Database EmotionalSongs

# CODICE DI CREAZIONE DEL DATABASE POSTGRESQL

CREATE TABLE CANZONI (

ID\_UNIVOCO VARCHAR(20) PRIMARY KEY,

REPO\_INDEX INT UNIQUE,

TITOLO VARCHAR NOT NULL,

AUTORE VARCHAR NOT NULL,

ANNO INT NOT NULL

);

CREATE TABLE UTENTI\_REGISTRATI(

ID\_USER VARCHAR(20) PRIMARY KEY,

EMAIL VARCHAR UNIQUE NOT NULL,

CF VARCHAR(16) UNIQUE NOT NULL,

PASSWORD VARCHAR(20) NOT NULL,

NOME VARCHAR NOT NULL,

COGNOME VARCHAR NOT NULL,

DATA\_NASCITA DATE NOT NULL,

INDIRIZZO VARCHAR NOT NULL DEFAULT ‘not found’,

LOGGED BOOL NOT NULL DEFAULT FALSE;

);

CREATE TABLE PLAYLIST (

ID\_PLAYLIST VARCHAR(20) PRIMARY KEY,

USER\_PROP\_ID VARCHAR(20) NOT NULL,

NOME\_PLAYLIST VARCHAR DEFAULT 'NewPlaylist'

);

ALTER TABLE PLAYLIST ADD FOREIGN KEY(USER\_PROP\_ID) REFERENCES UTENTI\_REGISTRATI(ID\_USER) ON UPDATE CASCADE ON DELETE CASCADE;

CREATE TABLE CONTENUTO\_PLAYLIST(

PLAYLIST\_ID VARCHAR(20),

CANZONE\_ID VARCHAR(20),

PRIMARY KEY(PLAYLIST\_ID,CANZONE\_ID)

);

ALTER TABLE CONTENUTO\_PLAYLIST ADD FOREIGN KEY(PLAYLIST\_ID) REFERENCES PLAYLIST(ID\_PLAYLIST) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE CONTENUTO\_PLAYLIST ADD FOREIGN KEY(CANZONE\_ID) REFERENCES CANZONI(ID\_UNIVOCO) ON DELETE CASCADE ON UPDATE CASCADE;

CREATE TABLE EMOZIONI (

USER\_PROP\_ID VARCHAR(20),

CANZONE\_ID VARCHAR(20),

COMMENTO VARCHAR(256),

MERAVIGLIA INT CHECK(MERAVIGLIA BETWEEN 1 AND 5) NOT NULL,

SOLENNITA INT CHECK(SOLENNITA BETWEEN 1 AND 5) NOT NULL,

TENEREZZA INT CHECK(TENEREZZA BETWEEN 1 AND 5) NOT NULL,

NOSTALGIA INT CHECK(NOSTALGIA BETWEEN 1 AND 5) NOT NULL,

PACATEZZA INT CHECK(PACATEZZA BETWEEN 1 AND 5) NOT NULL,

POTERE INT CHECK(POTERE BETWEEN 1 AND 5) NOT NULL,

GIOIA INT CHECK(GIOIA BETWEEN 1 AND 5) NOT NULL,

TENSIONE INT CHECK(TENSIONE BETWEEN 1 AND 5) NOT NULL,

TRISTEZZA INT CHECK(TRISTEZZA BETWEEN 1 AND 5) NOT NULL,

PRIMARY KEY(USER\_PROP\_ID,CANZONE\_ID)

);

ALTER TABLE EMOZIONI ADD FOREIGN KEY(CANZONE\_ID) REFERENCES CANZONI(ID\_UNIVOCO) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE EMOZIONI ADD FOREIGN KEY(USER\_PROP\_ID) REFERENCES UTENTI\_REGISTRATI(ID\_USER) ON DELETE CASCADE ON UPDATE CASCADE;

# SCHEMA UML DATABASE

# 

# NOTE DI PROGETTO DEL DATABASE

Si è deciso, in fase di progettazione, di:

* Inserire nella tabella UTENTI\_REGISTRATI la colonna LOGGED tipo BOOL per permettere al server di controllare che un utente non sia loggato da più terminali contemporaneamente, di default settato FALSE.
* Per questioni di gestione della visualizzazione delle canzoni del repositori si è deciso ad ogni CANZONE di assegnare un indice univoco – colonna: REPO\_INDEX , unique, tipo numerico.
* Essendo che una playlist può contenere tante canzoni, ogni playlist è assegnata ad un utente solo, ma ogni utente può avere più playlist, si è deciso di dividere la rappresentazione database della playlist in due tabelle distinte:
  + PLAYLIST
    - Contenente l’ id univoco (generato dal software che si appoggia al DB) della playlist, il suo nome (scelto dal utente) e l’ id del proprietario.
    - Sostanzialmente rappresenta l’ associazione utente <-> playlist.
  + CONTENUTO\_PLAYLIST
    - Contenente, per ogni riga, l’ id della playlist e l’ id di una delle canzoni che essa contiene;
    - Ogni playlist è rappresentata da un insieme di righe avente in comune il valore della colonna PLAYLIST\_ID.
* Non essendo stata richiesto che il software esegua alcun campionamento per zona geografica, si è deciso di rappresentare l’ indirizzo del utente nella colonna INDIRIZZO di UTENTI\_REGISTRATI come semplice valore letterale generico, i cui controlli di validità sono rimandati al software di gestione del DB: ServerES.