**CREAZIONE TABELLE**

**CREATE TABLE UTENTE**(

IDUtente integer PRIMARY KEY

Nome varchar NOT NULL

Cognome varchar NOT NULL

CF varchar NOT NULL

Via varchar NOT NULL

Numero varchar NOT NULL

Città varchar NOT NULL

CAP varchar NOT NULL

Email varchar NOT NULL

NomeUtente varchar NOT NULL

Password varchar NOT NULL

);

Rivedere IDUtente se va bene integer

Numero varchar: siccome può contenere anche lettere 🡪 13D

CAP varchar: siccome possono contenere come primo valore lo 0 che però potrebbe venire troncato

**CREATE TABLE PLAYLIST**(

IDPlaylist integer PRIMARY KEY

Nome varchar NOT NULL

IDUtente integer NOT NULL references UTENTE(IDUtente) ON UPDATE CASCADE, ON DELETE CASCADE????

);

**CREATE TABLE CANZONE**(

IDCanzone integer PRIMARY KEY

Autore varchar NOT NULL

Anno integer NOT NULL

Titolo varchar NOT NULL

);

**CREATE TABLE CONTIENE**(

IDPlaylist references PLAYLIST(IDPlaylist) ON UPDATE CASCADE, ON DELETE CASCADE???

IDCanzone references CANZONE(IDCanzone) ON UPDATE CASCADE, ON DELETE CASCADE???

);

**CREATE TABLE EMOZIONE**(

ID\_EMOZIONE integer PRIMARY KEY

VotoStupore integer

CommentoStupore varchar

VotoSolennità integer

CommentoSolennità varchar

VotoTenerezza integer

CommentoTenerezza varchar

VotoNostalgia integer

CommentoNostalgia varchar

VotoCalma integer

CommentoCalma varchar

VotoPotenza integer

CommentoPotenza varchar

VotoGioia integer

CommentoGioia varchar

VotoNervosismo integer

CommentoNervosismo varchar

VotoTristezza integer

CommentoTristezza varchar

IDCanzone references CANZONE(IDCanzone) ON UPDATE CASCADE, ON DELETE CASCADE

IDUtente references UTENTE(IDUtente) ON UPDATE CASCADE, ON DELETE CASCADE

);

* Verificare varchar che ci stiano 256 caratteri

**QUERY**

* Ricerca canzone per titolo, autore, anno
  + Singole e composte

SELECT Titolo, Autore, Anno

FROM CANZONE

WHERE Titolo = “TitoloInserito” AND Autore = “AutoreInserito” AND Anno = “AnnoInserito”

-

SELECT Titolo, Autore, Anno

FROM CANZONE

WHERE Titolo = “TitoloInserito”

-

SELECT Titolo, Autore, Anno

FROM CANZONE

WHERE Autore = “AutoreInserito”

-

SELECT Titolo, Autore, Anno

FROM CANZONE

WHERE Anno = “AnnoInserito”

-

SELECT Titolo, Autore, Anno

FROM CANZONE

WHERE Titolo = “TitoloInserito” AND Autore = “AutoreInserito”

-

SELECT Titolo, Autore, Anno

FROM CANZONE

WHERE Titolo = “TitoloInserito” AND Anno = “AnnoInserito”

-

SELECT Titolo, Autore, Anno

FROM CANZONE

WHERE Titolo = Autore = “AutoreInserito” AND Anno = “AnnoInserito”

* Login
  + NomeUtente e password
  + Verificare se esiste utente

SELECT IDUtente, NomeUtente

FROM UTENTE

WHERE NomeUtente = “NomeUtenteInserito” AND Password = “PasswordInserita”

* Creazione nuovo utente
  + Controllo già esistenza ed aggiunta eventuale
  + Controllo tramite mail che sarà univoca

SELECT COUNT (IDUtente)

FROM UTENTE

WHERE Email = “EmailInserita”

* Se questa query restituisce qualcosa allora significa che esiste già un utente con quella mail e quindi non si può creare un altro utente, altrimenti, se la query è vuota andiamo ad inserire il nuovo utente

INSERT INTO UTENTE (Nome, Cognome, CF, Via, Numero, Città, CAP, Email, NomeUtente, Password)

VALUES (NomeInserito, CognomeInserito, CFInserito, ViaInserita, NumeroInserito, CittàInserita, CAPInserito, EmailInserita, NomeUtenteGenerato, PasswordInserita)

* GetID utente

SELECT \*

FROM UTENTE

WHERE IDUtente = IDUtenteInserito

* Ricerca playlist
  + Tramite identificativo si stampano tutte le canzoni contenute in una playlist

SELECT Titolo, Autore, Anno

FROM CANZONE

WHERE IDCanzone IN (

SELECT IDCanzone

FROM CONTIENE

GROUP BY IDPlaylist

AND

IDPlaylist = (

SEELCT IDCanzone

FROM PLAYLIST

WHERE Nome = NomeInserito AND IDPlaylist = IDInserito

)

)

* Crea playlist
  + Se esiste già una playlist con quel nome, errore
  + Altrimenti si crea

SELECT COUNT(\*)

FROM PLAYLIST

WHERE Nome = NomeInserito AND IDUtente = IDUtente

-

INSERT INTO PLAYLIST(Nome, IDUtente)

VALUES(NomeInserito, IDUtente)

* Aggiungere canzoni playlist
  + Verifichiamo se esiste quella playlist (se count è 0 allora non esiste)
  + Verifichaimo se dentro a quella playlist c’è quella canzone
  + Inseriamo la canzone nella playlist

SELECT COUNT(\*)

FROM PLAYLIST

WHERE Nome = NomeInserito AND IDUtente = IDUtente

-

SELECT COUNT(\*)

FROM CONTIENE

WHERE IDCanzone = IDCanzoneInserito

GROUP BY IDPlaylist, IDCanzone

* Visualizzare tutte le canzoni

SELECT \*

FROM CONTIENE

WHERE IDPlaylist = (

SELECT IDPlaylist

FROM PLAYLIST

WHERE Nome = NomeInserito)

* Inserisci emozione
  + IDUtente, IDCanzone
  + Utilizziamo la query di ricerca per l’IDCanzone
  + L’IDUtente lo prendiamo dalla sessione

INSERT INTO EMOZIONE(VotoStupore, CommentoStupore, VotoSolennità, CommentoSolennità, VotoTenerezza, CommentoTenerezza, VotoNostalgia, CommentoNostalgia,VotoCalma, CommentoCalma, VotoPotenza, CommentoPotenza, VotoGioia, CommentoGioia, VotoNervosismo, CommentoNervosismo, VotoTristezza, CommentoTristezza, IDCanzone, IDUtente

)

VALUES (

VotoStuporeInserito, CommentoStuporeInserito, VotoSolennitàInserito, CommentoSolennitàInserito, VotoTenerezzaInserito, CommentoTenerezzaInserito, VotoNostalgiaInserito, CommentoNostalgiaInserito,VotoCalmaInserito, CommentoCalmaInserito, VotoPotenzaInserito, CommentoPotenzaInserito, VotoGioiaInserito, CommentoGioiaInserito, VotoNervosismoInserito, CommentoNervosismoInserito, VotoTristezzaInserito, CommentoTristezzaInserito, IDCanzone, IDUtente

)

* Contiene
* Ricerca emozioni relative ad una canzone
  + Utilizziamo la query di ricerca per l’IDCanzone

SELECT \*

FROM EMOZIONE

WHERE IDCanzone = IDCanzone

* Verifica che un utente non crei più playlist con lo stesso nome

SELECT COUNT(\*)

FROM PLAYLIST

WHERE Nome = NomeInserito AND IDUtente = IDUtente

* Elenco playlist di un utente
  + Ricerca IDUtente tramite sessione

SELECT Nome

FROM PLAYLIST

WHERE IDUtente = IDUtente

* Restituzione nome playlist tramite idPlaylist

SELECT Nome

FROM PLAYLIST

WHERE IDPlaylist = IDPlaylist