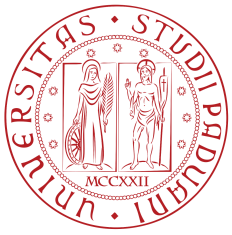
Progetto di Basi di Dati
SoundExp

Ivan Antonino Arena 2000546
Filippo Bomben 2008461

A.A. 2021/2022



Indice

1 Abstract

SoundExp è un servizio di streaming di contenuti audio musicali e d'intrattenimento creati da terzi e fruibili da ogni utente, con restrizioni in base al tipo di abbonamento. Per abbonarsi, l'utente può utilizzare il metodo di pagamento che preferisce tra carta e metodi digitali (PayPal, Google Pay e Apple Pay), o utilizzarli entrambi. I creatori di contenuti si dividono in musicisti e podcaster, rispettivamente al tipo di prodotto che offrono. Essi sono retribuiti mensilmente dalla compagnia in base al numero di ascolti generati.
Ogni utente ha la possibilità di salvare i contenuti che più preferisce, siano essi brani o podcast. L'utente inoltre può decidere di creare delle playlist oppure di ascoltare quelle create dagli altri utenti.
Sono disponibili tre tipi di abbonamento che offrono costi e servizi differenti:

- **Music:** permette accesso illimitato alla libreria di brani musicali;
- **Podcast:** permette accesso illimitato alla libreria di podcast;
- **Full:** permette accesso illimitato ad entrambe le librerie;

La fatturazione del servizio può avere cadenza annuale o mensile, secondo la preferenza dell'utente.

2 Analisi requisiti

2.1 Descrizione testuale dei requisiti

Nella base di dati sono presenti gli **Utenti** registrati alla piattaforma, di cui sono noti:

- Nome
- Cognome
- Indirizzo e-mail
- Password
- Abbonamento
- Username
- Frequenza addebito
- Scadenza abbonamento
- Brani preferiti
- Episodi preferiti
- Playlist personali

Sono disponibili quattro **Metodi di pagamento**: Carta e Digitali di cui sono noti:

- Numero carta
- Circuito
- Scadenza
- CCV
- Intestatario

Pagamento Digitale tra cui Paypal, Google Pay, Apple Pay, di cui sono noti:

- Indirizzo e-mail

- Password

Sono, inoltre presenti gli **Artisti** che collaborano con la piattaforma, che possono essere **Musicisti** o **Podcaster**, di cui sono noti:

- Nome
- Indirizzo e-mail
- Password
- BIC
- IBAN
- Indirizzo di fatturazione
- Ascoltatori mensili

La piattaforma mette a disposizione degli utenti il materiale fornito dagli artisti, **Brani**, di cui sono noti:

- Titolo
- Artista
- Album
- Numero traccia
- Durata
- Anno di uscita
- Genere
- Numero di riproduzioni

O **Episodi** di cui sono noti:

- Titolo
- Podcaster
- Podcast
- Numero episodio
- Durata
- Anno di uscita
- Genere
- Numero di riproduzioni

Gli utenti possono acquistare tre diversi tipi di **Abbonamento**: Music, Podcast e Full. Per ogni abbonamento sono noti:

- ID
- Nome
- Prezzo mensile
- Prezzo annuale

SoundExp offre delle **Playlist** di brani, che possono anche essere create dagli utenti e di cui sono noti:

- Titolo
- Autore
- Data di creazione

Ogni utente ha la possibilità di salvare i brani o gli episodi che più ama rispettivamente nei propri **Brani preferiti**, di cui sono noti:

- Titolo
- Autore

ed **Episodi preferiti** di cui sono noti:

- Titolo
- Podcaster

Ogni artista viene retribuito da SoundExp in base agli ascoltatori mensili. Si tiene traccia dei **Pagamenti** effettuati, di cui sono noti:

- ID Transazione
- Beneficiario
- IBAN
- Importo
- Data esecuzione

2.2 Operazioni tipiche

OPERAZIONE	TIPO	FREQUENZA
Ricerca di un brano o di un album	L	10000 volte al giorno
Ricerca di un episodio o di un podcast	L	2500 volte al giorno
Aggiornamento degli ascoltatori mensili di ogni artista	S	300 volte al giorno
Inserimento del pagamento di un artista	S	300 volte al mese
Creazione di una playlist	S	50 volte al mese
Classifica dei brani o episodi più ascoltati	L	50 volte all'anno
Controllare gli abbonamenti con scadenza imminente	L	1000 volte al giorno
Visualizzare il reddito annuale di un artista	L	300 volte all'anno
Controllare il numero di abbonati "Platinum"	L	30 volte al mese
Iscrizione di un nuovo utente	S	10 volte al giorno
Inserimento di un nuovo album o podcast	S	100 volte all'anno

3 Progettazione concettuale

3.1 Lista entità

- **Utenti**
  - Nome varchar(50) NOT NULL
  - Cognome varchar(50) NOT NULL
  - Username varchar(50) PRIMARY KEY
  - E-mail varchar(50) NOT NULL UNIQUE
  - Password varchar(12) NOT NULL
  - Abbonamento char(1) NOT NULL
  - Frequenza addebito char(1) NOT NULL
  - Scadenza abbonamento date NOT NULL
- **Metodi di Pagamento** (generalizzazione di **Carte** e **Digitali**)
- **Carte**
  - Numero carta varchar(19) PRIMARY KEY
  - Circuito varchar(20) NOT NULL
  - Scadenza date NOT NULL
  - CCV char(3) NOT NULL
  - Intestatario varchar(50) NOT NULL
- **Digitali** (generalizzazione di **PayPal**, **Google Pay** e **Apple Pay**)
  - E-mail varchar(50) PRIMARY KEY
  - Password varchar(16) NOT NULL
- **Artisti** (generalizzazione di **Musicisti** e **Podcaster**)
  - Nome varchar(50) NOT NULL UNIQUE
  - E-mail varchar(50) NOT NULL
  - Password varchar(16) NOT NULL
  - BIC char(11) NOT NULL
  - IBAN varchar(34) PRIMARY KEY
  - Indirizzo di fatturazione (attributo composto):
    - Stato char(2) NOT NULL
    - Città varchar(20) NOT NULL
    - CAP varchar(5) NOT NULL
    - Via varchar(50) NOT NULL
    - Numero civico varchar(5) NOT NULL
- **Brani**
  - Titolo varchar(50) PRIMARY KEY
  - Artista varchar(50) NOT NULL
  - Album varchar(50) NOT NULL
  - Traccia smallint NOT NULL, > 0

  - Durata varchar(8) NOT NULL
  - Anno di uscita smallint NOT NULL
  - Genere varchar(12) NOT NULL
  - Riproduzioni int NOT NULL, >= 0
- **Episodi**
  - Titolo varchar(50) PRIMARY KEY
  - Podcaster varchar(50) NOT NULL
  - Podcast varchar(50) NOT NULL
  - Numero episodi smallint NOT NULL, > 0
  - Durata varchar(8) NOT NULL
  - Anno di uscita smallint NOT NULL
  - Genere varchar(15) NOT NULL
  - Riproduzioni int può essere NULL, >= 0
- **Abbonamenti**
  - Nome varchar(8) NOT NULL
  - Prezzo mensile decimal(10,2) NOT NULL
  - Prezzo annuale decimal(10,2) NOT NULL
- **Pagamenti**
  - ID Transazione varchar(20) PRIMARY KEY
  - Beneficiario varchar(50) NOT NULL
  - IBAN varchar(34) NOT NULL
  - Importo decimal(10,2) NOT NULL
  - Data esecuzione date NOT NULL
- **Playlist**
  - Nome varcar(50) PRIMARY KEY
  - Creatore varchar(50) PRIMARY KEY
  - Data creazione date NOT NULL

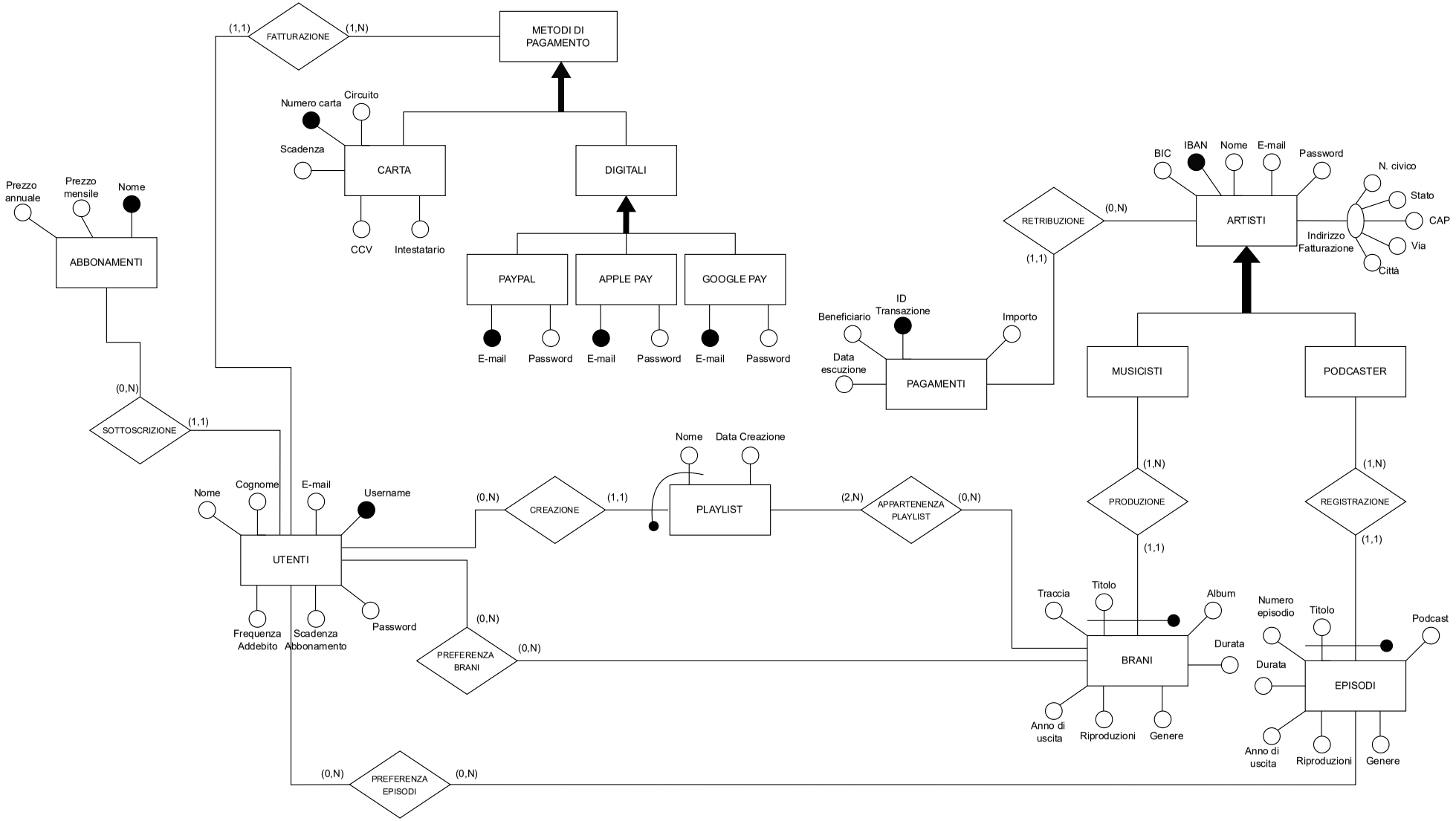
3.2 Tabella relazioni

Relazione	Entità coinvolte	Descrizione
Fatturazione	Utenti (1,1) Metodi di pagamento (1,N)	Ogni utente può utilizzare un metodo di pagamento digitale o con carta o entrambi.
Sottoscrizione	Utenti (1,1) Abbonamenti (0,N)	Ogni utente è sottoscritto ad un solo abbonamento. I tre tipi di abbonamenti possono essere tutti sottoscritti da qualunque numero di utenti
Retribuzione	Artisti (0,N) Pagamenti (1,1)	Un artista, se genera abbastanza ascolti, viene retribuito da SoundExp con un bonifico bancario univoco.
Produzione	Musicisti (1,N) Brani (1,1)	Ogni musicista deve pubblicare almeno un brano, che appartiene ad un solo musicista, il suo autore.
Registrazione	Podcaster (1,N) Episodi (1,1)	Ogni podcaster deve pubblicare almeno un episodio, che appartiene ad un solo podcaster, il suo autore.
Creazione	Utenti (0,N) Playlist (1,1)	Ogni utente può scegliere di creare una (o più) playlist. Ogni playlist appartiene all'utente che le ha create.
Appartenenza Playlist	Playlist (2,N) Brani (0,N)	Una playlist deve contenere almeno due brani. Un brano può appartenere a più playlist o a nessuna.
Preferenza Brani	Utenti (0,N) Brani (0,N)	Ogni utente può salvare tra i preferiti un qualunque numero di brani.
Preferenza Episodi	Utenti (0,N) Episodi (0,N)	Ogni utente può salvare tra i preferiti un qualunque numero di episodi.

3.3 Schema concettuale (E-R)

3.4 Vincoli non rappresentabili nello schema E-R:

- Se in Metodi di pagamento un utente ha *E-mail* NOT NULL allora *Numero Carta* può essere NULL e viceversa;
- Se un utente è titolare di un abbonamento di tipo Music non potrà salvare come preferiti episodi;
- Se un utente è titolare di un abbonamento di tipo Podcast non potrà salvare come preferiti brani musicali e non potrà creare o ascoltare **Playlist**;


(1,1)
FATTURAZIONE
(1,N)
METODI DI PAGAMENTO
Numero carta
Circuito
Scadenza
CARTA
DIGITALI
CCV
Intestatario
PAYPAL
APPLE PAY
GOOGLE PAY
E-mail
Password
E-mail
Password
E-mail
Password
Prezzo annuale
Prezzo mensile
Nome
ABBONAMENTI
(0,N)
SOTTOSCRIZIONE
(1,1)
Nome
Cognome
E-mail
Username
UTENTI
Frequenza Addebito
Scadenza Abbonamento
Password
(0,N)
CREAZIONE
(1,1)
Nome
Data Creazione
PLAYLIST
(2,N)
APPARTENENZA PLAYLIST
(0,N)
(0,N)
PREFERENZA BRANI
(0,N)
(0,N)
PREFERENZA EPISODI
(0,N)
RETRIBUZIONE
(1,1)
Beneficiario
ID Transazione
Importo
Data escuzione
PAGAMENTI
BIC
IBAN
Nome
E-mail
Password
ARTISTI
N. civico
Stato
CAP
Via
Città
Indirizzo Fatturazione
MUSICISTI
PODCASTER
(1,N)
PRODUZIONE
(1,1)
(1,N)
REGISTRAZIONE
(1,1)
Traccia
Titolo
Album
Durata
BRANI
Anno di uscita
Riproduzioni
Genere
Numero episodio
Titolo
Podcast
Durata
EPISODI
Anno di uscita
Riproduzioni
Genere

4 Progettazione logica

4.1 Ristrutturazione dello schema

4.1.1 Analisi delle ridondanze

L'attributo *Tipo* dell'entità **Artisti** è ridondante, in quanto la specializzazione dell'artista può essere dedotta dall'appartenenza o meno ad una delle due entità **Brani** e **Episodi**. Le operazioni che coinvolgono questo attributo sarebbero due:

- Contare il numero di musicisti (1 volta al giorno);
- Contare il numero di podcaster (1 volta al giorno);

Concetto	**Costrutto**	**Volumi**
Produzione	R	1000
Registrazione	R	250
Brani	E	1000
Episodi	E	250

Operazione 1 e 2 con *presenza* di ridondanza (le due operazioni agiscono sulla stessa entità):

Concetto	**Costrutto**	**Accessi**	**Tipo**
Artisti	E	1	L

Operazione 1 con *assenza* di ridondanza:

Concetto	**Costrutto**	**Accessi**	**Tipo**
Artisti	E	1	L
Produzione	R	1	L
Brani	E	1	L

Operazione 2 con *assenza* di ridondanza:

Concetto	**Costrutto**	**Accessi**	**Tipo**
Artisti	E	1	L
Registrazione	R	1	L
Episodi	E	1	L

Con assenza di ridondanza entrambe le operazioni effettuano ben tre accessi in lettura a tre differenti entità, contro l'unico accesso all'entità **Artisti** con presenza di ridondanza. Si conclude perciò che conviene mantenere la ridondanza, anche perché, nonostante la differenza nel numero di accessi, l'attributo *Tipo* è un solo carattere ed occupa quindi solamente 1 byte.

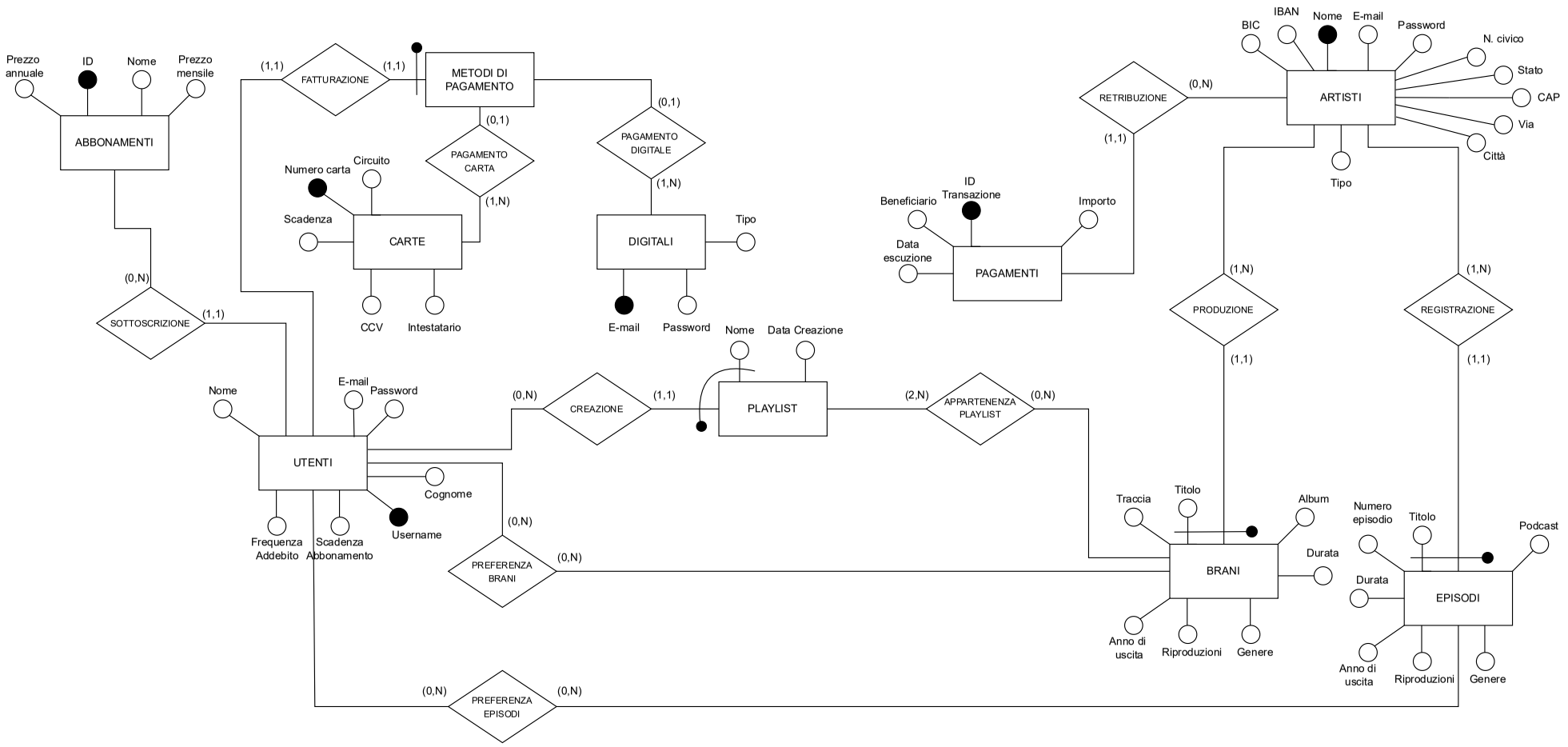
4.1.2 Eliminazione delle generalizzazioni

- La generalizzazione **Digitali** può essere eliminata poiché tutte le sue specializzazioni sono condividono gli stessi attributi, per cui le si può accorpare in un'unica entità, aggiungendo un attributo per distinguerne la tipologia.
- La generalizzazione **Metodi di pagamento** può essere eliminata poiché tutte le sue specializzazioni sono delle entità diverse, che necessitano di essere distinte tra loro. Vengono, quindi, aggiunti un attributo per identificare il metodo di pagamento digitale ed uno per identificare la carta.
- Le entità **Musicista** e **Podcaster**, specializzazioni dell'entità **Artista**, sono eliminabili in quanto forniscono solamente un'informazione aggiuntiva alla generalizzazione, facilmente sostituibile tramite un attributo.

4.1.3 Scelta degli identificatori primari

- Per risparmiare spazio, invece di identificare gli **Abbonamenti** tramite il loro nome, lo facciamo tramite un nuovo attributo *ID*.

4.2 Schema relazionale (ristrutturato)


Prezzo annuale
ID
Nome
Prezzo mensile
ABBONAMENTI
(0,N)
SOTTOSCRIZIONE
(1,1)
(1,1)
FATTURAZIONE
(1,1)
METODI DI PAGAMENTO
(0,1)
PAGAMENTO CARTA
(1,N)
Numero carta
Circuito
Scadenza
CARTE
CCV
Intestatario
(0,1)
PAGAMENTO DIGITALE
(1,N)
DIGITALI
Tipo
E-mail
Password
Beneficiario
ID Transazione
Importo
Data escuzione
PAGAMENTI
(1,1)
RETRIBUZIONE
(0,N)
BIC
IBAN
Nome
E-mail
Password
N. civico
Stato
CAP
Via
Città
ARTISTI
Tipo
(1,N)
PRODUZIONE
(1,1)
(1,N)
REGISTRAZIONE
(1,1)
Nome
E-mail
Password
UTENTI
Cognome
Username
Frequenza Addebito
Scadenza Abbonamento
(0,N)
CREAZIONE
(1,1)
Nome
Data Creazione
PLAYLIST
(2,N)
APPARTENENZA PLAYLIST
(0,N)
(0,N)
PREFERENZA BRANI
(0,N)
Traccia
Titolo
Album
Durata
BRANI
Anno di uscita
Riproduzioni
Genere
Numero episodio
Titolo
Podcast
Durata
EPISODI
Anno di uscita
Riproduzioni
Genere
(0,N)
PREFERENZA EPISODI
(0,N)

4.2.1 Descrizione dello schema e vincoli di integrità

- **Utenti**(Username, Nome, Cognome, Email, Password, FrequenzaAddebito, ScadenzaAbbonamento, Abbonamento)
  - Utenti.Abbonamento → Abbonamenti.ID
- **Abbonamenti**(ID, Nome, PrezzoAnnuale, PrezzoMensile)
- **MetodiDiPagamento**(Username, Email, NumeroCarta)
  - MetodiDiPagamento.Username → Utenti.Username
  - MetodiDiPagamento.NumeroCarta → Carte.NumeroCarta
  - MetodiDiPagamento.Email → Digitali.Email
- **Carte**(NumeroCarta, Circuito, Scadenza, CCV, Intestatario)
- **Digitali**(Email, Password, Tipo)
- **Artisti**(Nome, IBAN, BIC, Email, Password, Tipo, NCivico, Stato, CAP, Città, Via)
- **Brani**(Titolo, Artista, Album, Traccia, Durata, AnnoUscita, Riproduzioni, Genere)
  - Brani.Artista → Artisti.Nome
- **Episodi**(Titolo, Podcaster, Podcast, NumeroEpisodio, Durata, AnnoUscita, Riproduzioni, Genere)
  - Episodi.Podcaster → Artisti.Nome
- **Pagamenti**(IDTransazione, Beneficiario, IBAN, Importo, DataEsecuzione)
  - Pagamenti.IBAN → Artisti.IBAN
- **Playlist**(Nome, Creatore, DataCreazione)
  - Playlist.Creatore → Utenti.Username
- **AppartenenzaPlaylist**(Nome, Creatore, Brano, Artista)
  - AppartenenzaPlaylist.Nome → Playlist.Nome
  - AppartenenzaPlaylist.Creatore → Playlist.Creatore
  - AppartenenzaPlaylist.Titolo → Brani.Titolo
  - AppartenenzaPlaylist.Artista → Brani.Artista
- **PreferenezaBrani**(Titolo, Artista, Proprietario)
  - PreferenezaBrani.Proprietario → Utenti.Username
  - PreferenezaBrani.Titolo → Brani.Titolo
  - PreferenezaBrani.Artista → Brani.Artista
- **PreferenzaEpisodi**(Titolo, Podcaster, Proprietario)
  - PreferenzaEpisodi.Proprietario → Utenti.Username
  - PreferenzaEpisodi.Titolo → Episodi.Titolo
  - PreferenzaEpisodi.Podcaster → Episodi.Podcaster

5 Definizione delle Query e degli indici associate

5.1 Query

1. **Mostrare il nome di ogni artista e l'importo dell'ultimo bonifico effettuato a loro favore, con la rispettiva data di esecuzione, in ordine decrescente di importo.**

```
SELECT a.nome, p.importo, p.dataesecuzione FROM artisti AS a
JOIN pagamenti AS p ON a.iban = p.iban
WHERE p.dataesecuzione = (SELECT MAX(p.dataesecuzione) FROM pagamenti AS p)
ORDER BY p.importo DESC;
```

nome	importo	data
The Doors	195978.30	2021-05-01
Billie Eilish	168880.36	2021-05-01
N.W.A.	146551.57	2021-05-01
Sting	128313.85	2021-05-01
Green Day	127431.22	2021-05-01
One Direction	121584.25	2021-05-01
M83	110885.44	2021-05-01
Radiohead	108055.45	2021-05-01
Pink Floyd	106321.18	2021-05-01
Muschio Selvaggio	100128.61	2021-05-01
Travis Scott	94034.61	2021-05-01
Joe Rogan	89283.68	2021-05-01

2. **Mostrare tutti gli artisti con più di 5 milioni di ascolti totali.**

```
(SELECT a.nome as artista, SUM(b.riproduzioni) AS ascolti FROM artisti AS a
JOIN brani AS b ON a.nome = b.artista
GROUP BY a.nome HAVING SUM(b.riproduzioni) > 5000000)
UNION
(SELECT a.nome as artista, SUM(e.riproduzioni) AS ascolti FROM artisti AS a
JOIN episodi AS e ON a.nome = e.podcaster
GROUP BY a.nome HAVING SUM(e.riproduzioni) > 5000000);
```

artista	ascolti
Joe Rogan	425357185
Eminem	6321545
Paky	6309133
Radiohead	7054310
Pink Floyd	6723822
Dire Straits	7231970
The Police	7570184
Muschio Selvaggio	59925995
tha Supreme	5705316
Billie Eilish	5998105
One Direction	6118993
Oasis	6713780
Alessandro Barbero	7584299

3. **Mostrare username, nome e cognome di tutti gli utenti che pagano l'abbonamento con Google Pay e ordinarli per cognome.**

```
SELECT u.username, u.email, u.nome, u.cognome, d.tipo FROM utenti AS u
JOIN metodidipagamento AS m ON u.username = m.username
JOIN digitali AS d ON m.email = d.email
WHERE d.tipo = 'G'
ORDER BY u.cognome ASC;
```

username	email	nome	cognome	tipo
dalcido2g	dalcido2g@japanpost.jp	Deva	Alcido	G
pandinov	pandinov@npr.org	Pegeen	Andino	G
wannear14	wannear14@prweb.com	Wendie	Annear	G
mantognoni4	mantognoni4@de.vu	Maurizia	Antognoni	G
battfield0	battfield0@dailymotion.com	Ber	Attfield	G
cbingall2l	cbingall2l@ca.gov	Claudianus	Bingall	G
hbouldse	hbouldse@delicious.com	Hilary	Boulds	G
mbratch3	mbratch3@bluehost.com	Mela	Bratch	G
mbrazener1c	mbrazener1c@nps.gov	Millie	Brazener	G
rburburough17	rburburough17@paypal.com	Richardo	Burburough	G

4. **Mostrare il profitto totale per ogni tipo di abbonamento esistente.**

```
SELECT u1.abbonamento, SUM(introiti)
FROM ((SELECT u2.abbonamento, SUM(a.prezzoMensile) AS introiti FROM utenti AS u2
        JOIN abbonamenti AS a ON u2.abbonamento = a.id
        WHERE u2.frequenzaaddebito = 'M'
        GROUP BY u2.abbonamento)
    UNION
        (SELECT u3.abbonamento, SUM(a.prezzoAnnuale) AS introiti FROM utenti AS u3
        JOIN abbonamenti AS a ON u3.abbonamento = a.id
        WHERE u3.frequenzaaddebito = 'A'
        GROUP BY u3.abbonamento)
) AS r, utenti AS u1
GROUP BY u1.abbonamento;
```

abbonamento	sum
P	106764.00
M	75516.00
F	78120.00

5. **Mostrare username, nome, cognome ed e-mail di tutti gli utenti che hanno creato una playlist ed ordinarli per cognome.**

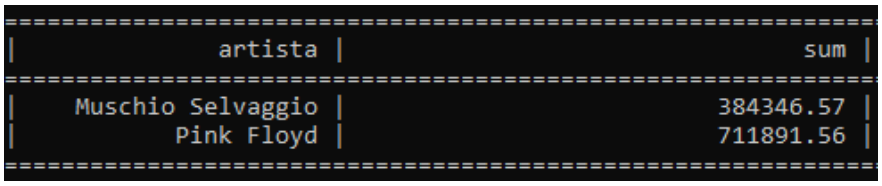
```
SELECT DISTINCT u.username, u.nome, u.cognome, u.email, p.nome
FROM utenti AS u
JOIN playlist AS p
    ON u.username = p.creatore
ORDER BY u.cognome ASC;
```

username	nome	cognome	email	nome
mantognoni4	Maurizia	Antognoni	mantognoni4@de.vu	Chilling
battfield0	Ber	Attfield	battfield0@dailymotion.com	Bass & Love
bbaldam2	Burgess	Baldam	bbaldam2@about.me	Sopa De Macaco
mbratch3	Mela	Bratch	mbratch3@bluehost.com	RaplsLife
lglawsop1	Latrina	Glawsop	lglawsop1@paginegialle.it	AmiciInVacanza

6. **Mostrare il musicista con almeno 10 brani prodotti e il podcaster con almeno 10 episodi registrati più pagati di sempre**.

```
(SELECT m.artista, SUM(p.importo)
FROM (SELECT b.artista
    FROM brani AS b
    GROUP BY b.artista
        HAVING COUNT(b.titolo) >= 10) AS m
JOIN artisti AS a
    ON a.nome = m.artista
JOIN pagamenti AS p
    ON p.iban = a.iban
```

```
GROUP BY m.artista
ORDER BY SUM(p.importo) DESC
LIMIT 1)
UNION
(SELECT pc.artista, SUM(p.importo)
FROM (SELECT e.podcaster AS artista
    FROM episodi AS e
    GROUP BY e.podcaster
        HAVING COUNT(e.titolo) >= 10) AS pc
JOIN artisti AS a
    ON a.nome = pc.artista
JOIN pagamenti AS p
    ON p.iban = a.iban
GROUP BY pc.artista
ORDER BY SUM(p.importo) DESC
LIMIT 1);
```



artista	sum
Muschio Selvaggio	384346.57
Pink Floyd	711891.56

5.2 Indici

La ricerca di uno specifico album è una query molto frequente, mentre l'inserimento di un album avviene solo poche volte all'anno (difficilmente un artista realizza più di un album ogni anno, solitamente accade ogni due o tre anni almeno) perciò risulta ragionevole, ipotizzando un utilizzo su larga scala, aggiungere un indice sull'attributo *album* della relazione **Brani**.

```
CREATE INDEX albums ON brani(album);
```

Per motivi analoghi, si indicizza anche l'attributo *podcast* della relazione **Episodi**.

```
CREATE INDEX podcasts ON episodi(podcast);
```

6 Documentazione codice C++

Compilare utilizzando il comando `g++ -o codice codice.c -L dependencies/lib -lpq` (illustrato nel file `compile.txt`) ed eseguire il file compilato `codice.exe`. Prima della compilazione è necessario sostituire nelle variabili globali `PG_HOST, PG_USER, PG_DB, PG_PASS, PG_PORT` rispettivamente l'indirizzo dell'host, il nome utente di PostgreSQL, il nome del database, la password e la porta utilizzata. Il programma utilizza la funzione `PQconnectdb` per connettersi al database, la funzione `PQstatus` per verificare lo stato della connessione, la funzione `PQfinish` per terminare la connessione. La funzione `void printResults(PGconn *conn, const char *query)` stampa i risultati di una data query, ricavati tramite `PQgetvalue`, dopo che questi sono stati verificati dalla funzione `void checkResults(PGresult *res, const PGconn *conn)`.

Il programma, quando eseguito, stampa a schermo tutte le query e, successivamente, richiede all'utente come input un numero facente riferimento alla query che egli vuole eseguire. Per la terza query è previsto in input anche un parametro, che permette all'utente di visualizzare gli abbonati che pagano con il mezzo di pagamento richiesto.