**Progetto di Basi di Dati: Piattaforma musicale**

**1. Raccolta delle specifiche della realtà d’interesse**

Descrizione

Lo scopo di questo progetto è realizzare una base di dati per la gestione di una piattaforma di streaming musicale.

L’accesso alla piattaforma può essere effettuato con la creazione di un Account Utente che viene identificato univocamente dalla combinazione di e-mailepassword. Inoltre, di un Account teniamo traccia anche di nome, cognome, data di nascita, nazionalità e preferenze. Per quanto riguarda le preferenze, se ne possono memorizzare molteplici. Tramite un Account è possibile sottoscrivere un abbonamento che permette di distinguere l’utente che lo utilizza come Utente Premium. La sottoscrizione viene effettuata in una certa data dalla quale ne deriva una data di scadenza dell’abbonamento che siamo intenzionati a memorizzare. Ogni abbonamento è identificato univocamente da un numero, può essere di un determinato tipo e ha un certo prezzo. Un utente può sottoscrivere al più un abbonamento per volta. Un Account Utente che non effettua un abbonamento, è considerato un utente di tipo normale e ha una data di scadenza che si intende memorizzare, stabilita alla sua iscrizione, e al cui raggiungimento non sarà possibile utilizzare la piattaforma senza sottoscrivere un abbonamento. I tipi di utente sono tutti e soli quelli elencati, ovvero un utente può essere esclusivamente di tipo normale o premium, e alla scadenza di un eventuale abbonamento, nell’intenzione di voler ancora utilizzare il servizio, il sistema è in attesa di una nuova sottoscrizione. Un utente, in una sua istanza di accesso, può connettersi a più dispositivi, i quali, per motivazione di privacy e sicurezza, vengono schedati e memorizzati in base al loro indirizzo ip, tipo di dispositivo (computer, tablet, telefono,…) e sono individuati da un codice. Con un dato dispositivo è possibile connettersi a più account utente. Per ogni connessione è importante tenere traccia dell’orario di connessione. Un Utente può creare diverse playlist, per ognuna di esse memorizziamo la data di creazione, il nome, il numero delle canzoni contenute al suo interno e un codice che la identifica. Le canzoni scritte dagli artisti memorizzati vengono mantenute nella piattaforma. Delle canzoni manteniamo il genere, NSFW (Una caratteristica che esprime uno stato che condiziona la possibilità di essere ascoltata da alcuni utenti in particolari condizioni) e sono distinte per nome, durata e codice identificativo dell’album in cui sono inserite, inoltre degli album memorizziamo il nome, la data di pubblicazione e il tipo. Nel nostro miniworld riserviamo anche un suo sottoinsieme alla definizione degli artisti. In particolare, teniamo traccia del nome, pseudonimo, cognome, e-mail, e loro codice univoco; ognuno di essi può comparire in diversi album, scrivere canzoni e, più nel dettaglio, appartenere ad un’etichetta discografica. Ogni etichetta è identificata da un nome e ha un tipo e un contatto.

Specifiche di Interesse

Nel realtà di interesse della nostra piattaforma musicale che vogliamo rappresentare, è necessario creare un Account utente mediante inserimento di e-mail, non presente come elemento di accesso di altri Account, e password con lunghezza di minimo 7 caratteri e massimo 15 (di cui almeno 1 carattere maiuscolo e almeno 1 carattere numerico). Nell’atto della sottoscrizione sono imposti alcuni limiti in merito all’età degli iscritti, infatti c’è bisogno di avere minimo 12 anni per poter creare l’account. Un Account Utente può essere esclusivamente di tipo “Normale” o di tipo “Premium”. Per quanto riguarda il primo otteniamo le potenzialità base della piattaforma, quali il numero limitato di ascolti liberi (skip o salti di canzoni), creazione di playlist o raccolte, seguire playlist create da altri utenti, possibilità di collegare il proprio account con vari dispositivi(Tv, smartphone, smartwatch, auto, ecc…).

Le limitazioni di un account utente non provvisto di abbonamento sono: impossibilità di ascoltare i brani in modalità offline e quindi anche di effettuare download di canzoni ed album, qualità ridotta dell’audio,ed oltre a questo ha un periodo limitato di prova entro il quale poter usare liberamente la piattaforma che dura 3 mesi. Di controparte, la sottoscrizione di un abbonamento permette di sbloccare tutte le caratteristiche bloccate in modalità normale, per cui si può avere a disposizione un'alta qualità dell’audio, skip senza limiti, ecc… . In caso di Account utente con sottoscrizione di abbonamento si può scegliere tra diverse tipologie, tra cui:

- Abbonamento Premium Individuale: pensato per utenti di ogni genere che intendono utilizzare il servizio in maniera individuale;

-Abbonamento Premium per Studenti: pensato per studenti attualmente iscritti ad un’università con un prezzo ridotto rispetto al Premium Individuale;

-Abbonamento Premium Duo: è un tipo di abbonamento perfetto per due persone che intendono condividere le proprie scelte in ambito musicale. Per cui si necessita di due Account da collegare in fase iniziale di sottoscrizione;

-Abbonamento Premium Famiglia: è pensato per una famiglia di al più 6 persone con condivisione degli n gusti musicali su n account coinvolti e consente ai membri tutori della famiglia di controllare i brani espliciti degli utenti minori.

Un Utente può sottoscriversi al più ad un Abbonamento per volta, ciò nonostante manteniamo anche lo storico di quelli scaduti.

Tutte le tipologie di Abbonamenti esposte offrono tutte le funzionalità Premium elencate in precedenza.

Un utente può connettersi a vari dispositivi (di ogni tipo) contemporaneamente ma con la limitazione di poter ascoltare musica su un dispositivo per volta , infatti è necessario effettuare uno switch sul dispositivo che ne richiede il controllo.

Ogni utente, indipendentemente dalla sua tipologia di Account, ha la possibilità di creare quante Playlist desidera, pur rispettando alcuni criteri, ad es. Il nome della playlist può avere un massimo di 30 caratteri e inoltre, siccome le playlist sono raggruppate all’interno del database per utente proprietario, non possono avere nello stesso gruppo lo stesso nome per disambiguare nella scelta e nella condivisione della playlist ad altri utenti, in aggiunta è presente un certo set o dizionario di parole vietate nella scelta di un nome di playlist. Le playlist sono condivise ad altri utenti che hanno criteri di scelta dei brani simili alle tipologie di brani che è possibile trovare in essa.

La differenza sull’inserimento di una canzone nel database sta nel fatto che non si è obbligati a porsi limiti per quanto riguarda la scelta di parole che compongono il nome, siccome c’è la possibilità di marcare una canzone come “Esplicita” o meno (Tramite l’attributo booleano NSFW (**NSFW** è l'acronimo di **Not safe for work**, ossia **non sicuro per il lavoro**; usato principalmente su [siti](https://it.wikipedia.org/wiki/Sito_web), [forum](https://it.wikipedia.org/wiki/Forum_(Internet)), [blog](https://it.wikipedia.org/wiki/Blog) e [chat](https://it.wikipedia.org/wiki/Chat), serve a indicare [URL](https://it.wikipedia.org/wiki/Uniform_Resource_Locator) o collegamenti a materiale esplicito, volgare o potenzialmente offensivo, in modo che l'utente possa evitare di incapparvi inavvertitamente durante l'orario di lavoro, a scuola o in situazioni di privacy limitata.) ), il cui stato di sarà visualizzato dall’utente tramite l’apparizione di un bollino affianco ad essa.

All’interno della piattaforma, si tiene traccia degli Artisti che possono comporre delle canzoni che verranno eventualmente inserite anch’esse nel database, ma più in generale, per varie ragioni, gli Artisti vengono memorizzati a prescindere dal momento in cui effettuano pubblicazioni, ragion per cui esistono dei fattori esterni di scelta quali ad es. (appartenenza ad un etichetta, popolarità, frequenza di composizione di brani, ecc…). È possibile ricercare specifiche Playlist, Canzoni, Album, Artisti, Utenti ed Etichette Discografiche in base al loro nome identificativo ed è possibile personalizzare ulteriormente la ricerca indicando, in particolare per le canzoni, un intervallo di secondi entro i quali devono rientrare i risultati ottenibili oppure è possibile differenziare l’output in base all’appartenenza ad una specifica etichetta, ecc… .

Altra interazione effettuabile con gli elementi principali della nostra piattaforma è l’indicazione di preferenze per un’Artista e/o una Canzone tramite la marcatura da parte dell’utente di un simbolo previsto per mantenere tale informazione. Più in dettaglio, con la marcatura riferita ad una Canzone, l’utente crea o modifica in maniera automatica una playlist che contiene tutte le canzoni per cui è stato mostrato interesse, mentre con la marcatura per un Artista, si allunga la lista di preferenze relative all’Utente.

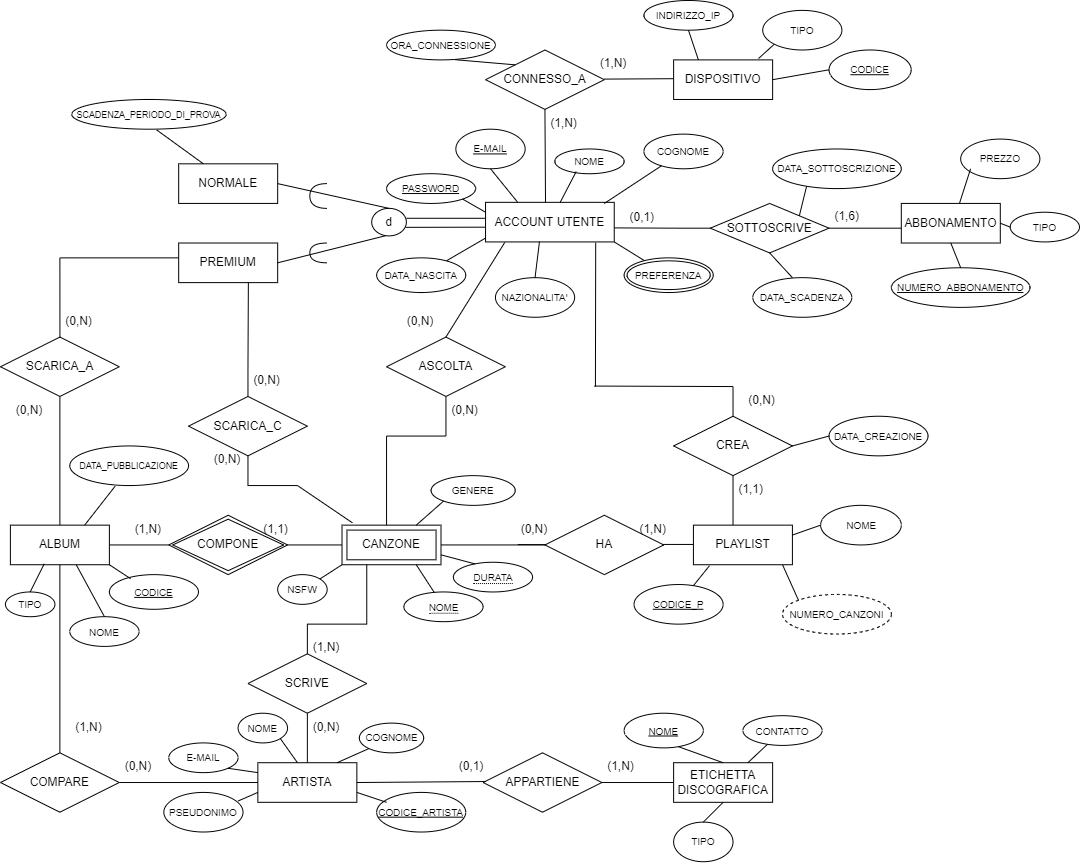
Così come nella Playlist creata implicitamente dal sistema e in tutte le playlist esiste un ordine preciso di ascolto alla terminazione di ogni brano contenuto in esse, e riguarda la cronologia di aggiunta alla Playlist. Tuttavia, tale ordine può essere mutato in base a delle scelte che l’Utente può effettuare in una schermata dedicata visualizzata alla selezione di ogni brano, ad es. può decidere di riascoltare il brano corrente quante volte desidera oppure può selezionare la modalità “Riproduzione Casuale” (la cosiddetta Shuffle) che permette, tramite un algoritmo di scelta randomica innescato alla fine del brano, di selezionare brani in maniera casuale. Si può anche scegliere di terminare la nostra serie di ascolti alla fine della playlist o ricominciare dal primo brano di essa in ripetizione o “loop”.

Glossario Dei Termini

| Termine | Significato |
| --- | --- |
| Account Utente | L’ Account Utente è il concetto che permette di accedere alla piattaforma musicale. |
| Normale | E’ una modalità che consente di eseguire delle azioni base all’interno della piattaforma. |
| Premium | Tale modalità è resa disponibile ad un utente solo se esso sottoscrive un abbonamento di qualsiasi tipo e rende possibile l’utilizzo di tutte le funzionalità offerte dalla piattaforma. |
| Abbonamento | Strumento che concede all’utente base di beneficiare dello sblocco di funzionalità aggiuntive per un determinato periodo di tempo. |
| Playlist | Raccolta di canzoni che può essere creata da un utente oppure generata automaticamente in base ai gusti dell’utente. |
| Canzone | Brano messo a disposizione dalla piattaforma per tutti gli utenti (eccetto limitazioni come la NSFW per minorenni ad es.) |
| Album | Pubblicazione a nome di uno o più artisti che contiene una collezione di canzoni scritte esclusivamente da essi. |
| Artista | Autore dei brani che vengono memorizzati dalla piattaforma. |

**2. Progettazione concettuale della Base di Dati**

Schema EER



Dizionario delle Entità

**Legenda**: sottoclasse, attributo multivalore, attributo ridondante, entità debole, chiave candidata.

| Entità | Descrizione | Attributi | Identificatore |
| --- | --- | --- | --- |
| Account Utente | Profilo con cui accedere alla piattaforma. | * nome * cognome * e-mail * password * nazionalita * data\_nascita * preferenza | e-mail  password |
| Normale | Modalità di utilizzo di base | * Scadenza\_Periodo\_di\_prova | / |
| Premium | Modalità di utilizzo completo delle funzionalità | / | / |
| Abbonamento | Contratto di una certa durata stipulato con la piattaforma che sblocca la modalità Premium | * numero abbonamento * prezzo * tipo | numero abbonamento |
| Dispositivo | Macchina attraverso la quale potersi collegare alla piattaforma | * codice * indirizzo ip * tipo | codice |
| Playlist | Raccolta di canzoni che può essere creata da un utente (o in maniera automatica al verificarsi di alcune condizioni) | * codice\_p * numero canzoni * nome | codice\_p |
| Canzone | Brano che è possibile riprodurre da parte di tutti gli utenti o solo di un suo sottogruppo | * nome * durata * genere * nsfw | nome  durata |
| Album | Raccolta di brani pubblicata da uno o più autori | * codice * nome * tipo * data\_pubblicazione | codice |
| Artista | Autori dei brani che vengono memorizzati dalla piattaforma | * codice artista * nome * cognome * pseudonimo | codice artista |
| Etichetta Discografica | Marchio di una casa discografica a cui possono appartenere uno o più autori | * nome * contatto * tipo | nome |

Dizionario delle Relazioni

| Relazione | Descrizione | Entità coinvolte | Attributi |
| --- | --- | --- | --- |
| Sottoscrive | Un utente sottoscrive un abbonamento alla piattaforma | Account Utente (0,1)  Abbonamento (1,1) | Data Sottoscrizione  Data Scadenza |
| Crea | Un utente crea una playlist musicale | Account Utente (0,N)  Playlist (1,1) | / |
| Ascolta | Riproduzione da parte dell’utente di un brano | Account Utente (0,N)  Canzone (0,N) | / |
| Connesso\_A | Associazione di un utente ad un particolare dispositivo | Account Utente (1,N)  Dispositivo (1,N) | Ora\_Connessione |
| Ha | Relazione di appartenenza di una canzone ad una playlist | Playlist (1,N)  Canzone (0,N) | / |
| Scarica\_C | Un utente di tipo Premium effettua il download di un brano | Premium (0,N)  Canzone (0,N) | / |
| Scarica\_A | Un utente di tipo Premium effettua il download di un album | Premium (0,N)  Album (0,N) | / |
| Scrive | Associazione tra una canzone ed uno o più autori | Canzone (1,N)  Artista (0,N) | / |
| Compone | Relazione di appartenenza di una canzone ad un album | Canzone (1,1)  Album (1,N) | / |
| Compare | Un artista collabora in un album | Artista (0,N)  Album (1,N) | / |
| Appartiene | Etichetta discografica a cui un artista fa riferimento | Artista (0,1)  Etichetta Discografica (1,N) | / |

Vincoli non esprimibili nello schema

Di seguito vengono specificati vincoli che non è possibile rappresentare nello schema EER:

* L’ attributo “password” dell’entità Account Utente deve avere lunghezza massima di 15 caratteri;
* L’attributo “Data\_Nascita” dell’entità Account Utente deve esprimere un età maggiore di 12 anni;
* L’attributo “Nome” dell’entità Playlist deve avere al più 30 caratteri e deve essere univoco.

**3. Definizione delle procedure per la gestione della base di dati:**

Tavola dei volumi

| Concetto | Tipo | Carico Applicativo |
| --- | --- | --- |
| Account Utente | E | 30 |
| Normale | E | 10 |
| Premium | E | 20 |
| Abbonamento | E | 60 |
| Dispositivo | E | 100 |
| Playlist | E | 150 |
| Canzone | E | 500 |
| Album | E | 50 |
| Artista | E | 200 |
| Etichetta Discografica | E | 50 |
| Sottoscrive | R | 20 |
| Ascolta | R | 1500 |
| Scarica\_A | R | 100 |
| Scarica\_C | R | 1000 |
| Connesso A | R | 450 |
| Crea | R | 150 |
| Ha | R | 1500 |
| Compone | R | 500 |
| Compare | R | 100 |
| Scrive | R | 1000 |
| Appartiene | R | 100 |

Tavola delle Operazioni

|  | Operazione | Tipo | Frequenza |
| --- | --- | --- | --- |
| 1 | Creare un nuovo account | I | 10/mm |
| 2 | Sottoscrivere un abbonamento | I | 3/mm |
| **3** | Crea una playlist | I | 5/mm |
| **4** | Aggiunta di una canzone in una playlist | I | 50/mm |
| 5 | Inserimento di una canzone | I | 40/mm |
| 6 | Inserimento di un artista | I | 20/mm |
| **7** | Visualizzazione di tutte le informazioni relative a tutte le playlist, incluso il numero di canzoni per ognuna | B | 3/mm |
| 8 | Aggiunta di un’etichetta discografica al database | I | 5/mm |
| 9 | Utente Premium seleziona una canzone da scaricare | I | 20/mm |
| 10 | Inserimento di un Album | I | 5/mm |
| 11 | Trovare l’indirizzo IP di un dispositivo | I | 50/mm |
| 12 | Trovare il numero di canzoni Esplicite (NSFW) | B | 6/mm |
| 13 | Utente Premium seleziona un Album da scaricare | I | 5/mm |
| 14 | Trovare tutti gli utenti di una data nazionalità che non ascoltano un dato genere musicale | I | 10/mm |
| 15 | Trovare tutte le playlist che non contengono canzoni NSFW | I | 3/mm |

**4. Progettazione Logica**

Analisi delle Ridondanze

Il nostro dato ridondante è “Numero\_Canzoni” dell’entità Playlist. E’ ridondante perché è derivabile dal calcolo del numero di occorrenze di una data playlist nella relazione “Ha” tra le due entità Playlist e Canzone. Supponendo che l’attributo sia memorizzato con **2 byte** in memoria e che sono presenti 150 playlist nel nostro database, la sua memorizzazione occuperebbe in totale **300 byte**. E’ necessario decidere se mantenere o meno il dato ridondante, la quale decisione viene effettuate mediante la costruzione delle Tavole degli accessi per ogni operazione che coinvolge l’eventuale ridondante. Ciò che determina il risultato è la differenza della somma del numero di accessi in entrambe le situazioni (presenza o meno del dato ridondante).

(Gli accessi in scrittura per convenzione valgono doppio; ovvero 1S=2 accessi

e uno in lettura vale come accesso singolo; 1L=1 accesso)

Tavola degli Accessi

Operazione 3 (Con ridondanza)

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Playlist | E | 1 | S |
| Crea | R | 1 | S |
| Ha | R | (1500/150)\*1=10 | S |

**Totale** = (10+1+1)S = 24 accessi per operazione

Applicando la frequenza con la quale viene svolta l’operazione otteniamo

(24\*5/mm) accessi= 120 accessi/mm + 300 byte di memorizzazione

Operazione 3 (Senza Ridondanza)

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Playlist | E | 1 | S |
| Crea | R | 1 | S |
| Ha | R | (1500/150)\*1=10 | S |

**Totale** = (10+1+1)S = 24 accessi per operazione

Applicando la frequenza con la quale viene svolta l’operazione otteniamo

(24\*5/mm) accessi= 120 accessi/mm

Operazione 4 (Con ridondanza)

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Ha | R | (1500/500)=3 | S |
| Playlist | E | 3 | L |
| Playlist | E | 3 | S |

**Totale** = 3S + 3L+ 3S = 6S + 3L = 12+3 = 15 accessi

Aggiungendo la frequenza di esecuzione dell’operazione otteniamo:

(15\*50/mm) accessi= 750 accessi/mm + 300 Byte di memoria usati per mantenere l’attributo

Operazione 4 (Senza Ridondanza)

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Ha | R | 3 | S |

**Totale** = 3S = 6 accessi

Aggiungiamo la frequenza di operazione:

(6\*50/mm) accessi = 300 accessi/mm

Operazione 7 (Con Ridondanza)

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Playlist | E | 150 | L |

**Totale** = 150L = 150 accessi

Ai quali, aggiungendo la frequenza operativa diventano:

(150\*3/mm) accessi = 450 accessi/mm + 300 byte di memorizzazione per l’attributo ridondante

Operazione 7 (Senza Ridondanza)

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Playlist | E | 150 | L |
| Ha | R | (1500/150)\*150=1500 | L |

**Totale** = (1500 + 150) L = 1650 accessi

Aggiungendo la frequenza operativa diventano:

(1650\*3/mm) accessi = 4950 accessi/mm

Confrontiamo ora entrambi gli scenari (con e senza ridondanza):

**Con ridondanza**:

Totale Accessi = (450+750+120) accessi/mm = 1320 accessi/mm

Memoria Occupata = 300 byte (in base al carico applicativo dell’entità contenente l’attributo)

**Senza Ridondanza**:

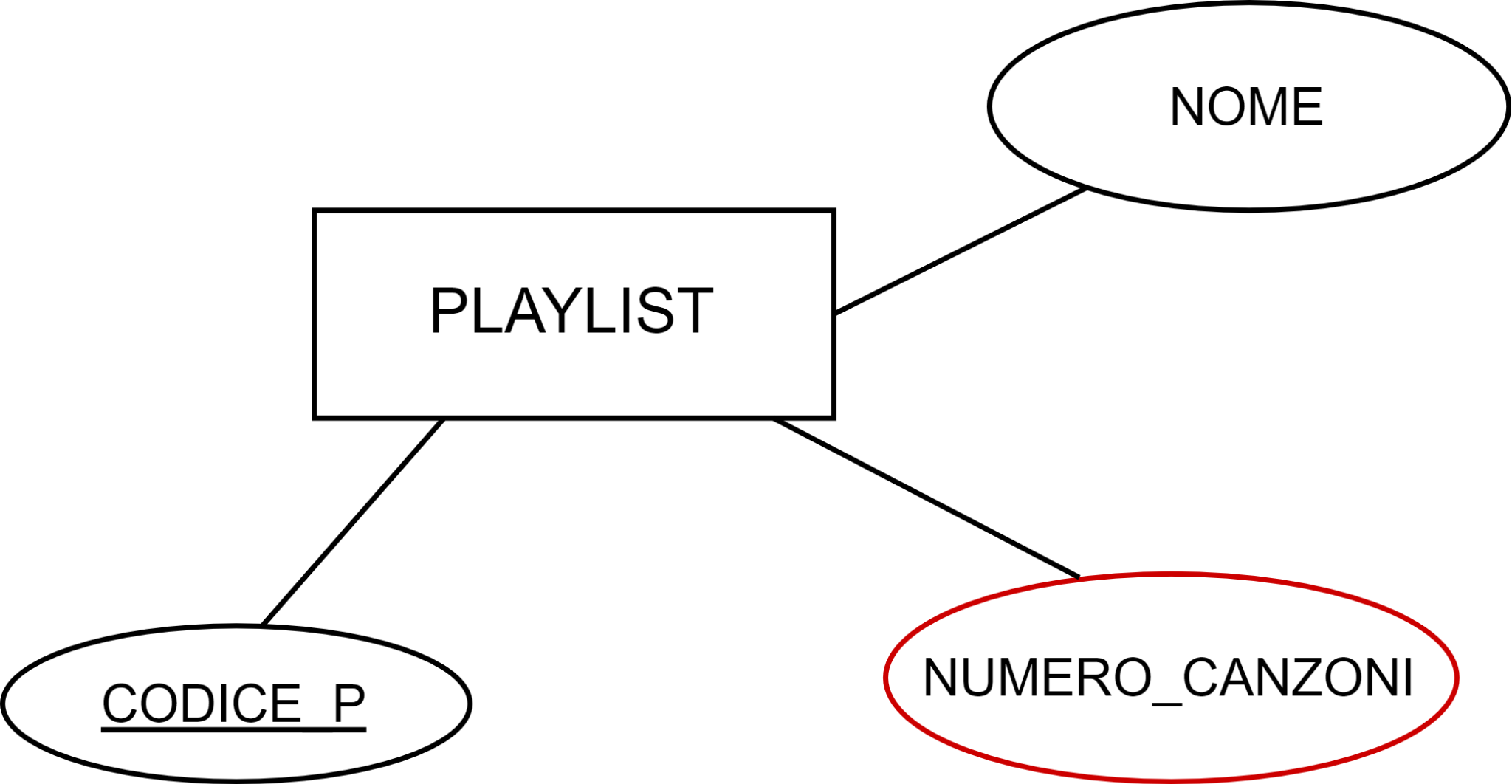
Totale Accessi = (4950+300+120) accessi/mm = 5370 accessi/mm

Senza ulteriori occupazioni in memoria

Per cui dato che

1320 accessi/mm con dato ridondante << 5370 accessi/mm senza dato ridondante

Concludiamo che conviene mantenere la ridondanza per cui l’entità playlist non viene modificata.

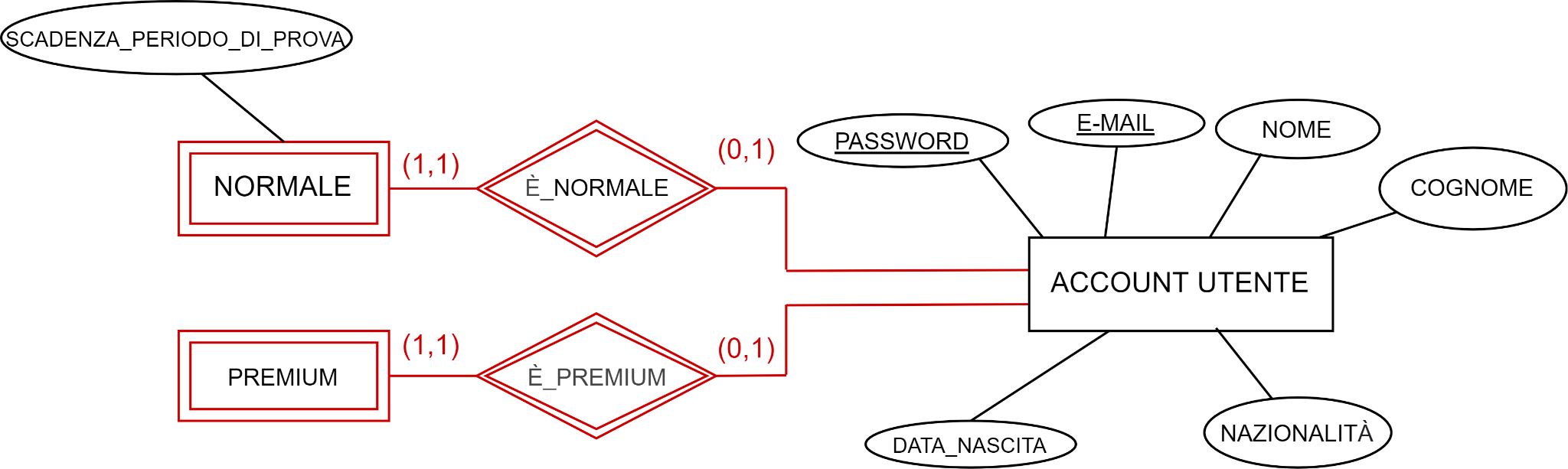


Eliminazione delle Gerarchie

Da come possiamo osservare dallo schema ER, è presente una gerarchia che nasce dall’entità “Account Utente”, che presenta come sottoclassi l’entità “Normale” e l’entità “Premium”.

Delle strategie conosciute, operiamo in modo da sostituire la gerarchia con due relazioni (una per ogni sottoclasse) perché esistono operazioni che distinguono tra entità padre e entità figlie (vedere la n.9 ad es.) ed inoltre possediamo alcune caratteristiche e relazioni che coinvolgono solo le occorrenze delle classi figlie, portando altrimenti alla generazione di eventuali valori nulli, oppure aumentando il numero di accessi per stabilire il tipo di utente se si è scelto di seguire altre strategie

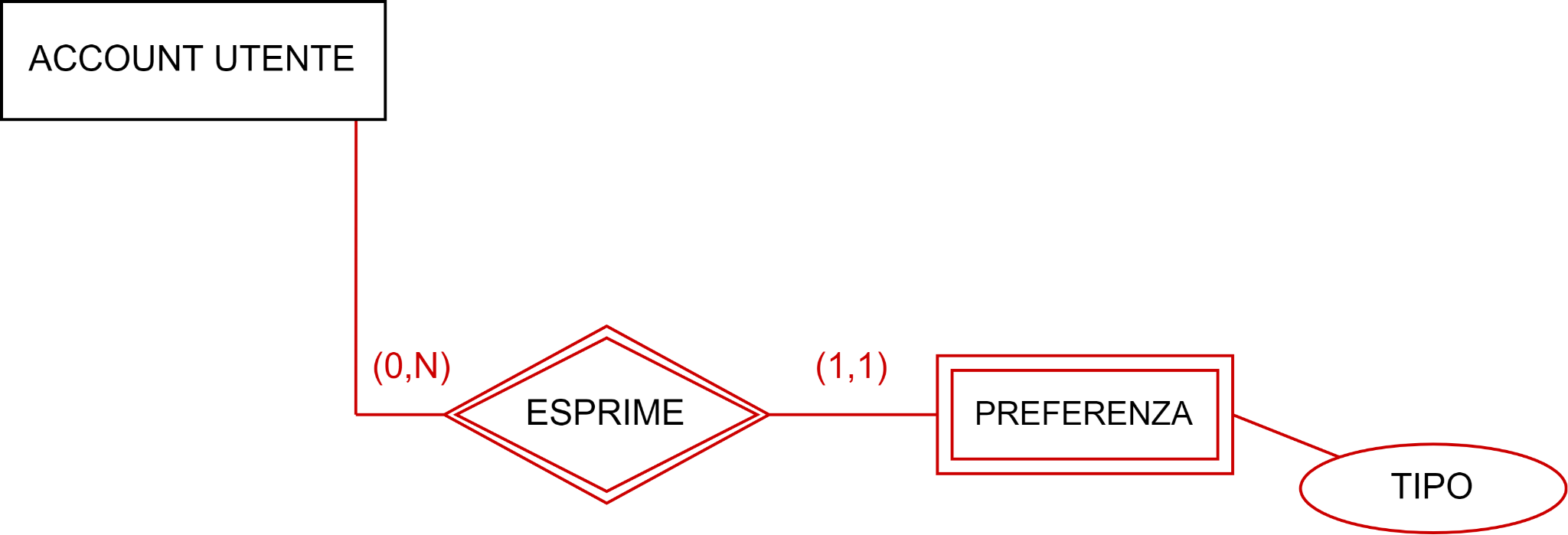
Ovviamente, dato che la gerarchia era di tipo totale e disgiunta, il vincolo di cardinalità della relazione è del tipo (1,1).



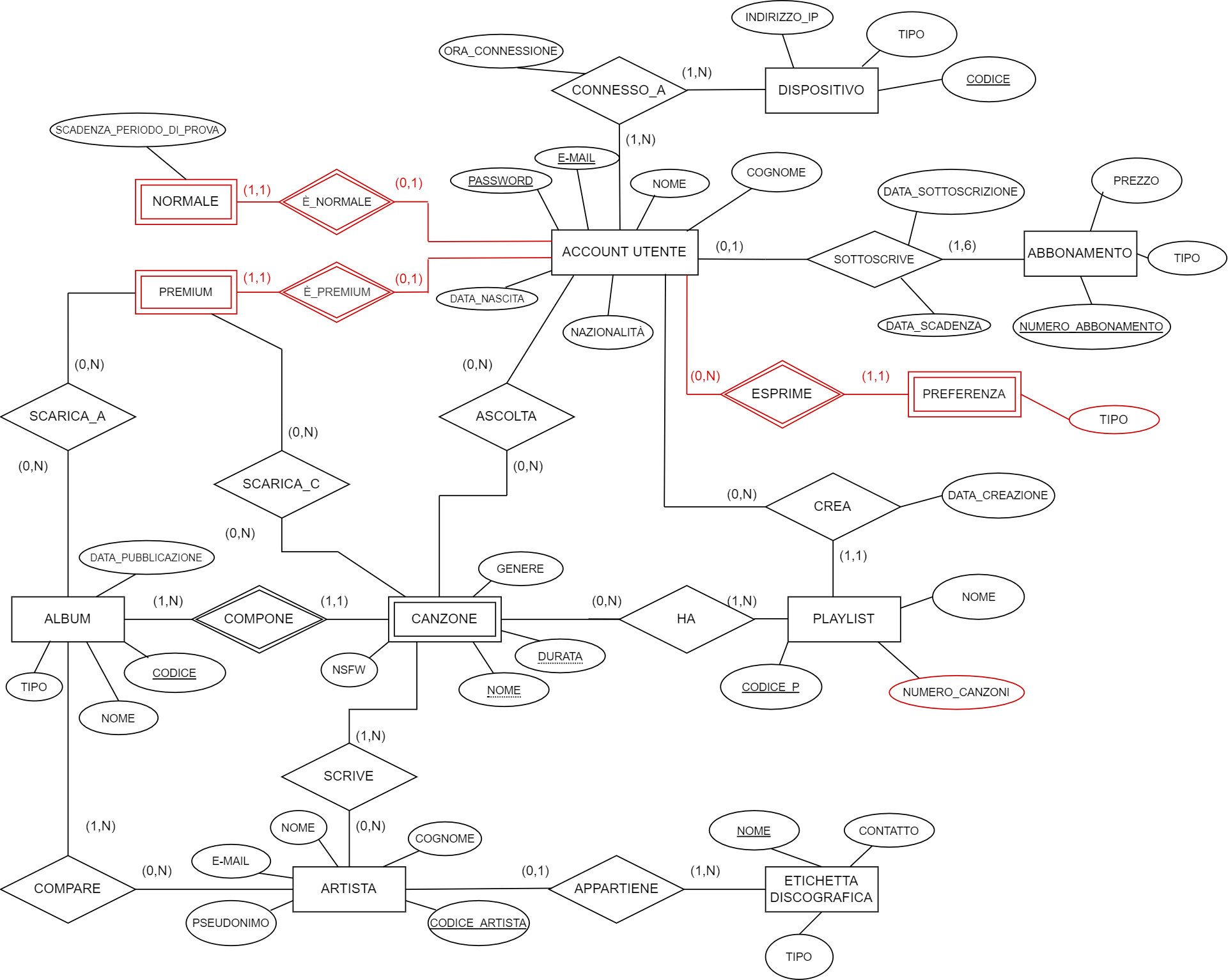
Eliminazione dell’Attributo multivalore

Nello schema ER visto in principio, esiste un attributo multivalore denominato “Preferenza” sull’entità “Account Utente”

Dato che nel modello relazionale, non è possibile rappresentarne uno di questo tipo, siamo costretti a sostituirlo con altri costrutti base del modello EER. Per cui scegliamo di definire una nuova entità debole “Preferenza” che ha un attributo di nome “Tipo”.



Schema EER Ristrutturato

Terminata la Fase di Ristrutturazione, lo schema che ne deriva è il seguente:

Schema Relazionale

Si passa ora alla fase di mapping:

**Account Utente** (E-mail,Password,Nome,Cognome,Data\_Nascita,Nazionalita)

**Abbonamento** (Numero\_Abbonamento, Prezzo, Tipo)

**Sottoscrive** (Account Utente.Password↑, AccountUtente.E-mail↑, Abbonamento.Numero\_Abbonamento↑, data\_sottoscrizione, data\_scadenza)

**Dispositivo** (Codice,Tipo,Indirizzo\_IP)

**Connesso\_A** (Account Utente.Password↑, AccountUtente.E-mail↑, Dispositivo.Codice↑, Ora\_Connessione)

**Normale** (Scadenza\_Periodo\_di\_Prova, Account Utente.Password↑, Account Utente.E-mail↑)

**Premium** (Account Utente.Password↑, Account Utente.E-mail↑)

**Album** (Codice, Nome, Tipo, Data\_Pubblicazione)

**Canzone** (Durata, Nome, Album.Codice↑, NSFW, Genere)

**Scarica\_A** (Album.Codice↑,Premium.Account Utente.Password↑, Premium.AccountUtente.E-mail↑)

**Scarica\_C** (Premium.Account Utente.Password↑, Premium.AccountUtente.E-mail↑, Canzone.Durata↑, Canzone.Nome↑, Canzone.Album.Codice↑)

**Ascolta** (Account\_Utente.E-mail↑, Account\_Utente.Password↑, Canzone.Durata↑, Canzone.Nome↑, Canzone.Album.Codice↑)

**Preferenza** (Tipo, Account\_Utente.E-mail↑, Account\_Utente.Password↑)

**Playlist** (Codice\_P, Nome, Numero\_Canzoni, Data\_Creazione, Account Utente.Password↑,

Account Utente.E-mail↑)

**Ha** (Playlist.Codice\_P↑, Canzone.Nome↑, Canzone.Durata↑, Canzone.Album.Codice↑)

**Artista** (Codice\_Artista, Pseudonimo, Nome, Cognome, E-mail)

**Compare** (Album.Codice↑, Artista.Codice\_Artista↑)

**Scrive** (Canzone.Nome↑, Canzone.Durata↑, Canzone.Album.Codice↑, Artista.Codice\_Artista↑)

**Etichetta Discografica** (Nome,Contatto,Tipo)

**Appartiene** (Etichetta Discografica.Nome↑, Artista.Codice\_Artista↑)

Normalizzazione

Osservando lo schema EER Ristrutturato, esso si trova in 2NF in quanto:

È già in **prima forma normale** dato che i valori inseriti negli schemi sono tutti atomici dopo la rimozione dell’ Attributo multivalore “Preferenze” dall’entità Account utente.

È in **seconda forma normale** in quanto, oltre ad essere in 1NF, tutte le volte che abbiamo chiavi primarie composte da più attributi, tutti gli attributi non primi sono pienamente funzionalmente dipendenti da esse e tutte le dipendenze funzionali che coinvolgono la chiave sono piene e non parziali.

Deve essere normalizzato in 3NF in quanto esistono le due seguenti dipendenze transitive che hanno sia come parte sinistra che destra della dipendenza un attributo non primo:

(Artista.Nome,Artista.Cognome) -> (Artista.Pseudonimo)

e

(Abbonamento.Tipo) -> (Abbonamento.Prezzo)

e ciò va risolto con la scomposizione degli schema in cui sono contenuti in due schemi di relazione per ognuno formati in maniera seguente

**Per la prima dipendenza:**

**Artista** (Codice\_Artista, Artista1.Nome↑, Artista1.Cognome↑)

**Artista1** (Nome,Cognome,Pseudonimo)

**Per la seconda dipendenza:**

**Abbonamento** (Numero\_Abbonamento, Abbonamento1.Tipo↑)

**Abbonamento1** (Tipo,Prezzo)

**5. Realizzazione della base di dati con MySQL**

Di seguito il listato delle istruzioni MySQL per la creazione della base di dati:

drop schema PiattaformaMusicale;

create schema PiattaformaMusicale;

use PiattaformaMusicale;

create table Account\_Utente

(

email varchar(50) NOT NULL,

password varchar(15) NOT NULL,

nome varchar(20) NOT NULL,

cognome varchar(20) NOT NULL,

data\_nascita date NOT NULL,

nazionalità varchar(20) NOT NULL,

PRIMARY KEY (email,password)

);

create table Abbonamento1

(

tipo varchar(20) NOT NULL,

prezzo double NOT NULL,

PRIMARY KEY (tipo)

);

create table Abbonamento

(

numero\_Abbonamento int NOT NULL,

tipo varchar(20) NOT NULL,

PRIMARY KEY (numero\_Abbonamento),

FOREIGN KEY (tipo) REFERENCES Abbonamento1(tipo)

);

create table Sottoscrive

(

email varchar(50) NOT NULL,

password varchar(15) NOT NULL,

numero\_Abbonamento int NOT NULL,

data\_sottoscrizione date NOT NULL,

data\_scadenza date NOT NULL,

PRIMARY KEY (email,password,numero\_Abbonamento),

FOREIGN KEY (email,password) REFERENCES Account\_Utente (email,password),

FOREIGN KEY (numero\_Abbonamento) REFERENCES Abbonamento (numero\_Abbonamento)

);

create table Dispositivo

(

codice int NOT NULL,

tipo varchar(20) NOT NULL,

indirizzo\_ip varchar(30) NOT NULL,

PRIMARY KEY (codice)

);

create table Connesso\_A

(

password varchar(15) NOT NULL,

email varchar(50) NOT NULL,

codice int NOT NULL,

ora\_connessione time NOT NULL,

PRIMARY KEY (password,email,codice),

FOREIGN KEY (email,password) REFERENCES Account\_Utente (email,password),

FOREIGN KEY (codice) REFERENCES Dispositivo (codice)

);

create table Normale

(

scadenza\_periodo\_di\_prova date NOT NULL,

password varchar(15) NOT NULL,

email varchar(50) NOT NULL,

PRIMARY KEY (password,email),

FOREIGN KEY (email,password) REFERENCES Account\_Utente (email,password)

);

create table Premium

(

password varchar(15) NOT NULL,

email varchar(50) NOT NULL,

PRIMARY KEY (password,email),

FOREIGN KEY (email,password) REFERENCES Account\_Utente (email,password)

);

create table Album

(

codice int NOT NULL,

nome varchar(20) NOT NULL,

tipo varchar(30) NOT NULL,

data\_pubblicazione date NOT NULL,

PRIMARY KEY (codice)

);

create table Canzone

(

durata time NOT NULL,

codice int NOT NULL,

nome varchar(30) NOT NULL,

NSFW bit(1) NOT NULL,

genere varchar(20) NOT NULL,

PRIMARY KEY (durata,nome,codice),

FOREIGN KEY (codice) REFERENCES Album(codice)

);

create table Scarica\_A

(

codice int NOT NULL,

password varchar(15) NOT NULL,

email varchar(50) NOT NULL,

PRIMARY KEY (codice,password,email),

FOREIGN KEY (codice) REFERENCES Album(codice),

FOREIGN KEY (email,password) REFERENCES Premium (email,password)

);

create table Scarica\_C

(

password varchar(15) NOT NULL,

email varchar(50) NOT NULL,

durata time NOT NULL,

nome varchar(30) NOT NULL,

codice int NOT NULL,

PRIMARY KEY (password,email,durata,nome,codice),

FOREIGN KEY (email,password) REFERENCES Premium (email,password),

FOREIGN KEY (durata,nome,codice) REFERENCES Canzone (durata,nome,codice)

);

create table Ascolta

(

password varchar(15) NOT NULL,

email varchar(50) NOT NULL,

durata time NOT NULL,

nome varchar(30) NOT NULL,

codice int NOT NULL,

PRIMARY KEY (password,email,durata,nome,codice),

FOREIGN KEY (email,password) REFERENCES Account\_Utente (email,password),

FOREIGN KEY (durata,nome,codice) REFERENCES Canzone (durata,nome,codice)

);

create table Preferenza

(

password varchar(15) NOT NULL,

email varchar(50) NOT NULL,

tipo varchar(20) NOT NULL,

PRIMARY KEY (password,email,tipo),

FOREIGN KEY (email,password) REFERENCES Account\_Utente (email,password)

);

create table Playlist

(

codice\_P int NOT NULL,

nome varchar(30) NOT NULL,

numero\_canzoni int NOT NULL,

data\_Creazione date NOT NULL,

password varchar(15) NOT NULL,

email varchar(50) NOT NULL,

PRIMARY KEY (codice\_P),

FOREIGN KEY (email,password) REFERENCES Account\_Utente (email,password)

);

create table Ha

(

codice\_P int NOT NULL,

nome varchar(30) NOT NULL,

durata time NOT NULL,

codice int NOT NULL,

PRIMARY KEY (codice\_P,nome,durata,codice),

FOREIGN KEY (codice\_P) REFERENCES Playlist(codice\_p),

FOREIGN KEY (durata,nome,codice) REFERENCES Canzone (durata,nome,codice)

);

create table Artista1

(

nome varchar(20) NOT NULL,

cognome varchar(20) NOT NULL,

pseudonimo varchar(20) NOT NULL,

PRIMARY KEY (nome,cognome)

);

create table Artista

(

codice\_Artista int NOT NULL,

nome varchar(20) NOT NULL,

cognome varchar(20) NOT NULL,

email varchar(50) NOT NULL,

PRIMARY KEY (codice\_Artista),

FOREIGN KEY (nome,cognome) REFERENCES Artista1 (nome,cognome)

);

create table Compare

(

codice int NOT NULL,

codice\_Artista int NOT NULL,

PRIMARY KEY (codice,codice\_Artista),

FOREIGN KEY (codice) REFERENCES Album (codice),

FOREIGN KEY (codice\_Artista) REFERENCES Artista (Codice\_Artista)

);

create table Scrive

(

durata time NOT NULL,

codice int NOT NULL,

nome varchar(30) NOT NULL,

codice\_Artista int NOT NULL,

PRIMARY KEY (durata,codice,nome,codice\_Artista),

FOREIGN KEY (durata,nome,codice) REFERENCES Canzone (durata,nome,codice),

FOREIGN KEY (codice\_Artista) REFERENCES Artista (codice\_Artista)

);

create table Etichetta\_Discografica

(

nome varchar(30) NOT NULL,

contatto varchar(50) NOT NULL,

tipo varchar(30) NOT NULL,

PRIMARY KEY (nome)

);

create table Appartiene

(

nome varchar(30) NOT NULL,

codice\_Artista int NOT NULL,

PRIMARY KEY (nome,codice\_Artista),

FOREIGN KEY (nome) REFERENCES Etichetta\_Discografica (nome),

FOREIGN KEY (codice\_Artista) REFERENCES Artista (codice\_Artista)

);

**6. Implementazione Query SQL**

**Operazione 1:**

INSERT INTO Account\_Utente (e-mail,Password,Nome,Cognome,Data\_Nascita,Nazionalita)

VALUES

(?,?,?,?,?,?);

**Operazione 2:**

INSERT INTO Sottoscrive (password,e-mail,numero\_Abbonamento,data\_sottoscrizione,data\_scadenza)

VALUES

(?,?,?,?,?);

**Operazione 3:**

INSERT INTO Playlist (codice\_P,nome,numero\_canzoni,data\_creazione,password,e-mail)

VALUES

(?,?,?,?,?,?);

**Operazione 4:**

INSERT INTO Ha (codice\_P,nome,durata,codice)

VALUES

(?,?,?,?);

**Operazione 5:**

INSERT INTO Canzone

(nome,durata,codice,NSFW,genere)

VALUES

(?,?,?,?,?);

**Operazione 6:**

INSERT INTO Artista

(codice\_Artista,nome,cognome,e-mail)

VALUES

(?,?,?,?);

**Operazione 7:**

SELECT \*

FROM Playlist;

**Operazione 8:**

INSERT INTO Etichetta\_Discografica (nome,contatto,tipo)

VALUES

(?,?,?);

**Operazione 9:**

INSERT INTO Scarica\_C

(password,e-mail,nome,durata,codice)

VALUES

(?,?,?,?,?);

**Operazione 10:**

INSERT INTO Album (codice,nome,tipo,data\_pubblicazione)

VALUES

(?,?,?,?);

**Operazione 11:**

SELECT D.indirizzo\_ip

FROM Dispositivo AS D

WHERE D.codice=?;

**Operazione 12:**

SELECT COUNT(\*)

FROM Canzone AS C

WHERE C.NSFW=1;

**Operazione 13:**

INSERT INTO Scarica\_A

(codice,password,e-mail)

VALUES

(?,?,?);

**Operazione 14:**

SELECT A.nome,A.cognome

FROM Account\_Utente AS A

WHERE A.nazionalità=? AND

NOT EXISTS(SELECT \*

FROM ASCOLTA AS k,Canzone AS C

WHERE k.email=A.email AND

k.password=A.password AND

k.durata=C.durata AND

k.nome=C.nome AND

k.codice=C.codice AND

C.genere=?)

**Operazione 15:**

SELECT P.nome

FROM Playlist AS P

WHERE NOT EXISTS (SELECT \*

FROM Canzone AS C, Ha AS H

WHERE C.durata=H.durata AND

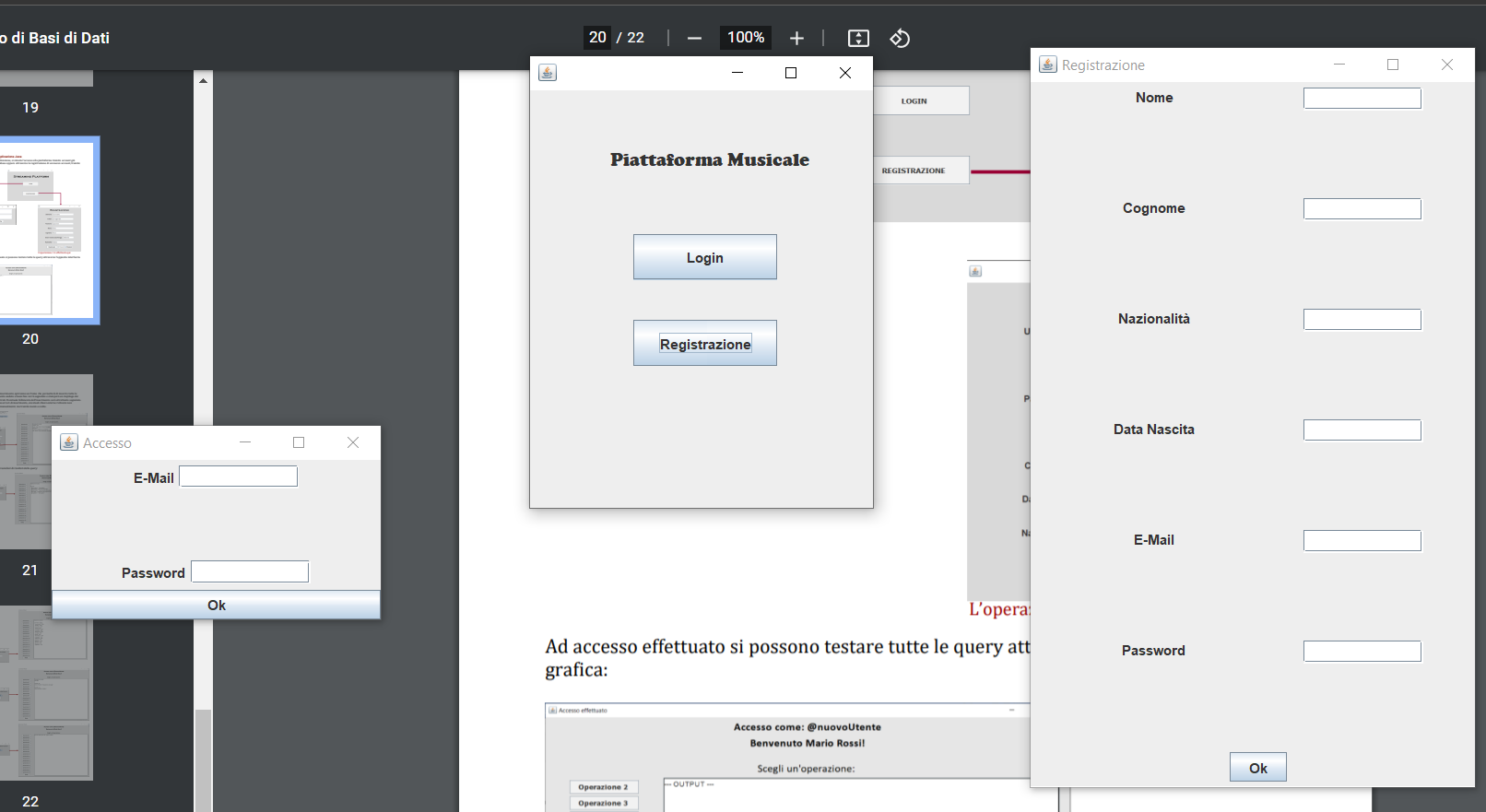
C.nome=H.nome AND

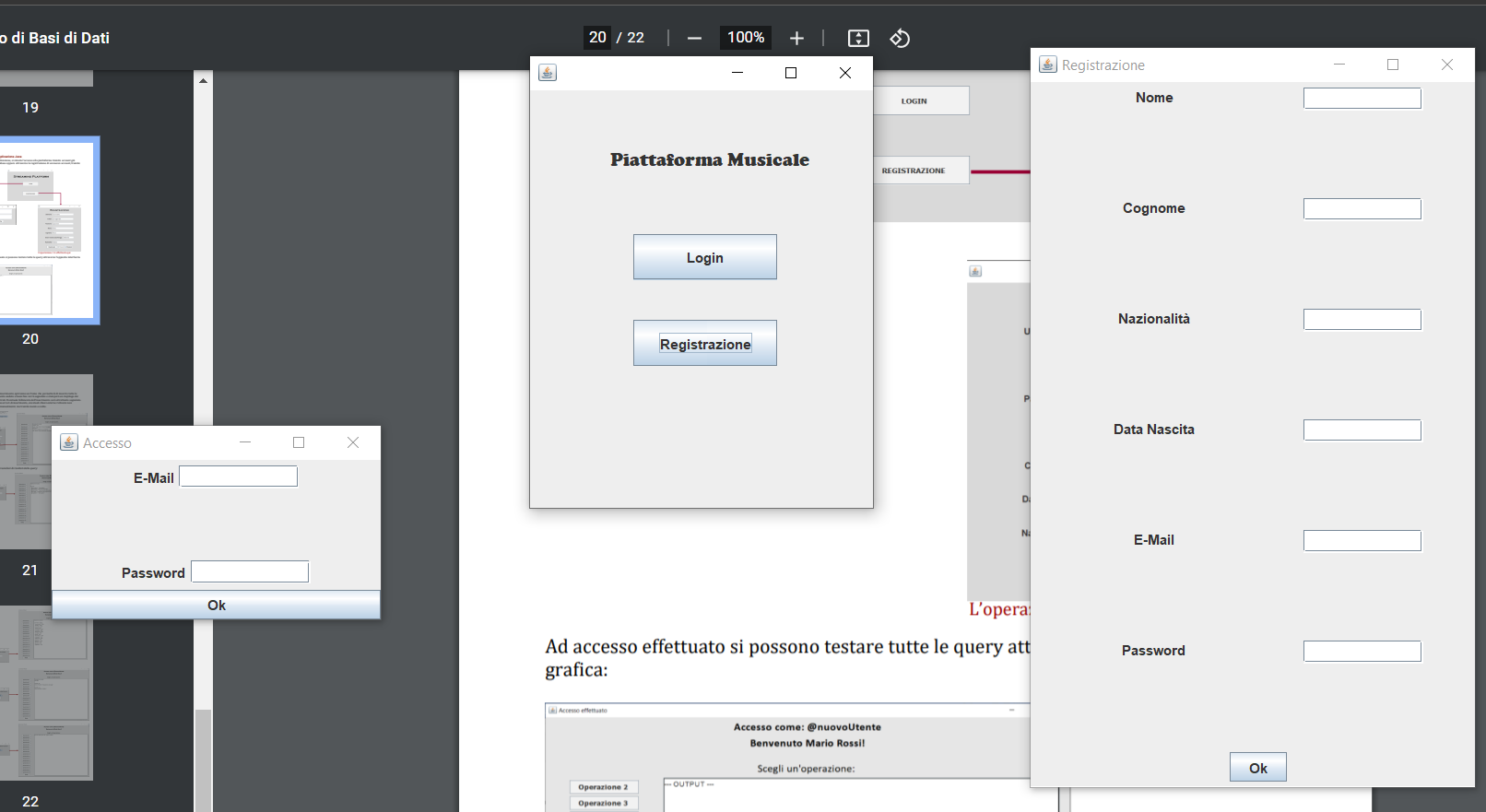
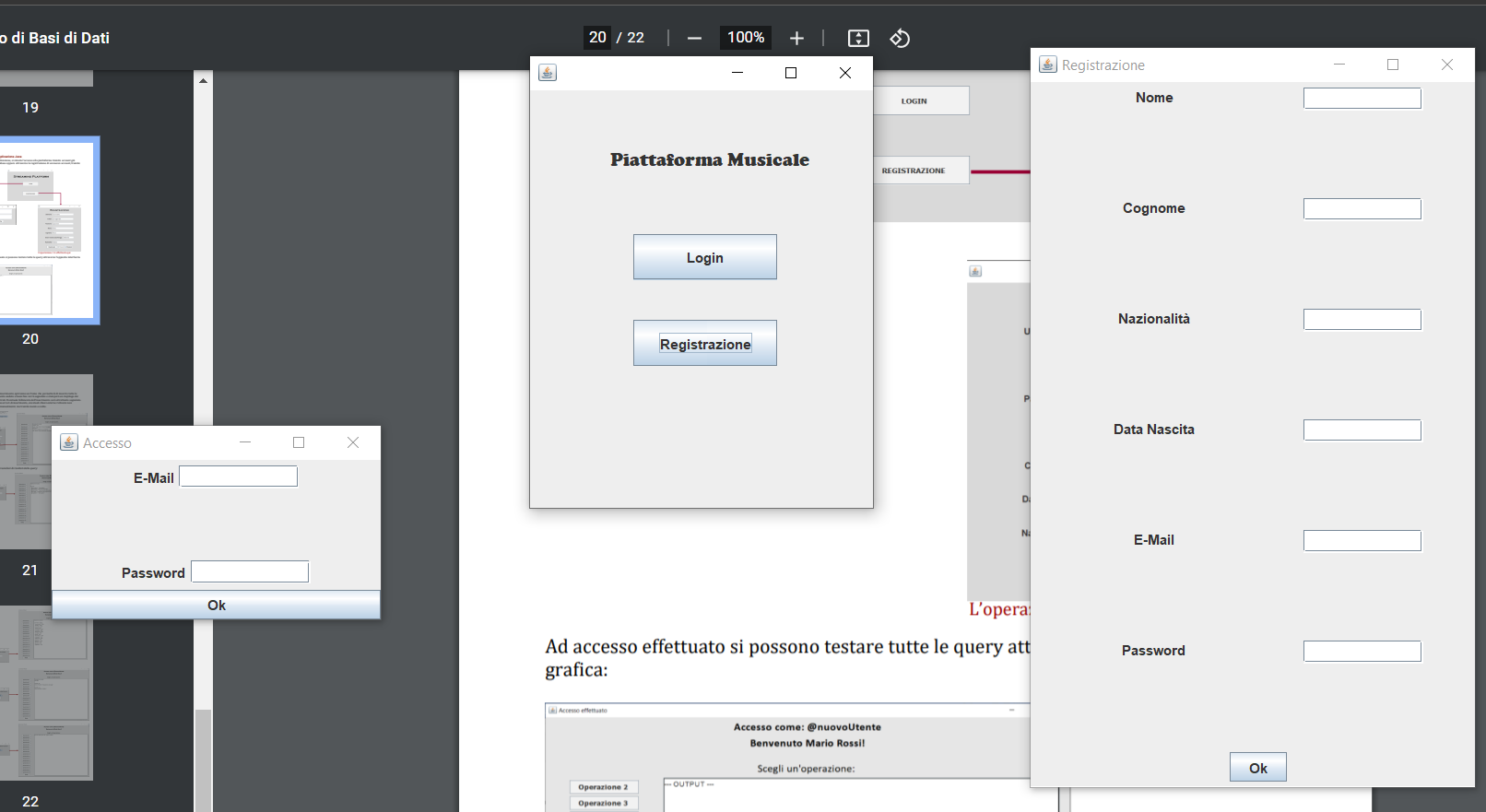
C.codice=H.codice AND

P.codice\_P=H.codice\_P AND

C.NSFW=1)

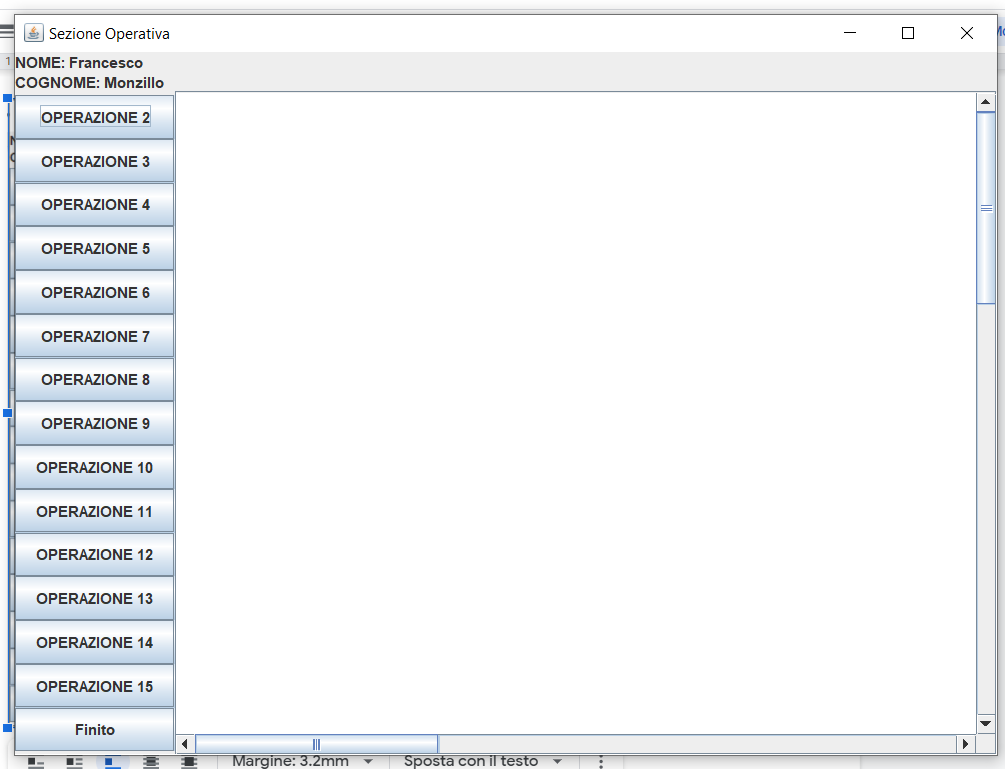
**7. Test dell’applicazione JAVA**

All’avvio dell’applicazione, si simula l’accesso alla piattaforma tramite account già esistente nel database oppure attraverso la registrazione di un nuovo account, tramite i seguenti frame:

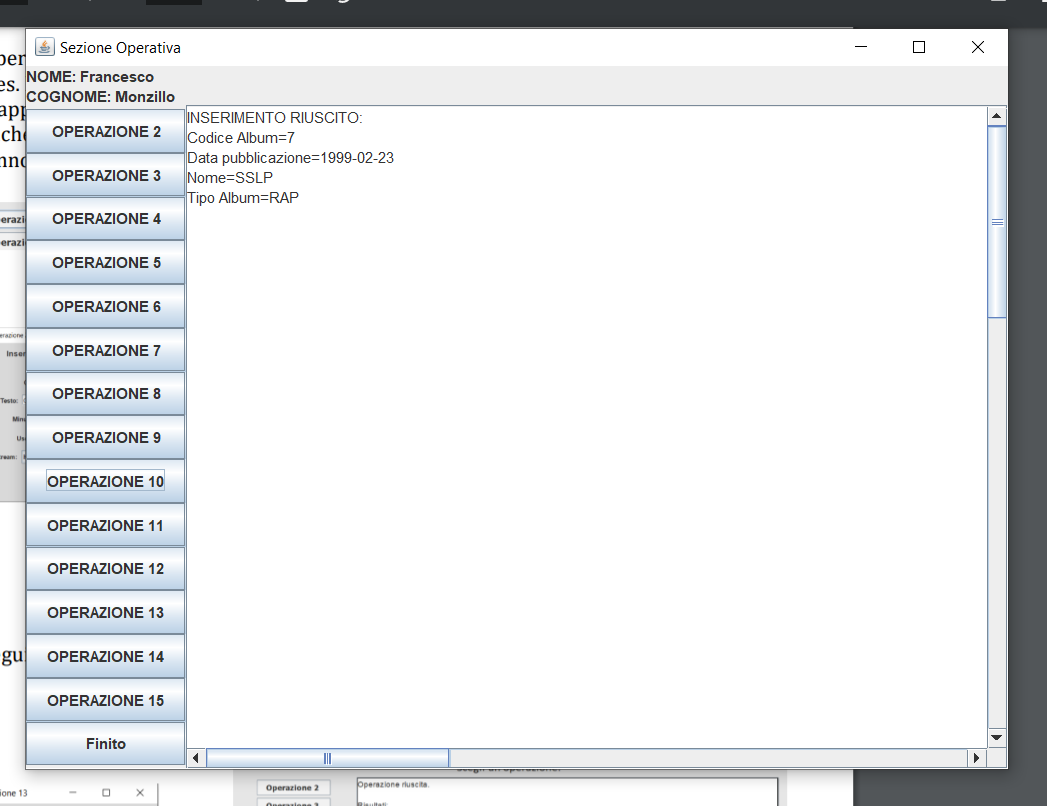
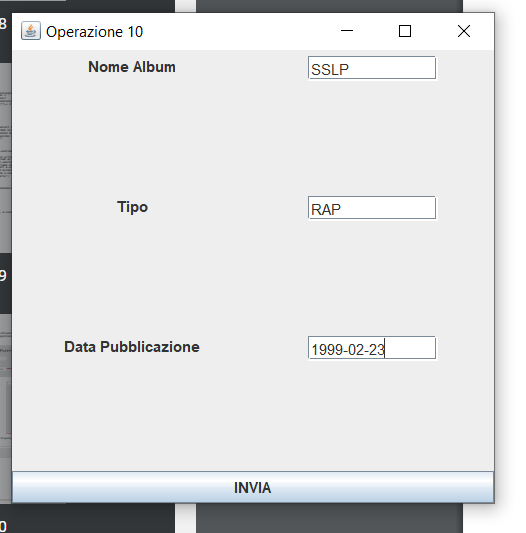


**L’operazione 1 è effettuata qui**

Ad accesso effettuato è possibile testare tutte le query attraverso l’apposita interfaccia grafica:

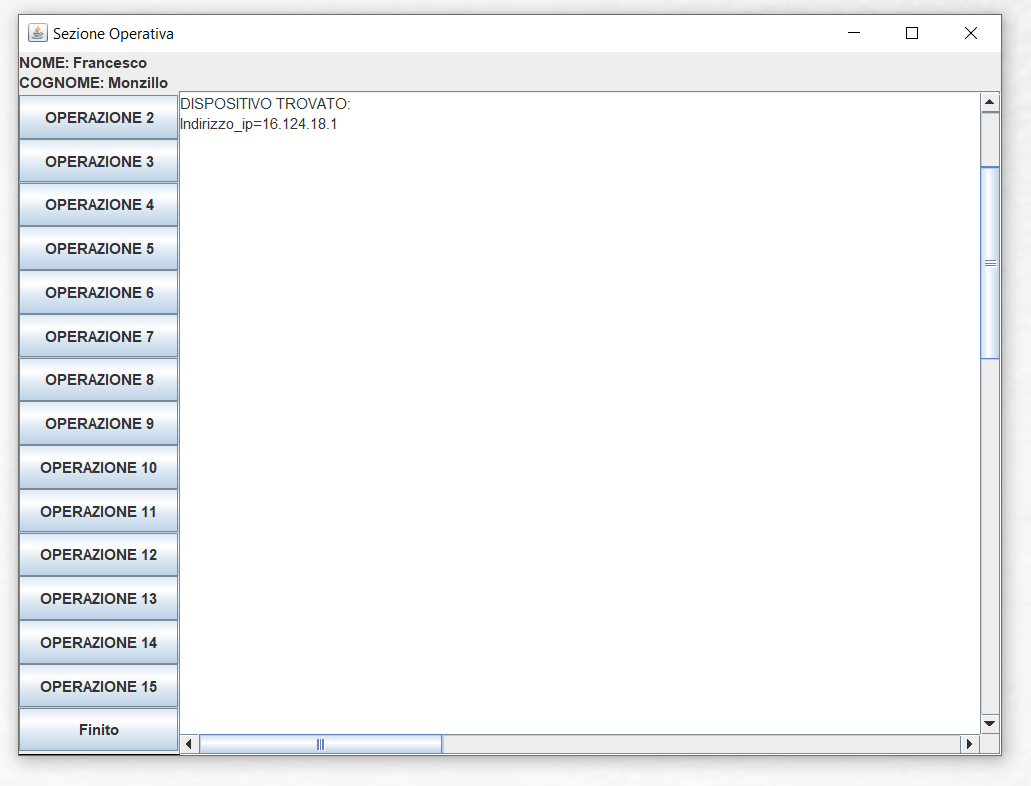
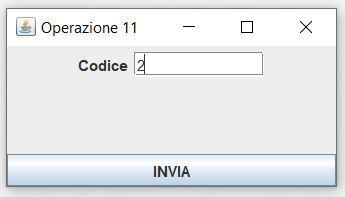


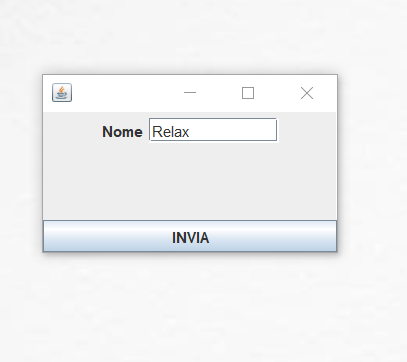
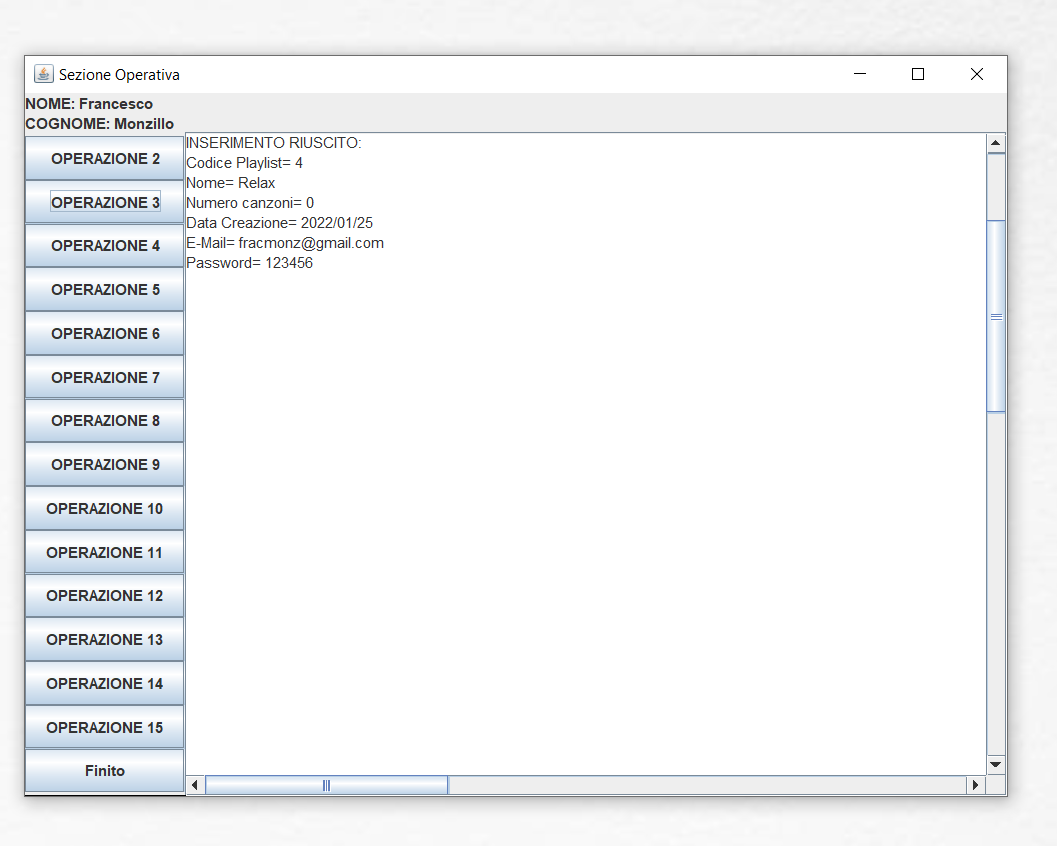
Le operazioni di inserimento apriranno un frame che permetterà di inserire tutte le values. L’inserimento andato a buon fine verrà segnalato e stamperà un riepilogo dei dati appena registrati. Eventuale fallimento dell’inserimento sarà altrettanto segnalato.



Operazione 10

Di seguito altri screenshot di varie operazioni:

Operazione 7



Operazione 3

**AUTORI**

Francesco Monzillo - 0512110536

Jessica Zampetti - 0512109894