallor in concerto','00:01:33','Non-Music','Originale','1','01/01/1981','Squallor'),  
('2','Damme e denare','00:06:01','Non-Music','Originale','1','01/01/1981','Squallor'),  
('3','Madonina','00:04:05','Non-Music','Originale','1','01/01/1981','Squallor'),  
('4','Cornutone','00:04:13','Non-Music','Originale','1','01/01/1981','Squallor'),  
('5','Torna a casa Mexico','00:05:17','Non-Music','Originale','1','01/01/1981','Squallor'),  
('6','Pret-a-porter','00:05:45','Non-Music','Originale','1','01/01/1981','Squallor'),  
('7','Pierpaolo n3','00:05:59','Non-Music','Originale','1','01/01/1981','Squallor'),  
('8','Tombeado','00:05:34','Non-Music','Originale','1','01/01/1981','Squallor'),  
('9','tango 13','00:06:17','Non-Music','Originale','1','01/01/1981','Squallor'),  
('10','THESCOTS','00:02:45','Hip-hop/Rap','Originale','2','24/04/2020','THESCOTTS'),  
('11','STARGAZING','00:04:31','Hip-Hop/Trap','Originale','3','13/08/2018','Travis Scott'),  
('12','SICKO MODE','00:04:13','Hip-Hop/Trap','Originale','3','13/08/2018','Travis Scott'),  
('13','STOP CRYING','00:05:39','Hip-Hop/Trap','Originale','3','13/08/2018','Travis Scott'),  
('14','SKELETONS','00:02:26','Hip-Hop/Trap','Originale','3','13/08/2018','Travis Scott'),  
('15','5% TINT','00:03:16','Hip-Hop/Trap','Originale','3','13/08/2018','Travis Scott'),  
('16','BUTTERFLY EFFECT','00:03:11','Hip-Hop/Trap','Originale','3','13/08/2018','Travis Scott'),  
('17','CAROUSEL','00:03:00','Hip-Hop/Trap','Originale','3','13/08/2018','Travis Scott'),  
('18','WAKE UP','00:03:52','Hip-Hop/Trap','Originale','3','13/08/2018','Travis Scott'),  
  
('19','In My Dreams','00:03:19','Hip-Hop/Alternative Rock','Originale','4','15/08/2009','Kid Cudi'),  
('20','Simple As...','00:02:31','Hip-Hop/Alternative Rock','Originale','4','15/08/2009','Kid Cudi'),  
('21','Day N Nite','00:03:41','Hip-Hop/Alternative Rock','Originale','4','15/08/2009','Kid Cudi'),  
('22','Day N Nite(Crookers Remix)','00:04:42','Hip-Hop/Alternative Rock','Cover','4','15/08/2009','Kid Cudi'),  
('23','Pursuit of Happiness','00:04:55','Hip-Hop/Alternative Rock','Originale','4','15/08/2009','Kid Cudi'),  
('24','Solo Dolo','00:04:26','Hip-Hop/Alternative Rock','Originale','4','15/08/2009','Kid Cudi'),  
  
('25','Blowin in the wind','00:02:48','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),

('26','The tymes they are a-changing','00:03:14','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),  
 ('27','Dont think Twice, its all right','00:03:40','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),  
 ('28','Mr. Tambourine Man','00:05:29','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),  
 ('29','Like a Rolling Stone','00:06:10','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),  
 ('30','Subterranean Homesick blues','00:02:20','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),  
 ('31','Lay Lady Lay','00:03:19','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),  
 ('32','I shall Be released','00:03:04','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),  
 ('33','Knockin on heavens door','00:02:31','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),  
 ('34','Hurricane','00:08:33','Rock/Folk Music','Originale','5','30/09/2000','Bob Dylan'),  
 ('35','Un angioletto come te','00:04:33','Pop','Cover','6','30/09/2015','Francesco De Gregorio'),  
 ('36','Come il giorno','00:03:58','Pop','Cover','6','30/09/2015','Francesco De Gregorio'),  
 ('37','Acido Seminterrato','00:02:08','Pop','Cover','6','30/09/2015','Francesco De Gregorio'),  
 ('38','Civil War','00:07:42','Hard Rock/Rock Clasico','Originale','7','17/08/1991','Guns N Roses'),  
 ('39','Knockin on heavens door','00:05:36','Hard Rock/Rock Clasico','Cover','7','17/08/1991','Guns N Roses'),  
 ('40','Shotgun Blues','00:03:23','Hard Rock/Rock Clasico','Originale','7','17/08/1991','Guns N Roses'),  
 ('41','Personal Jesus','00:04:55','Rock elettronico','Originale','9','19/03/1990','Depeche Mode'),  
 ('42','Enjoy the Silence','00:06:13','Rock elettronico','Originale','9','19/03/1990','Depeche Mode'),  
 ('43','Waiting for the night','00:06:07','Rock elettronico','Originale','9','19/03/1990','Depeche Mode'),  
 ('44','Personal Jesus','00:03:20','Country','Cover','8','01/01/2002','Johnny Cash'),  
 ('45','Hurt','00:03:39','Country','Cover','8','01/01/2002','Johnny Cash'),  
 ('46','Big Iron','00:03:59','Country','Cover','8','01/01/2002','Johnny Cash'),  
 ('47','Lui e Lei','00:03:13','indie/Folk','Originale','10','01/01/1970','Francesco Guccini'),  
 ('48','Vedi Cara','00:04:58','indie/Folk','Originale','10','01/01/1970','Francesco Guccini'),  
 ('49','Primavera di Praga','00:03:38','indie/Folk','Originale','10','01/01/1970','Francesco Guccini'),

('50','Il Compleanno','00:03:31','indie/Folk','Originale','10','01/01/1970','Francesco Guccini'),  
 ('51','Giorno d estate','00:03:48','indie/Folk','Originale','12','01/01/1976','Francesco Guccini'),  
 ('52','L avvelenata','00:04:35','indie/Folk','Originale','12','01/01/1976','Francesco Guccini'),  
 ('53','Canzone di notte  
 n.2','00:04:53','indie/Folk','Originale','12','01/01/1976','Francesco Guccini'),  
 ('54','Canzone Quasi d  
 amore','00:04:10','indie/Folk','Originale','12','01/01/1976','Francesco Guccini'),  
 ('55','L avvelenata','00:04:36','indie/Folk','Remastering','11','10/10/2006','Francesco Guccini'),  
 ('56','Eskimo','00:08:14','indie/Folk','Remastering','11','10/10/2006','Francesco Guccini'),  
 ('57','Farewell','00:05:15','indie/Folk','Remastering','11','10/10/2006','Francesco Guccini'),  
 ('58','Vedi Cara','00:04:51','indie/Folk','Remastering','11','10/10/2006','Francesco Guccini'),  
 ('59','Bologna','00:04:42','indie/Folk','Remastering','11','10/10/2006','Francesco Guccini'),  
 ('60','Canzone di notte  
 n.2','00:04:53','indie/Folk','Remastering','11','10/10/2006','Francesco Guccini'),  
 ('61','Canzone per un amica','00:03:20','Rock/Pop','Cover','13','15/05/1992','Nomadi'),  
 ('62','Io Vagabondo','00:03:25','Rock/Pop','Remastering','13','15/05/1992','Nomadi'),  
 ('63','Dio è morto','00:02:54','Rock/Pop','Cover','13','15/05/1992','Nomadi'),  
 ('64','Il cielo in una stanza','00:02:52','Musica D  
 artista','Originale','15','01/06/1960','Mina'),  
 ('65','Sapore Di sale','00:03:33','Musica D artista','Remastering','14','01/01/2001','Gino Paoli'),  
 ('66','Senza Fine','00:02:47','Musica D artista','Remastering','14','01/01/2001','Gino Paoli'),  
 ('67','Il cielo in una stanza','00:01:59','Pop','Cover','14','01/01/2001','Gino Paoli'),  
 ('68','Pesci Rossi','00:02:20','Pop','Cover','16','01/01/1997','Mina'),  
 ('69','La Nonna Magdalena','00:03:20','Pop','Cover','16','01/01/1997','Mina'),  
 ('70','Il cielo in una stanza','00:02:59','Pop','Remastering','16','01/01/1997','Mina'),  
 ('71','Pesci Rossi','00:02:22','Pop','Remastering','16','01/01/1997','Mina'),  
 ('72','La Nonna Magdalena','00:02:10','Pop','Remastering','16','01/01/1997','Mina'),  
 ('73','Tintarella di Luna','00:03:06','Pop','Remastering','16','01/01/1997','Mina'),  
 ('74','Acqua e sale','00:04:42','Folk/Pop','Originale','17','14/05/1998','MinaCelentano'),  
 ('75','Brivido  
 Felino','00:03:44','Folk/Pop','Originale','17','14/05/1998','MinaCelentano'),  
 ('76','Si è spento il sole','00:02:56','Rock/Pop Rock','Cover','18','10/11/2006','Adriano Celentano'),  
 ('77','Azzurro','00:03:43','Rock/Pop Rock','Remastering','18','10/11/2006','Adriano Celentano'),

('78','Il tempo se ne va','00:04:55','Rock/Pop  
 Rock','Remastering','18','10/11/2006','Adriano Celentano'),  
 ('79','24 mila baci','00:02:14','Rock/Pop Rock','Cover','18','10/11/2006','Adriano  
 Celentano'),  
 ('80','Sapore Di sale','00:03:33','Musica D artista','Originale','19','01/01/1964','Gino  
 Paoli');

**/\*popolamento tabella ascolti \*/**

**INSERT INTO ASCOLTI** ( Ascoltatoreid, Ora, tracciaid, albumid, versione, artista)  
 values

('Francesca\_pgl','Mattina','10','2','24/04/2020','THESCOTTS'),  
 ('Francesca\_pgl','Sera','70','16','01/01/1997','Mina'),  
 ('Francesca\_pgl','Mattina','23','4','15/08/2009','Kid Cudi'),  
 ('Francesca\_pgl','Sera','70','16','01/01/1997','Mina'),  
 ('Francesca\_pgl','Pomeriggio','48','10','01/01/1970','Francesco Guccini'),  
 ('Francesca\_pgl','Mattina','70','16','01/01/1997','Mina'),  
 ('Francesca\_pgl','Pomeriggio','70','16','01/01/1997','Mina'),  
 ('Torci','Pomeriggio','10','2','24/04/2020','THESCOTTS'),  
 ('Torci','Notte','23','4','15/08/2009','Kid Cudi'),  
 ('Torci','Pomeriggio','10','2','24/04/2020','THESCOTTS'),  
 ('Torci','Pomeriggio','23','4','15/08/2009','Kid Cudi'),  
 ('Torci','Notte','10','2','24/04/2020','THESCOTTS'),  
 ('Torci','Notte','10','2','24/04/2020','THESCOTTS'),  
 ('Torci','Notte','23','4','15/08/2009','Kid Cudi'),  
 ('Francesca\_pgl','Notte','52','12','01/01/1976','Francesco Guccini'),  
 ('Francesca\_pgl','Pomeriggio','52','12','01/01/1976','Francesco Guccini'),  
 ('Torci','Pomeriggio','10','2','24/04/2020','THESCOTTS'),  
 ('Torci','Pomeriggio','10','2','24/04/2020','THESCOTTS'),  
 ('Francesca\_pgl','Mattina','70','16','01/01/1997','Mina'),  
 ('Francesca\_pgl','Mattina','70','16','01/01/1997','Mina'),  
 ('Francesca\_pgl','Pomeriggio','52','12','01/01/1976','Francesco Guccini'),  
 ('Francesca\_pgl','Pomeriggio','48','10','01/01/1970','Francesco Guccini'),  
 ('Torci','Mattina','48','10','01/01/1970','Francesco Guccini');

**/\*tabella followers\*/**

**INSERT INTO FOLLOWERS ( utente1 , utente2 )**

values

```
('Torci','Kid Cudi'),
('Torci','Travis Scott'),
('Torci','THESCOTTS'),
('Torci','Squallor'),
('Torci','Francesco De Gregorio'),
('Torci','Bob Dylan'),
('Torci','Gino Paoli'),
('Torci','Francesca_pgl'),
('Francesca_pgl','Torci'),
('Francesca_pgl','Gino Paoli'),
('Francesca_pgl','Mina'),
('Francesca_pgl','MinaCelentano'),
('Francesca_pgl','Adriano Celentano'),
('Francesca_pgl','Nomadi'),
('Francesca_pgl','Depeche Mode'),
('Kid Cudi','Torci'),
('Travis Scott','Torci'),
('THESCOTTS','Torci'),
('Squallor','Torci'),
('Francesco De Gregorio','Torci'),
('Bob Dylan','Torci'),
('Gino Paoli','Francesca_pgl'),
('Mina','Francesca_pgl'),
('MinaCelentano','Francesca_pgl'),
('Adriano Celentano','Francesca_pgl'),
('Nomadi','Francesca_pgl'),
('Depeche Mode','Francesca_pgl');
```