PODCAST-STREAMING APPLICATION DATABASE

DAMG 6210 – FINAL PROJECT
GROUP 4

Kruthika Kolume	002786400
Pramod Begur Nagaraj	002708842
Shravya Gunda	002197043
Swetha Paturu	002747560
Vidit Vinay Jain	002965764

DESCRIPTION

- Podcast-streaming applications.
- The database stores information about podcasts and user preferences such as searches, ratings, and playlists.
- This allows users to search, listen, download, rate, and create personal playlists and favorite lists.
- The database is designed for an efficient user experience and secure data storage.

OBJECTIVES

Persist information related to podcasts, such as Speakers, Ratings, Genres, Collections, and Subtitles

Manage user information and preferences, such as Playlists, Favorites, and Downloads.

The database should provide a user-friendly interface that allows users to easily search and browse for podcasts.

The database should be secure and protect user data like login credentials.

BUSINESS RULES

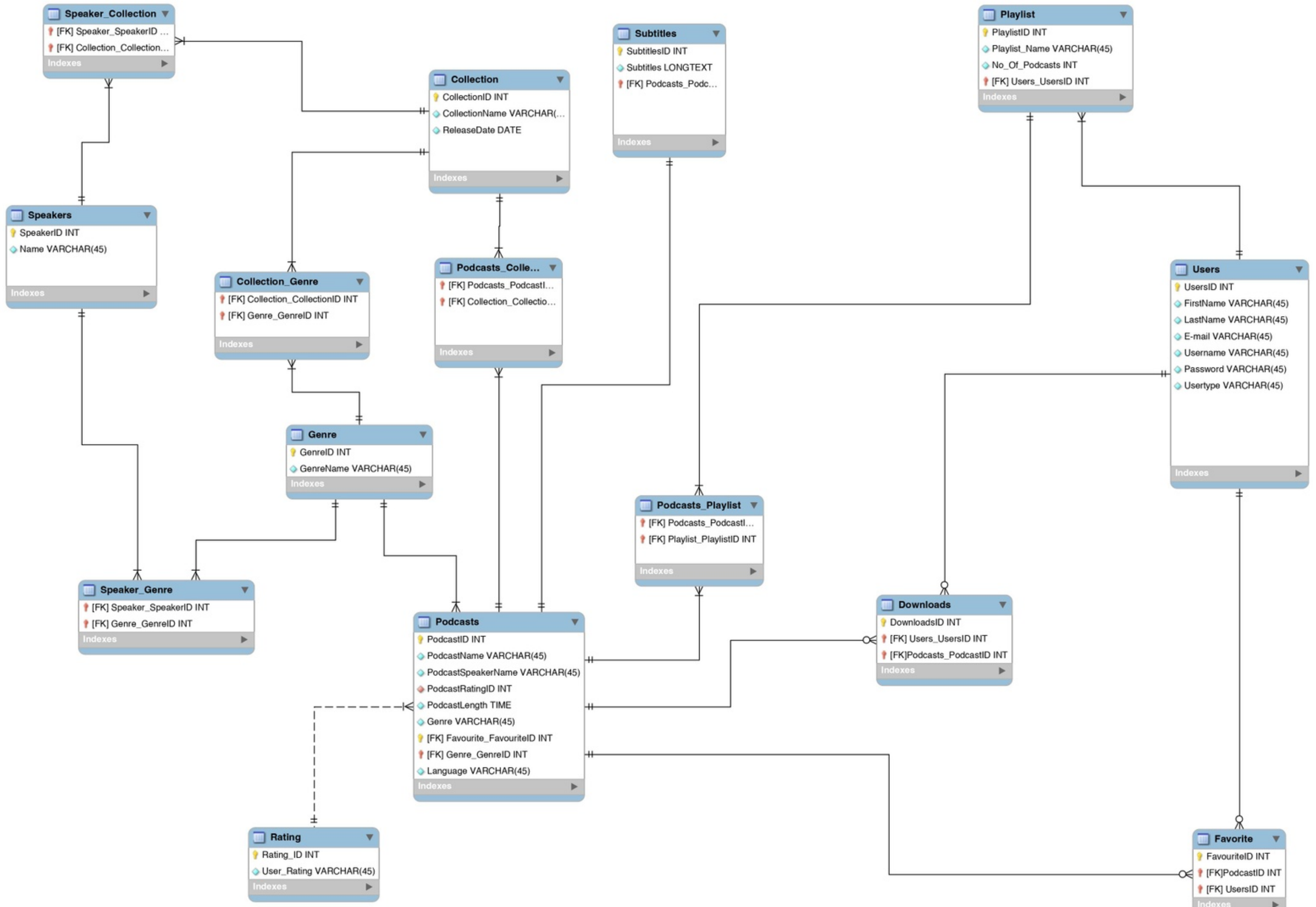
The Podcast App offers a comprehensive listening experience for users who are looking to discover and enjoy a wide variety of podcasts.

1. **Playlists:** Users can create their own playlists, which are private and personalized lists of podcasts.
2. **Favorites:** The "Favorites" feature is a many-to-one relationship that enables users to mark their favorite podcasts with a "heart."
3. **Downloads:** This feature allows users to download their favorite podcasts, enabling them to access the content offline at any time.
4. **Podcast Management:** With the ability to download, add to playlists, and add to favorite lists, the Podcast App provides a comprehensive and easy-to-use interface
5. **User Registration and Login:** To use the app, users must first register and then log in.

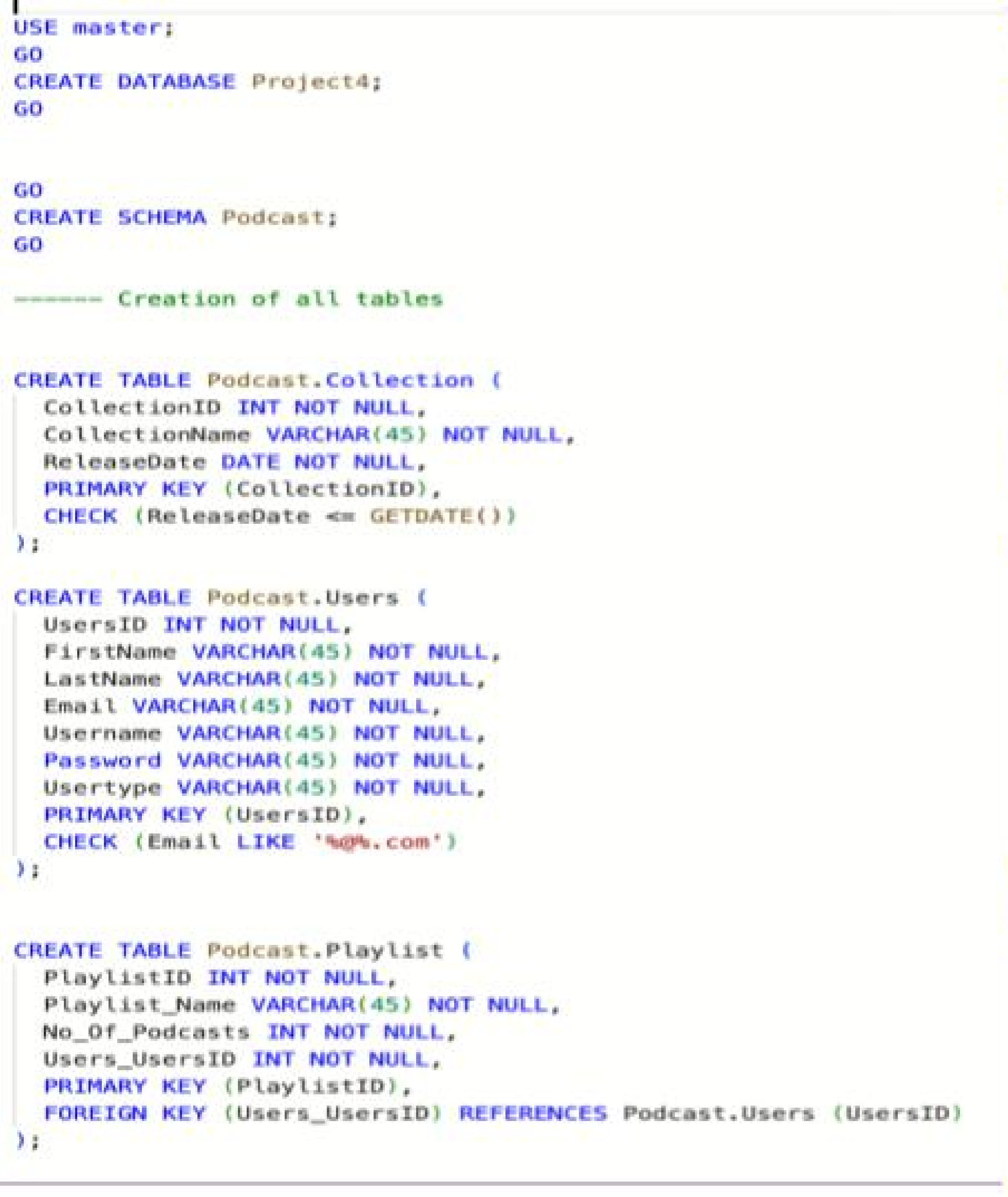
ENTITIES: 15

- Users
- Playlist
- Downloads
- Favorites
- Podcasts
- Podcast_Playlist
- Subtitles
- Genre
- Collection_Genre
- Collection
- Speaker_Genre
- Podcast_Collection
- Speaker_Collection
- Speakers
- Ratings

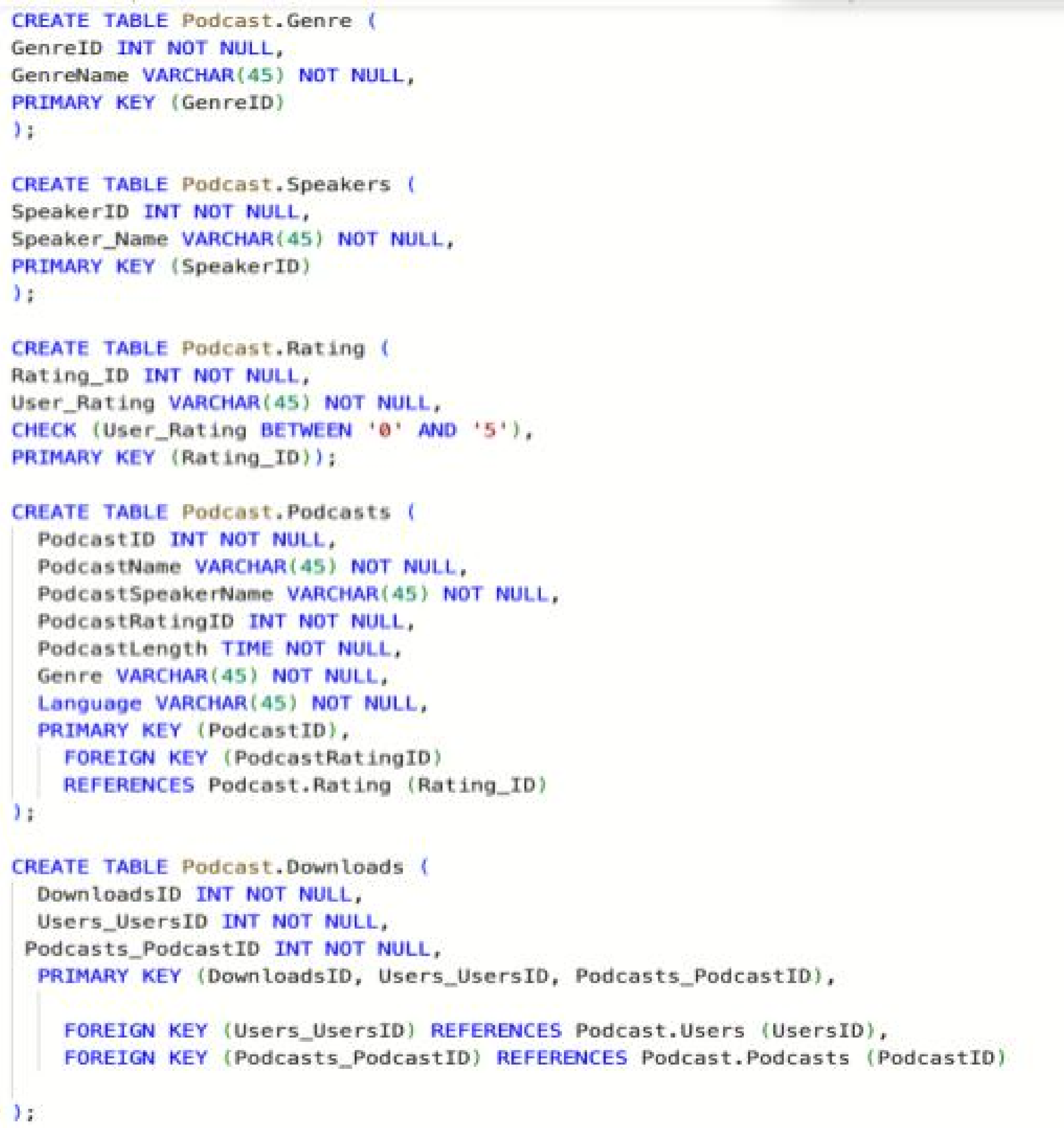
ERD


Speaker_Collection
[FK] Speaker_SpeakerID …
[FK] Collection_Collection…
Indexes
Collection
CollectionID INT
CollectionName VARCHAR(…
ReleaseDate DATE
Indexes
Subtitles
SubtitlesID INT
Subtitles LONGTEXT
[FK] Podcasts_Podc…
Indexes
Playlist
PlaylistID INT
Playlist_Name VARCHAR(45)
No_Of_Podcasts INT
[FK] Users_UsersID INT
Indexes
Speakers
SpeakerID INT
Name VARCHAR(45)
Indexes
Collection_Genre
[FK] Collection_CollectionID INT
[FK] Genre_GenreID INT
Indexes
Podcasts_Colle…
[FK] Podcasts_PodcastI…
[FK] Collection_Collectio…
Indexes
Users
UsersID INT
FirstName VARCHAR(45)
LastName VARCHAR(45)
E-mail VARCHAR(45)
Username VARCHAR(45)
Password VARCHAR(45)
Usertype VARCHAR(45)
Indexes
Genre
GenreID INT
GenreName VARCHAR(45)
Indexes
Podcasts_Playlist
[FK] Podcasts_PodcastI…
[FK] Playlist_PlaylistID INT
Indexes
Speaker_Genre
[FK] Speaker_SpeakerID INT
[FK] Genre_GenreID INT
Indexes
Podcasts
PodcastID INT
PodcastName VARCHAR(45)
PodcastSpeakerName VARCHAR(45)
PodcastRatingID INT
PodcastLength TIME
Genre VARCHAR(45)
[FK] Favourite_FavouriteID INT
[FK] Genre_GenreID INT
Language VARCHAR(45)
Indexes
Downloads
DownloadsID INT
[FK] Users_UsersID INT
[FK]Podcasts_PodcastID INT
Indexes
Rating
Rating_ID INT
User_Rating VARCHAR(45)
Indexes
Favorite
FavouriteID INT
[FK]PodcastID INT
[FK] UsersID INT
Indexes

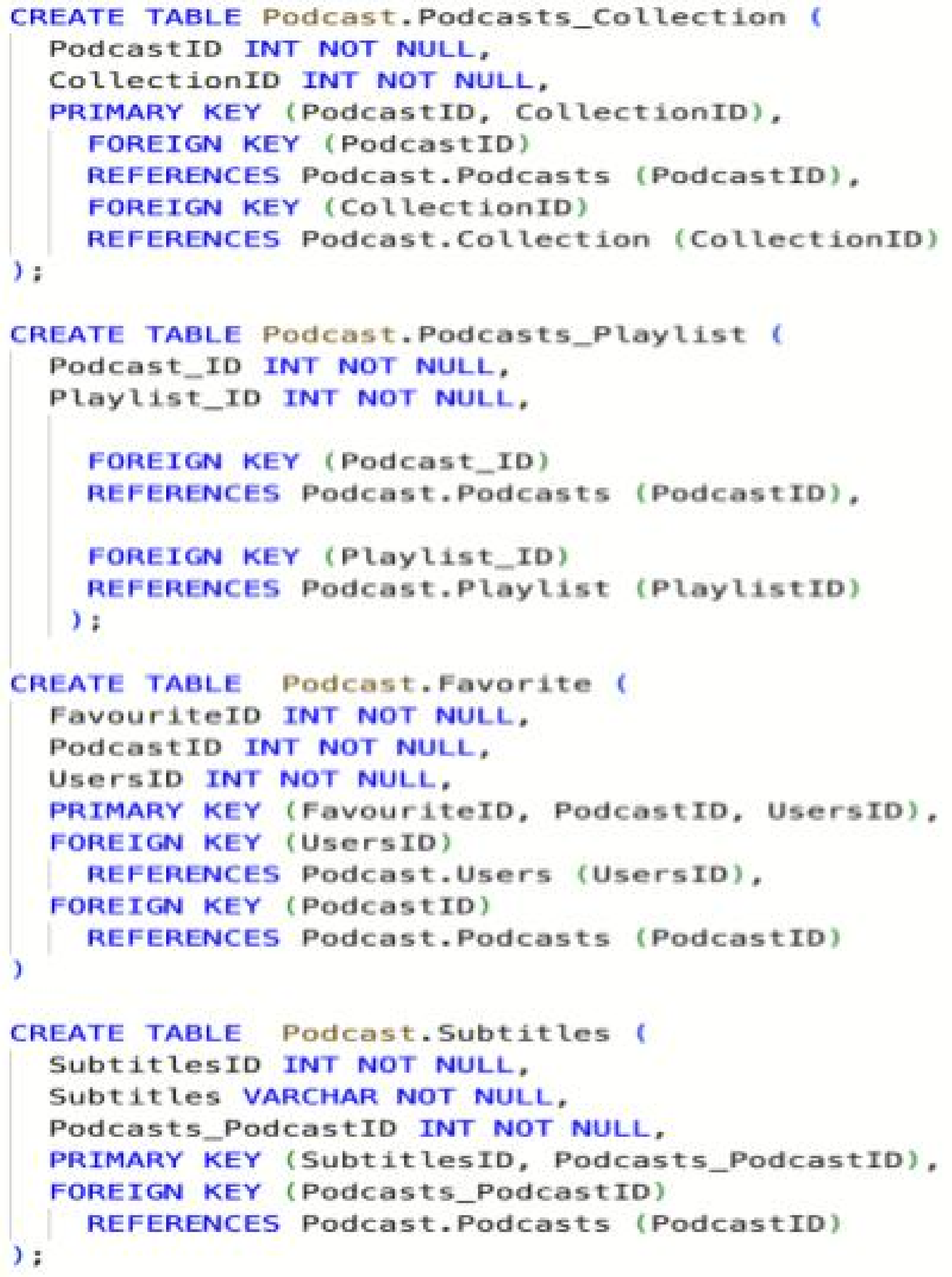
SQL DDL



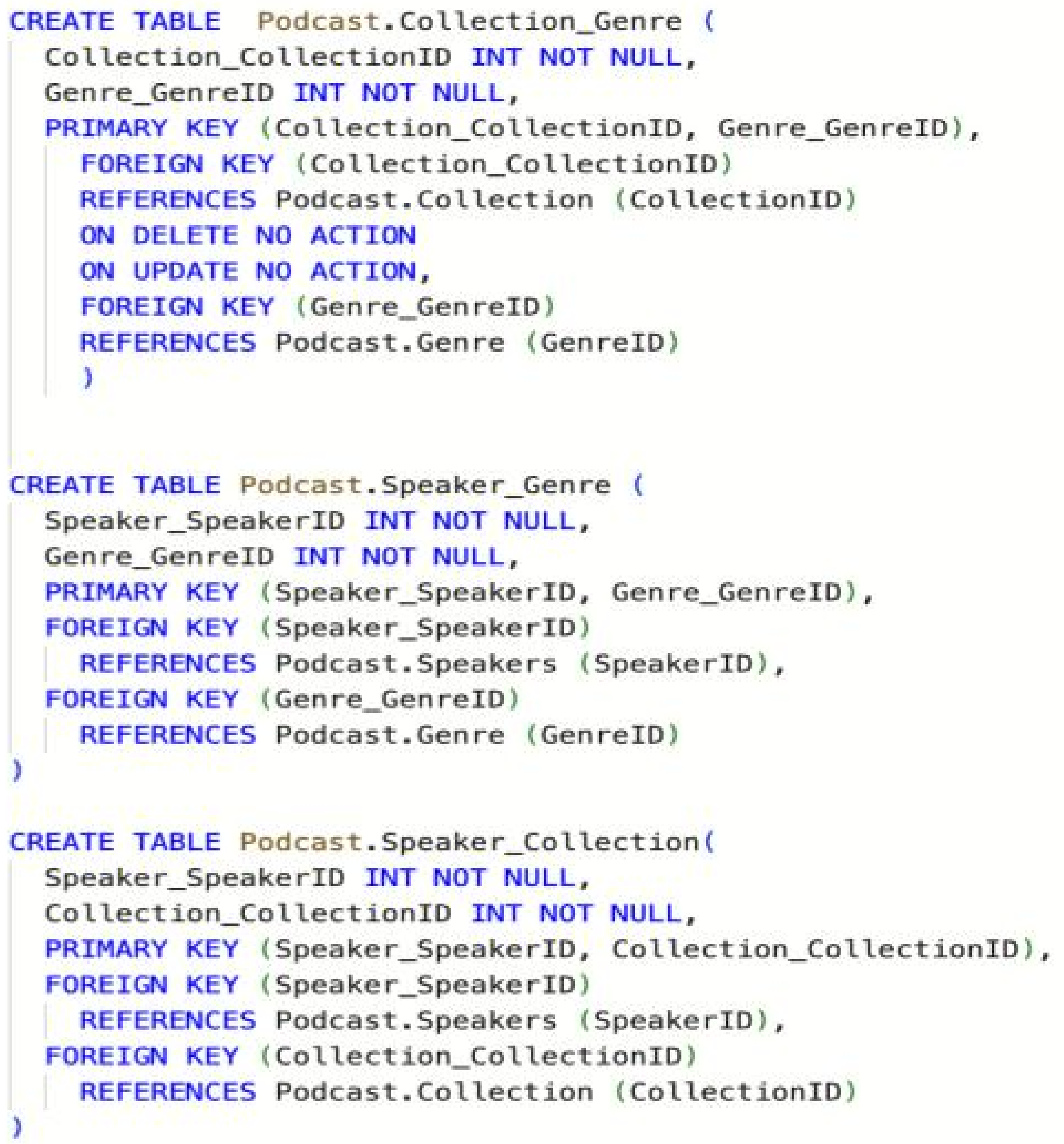
```sql
USE master;
GO
CREATE DATABASE Project4;
GO


GO
CREATE SCHEMA Podcast;
GO

------- Creation of all tables


CREATE TABLE Podcast.Collection (
  CollectionID INT NOT NULL,
  CollectionName VARCHAR(45) NOT NULL,
  ReleaseDate DATE NOT NULL,
  PRIMARY KEY (CollectionID),
  CHECK (ReleaseDate <= GETDATE())
);

CREATE TABLE Podcast.Users (
  UsersID INT NOT NULL,
  FirstName VARCHAR(45) NOT NULL,
  LastName VARCHAR(45) NOT NULL,
  Email VARCHAR(45) NOT NULL,
  Username VARCHAR(45) NOT NULL,
  Password VARCHAR(45) NOT NULL,
  Usertype VARCHAR(45) NOT NULL,
  PRIMARY KEY (UsersID),
  CHECK (Email LIKE '%@%.com')
);


CREATE TABLE Podcast.Playlist (
  PlaylistID INT NOT NULL,
  Playlist_Name VARCHAR(45) NOT NULL,
  No_Of_Podcasts INT NOT NULL,
  Users_UsersID INT NOT NULL,
  PRIMARY KEY (PlaylistID),
  FOREIGN KEY (Users_UsersID) REFERENCES Podcast.Users (UsersID)
);
```



```sql
CREATE TABLE Podcast.Genre (
GenreID INT NOT NULL,
GenreName VARCHAR(45) NOT NULL,
PRIMARY KEY (GenreID)
);

CREATE TABLE Podcast.Speakers (
SpeakerID INT NOT NULL,
Speaker_Name VARCHAR(45) NOT NULL,
PRIMARY KEY (SpeakerID)
);

CREATE TABLE Podcast.Rating (
Rating_ID INT NOT NULL,
User_Rating VARCHAR(45) NOT NULL,
CHECK (User_Rating BETWEEN '0' AND '5'),
PRIMARY KEY (Rating_ID));

CREATE TABLE Podcast.Podcasts (
  PodcastID INT NOT NULL,
  PodcastName VARCHAR(45) NOT NULL,
  PodcastSpeakerName VARCHAR(45) NOT NULL,
  PodcastRatingID INT NOT NULL,
  PodcastLength TIME NOT NULL,
  Genre VARCHAR(45) NOT NULL,
  Language VARCHAR(45) NOT NULL,
  PRIMARY KEY (PodcastID),
    FOREIGN KEY (PodcastRatingID)
    REFERENCES Podcast.Rating (Rating_ID)
);

CREATE TABLE Podcast.Downloads (
  DownloadsID INT NOT NULL,
  Users_UsersID INT NOT NULL,
 Podcasts_PodcastID INT NOT NULL,
  PRIMARY KEY (DownloadsID, Users_UsersID, Podcasts_PodcastID),

    FOREIGN KEY (Users_UsersID) REFERENCES Podcast.Users (UsersID),
    FOREIGN KEY (Podcasts_PodcastID) REFERENCES Podcast.Podcasts (PodcastID)

);
```

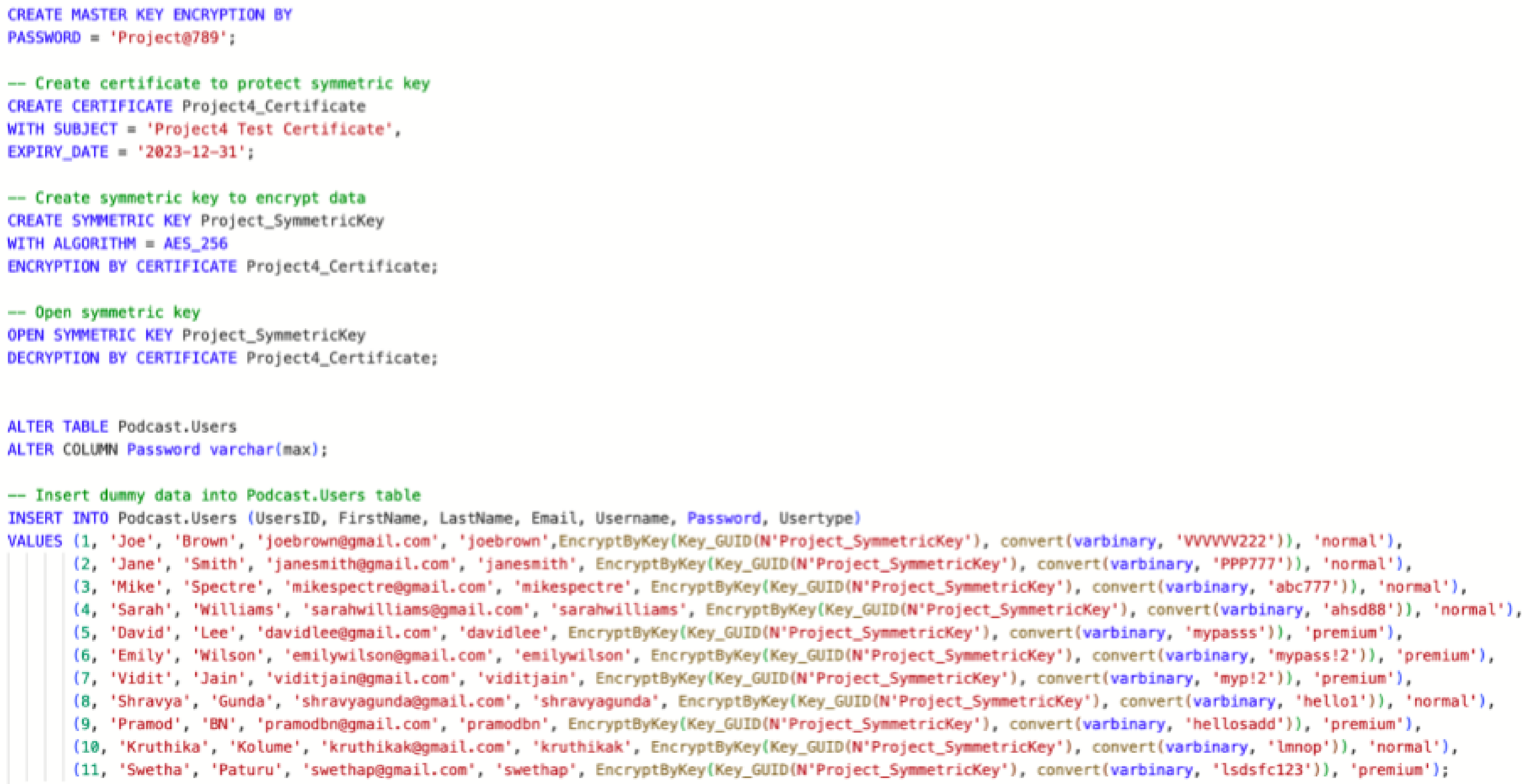


```sql
CREATE TABLE Podcast.Podcasts_Collection (
  PodcastID INT NOT NULL,
  CollectionID INT NOT NULL,
  PRIMARY KEY (PodcastID, CollectionID),
    FOREIGN KEY (PodcastID)
    REFERENCES Podcast.Podcasts (PodcastID),
    FOREIGN KEY (CollectionID)
    REFERENCES Podcast.Collection (CollectionID)
);

CREATE TABLE Podcast.Podcasts_Playlist (
  Podcast_ID INT NOT NULL,
  Playlist_ID INT NOT NULL,

    FOREIGN KEY (Podcast_ID)
    REFERENCES Podcast.Podcasts (PodcastID),

    FOREIGN KEY (Playlist_ID)
    REFERENCES Podcast.Playlist (PlaylistID)
   );
CREATE TABLE  Podcast.Favorite (
  FavouriteID INT NOT NULL,
  PodcastID INT NOT NULL,
  UsersID INT NOT NULL,
  PRIMARY KEY (FavouriteID, PodcastID, UsersID),
  FOREIGN KEY (UsersID)
    REFERENCES Podcast.Users (UsersID),
  FOREIGN KEY (PodcastID)
    REFERENCES Podcast.Podcasts (PodcastID)
)

CREATE TABLE  Podcast.Subtitles (
  SubtitlesID INT NOT NULL,
  Subtitles VARCHAR NOT NULL,
  Podcasts_PodcastID INT NOT NULL,
  PRIMARY KEY (SubtitlesID, Podcasts_PodcastID),
  FOREIGN KEY (Podcasts_PodcastID)
    REFERENCES Podcast.Podcasts (PodcastID)
);
```



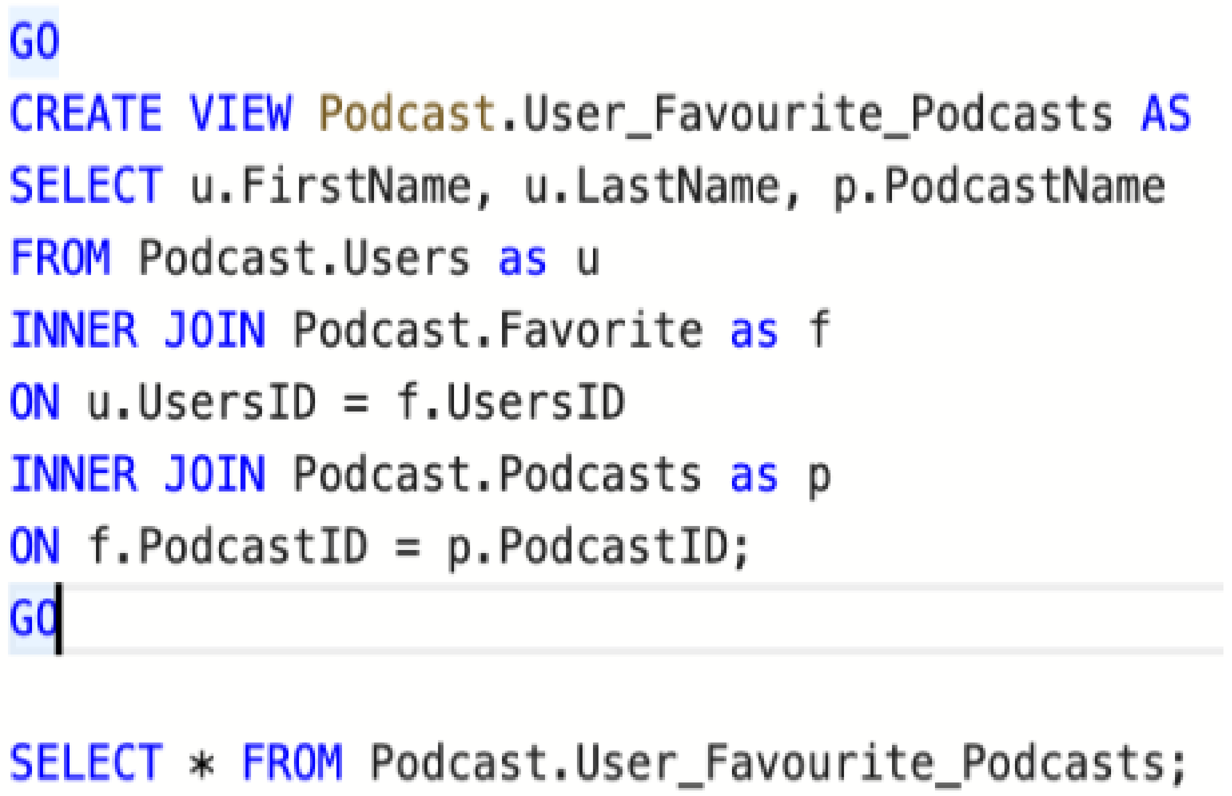
```sql
CREATE TABLE  Podcast.Collection_Genre (
  Collection_CollectionID INT NOT NULL,
  Genre_GenreID INT NOT NULL,
  PRIMARY KEY (Collection_CollectionID, Genre_GenreID),
    FOREIGN KEY (Collection_CollectionID)
    REFERENCES Podcast.Collection (CollectionID)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    FOREIGN KEY (Genre_GenreID)
    REFERENCES Podcast.Genre (GenreID)
    )


CREATE TABLE Podcast.Speaker_Genre (
  Speaker_SpeakerID INT NOT NULL,
  Genre_GenreID INT NOT NULL,
  PRIMARY KEY (Speaker_SpeakerID, Genre_GenreID),
  FOREIGN KEY (Speaker_SpeakerID)
    REFERENCES Podcast.Speakers (SpeakerID),
  FOREIGN KEY (Genre_GenreID)
    REFERENCES Podcast.Genre (GenreID)
)

CREATE TABLE Podcast.Speaker_Collection(
  Speaker_SpeakerID INT NOT NULL,
  Collection_CollectionID INT NOT NULL,
  PRIMARY KEY (Speaker_SpeakerID, Collection_CollectionID),
  FOREIGN KEY (Speaker_SpeakerID)
    REFERENCES Podcast.Speakers (SpeakerID),
  FOREIGN KEY (Collection_CollectionID)
    REFERENCES Podcast.Collection (CollectionID)
)
```

SQL ENCRYPTION

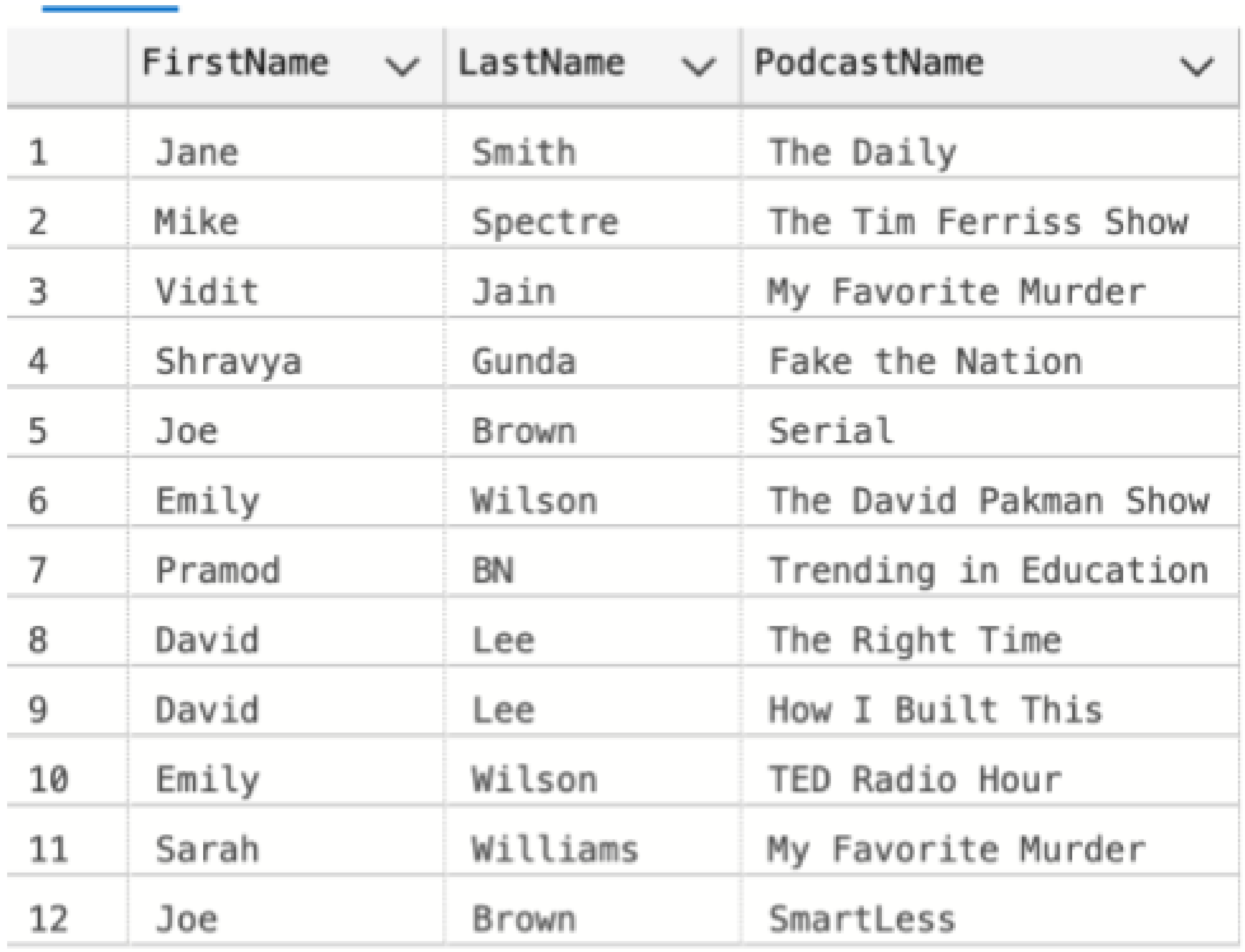


```sql
CREATE MASTER KEY ENCRYPTION BY
PASSWORD = 'Project@789';

-- Create certificate to protect symmetric key
CREATE CERTIFICATE Project4_Certificate
WITH SUBJECT = 'Project4 Test Certificate',
EXPIRY_DATE = '2023-12-31';

-- Create symmetric key to encrypt data
CREATE SYMMETRIC KEY Project_SymmetricKey
WITH ALGORITHM = AES_256
ENCRYPTION BY CERTIFICATE Project4_Certificate;

-- Open symmetric key
OPEN SYMMETRIC KEY Project_SymmetricKey
DECRYPTION BY CERTIFICATE Project4_Certificate;



ALTER TABLE Podcast.Users
ALTER COLUMN Password varchar(max);

-- Insert dummy data into Podcast.Users table
INSERT INTO Podcast.Users (UsersID, FirstName, LastName, Email, Username, Password, Usertype)
VALUES (1, 'Joe', 'Brown', 'joebrown@gmail.com', 'joebrown',EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'VVVVVV222')), 'normal'),
       (2, 'Jane', 'Smith', 'janesmith@gmail.com', 'janesmith', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'PPP777')), 'normal'),
       (3, 'Mike', 'Spectre', 'mikespectre@gmail.com', 'mikespectre', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'abc777')), 'normal'),
       (4, 'Sarah', 'Williams', 'sarahwilliams@gmail.com', 'sarahwilliams', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'ahsd88')), 'normal'),
       (5, 'David', 'Lee', 'davidlee@gmail.com', 'davidlee', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'mypasss')), 'premium'),
       (6, 'Emily', 'Wilson', 'emilywilson@gmail.com', 'emilywilson', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'mypass!2')), 'premium'),
       (7, 'Vidit', 'Jain', 'viditjain@gmail.com', 'viditjain', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'myp!2')), 'premium'),
       (8, 'Shravya', 'Gunda', 'shravyagunda@gmail.com', 'shravyagunda', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'hello1')), 'normal'),
       (9, 'Pramod', 'BN', 'pramodbn@gmail.com', 'pramodbn', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'hellosadd')), 'premium'),
       (10, 'Kruthika', 'Kolume', 'kruthikak@gmail.com', 'kruthikak', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'lmnop')), 'normal'),
       (11, 'Swetha', 'Paturu', 'swethap@gmail.com', 'swethap', EncryptByKey(Key_GUID(N'Project_SymmetricKey'), convert(varbinary, 'lsdsfc123')), 'premium');
```

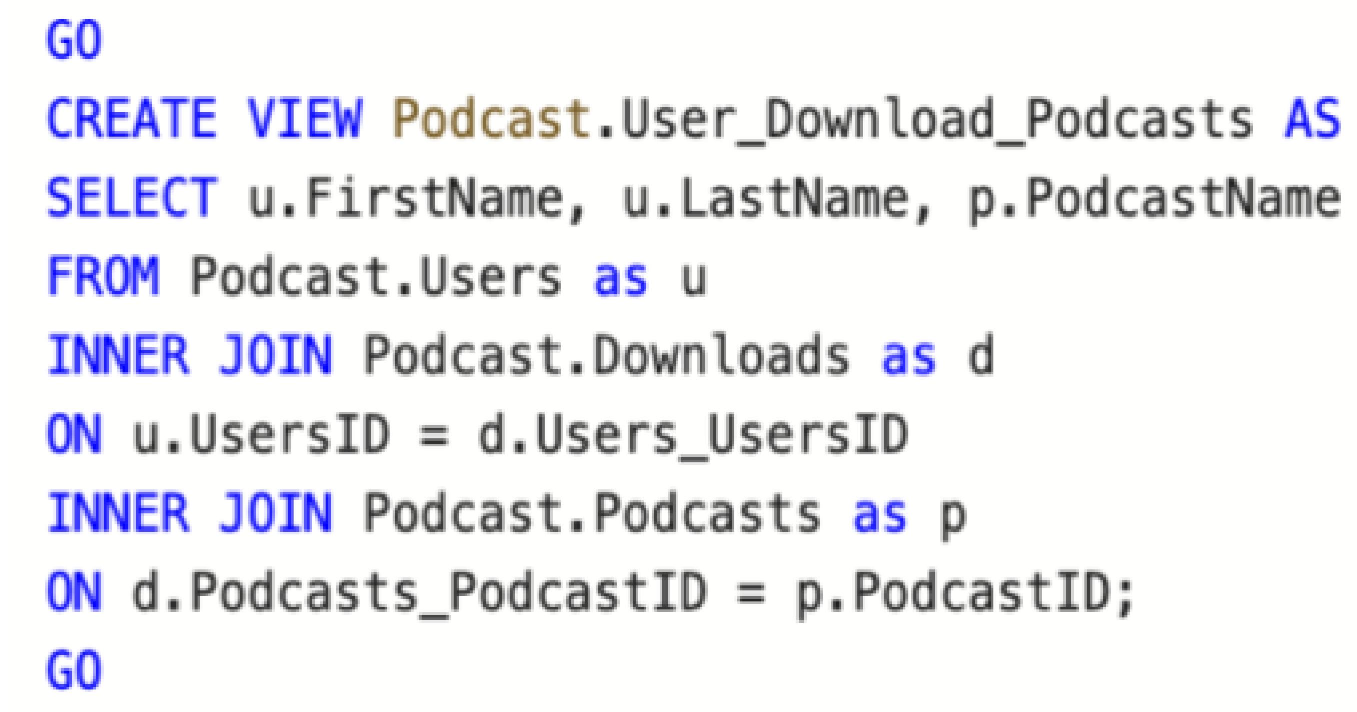
SQL VIEW



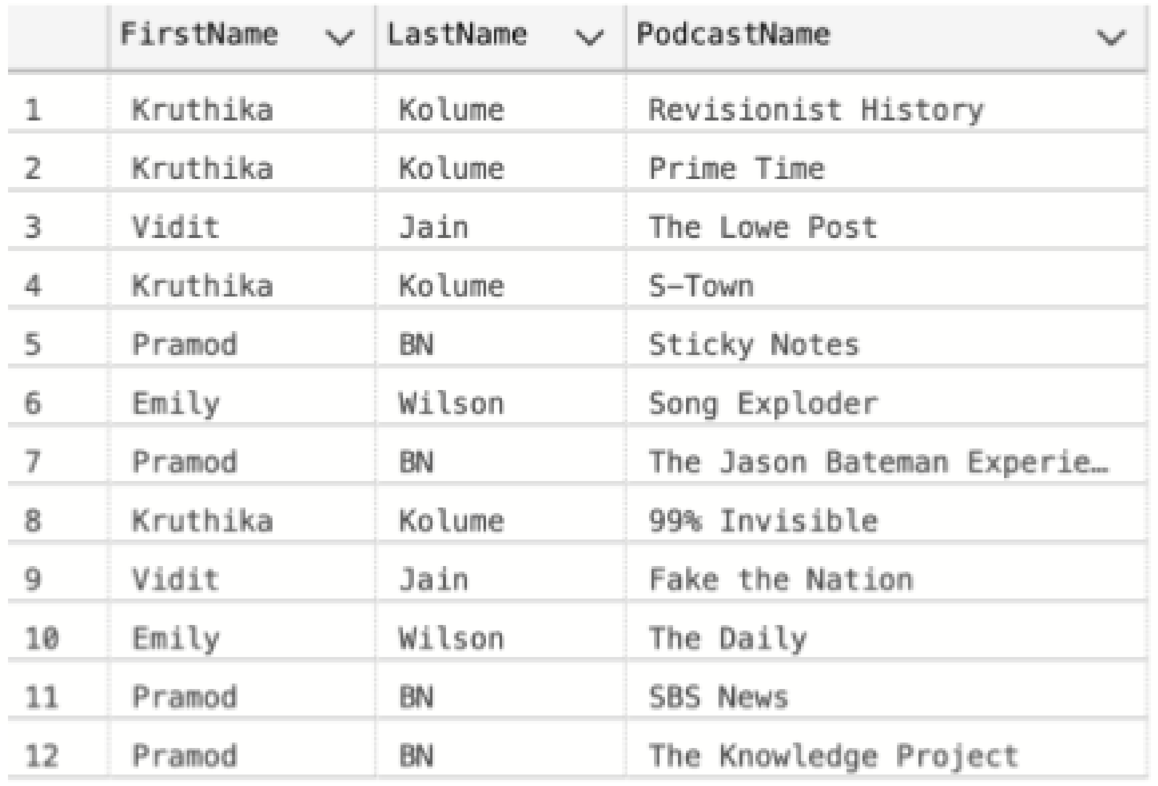
```sql
GO
CREATE VIEW Podcast.User_Favourite_Podcasts AS
SELECT u.FirstName, u.LastName, p.PodcastName
FROM Podcast.Users as u
INNER JOIN Podcast.Favorite as f
ON u.UsersID = f.UsersID
INNER JOIN Podcast.Podcasts as p
ON f.PodcastID = p.PodcastID;
GO

SELECT * FROM Podcast.User_Favourite_Podcasts;
```



	FirstName	LastName	PodcastName
1	Jane	Smith	The Daily
2	Mike	Spectre	The Tim Ferriss Show
3	Vidit	Jain	My Favorite Murder
4	Shravya	Gunda	Fake the Nation
5	Joe	Brown	Serial
6	Emily	Wilson	The David Pakman Show
7	Pramod	BN	Trending in Education
8	David	Lee	The Right Time
9	David	Lee	How I Built This
10	Emily	Wilson	TED Radio Hour
11	Sarah	Williams	My Favorite Murder
12	Joe	Brown	SmartLess

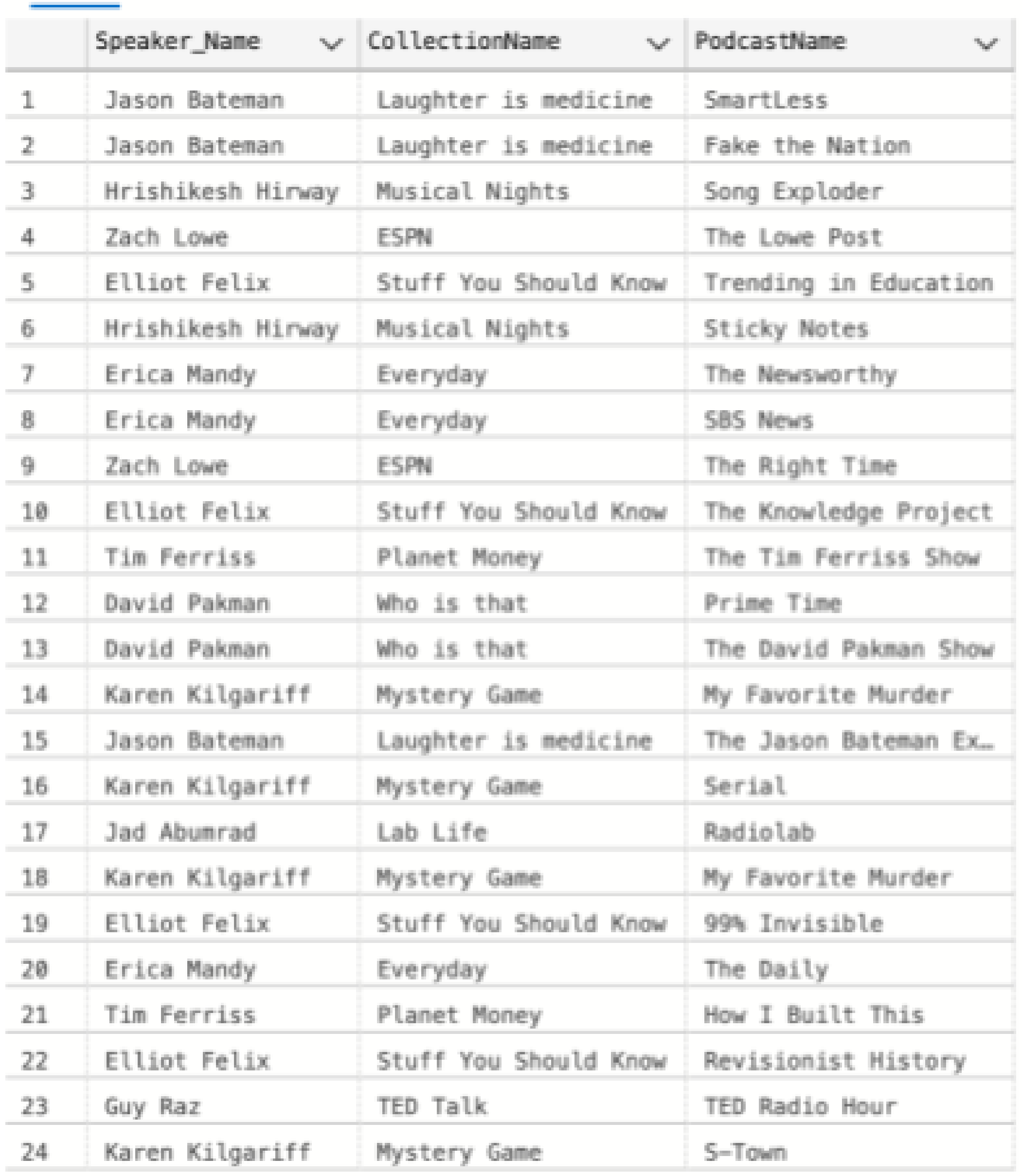


```
GO
CREATE VIEW Podcast.User_Download_Podcasts AS
SELECT u.FirstName, u.LastName, p.PodcastName
FROM Podcast.Users as u
INNER JOIN Podcast.Downloads as d
ON u.UsersID = d.Users_UsersID
INNER JOIN Podcast.Podcasts as p
ON d.Podcasts_PodcastID = p.PodcastID;
GO
```



	FirstName	LastName	PodcastName
1	Kruthika	Kolume	Revisionist History
2	Kruthika	Kolume	Prime Time
3	Vidit	Jain	The Lowe Post
4	Kruthika	Kolume	S-Town
5	Pramod	BN	Sticky Notes
6	Emily	Wilson	Song Exploder
7	Pramod	BN	The Jason Bateman Experie…
8	Kruthika	Kolume	99% Invisible
9	Vidit	Jain	Fake the Nation
10	Emily	Wilson	The Daily
11	Pramod	BN	SBS News
12	Pramod	BN	The Knowledge Project

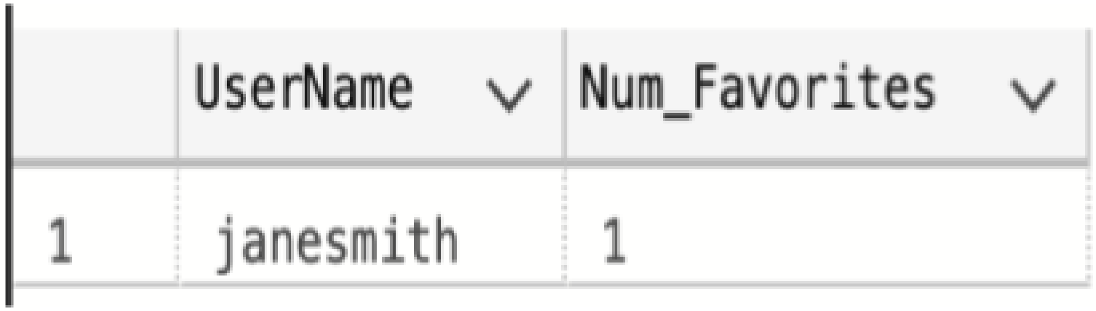
```
GO
CREATE VIEW Podcast.Speaker_Collection_Podcast AS
SELECT a.Speaker_Name, b.CollectionName, s.PodcastName
FROM Podcast.Speakers AS a
INNER JOIN Podcast.Speaker_Collection AS aa
ON a.SpeakerID = aa.Speaker_SpeakerID
INNER JOIN Podcast.Collection b
ON aa.Collection_CollectionID = b.CollectionID
INNER JOIN Podcast.Podcasts_Collection sa
ON sa.CollectionID = b.CollectionID
INNER JOIN Podcast.Podcasts s
ON s.PodcastID = sa.PodcastID;
GO

SELECT * FROM Podcast.Speaker_Collection_Podcast;
```



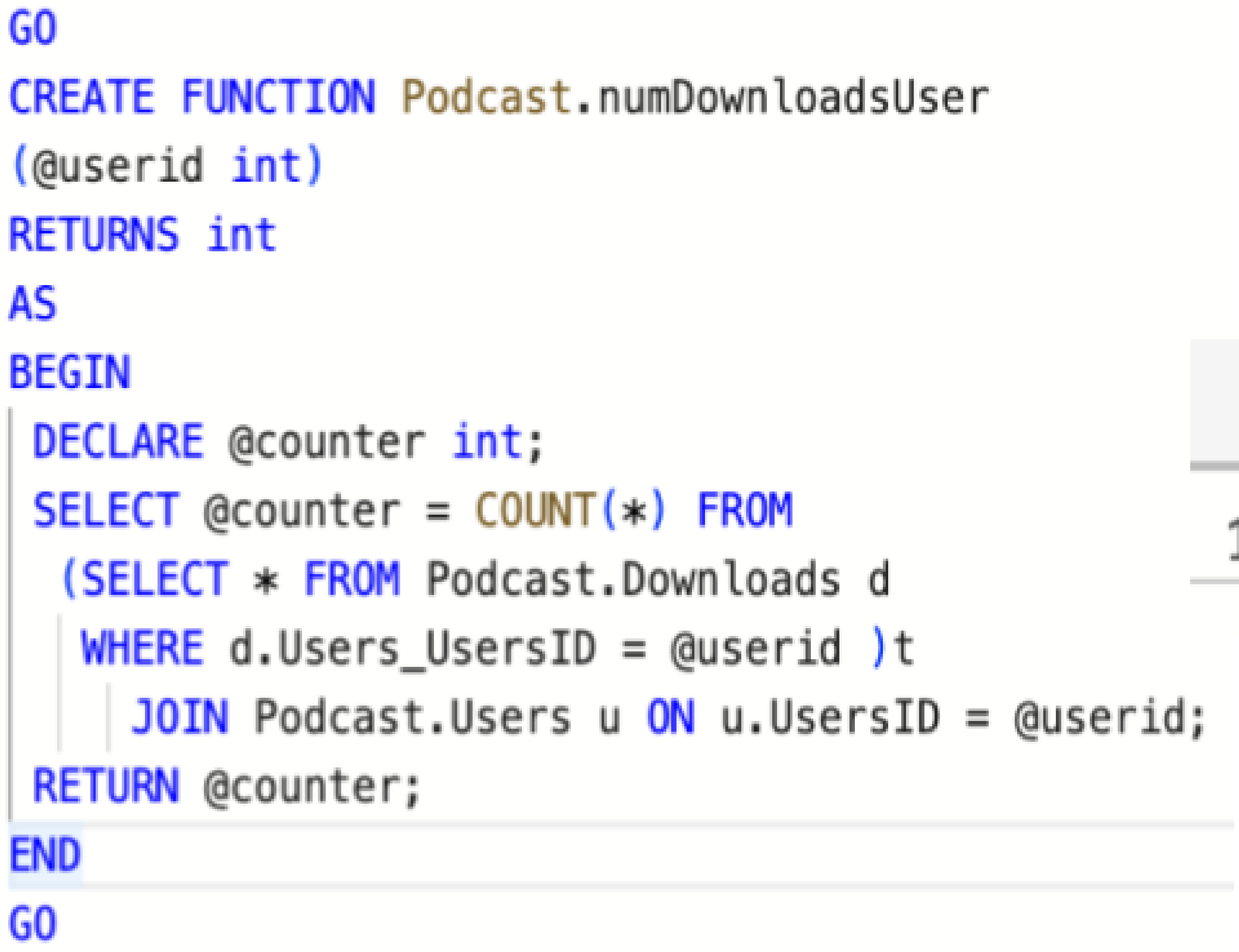
	Speaker_Name	CollectionName	PodcastName
1	Jason Bateman	Laughter is medicine	SmartLess
2	Jason Bateman	Laughter is medicine	Fake the Nation
3	Hrishikesh Hirway	Musical Nights	Song Exploder
4	Zach Lowe	ESPN	The Lowe Post
5	Elliot Felix	Stuff You Should Know	Trending in Education
6	Hrishikesh Hirway	Musical Nights	Sticky Notes
7	Erica Mandy	Everyday	The Newsworthy
8	Erica Mandy	Everyday	SBS News
9	Zach Lowe	ESPN	The Right Time
10	Elliot Felix	Stuff You Should Know	The Knowledge Project
11	Tim Ferriss	Planet Money	The Tim Ferriss Show
12	David Pakman	Who is that	Prime Time
13	David Pakman	Who is that	The David Pakman Show
14	Karen Kilgariff	Mystery Game	My Favorite Murder
15	Jason Bateman	Laughter is medicine	The Jason Bateman Ex…
16	Karen Kilgariff	Mystery Game	Serial
17	Jad Abumrad	Lab Life	Radiolab
18	Karen Kilgariff	Mystery Game	My Favorite Murder
19	Elliot Felix	Stuff You Should Know	99% Invisible
20	Erica Mandy	Everyday	The Daily
21	Tim Ferriss	Planet Money	How I Built This
22	Elliot Felix	Stuff You Should Know	Revisionist History
23	Guy Raz	TED Talk	TED Radio Hour
24	Karen Kilgariff	Mystery Game	S-Town

SQL FUNCTION

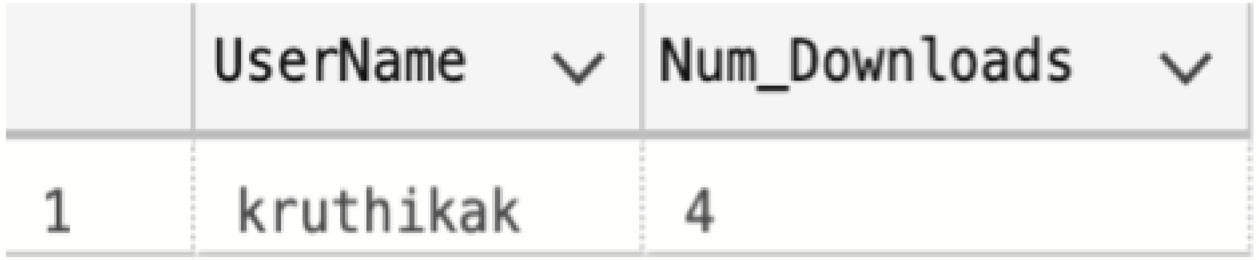
```
GO
CREATE FUNCTION Podcast.numFavoritesUser
(@userid int)
RETURNS int
AS
BEGIN
 DECLARE @counter int;
 SELECT @counter = COUNT(*) FROM
  (SELECT * FROM Podcast.Favorite f
   WHERE f.UsersID = @userid )t
     JOIN Podcast.Users u ON u.UsersID = @userid;
 RETURN @counter;
END;
GO
```



	UserName	Num_Favorites
1	janesmith	1

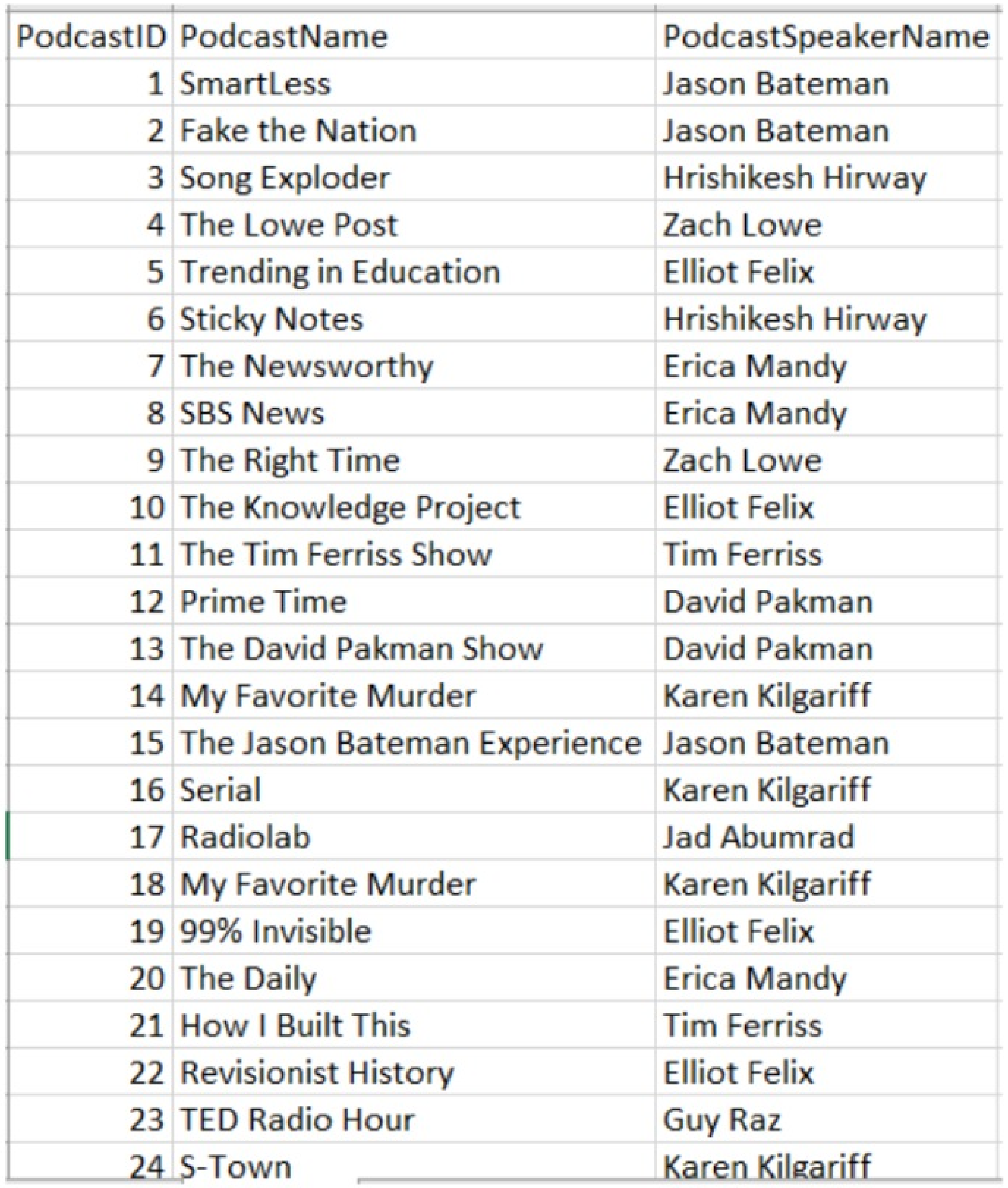


```
GO
CREATE FUNCTION Podcast.numDownloadsUser
(@userid int)
RETURNS int
AS
BEGIN
 DECLARE @counter int;
 SELECT @counter = COUNT(*) FROM
  (SELECT * FROM Podcast.Downloads d
   WHERE d.Users_UsersID = @userid )t
     JOIN Podcast.Users u ON u.UsersID = @userid;
 RETURN @counter;
END
GO
```



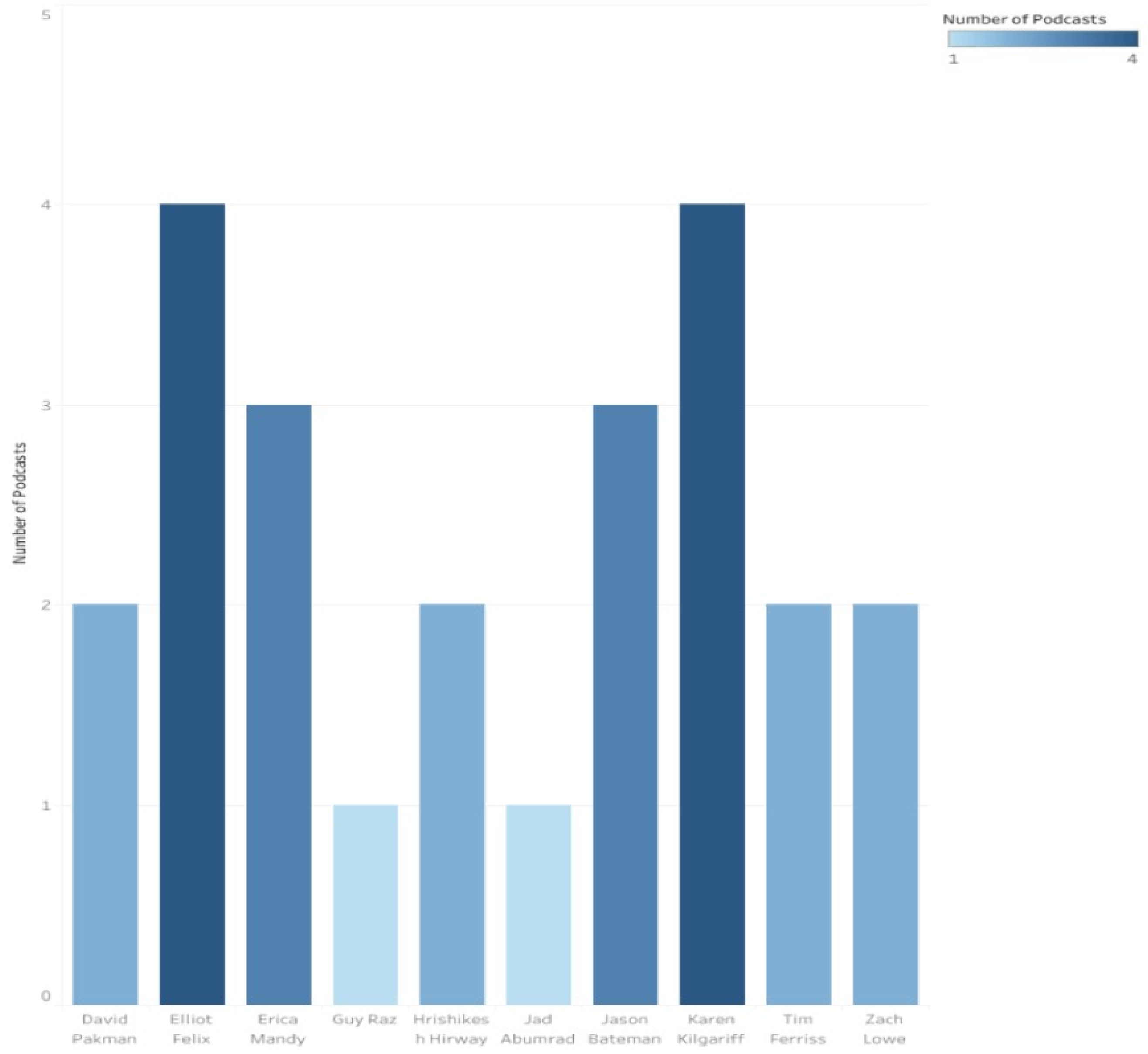
	UserName	Num_Downloads
1	kruthikak	4

REPORTS

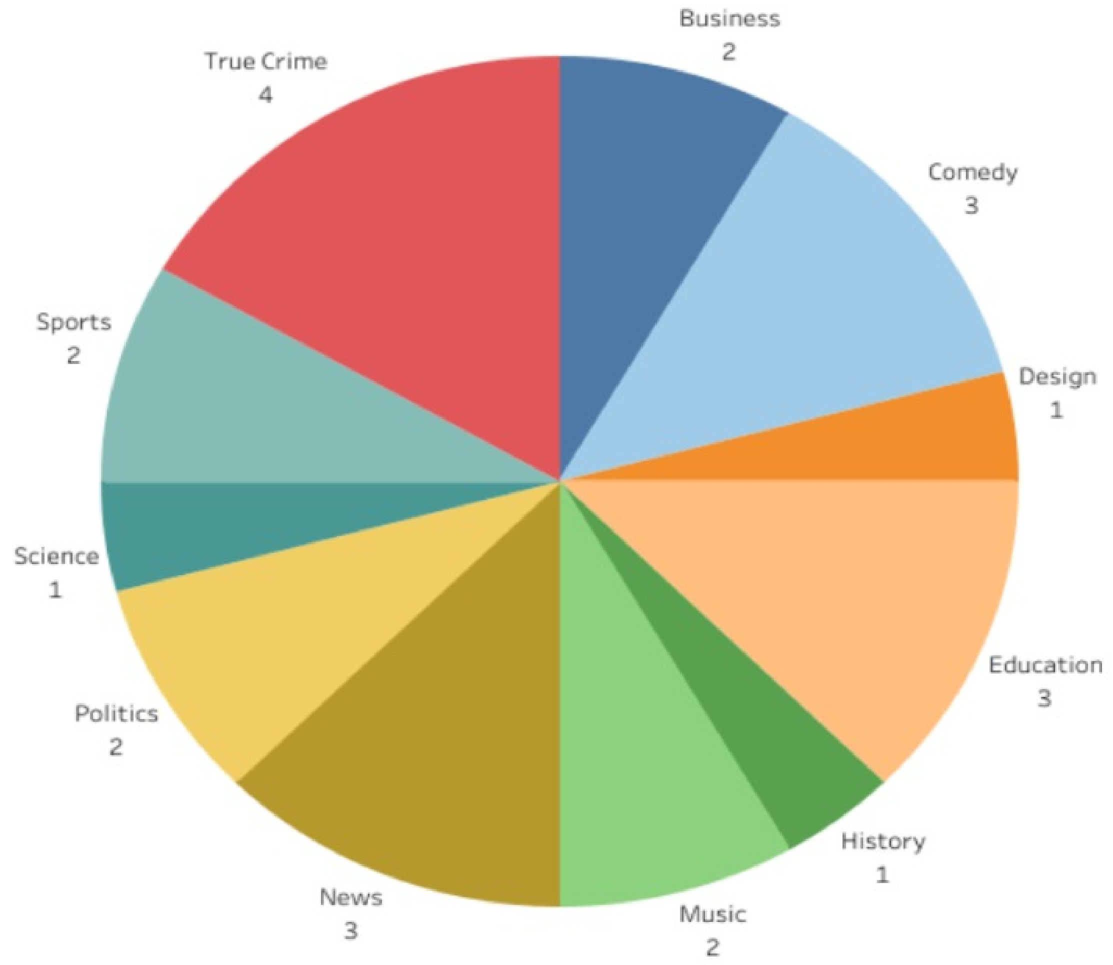


PodcastID	PodcastName	PodcastSpeakerName
1	SmartLess	Jason Bateman
2	Fake the Nation	Jason Bateman
3	Song Exploder	Hrishikesh Hirway
4	The Lowe Post	Zach Lowe
5	Trending in Education	Elliot Felix
6	Sticky Notes	Hrishikesh Hirway
7	The Newsworthy	Erica Mandy
8	SBS News	Erica Mandy
9	The Right Time	Zach Lowe
10	The Knowledge Project	Elliot Felix
11	The Tim Ferriss Show	Tim Ferriss
12	Prime Time	David Pakman
13	The David Pakman Show	David Pakman
14	My Favorite Murder	Karen Kilgariff
15	The Jason Bateman Experience	Jason Bateman
16	Serial	Karen Kilgariff
17	Radiolab	Jad Abumrad
18	My Favorite Murder	Karen Kilgariff
19	99% Invisible	Elliot Felix
20	The Daily	Erica Mandy
21	How I Built This	Tim Ferriss
22	Revisionist History	Elliot Felix
23	TED Radio Hour	Guy Raz
24	S-Town	Karen Kilgariff

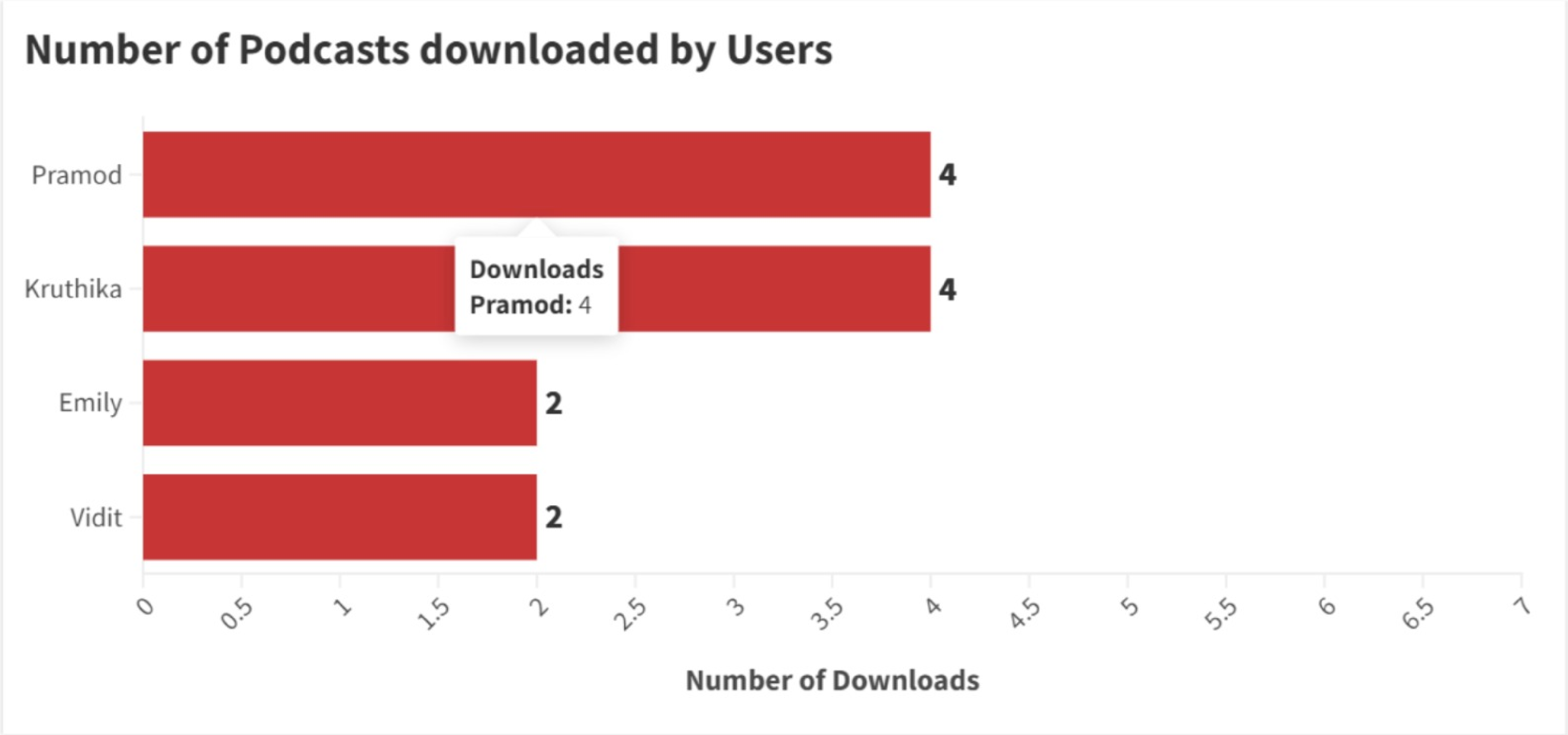
Number of Podcasts per Speaker



Number of Podcasts in each Genre



Genre
Business
Comedy
Design
Education
History
Music
News
Politics
Science
Sports
True Crime


Number of Podcasts downloaded by Users
Pramod
Kruthika
Emily
Vidit
4
4
2
2
Downloads
Pramod: 4
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7
Number of Downloads