

CPSC 304 Project Cover Page

University of British Columbia, Vancouver

Department of Computer Science

Milestone: 02

Date: October 20, 2023

Group Number: 102

Name	Student Number	CS Alias (Userid)	Preferred Email Address
William Chow	93966943	x3c4k	williamchow604@gmail.com
Simrit Nijjar	66234287	s9m3h	simritnijjar@outlook.com
Davis Horton	17572686	i2o5k	hortondavis1@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Table of Contents

Project Summary

ER Diagram

Changes

Justifications

Diagram

Schema

Functional Dependencies

Normalization

Step 01: Minimal Cover

Step 02: Decompose to BCNF

User (UserID, UserName, Email, Age, Birthday)

Subscription(StreamingServiceName, Tier, Duration, TotalCost, MonthlyCost)

Updated Schema

SQL DDL Statements

Notes

Statements

Insert Statements

Notes

Statements

Project Summary

A database that holds media made from different studios and offered by streaming services. A user can subscribe to a streaming services and save media to history or to watch later.

ER Diagram

Changes

Added missing non-key attributes to entities (we missed this initially)

Added "listID" Primary Key to WatchList (Allows different users to have watch lists with the same name)

Fixed partial key and Primary Key for Subscription and Streaming Service (They were swapped accidentally)

Added "Total Disjoint" to ISA relationship (Specifies that we can not have an entity of type Media)

Updated Participation Constraint between WatchList and createWatchList to be forced (Each watchlist must be associated with a single user)

Updated Participation Constraint between User and subscribesTo to be Many-to-Many (user should be able to subscribe to multiple streaming services).

Updated Participation Constraint between Media and isGenre to be forced (Each Media must be associated with a genre)

Updated Participation Constraint between Media and isHostedBy to be forced (Each Media must be associated with a Streaming Service)

Added MonthlyCost attribute Subscription

Added company attribute to StreamingService to specify where each streaming service name came from which company

Added email attribute to User for a Candidate Key

Removed weak entity that was attached to Movie as it wasn't useful with no attributes attached to Movie

Updated Type to be renamed to Version (Type is reserved)

Removed “historyID” in WatchHistory and replaced it with “dateAdded” as a primary key to specify that a media can be added to a watch history multiple times

Added “birthday” & “age” attribute to User for more Functional Dependencies

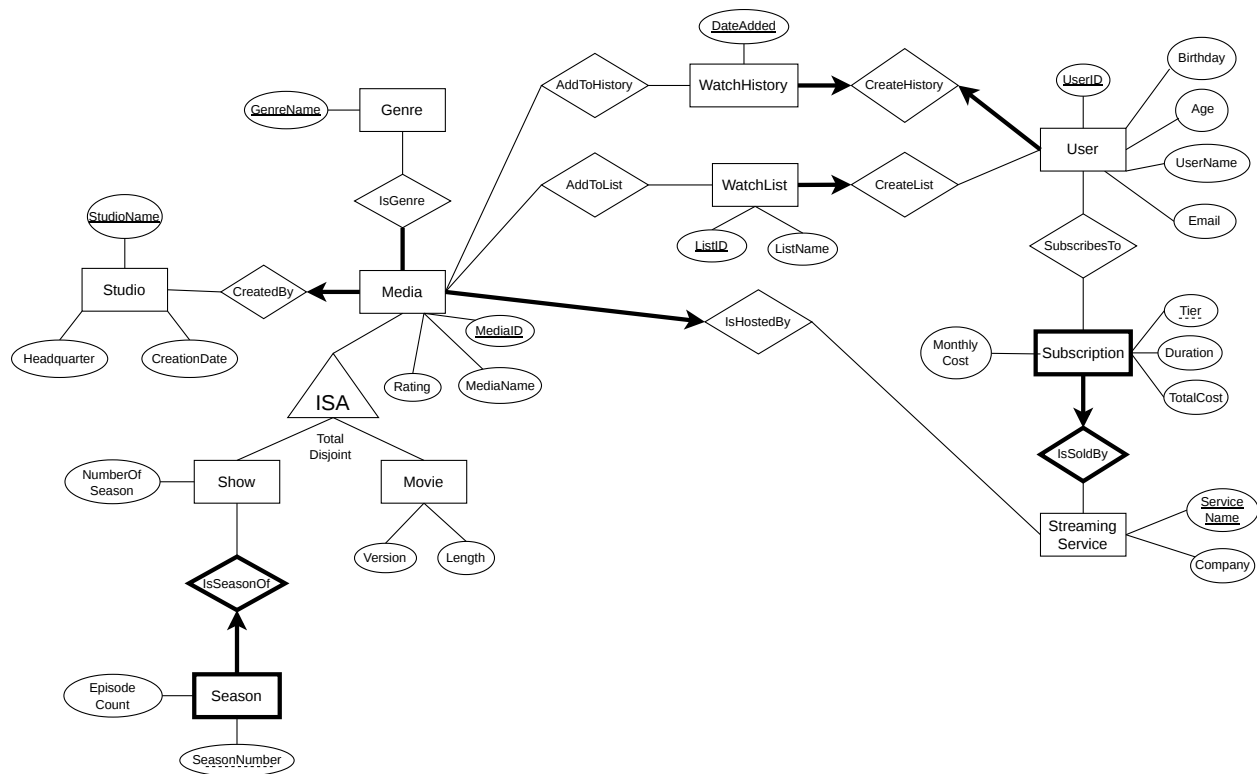
Updated naming convention in ER diagram to only include the attribute name instead of both the attribute name and table name

Justifications

Genre: We only have the genreName Primary Key because each genre only needs to be identified by its name for a user to understand its description when viewing a media item.

WatchHistory: Each user will only have a single history item associated with them and each history will only be associated with a single user. The history is only necessary to keep track of the media items that the user has watched so there is no other data that must be kept (other than the date that the Media was watched).

Diagram



Schema

```

Studio (
  StudioName: VARCHAR,
  Headquarter: VARCHAR, // Location (State/Province, Country)
  CreationDate: CHAR(10), // DATE FORMAT (yyyy-mm-dd)
  PRIMARY KEY: (StudioName),
  CANDIDATE KEY: (StudioName)
)
  
```

```

Genre (
  GenreName: CHAR,
  MediaID: INT,
  PRIMARY KEY: (GenreName, MediaID),
  CANDIDATE KEY: (GenreName, MediaID),
  FOREIGN KEY: (MediaID)
)
  
```

```

Season (
    MediaID: INT,
    SeasonNumber: INT,
    EpisodeCount: INT,
    PRIMARY KEY: (MediaID, SeasonNumber),
    CANDIDATE KEY: (MediaID, SeasonNumber),
    FOREIGN KEY: (MediaID) // References Show
)

```

```

User (
    UserID: INT,
    UserName: VARCHAR,
    Email: VARCHAR UNIQUE,
    Birthday: CHAR(10) // YYYY-MM-DD
    Age: INT
    PRIMARY KEY: (UserID),
    CANDIDATE KEYS: [(UserID), (Email)]
)

```

```

StreamingService (
    ServiceName: VARCHAR,
    Company: VARCHAR,
    PRIMARY KEY: (ServiceName),
    CANDIDATE KEY: (ServiceName)
)

```

```

AddToList (
    ListID: INT,
    MediaID: INT,
    PRIMARY KEY: (MediaID, ListID),
    CANDIDATE KEY: (MediaID, ListID),
    FOREIGN KEY: (MediaID)
)

```

```

Subscription (
    ServiceName: VARCHAR,
    Tier: VARCHAR,
    Duration: INT, // Months
    TotalCost: DOUBLE, // Dollars
    MonthlyCost: DOUBLE, // Dollars
    PRIMARY KEY: (ServiceName, Tier),

```

```
CANDIDATE KEY: (ServiceName, Tier),  
FOREIGN KEY: (ServiceName)  
)
```

```
Media (  
  MediaID: INT,  
  MediaName: VARCHAR,  
  Rating: INT,  
  StudioName: VARCHAR NOT NULL,  
  ServiceName: VARCHAR NOT NULL,  
  PRIMARY KEY: (MediaID),  
  CANDIDATE KEY: (MediaID),  
  FOREIGN KEYS: [(StudioName), (ServiceName)],  
)
```

```
Show (  
  MediaID: INT,  
  NumberOfSeasons: INT,  
  PRIMARY KEY (MediaID),  
  CANDIDATE KEY: (MediaID),  
  FOREIGN KEY (MediaID)  
)
```

```
Movie (  
  MediaID: INT,  
  Version: VARCHAR,  
  Length: INT, // minutes  
  PRIMARY KEY: (MediaID, Version),  
  CANDIDATE KEY: (MediaID, Version),  
  FOREIGN KEY: (MediaID)  
)
```

```
WatchHistory (  
  UserID: INT,  
  MediaID: INT,  
  DateAdded: CHAR(10), // DATE (yyyy-mm-dd)  
  PRIMARY KEY: (UserID, MediaID, DateAdded),  
  CANDIDATE KEY: (UserID, MediaID, DateAdded),  
  FOREIGN KEYS: [(UserID), (MediaID)]  
)
```

```
WatchList (  
    listID: INT,  
    listName: VARCHAR,  
    UserID: INT NOT NULL,  
    PRIMARY KEY: (ListID),  
    CANDIDATE KEY: (ListID),  
    FOREIGN KEY: (UserID)  
)
```

```
SubscribesTo (  
    UserID: INT,  
    ServiceName: VARCHAR,  
    Tier: VARCHAR,  
    PRIMARY KEY: (UserID, ServiceName),  
    CANDIDATE KEY: (UserID, ServiceName),  
    FOREIGN KEY: (UserID),  
    FOREIGN KEY: (ServiceName, Tier) // REFERENCES Subscription  
)
```

Functional Dependencies

Studio_Name → (Studio_Headquarters, Studio_CreationDate)

Media_ID → (Media_Name, Media_Rating, Studio_Name, StreamingService_Name)

Media_ID → Show_NumberOfSeasons

(Media_ID, Movie_Version) → Movie_Length

(Media_ID, Season_SeasonNumber) → Season_EpisodeCount

User_ID → (User_Name, User_Email, User_Birthday, User_Age)

User_Email → (User_Name, User_ID, User_Birthday, User_Age)

User_Birthday → User_Age

StreamingService_Name → StreamingService_Company

(StreamingService_Name, Subscription_Tier) → (Subscription_Duration,
Subscription_TotalCost, Subscription_MonthlyCost)

(Subscription_Duration, Subscription_Cost) → Subscription_MonthlyCost

(User_ID, StreamingService_Name) → Subscription_Tier

List_ID → (List_Name, User_ID)

Normalization

Step 01: Minimal Cover

- Studio_Name → Studio_Headquarters
- Studio_Name → Studio_CreationDate
- Media_ID → Media_Rating
- Media_ID → Media_Name
- Media_ID → Studio_Name
- Media_ID → StreamingService_name
- Media_ID → Show_NumberOfSeasons
- (Media_ID, Movie_Version) → Movie_Length
- (Media_ID, Season_Number) → Season_EpisodeCount
- User_ID → User_Name
- User_ID → User_Email
- User_ID → User_Birthday
- ~~User_ID → User_Age~~
- ~~User_Email → User_Name~~
- User_Email → User_ID
- ~~User_Email → User_Birthday~~
- StreamingService_Name → StreamingService_Company
- (StreamingService_Name, Subscription_Tier) → Subscription_Duration
- (StreamingService_Name, Subscription_Tier) → Subscription_TotalCost
- ~~StreamingService_Name, Subscription_Tier → Subscription_MonthlyCost~~
- (Subscription_Duration, Subscription_Cost) → Subscription_MonthlyCost
- (User_ID, StreamingService_Name) → Subscription_Tier
- List_ID → List_Name
- List_ID → User_ID

- ~~User_Email~~ → User_Age
- User_Birthday → User_Age

Step 02: Decompose to BCNF

User (UserID, UserName, Email, Age, Birthday)

Functional Dependencies

UserID → (UserName, Email, Age, Birthday)

UserEmail → (UserName, UserID, Age, Birthday)

Birthday → Age } **Violates 3NF**

Minimal Cover

Key: {UserID} OR {UserEmail}

$UserID^+ = \{Email, UserID, UserName, Age, Birthday\}$

$UserEmail^+ = \{Email, UserID, UserName, Age, Birthday\}$

UserID → UserName

UserID → Email

UserID → Birthday

Email → UserID

Birthday → Age

Decomposition

User (UserID, UserName, Email, Birthday)

Age (Birthday, Age)

Subscription(StreamingServiceName, Tier, Duration, TotalCost, MonthlyCost)

Functional Dependencies

(StreamingServiceName, Tier) → (Duration, TotalCost, MonthlyCost)

(Duration, TotalCost) → MonthlyCost } **Violates 3NF**

Minimal Cover

Key: (StreamingServiceName, Tier)

$(\text{StreamingServiceName}, \text{Tier})^+ = \{\text{StreamingServiceName}, \text{Tier}, \text{Duration}, \text{TotalCost}, \text{MonthlyCost}\}$

$(\text{Duration}, \text{TotalCost})^+ = \{\text{MonthlyCost}\}$

$(\text{StreamingServiceName}, \text{Tier}) \rightarrow \text{Duration}$

$(\text{StreamingServiceName}, \text{Tier}) \rightarrow \text{TotalCost}$

$(\text{Duration}, \text{TotalCost}) \rightarrow \text{MonthlyCost}$

Decomposition

Subscription (StreamingServiceName, Tier, Duration, TotalCost)

Cost (Duration, TotalCost, MonthlyCost)

Updated Schema

```
Studio (  
  StudioName: VARCHAR,  
  Headquarter: VARCHAR,    // Location (State/Province, Country)  
  CreationDate: CHAR(10),  // DATE FORMAT (yyyy-mm-dd)  
  PRIMARY KEY: (StudioName),  
  CANDIDATE KEY: (StudioName)  
)
```

```
Genre (  
  GenreName: CHAR,  
  MediaID: INT,  
  PRIMARY KEY: (GenreName, MediaID),  
  CANDIDATE KEY: (GenreName, MediaID),  
  FOREIGN KEY: (MediaID)  
)
```

```
Season (  
  MediaID: INT,  
  SeasonNumber: INT,  
  EpisodeCount: INT,  
  PRIMARY KEY: (MediaID, SeasonNumber),  
  CANDIDATE KEY: (MediaID, SeasonNumber),
```

```
FOREIGN KEY: (MediaID) // References Show
)
```

```
User (
  UserID: INT,
  UserName: VARCHAR,
  Email: VARCHAR UNIQUE,
  Birthday: CHAR(10), // YYYY-MM-DD
  PRIMARY KEY: (UserId),
  CANDIDATE KEYS: [(UserID), (Email)]
)
```

```
Age (
  Birthday: CHAR(10), // YYYY-MM-DD
  Age: INT,
  PRIMARY KEY: (Birthday),
  FOREIGN KEY: (Birthday) // References User
)
```

```
StreamingService (
  ServiceName: VARCHAR,
  Company: VARCHAR,
  PRIMARY KEY: (ServiceName),
  CANDIDATE KEY: (ServiceName)
)
```

```
AddToList (
  ListID: INT,
  MediaID: INT,
  PRIMARY KEY: (MediaID, ListID),
  CANDIDATE KEY: (MediaID, ListID),
  FOREIGN KEY: (MediaID)
)
```

```
Subscription (
  ServiceName: VARCHAR,
  Duration: INT, // Months
  TotalCost: DOUBLE, // Dollars
  PRIMARY KEY: (ServiceName, Tier),
  FOREIGN KEY: (ServiceName)
)
```

```
Cost (
```

```

Duration: INT,          // Months
TotalCost: DOUBLE,     // Dollars
MonthlyCost: DOUBLE,   // Dollars
PRIMARY KEY: (Duration, TotalCost),
FOREIGN KEY: (Duration, TotalCost) // References Subscription
)

```

```

Media (
  MediaID: INT,
  MediaName: VARCHAR,
  Rating: INT,
  StudioName: VARCHAR NOT NULL,
  ServiceName: VARCHAR NOT NULL,
  PRIMARY KEY: (MediaID),
  CANDIDATE KEY: (MediaID),
  FOREIGN KEYS: [(StudioName), (ServiceName)],
)

```

```

Show (
  MediaID: INT,
  NumberOfSeasons: INT,
  PRIMARY KEY (MediaID),
  CANDIDATE KEY: (MediaID),
  FOREIGN KEY (MediaID)
)

```

```

Movie (
  MediaID: INT,
  Version: VARCHAR,
  Length: INT, // minutes
  PRIMARY KEY: (MediaID, Version),
  CANDIDATE KEY: (MediaID, Version),
  FOREIGN KEY: (MediaID)
)

```

```

WatchHistory (
  UserID: INT,
  MediaID: INT,
  DateAdded: CHAR(10), // DATE (yyyy-mm-dd)
  PRIMARY KEY: (UserID, MediaID, DateAdded),
  CANDIDATE KEY: (UserID, MediaID, DateAdded),
)

```

```
FOREIGN KEYS: [(UserID), (MediaID)]
)
```

```
WatchList (
  listID: INT,
  listName: VARCHAR,
  UserID: INT NOT NULL,
  PRIMARY KEY: (ListID),
  CANDIDATE KEY: (ListID),
  FOREIGN KEY: (UserID)
)
```

```
SubscribesTo (
  UserID: INT,
  ServiceName: VARCHAR,
  Tier: VARCHAR,
  PRIMARY KEY: (UserID, ServiceName),
  CANDIDATE KEY: (UserID, ServiceName),
  FOREIGN KEY: (UserID),
  FOREIGN KEY: (ServiceName, Tier) // REFERENCES Subscription
)
```

SQL DDL Statements

Notes

User → **StreamingUser** to avoid Reserved Word

Show → **TVShow** to avoid Reserved Word

Statements

```
CREATE TABLE Studio (
  StudioName VARCHAR PRIMARY KEY,
  Headquarter VARCHAR, -- Location (State/Province, Country)
  CreationDate CHAR(10) -- DATE FORMAT (yyyy-mm-dd)
)
```

```
CREATE TABLE Genre (  
    GenreName VARCHAR,  
    MediaID INT,  
    PRIMARY KEY (GenreName, MediaID),  
    FOREIGN KEY (MediaID)  
        REFERENCES Media  
)
```

```
CREATE TABLE Season (  
    MediaID INT,  
    SeasonNumber INT,  
    EpisodeCount INT,  
    PRIMARY KEY (MediaID, SeasonNumber),  
    FOREIGN KEY (MediaID)  
        REFERENCES TVShow  
)
```

```
CREATE TABLE StreamingUser (  
    UserID INT PRIMARY KEY,  
    UserName VARCHAR,  
    Email VARCHAR UNIQUE,  
    Birthday CHAR(10) -- YYYY-MM-DD  
)  
  
CREATE TABLE Age (  
    Birthday CHAR(10) PRIMARY KEY, -- YYYY-MM-DD  
    Age INT,  
    FOREIGN KEY (Birthday)  
        REFERENCES StreamingUser1(Birthday)  
)
```

```
CREATE TABLE StreamingService (  
    ServiceName VARCHAR PRIMARY KEY,  
    Company VARCHAR  
)
```

```
CREATE TABLE AddToList (  
    ListID INT,  
    MediaID INT,  
    PRIMARY KEY (MediaID, ListID),  
    FOREIGN KEY (MediaID)
```

```
REFERENCES Media
)
```

```
CREATE TABLE Subscription (
    ServiceName VARCHAR,
    Tier VARCHAR,
    Duration INT,          -- Months
    TotalCost DOUBLE,     -- Dollars
    PRIMARY KEY (ServiceName, Tier),
    FOREIGN KEY (ServiceName)
        REFERENCES StreamingService
)
```

```
CREATE TABLE Cost (
    Duration INT,          -- Months
    TotalCost DOUBLE,     -- Dollars
    MonthlyCost DOUBLE,   -- Dollars
    PRIMARY KEY (Duration, TotalCost),
    FOREIGN KEY (Duration, TotalCost)
        REFERENCES Subscription(Duration, TotalCost)
)
```

```
CREATE TABLE Media (
    MediaID INT PRIMARY KEY,
    MediaName VARCHAR,
    Rating INT,
    StudioName VARCHAR NOT NULL,
    ServiceName VARCHAR NOT NULL,
    FOREIGN KEY (StudioName)
        REFERENCES Studio,
    FOREIGN KEY (ServiceName)
        REFERENCES StreamingService
)
```

```
CREATE TABLE TVShow (
    MediaID INT PRIMARY KEY,
    NumberOfSeasons INT,
    FOREIGN KEY (MediaID)
        REFERENCES Media
)
```

```
CREATE TABLE Movie (
    MediaID INT,
```



```

Version VARCHAR,
Length INT, -- minutes
PRIMARY KEY (MediaID, Version),
FOREIGN KEY (MediaID)
    REFERENCES Media
)

```

```

CREATE TABLE WatchHistory (
    UserID INT,
    MediaID INT,
    DateAdded CHAR(10), -- DATE (yyyy-mm-dd)
    PRIMARY KEY (UserID, MediaID, DateAdded),
    FOREIGN KEY (UserID)
        REFERENCES StreamingUser,
    FOREIGN KEY (MediaID)
        REFERENCES Media
)

```

```

CREATE TABLE WatchList (
    ListID INT PRIMARY KEY,
    ListName VARCHAR,
    UserID INT NOT NULL,
    FOREIGN KEY (UserID)
        REFERENCES StreamingUser
)

```

```

CREATE TABLE SubscribesTo (
    UserID INT,
    ServiceName VARCHAR,
    Tier VARCHAR,
    PRIMARY KEY (UserID, ServiceName),
    FOREIGN KEY (UserID)
        REFERENCES StreamingUser,
    FOREIGN KEY (ServiceName, Tier)
        REFERENCES Subscription (ServiceName, Tier)
)

```

Insert Statements

Notes

This version of INSERT is supported in MySQL, but not Oracle. To change to Oracle, add INSERT statements for each inserted tuple.

Statements

```
INSERT INTO Studio (StudioName, Headquarter, CreationDate)
VALUES ('Universal Pictures', 'California, USA', 1912-03-15),
       ('Sony Pictures', 'California, USA', 1987-09-21),
       ('HBO', 'Maryland, USA', 1980-05-11),
       ('Cartoon Network', 'California, USA', 2010-01-30),
       ('MAPPA', 'Tokyo, Japan', 1972-02-12);
```

```
INSERT INTO Genre (GenreName, MediaID)
VALUES ('Drama', 1),
       ('Action', 2),
       ('Horror', 3),
       ('Comedy', 4),
       ('Superhero', 5),
       ('Drama', 6),
       ('Thriller', 7),
       ('Superhero', 8),
       ('Drama', 9),
       ('Adventure', 10);
```

```
INSERT INTO Media (MediaID, MediaName, Rating, StudioName, ServiceName)
VALUES (1, 'Game of Thrones', 8, 'HBO', 'Max'),
       (2, 'Attack on Titan', 8, 'MAPPA', 'Netflix'),
       (3, 'Teen Titans', 7, 'Cartoon Network', 'Crave TV'),
       (4, 'Bob', 3, 'Universal Pictures', 'Prime Video'),
       (5, 'Spiderman 2', 9, 'Sony Pictures', 'Crave TV'),
       (6, 'The Wire', 7, 'HBO', 'Max'),
       (7, 'Kingsman', 4, 'Sony Pictures', 'Netflix'),
       (8, 'The Amazing Spiderman', 8, 'Sony Pictures', 'DisneyPlus'),
       (9, 'Breaking Bad', 10, 'Universal Pictures', 'Netflix'),
       (10, 'Avater', 3, 'Cartoon Network', 'Crave TV');
```

```
INSERT INTO Movie (MediaID, Version, Length) -- IsA for Media
VALUES (4, 'Home Video', 120),
       (5, 'Theatrical Release', 110),
       (7, 'Home Video', 132),
```

```
(8, 'Theatrical Release', 142),  
(10, 'Directors Cut', 200);
```

```
INSERT INTO TVShow (MediaID, NumberOfSeasons) -- IsA for Media  
VALUES (1, 6),  
       (2, 4),  
       (3, 7),  
       (6, 8),  
       (9, 6);
```

```
INSERT INTO Season (MediaID, SeasonNumber, EpisodeCount) -- Weak entity for TVShow  
VALUES (1, 1, 24),  
       (1, 2, 23),  
       (2, 1, 12),  
       (2, 2, 13),  
       (3, 1, 10);
```

```
INSERT INTO StreamingUser (UserID, UserName, Email, Birthday)  
VALUES (1, 'bobSmith', 'bobsmith@gmail.com', 1961-01-05),  
       (2, 'blueDesk5', 'blueDesk@gmail.com', 1976-05-25),  
       (3, 'orangeBug', 'orangeBug@gmail.com', 2005-03-14),  
       (4, 'greenCandy', 'greenCandy@gmail.com', 2002-07-30),  
       (5, 'blackCookie', 'blackCookie@gmail.com', 1989-10-01);
```

```
INSERT INTO Age (Birthday, Age)  
VALUES (1961-01-05, 62),  
       (1976-05-25, 47),  
       (2005-03-14, 18),  
       (2002-07-30, 21),  
       (1989-10-01, 34);
```

```
INSERT INTO StreamingService (ServiceName, Company)  
VALUES ('Disney Plus', 'Disney'),  
       ('Netflix', 'Netflix'),  
       ('Prime Video', 'Amazon'),  
       ('Max', 'HBO'),  
       ('Crave TV', 'Bell Media');
```

```
INSERT INTO Subscription (ServiceName, Tier, Duration, TotalCost)
VALUES ('Disney Plus', 'Premium', 6, 90.00),
       ('Netflix', 'Basic', 1, 12.99),
       ('Max', '4k', 2, 40.00),
       ('Disney Plus', 'Basic', 3, 33.00),
       ('Crave TV', 'HD', 12, 120.00);
```

```
INSERT INTO Cost (Duration, TotalCost, MonthlyCost)
VALUES (6, 90.00, 15.00),
       (1, 12.99, 12.99),
       (2, 40.00, 20.00),
       (3, 33.00, 11.00),
       (12, 120.00, 10.00);
```

```
INSERT INTO SubscribesTo (UserID, ServiceName, Tier)
VALUES (1, 'Max', '4k'),
       (2, 'Netflix', 'Basic'),
       (3, 'Disney Plus', 'Premium'),
       (4, 'Crave Tv', 'HD'),
       (5, 'Disney Plus', 'Basic');
```

```
INSERT INTO WatchList (ListID, ListName, UserID)
VALUES (1, 'Superhero Movies', 1),
       (2, 'Must See', 2),
       (3, 'Recommended', 3),
       (4, 'Personal', 4),
       (5, 'My List', 5);
```

```
INSERT INTO AddToList (ListID, MediaID)
VALUES (1, 6),
       (2, 4),
       (3, 8),
       (4, 1),
       (5, 3);
```

```
INSERT INTO WatchHistory (UserID, MediaID, DateAdded)
VALUES (1, 2, 2023-09-02),
       (2, 8, 2023-08-22),
       (3, 1, 2023-10-05),
```

```
(4, 7, 2023-06-17),  
(5, 4, 2023-10-02);
```