University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___2_____

Date: June 12, 2023 ___________

Group Number: __11________

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Carson Lu	37798238	r3y3a	carson_lu@outlook.com
Jerry Fan	90420811	j0d7s	jerryxjfan@hotmail.com
Shekinah Titus	38809083	p5y2b	shekinahtitus7@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

2. A brief (~2-3 sentences) summary of your project.

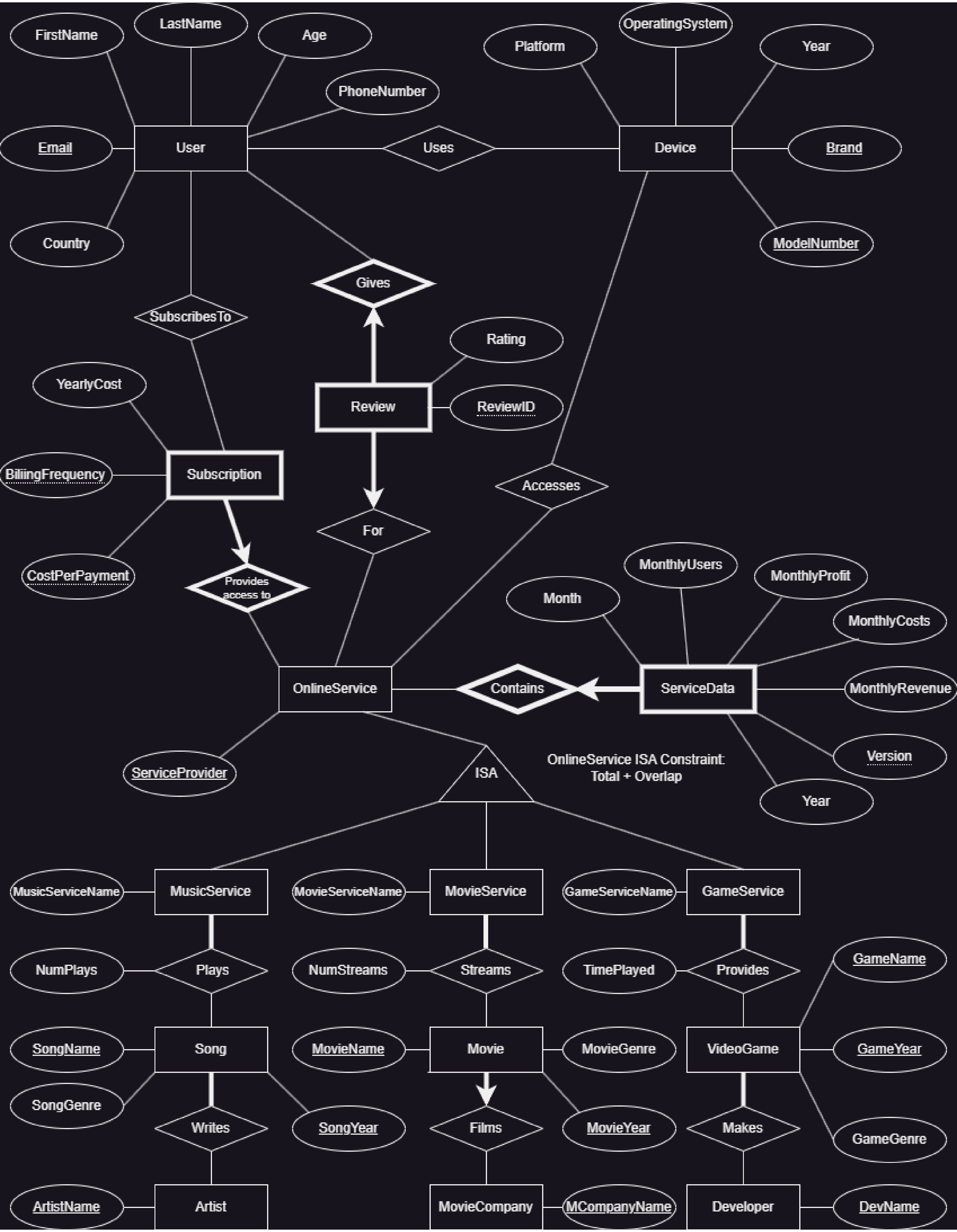
This project is a database for online services, both those with and without subscriptions, this specifically focuses on data about the online service and the users of these services. This database is useful for advertisers and product designers as it provides many statistics about an online service which allows them to understand usage details of their service by accessing data such as the demographic of users, devices being used, what titles are most popular, etc..

3. The ER diagram you are basing your item #3 (below) on.

This ER diagram may be the same as your milestone 1 submission or it might be different. If you have made changes from the version submitted in milestone 1, attach a note indicating what changes have been made and why.
If you have decided not to implement the suggestions given by your project mentor, please be sure to leave a note stating why. This is not to say that you must do everything that your project mentor says. In many instances, there are trade-offs between design choices and your decision may be influenced by different factors. That being said, your TAs will often leave suggestions that are meant to help massage your project into a form that will fit with the requirements in future project milestones. If you choose not to take their advice, it would be helpful for them to know why in order to better assist the group moving forward.

No suggestions were given in milestone 1 but some changes were made:

- Removed ISA from the top right, replaced it with the "platform" attribute
- Subscription is now a weak entity for OnlineService and has two PKs being BillingFrequency and Price
- Added PhoneNumber attribute to User
- OnlineService now has a PK of ServiceProvider (company name essentially) and it must be one or more of the services
- ISA for OnlineService now has its actual name of the service which may or may not be different from the name of the company
- Added year for Song, Movie, and VideoGame for date created as a PK
- Changed MovieCompany Name attribute to be MCompanyName and changed company name of Developer to Devname
- Added more attributes for ServiceData


FirstName
LastName
Age
PhoneNumber
Platform
OperatingSystem
Year
Email
User
Uses
Device
Brand
Country
ModelNumber
Gives
SubscribesTo
Rating
YearlyCost
Review
ReviewID
BiliingFrequency
Subscription
Accesses
For
CostPerPayment
Provides access to
MonthlyUsers
MonthlyProfit
Month
MonthlyCosts
OnlineService
Contains
ServiceData
MonthlyRevenue
Version
ServiceProvider
ISA
OnlineService ISA Constraint:
Total + Overlap
Year
MusicServiceName
MusicService
MovieServiceName
MovieService
GameServiceName
GameService
NumPlays
Plays
NumStreams
Streams
TimePlayed
Provides
GameName
SongName
Song
MovieName
Movie
MovieGenre
VideoGame
GameYear
SongGenre
Writes
SongYear
Films
MovieYear
Makes
GameGenre
ArtistName
Artist
MovieCompany
MCompanyName
Developer
DevName

4. The schema derived from your ER diagram (above).

For the translation of the ER diagram to the relational model, follow the same instructions as in your lectures. The process should be reasonably straightforward. For each table:

a. List the table definition (e.g., Table1(attr1: domain1, attr2: domain2, ...)). Make sure to include the domains for each attribute.

ENTITIES:

User(Email: char[50], FirstName: char[30], LastName: char[30], Age: integer, Country: char[60], PhoneNumber: integer)

Device(ModelNumber: integer, Brand: char[20], Year: integer, OperatingSystem: char[20], Platform: char[20])

Subscription(ServiceProvider : char[20], BillingFrequency: char[20], CostPerPayment: double, YearlyCost: double)

Review(Email: char[50], ReviewID: integer, ServiceProvider: char[20], Rating: integer)

ServiceDataContains(ServiceProvider: char[20], Version: char[10], MonthlyUsers: integer, MonthlyRevenue: double, MonthlyCosts: double, MonthlyProfit: double, Year: integer, Month: integer)

OnlineService(ServiceProvider: char[20])

MusicService(ServiceProvider: char[20], MusicServiceName: char[20])

MovieService(ServiceProvider: char[20], MovieServiceName: char[20])

GameService(ServiceProvider: char[20], GameServiceName: char[20])

Song(SongName: char[20], SongYear: integer, SongGenre: char[20])

Movie(MovieName: char[20], MCompanyName: char[20], MovieYear: integer, MovieGenre: char[20])

VideoGame(GameName: char[20], GameYear: integer, GameGenre: char[20])

Artist(ArtistName: char[20])

MovieCompany(MCompanyName: char[20])

Developer(DevName: char[20])

RELATIONSHIPS:

Uses(Email: char[50], ModelNumber: integer, Brand: integer)
SubscribesTo(Email: char[50], ServiceProvider: char[20], BillingFrequency: char[20], Price: double)
Accesses(ModelNumber: integer, Brand: char[20], ServiceProvider: char[20])
Plays(ServiceProvider: char[20], SongName: char[20], SongYear: integer, NumPlays: integer)
Streams(ServiceProvider: char[20], MovieName: char[20], MovieYear: integer, NumStreams: integer)
Provides(ServiceProvider: char[20], GameName: char[20], GameYear: integer, TimePlayed: integer)
Writes(SongName: char[20], SongYear: integer, ArtistName: char[20])
Makes(GameName: char[20], GameYear: integer, DevName: char[20])

b. Specify the primary key (PK), candidate key, (CK) foreign keys (FK), and other constraints (e.g., not null, unique, etc.) that the table must maintain.

ENTITIES:

NOTE: all Primary keys are left out of the Candidate Keys section because that would be redundant. Also, all candidate keys are a tuple, that is, for ServiceDataContains (ServiceProvider, Year, Month) can determine all keys, not each one separately. This is the same for the Primary keys. Also note that in the tables, there is at most one set of candidate keys.

Entities/Relationships	Primary Keys	Candidate Keys	Foreign Keys	Other Constraints
User	Email	PhoneNumber		
Device	ModelNumber Brand			
Subscription	ServiceProvider BillingFrequency CostPerPayment		ServiceProvider	
Review	Email ReviewID		Email ServiceProvider	ServiceProvider NOT NULL
ServiceDataContains	ServiceProvider Version	ServiceProvider Year Month	ServiceProvider	

OnlineService	ServiceProvider			
MusicService	ServiceProvider	MusicServiceName	ServiceProvider	UNIQUE MusicServiceName NOT NULL MusicServiceName
MovieService	ServiceProvider	MovieServiceName	ServiceProvider	UNIQUE MovieServiceName NOT NULL MovieServiceName
GameService	ServiceProvider	GameServiceName	ServiceProvider	UNIQUE GameServiceName NOT NULL GameServiceName
Song	SongName SongYear			
Movie	MovieName MovieYear		MCompanyName	MCompanyName NOT NULL
VideoGame	GameName GameYear			
Artist	ArtistName			
MovieCompany	MCompanyName			
Developer	CompanyName			
Uses	Email ModelNumber Brand		Email ModelNumber Brand	
SubscribesTo	Email BillingFrequency CostPerPayment ServiceProvider		Email BillingFrequency CostPerPayment ServiceProvider	
Accesses	ModelNumber Brand ServiceProvider		ModelNumber Brand ServiceProvider	
Plays	ServiceProvider SongName SongYear		ServiceProvider SongName SongYear	(ServiceProvider) that references MusicService has a participation constraint
Streams	ServiceProvider MovieName MovieYear		ServiceProvider MovieName MovieYear	(ServiceProvider) that references MovieService has a participation constrain

Provides	ServiceProvider GameName GameYear		ServiceProvider GameName GameYear	(ServiceProvider) that references GameService has a participation constrain
Writes	SongName SongYear ArtistName		SongName SongYear ArtistName	(SongName, SongYear) that references Song that has a participation constraint
Makes	GameName GameYear DevName		GameName GameYear DevName	(GameName, GameYyear) that references VideoGame has a participation constraint

5. Functional Dependencies (FDs)

a. Identify the functional dependencies in your relations, including the ones involving all candidate keys (including the primary key).

PKs and CKs are considered functional dependencies and should be included in the list of FDs. You do not need to include trivial FDs such as A → A.
Note: In your list of FDs, there must be some kind of valid FD other those identified by a PK or CK. If you observe that no relations have FDs other than the PK and CK(s), then you will have to intentionally add some (meaningful) attributes to show valid FDs. We want you to get a good normalization exercise. Your design must go through a normalization process.

Entities/Relationships	FDs
User	Email -> FirstName, LastName, Age, Country, PhoneNumber PhoneNumber -> Email, FirstName, LastName, Age, Country
Device	ModelNumber, Brand -> Year, OperatingSystem, Platform OperatingSystem -> Platform
Subscription	BillingFrequency, CostPerPayment -> YearlyCost
Review	Email, ReviewID -> ServiceProvider, Rating ReviewID -> Rating
ServiceDataContains	ServiceProvider, Version -> MonthlyUsers, MonthlyRevenue, Month, MonthlyProfit, MonthlyCosts, Year Month, Year -> Version MonthlyCost, MonthlyRevenue -> MonthlyProfit

	MonthlyCost, MonthlyProfit -> MonthlyRevenue MonthlyProfit, MonthlyRevenue -> MonthlyCost
OnlineService	
MusicService	ServiceProvider -> MusicServiceName MusicServiceName -> ServiceProvider
MovieService	ServiceProvider -> MovieServiceName MovieServiceName-> ServiceProvider
GameService	ServiceProvider -> GameServiceName GameServiceName-> ServiceProvider
Song	SongName, SongYear -> SongGenre
Movie	MovieName, MovieYear -> MovieGenre
VideoGame	GameName, GameYear -> GameGenre
Artist	
MovieCompany	
Developer	
Uses	
SubscribesTo	
Accesses	
Plays	ServiceProvider, SongName, SongYear -> NumPlays
Streams	ServiceProvider, MovieName, MovieYear -> NumStreams
Provides	ServiceProvider, GameName, GameYear -> TimePlayed
Writes	
Makes	

6. Normalization

a. Normalize each of your tables to be in 3NF or BCNF. Give the list of tables, their primary keys, their candidate keys, and their foreign keys after normalization.

You should show the steps taken for the decomposition. Should there be errors, and no work is shown, no partial credit can be awarded without steps shown. The format should be the same as Step 3, with tables listed similar to Table1(attr1:domain1, attr2:domain2, …). ALL Tables must be listed, not only the ones post normalization.

We will be normalizing all relations (EXCEPT for ServiceDataContains) to be in 3NF using the synthesis method. All primary keys are declared in the SQL section.

<u>User, OnlineService, MusicService, MovieService, GameService, Song, Movie, VideoGame, Artist, MovieCompany, Developer, SubscribesTo, Accesses, Plays, Streams, Provides, Writes, Makes:</u> Does not have any non-primary key FDs, non-candidate key FDs, or non-trivial FDs; therefore, these relations are already in 3NF

User(Email: char[50], FirstName: char[30], LastName: char[30], Age: integer, Country: char[60], PhoneNumber: integer)
OnlineService(ServiceProvider: char[20])
MusicService(ServiceProvider: char[20], MusicServiceName: char[20])
MovieService(ServiceProvider: char[20], MovieServiceName: char[20])
GameService(ServiceProvider: char[20], GameServiceName: char[20])
Song(SongName: char[20], SongYear: integer, SongGenre: char[20])
Movie(MovieName: char[20], MCompanyName: char[20], MovieYear: integer, MovieGenre: char[20])
VideoGame(GameName: char[20], GameYear: integer, GameGenre: char[20])
Artist(ArtistName: char[20])
MovieCompany(MCompanyName: char[20])
Developer(DevName: char[20])

Uses(Email: char[50], ModelNumber: integer, Brand: integer)
SubscribesTo(Email: char[50], ServiceProvider: char[20], BillingFrequency: char[20], Price: double)
Accesses(ModelNumber: integer, Brand: char[20], ServiceProvider: char[20])
Plays(ServiceProvider: char[20], SongName: char[20], SongYear: integer, NumPlays: integer)
Streams(ServiceProvider: char[20], MovieName: char[20], MovieYear: integer, NumStreams: integer)
Provides(ServiceProvider: char[20], GameName: char[20], GameYear: integer, TimePlayed: integer)
Writes(SongName: char[20], SongYear: integer, ArtistName: char[20])
Makes(GameName: char[20], GameYear: integer, DevName: char[20])

<u>Device:</u>
Primary Key: ModelNumber, Brand

FDs: OperatingSystem -> Platform
ModelNumber,Brand --> Year
ModelNumber,Brand --> OperatingSystem

Can’t reduce anything, create FDs through synthesis:
R1(OperatingSystem, Platform)
R2(ModelNumber, Brand, Year, OperatingSystem)

Final answer:
Device1(OperatingSystem: char[20], Platform: char[20]),
Device2(ModelNumber: integer, Brand: char[20], Year: integer, OperatingSystem: char[20])

ServiceDataContains: (only BCNF decomposition)

Primary Key: ServiceProvider, Version
FDs:
ServiceProvider, Version -> MonthlyUsers, MonthlyCosts, MonthlyProfit, Year, Month
Month, Year -> Version
MonthlyRevenue, MonthlyCosts -> MonthlyProfit
MonthlyRevenue, MonthlyProfit -> MonthlyCosts
MonthlyCosts, MonthlyProfit -> MonthlyRevenue

MonthlyRevenue, MonthlyCosts -> MonthlyProfit is not a superkey and is not part of the key so this relation is not in BCNF.

R(ServiceProvider, Version, MonthlyUsers, MonthlyCosts, MonthlyProfit, MonthlyRevenue, Year, Month)

Decompose R using Month, Year -> Version
R1(ServiceProvider, MonthlyUsers, MonthlyCosts, MonthlyProfit, MonthlyRevenue, Year, Month)
R2(Month, Year, Version)

Decompose R1 using MonthlyRevenue, MonthlyCosts -> MonthlyProfit

R3(ServiceProvider, MonthlyUsers, MonthlyCosts, MonthlyRevenue, Year, Month)
R4(MonthlyRevenue, MonthlyCosts, MonthlyProfit)

BCNF is not violated in these so final answer is:
ServiceDataContains1(Month, Year, Version)
ServiceDataContains2(ServiceProvider, MonthlyUsers, MonthlyCosts, MonthlyRevenue, Year, Month)
ServiceDataContains3(MonthlyRevenue, MonthlyCosts, MonthlyProfit)

Review:

Primary Key: Email, ReviewID
FDs: ReviewID -> Rating
We can see that based on the first FD that it's not in 3NF. Because there's only one FD, we already have a minimal closure.
Using synthesis, we get:
With FDs: R1(ReviewId, Rating)
With primary key and unused attributes: R2(Email, ReviewID, ServiceProvider)

Final Answer: Review1(ReviewID: integer, Rating: integer), Review2(Email: char[50], ReviewID: integer, ServiceProvider: char[20])

Subscription:

Primary Key: ServiceProvider, BillingFrequency, CostPerPayment
FDs: BillingFrequency, CostPerPayment->YearlyCost
This FD is not in 3NF, and it is already the minimal closure.
Using synthesis with FDs: R1(BillingFrequency, CostPerPayment, YearlyCost)
with primary key: R2(ServiceProvider, BillingFrequency, CostPerPayment)

Final Answer:
Subscription1(BillingFrequency: char[20], CostPerPayment: double, YearlyCost: double),
Subscription2(ServiceProvider: char[20], BillingFrequency: char[20], CostPerPayment: double)

7. The SQL DDL statements required to create all the tables from item #6.

The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc.

NOTE: ASSERTIONS ARE NOT INCLUDED AS WE HAVE NOT LEARNED THEM YET, HOWEVER THEY ARE LISTED IN Q4, REQUIRED FOR THOSE WITH PARTICIPATION CONSTRAINTS AS THEY ARE MANY TO MANY:
Plays, Streams, Provides, Writes, Makes all need assertions but are not included

```
CREATE TABLE User(
      Email char(50) PRIMARY KEY,
      FirstName char(30),
      LastName char(30),
      Age integer,
      Country char(60),
      PhoneNumber integer UNIQUE);

-- NOTE DEVICE DECOMPOSED INTO TWO TABLES (they have been renamed from the decomposition):

CREATE TABLE DeviceOS(
      OperatingSystem char(20) PRIMARY KEY,
      Platform char(20));

CREATE TABLE DeviceModel(
      ModelNumber integer,
      Brand char(20),
      Year integer,
      OperatingSystem Char(20),
      PRIMARY KEY (ModelNumber, Brand));

-- DECOMPOSED SUBSCRIPTION TABLES:
CREATE TABLE SubscriptionCosts(
      BillingFrequency char(20),
      CostPerPayment double,
      YearlyCost double,
      PRIMARY KEY (BillingFrequency, CostPerPayment));

CREATE TABLE SubscriptionProvider(
      ServiceProvider char(20),
      BillingFrequency char(20),
      CostPerPayment double,
      PRIMARY KEY (ServiceProvider, BillingFrequency, CostPerPayment),
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService);

-- REVIEW DECOMPOSED TABLES:
CREATE TABLE ReviewDetails(
      ReviewID integer PRIMARY KEY,
      Rating integer);

CREATE TABLE ReviewOverview(
      Email char(50),
      ReviewID integer,
      ServiceProvider char(20),
      PRIMARY KEY (Email, ReviewID, ServiceProvider),
      FOREIGN KEY (Email) REFERENCES User
            ON DELETE CASCADE,
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService
            ON DELETE CASCADE);


-- SERVICE DATA CONTAINS DECOMPOSED TABLES:
CREATE TABLE ServiceDataContainsVersion(
      Month integer,
      Year integer,
      Version char(10),
      PRIMARY KEY (Month, Year));

CREATE TABLE ServiceDataContainsCost(
      MonthlyCost integer,
      MonthlyRevenue integer,
      MonthlyProfit integer,
      PRIMARY KEY (MonthlyCost, MonthlyRevenue));

CREATE TABLE ServiceDataContainsProvider(
      ServiceProvider char(20),
      Version char(10),
      MonthlyUsers integer,
      MonthlyCosts integer,
      MonthlyRevenue integer,
      Year integer,
      Month integer,
      PRIMARY KEY (ServiceProvider, Version),
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService
            ON DELETE CASCADE);

CREATE TABLE OnlineService(
      ServiceProvider char(20) PRIMARY KEY);

CREATE TABLE MusicService(
      ServiceProvider char(20) PRIMARY KEY,
      MusicServiceName char(20) UNIQUE NOT NULL,
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService);

CREATE TABLE MovieService(
      ServiceProvider char(20) PRIMARY KEY,
      MovieServiceName char(20) UNIQUE NOT NULL,
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService);

CREATE TABLE GameService(
      ServiceProvider char(20) PRIMARY KEY,
      GameServiceName char(20) UNIQUE NOT NULL,
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService);

CREATE TABLE Song(
      SongName char(20),
      SongYear integer,
      SongGenre char(20),
      PRIMARY KEY (SongName, SongYear));

CREATE TABLE Movie(
      MovieName char(20) NOT NULL,
      MovieYear integer NOT NULL,
      MovieGenre char(20),
      MCompanyName char(20),
      PRIMARY KEY (MovieName, MovieYear),
      FOREIGN KEY (MCompanyName) REFERENCES Movie
            ON DELETE CASCADE);

CREATE TABLE VideoGame(
      GameName char(20),
      GameYear integer,
      GameGenre char(20),
      PRIMARY KEY (GameName, GameYear));

CREATE TABLE Artist(
      ArtistName char(20) PRIMARY KEY);

CREATE TABLE MovieCompany(
      MCompanyName char(20) PRIMARY KEY);

CREATE TABLE Developer(
      DevName char(20) PRIMARY KEY);

CREATE TABLE Uses(
      Email char(50),
      ModelNumber integer,
      Brand integer,
      FOREIGN KEY (Email) REFERENCES User,
      FOREIGN KEY (ModelNumber, Brand) references Device,
      PRIMARY KEY (Email, ModelNumber, Brand));

CREATE TABLE SubscribesTo(
      Email char(50),
      ServiceProvider char(20),
      BillingFrequency char(20),
      CostPerPayment double,
      YearlyCost double,
      PRIMARY KEY (Email, BillingFrequency, CostPerPayment,
ServiceProvider),
      FOREIGN KEY (Email) REFERENCES email,
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService,
      FOREIGN KEY (BillingFrequency, CostPerPayment, ServiceProvider)
REFERENCES Subscription(BillingFrequency, CostPerPayment, ServiceProvider));

CREATE TABLE Accesses(
      ModelNumber integer,
      Brand char(20),
      ServiceProvider char(20),
      FOREIGN KEY (ModelNumber, Brand) REFERENCES Device,
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService,
      PRIMARY KEY (ModelNumber, Brand, ServiceProvider));

CREATE TABLE Plays(
      ServiceProvider char(20),
      SongName char(20),
      SongYear integer,
      NumPlays integer,
      PRIMARY KEY (ServiceProvider, SongName, SongYear),
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService,
      FOREIGN KEY (SongName, SongYear) REFERENCES Song);

CREATE TABLE Streams(
      ServiceProvider char(20),
      MovieName char(20),
      MovieYear integer,
      NumStreams integer,
      PRIMARY KEY (ServiceProvider, MovieName, MovieYear),
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService,
      FOREIGN KEY (MovieName, MovieYear) REFERENCES Movie);

CREATE TABLE Provides(
      ServiceProvider char(20),
      GameName char(20),
      GameYear integer,
      TimePlayed integer,
      PRIMARY KEY (ServiceProvider, GameName, GameYear)
      FOREIGN KEY (ServiceProvider) REFERENCES OnlineService,
      FOREIGN KEY (GameName, GameYear) REFERENCES GAME);

CREATE TABLE Writes(
      SongName char(20),
      SongYear integer,
      ArtistName char(20),
      PRIMARY KEY (SongName, SongYear, ArtistName),
      FOREIGN KEY (SongName, SongYear) REFERENCES Song,
      FOREIGN KEY (ArtistName) REFERENCES Artist);

CREATE TABLE Makes(
      GameName char(20),
      GameYear integer,
      DevName char(20),
      PRIMARY KEY (GameName, GameYear, DevName),
      FOREIGN KEY (GameName, GameYear) REFERENCES Game,
      FOREIGN KEY (DevName) REFERENCES DEVELOPER);
```

8. INSERT statements to populate each table with at least 5 tuples.

You will likely want to have more than 5 tuples so that you can have meaningful queries later on.

NOTE: some placeholder names are used and will be changed later when adding in actual tuples for tables

```
INSERT INTO User (Email, FirstName, LastName, Age, Country, PhoneNumber)
VALUES ('JohnDoe@gmail.com', 'John', 'Doe', 25, 'United States',
1234567890);
INSERT INTO User (Email, FirstName, LastName, Age, Country, PhoneNumber)
VALUES ('JaneSmith@outlook.com', 'Jane', 'Smith', 30, 'Canada', 9876543210);
INSERT INTO User (Email, FirstName, LastName, Age, Country, PhoneNumber)
VALUES ('MichaelJohnson@hotmail.com', 'Michael', 'Johnson', 40, 'Australia',
5678901234);
INSERT INTO User (Email, FirstName, LastName, Age, Country, PhoneNumber)
VALUES ('EmilyBrown@yahoo.com', 'Emily', 'Brown', 35, 'United Kingdom',
9012345678);
```

```
INSERT INTO User (Email, FirstName, LastName, Age, Country, PhoneNumber)
VALUES ('DavidWilson@webmail.com', 'David', 'Wilson', 28, 'Germany',
3456789012);

INSERT INTO DeviceOS (OperatingSystem, Platform) VALUES ('iOS', 'Mobile');
INSERT INTO DeviceOS (OperatingSystem, Platform) VALUES ('Windows', 'PC');
INSERT INTO DeviceOS (OperatingSystem, Platform) VALUES ('Android',
'Mobile');
INSERT INTO DeviceOS (OperatingSystem, Platform) VALUES ('macOS', 'PC');
INSERT INTO DeviceOS (OperatingSystem, Platform) VALUES ('Linux', 'PC');

INSERT INTO DeviceModel (ModelNumber, Brand, Year, OperatingSystem) VALUES
(12345, 'Samsung', 2022, 'Android');
INSERT INTO DeviceModel (ModelNumber, Brand, Year, OperatingSystem) VALUES
(54321, 'Google', 2023,'Linux');
INSERT INTO DeviceModel (ModelNumber, Brand, Year, OperatingSystem) VALUES
(67890, 'Apple', 2021, 'iOs');
INSERT INTO DeviceModel (ModelNumber, Brand, Year, OperatingSystem) VALUES
(98765, 'Microsoft', 2020, 'macOS');
INSERT INTO DeviceModel (ModelNumber, Brand, Year, OperatingSystem) VALUES
(23456, 'Sony', 2022, 'Windows');

INSERT INTO SubscriptionCosts (BillingFrequency, CostPerPayment, YearlyCost)
VALUES ('Monthly', 9.99, 119.88);
INSERT INTO SubscriptionCosts (BillingFrequency, CostPerPayment, YearlyCost)
VALUES ('Quarterly', 24.99, 99.96);
INSERT INTO SubscriptionCosts (BillingFrequency, CostPerPayment, YearlyCost)
VALUES ('Yearly', 89.99, 89.99);
INSERT INTO SubscriptionCosts (BillingFrequency, CostPerPayment, YearlyCost)
VALUES ('Monthly', 12.99, 155.88);
INSERT INTO SubscriptionCosts (BillingFrequency, CostPerPayment, YearlyCost)
VALUES ('Annual', 49.99, 49.99);

INSERT INTO SubscriptionProvider (ServiceProvider, BillingFrequency,
CostPerPayment) VALUES ('Netflix', 'Monthly', 9.99);
INSERT INTO SubscriptionProvider (ServiceProvider, BillingFrequency,
CostPerPayment) VALUES ('Hulu', 'Monthly', 11.99);
INSERT INTO SubscriptionProvider (ServiceProvider, BillingFrequency,
CostPerPayment) VALUES ('Amazon Prime', 'Annual', 119.00);
INSERT INTO SubscriptionProvider (ServiceProvider, BillingFrequency,
CostPerPayment) VALUES ('Disney+', 'Yearly', 79.99);
INSERT INTO SubscriptionProvider (ServiceProvider, BillingFrequency,
CostPerPayment) VALUES ('Xbox Game Pass', 'Monthly', 9.99);
```

```
INSERT INTO ReviewDetails (ReviewID, Rating) VALUES (1, 4);
INSERT INTO ReviewDetails (ReviewID, Rating) VALUES (2, 5);
INSERT INTO ReviewDetails (ReviewID, Rating) VALUES (3, 3);
INSERT INTO ReviewDetails (ReviewID, Rating) VALUES (4, 4);
INSERT INTO ReviewDetails (ReviewID, Rating) VALUES (5, 2);

INSERT INTO ReviewOverview (Email, ReviewID, ServiceProvider) VALUES
('JohnDoe@gmail.com', 1, 'Netflix');
INSERT INTO ReviewOverview (Email, ReviewID, ServiceProvider) VALUES
('JaneSmith@outlook.com', 2, 'Amazon Prime');
INSERT INTO ReviewOverview (Email, ReviewID, ServiceProvider) VALUES
('MichaelJohnson@hotmail.com', 3, 'Hulu');
INSERT INTO ReviewOverview (Email, ReviewID, ServiceProvider) VALUES
('EmilyBrown@yahoo.com', 4, 'Disney+');
INSERT INTO ReviewOverview (Email, ReviewID, ServiceProvider) VALUES
('DavidWilson@webmail.com', 5, 'Spotify');

INSERT INTO ServiceDataContainsVersion (Month, Year, Version) VALUES (1,
2023, '1.0');
INSERT INTO ServiceDataContainsVersion (Month, Year, Version) VALUES (2,
2023, '1.1');
INSERT INTO ServiceDataContainsVersion (Month, Year, Version) VALUES (3,
2023, '1.2');
INSERT INTO ServiceDataContainsVersion (Month, Year, Version) VALUES (4,
2023, '1.3');
INSERT INTO ServiceDataContainsVersion (Month, Year, Version) VALUES (5,
2023, '1.4');

INSERT INTO ServiceDataContainsCost (MonthlyCost, MonthlyRevenue,
MonthlyProfit) VALUES (1000, 5000, 4000);
INSERT INTO ServiceDataContainsCost (MonthlyCost, MonthlyRevenue,
MonthlyProfit) VALUES (1500, 6000, 4500);
INSERT INTO ServiceDataContainsCost (MonthlyCost, MonthlyRevenue,
MonthlyProfit) VALUES (1200, 5500, 4300);
INSERT INTO ServiceDataContainsCost (MonthlyCost, MonthlyRevenue,
MonthlyProfit) VALUES (1800, 7000, 5200);
INSERT INTO ServiceDataContainsCost (MonthlyCost, MonthlyRevenue,
MonthlyProfit) VALUES (900, 4500, 3600);

INSERT INTO ServiceDataContainsProvider (ServiceProvider, Version,
MonthlyUsers, MonthlyCosts, MonthlyRevenue, Year, Month) VALUES ('Netflix',
'1.0', 100000, 1000, 5000, 2023, 1);
```

```
INSERT INTO ServiceDataContainsProvider (ServiceProvider, Version,
MonthlyUsers, MonthlyCosts, MonthlyRevenue, Year, Month) VALUES ('Hulu',
'1.0', 80000, 1500, 6000, 2023, 2);
INSERT INTO ServiceDataContainsProvider (ServiceProvider, Version,
MonthlyUsers, MonthlyCosts, MonthlyRevenue, Year, Month) VALUES ('Amazon
Prime', '1.0', 120000, 1200, 5500, 2023, 3);
INSERT INTO ServiceDataContainsProvider (ServiceProvider, Version,
MonthlyUsers, MonthlyCosts, MonthlyRevenue, Year, Month) VALUES ('Disney+',
'1.0', 90000, 1800, 7000, 2023, 4);
INSERT INTO ServiceDataContainsProvider (ServiceProvider, Version,
MonthlyUsers, MonthlyCosts, MonthlyRevenue, Year, Month) VALUES ('Spotify',
'1.0', 110000, 900, 4500, 2023, 5);

-- Music Service Providers
INSERT INTO OnlineService (ServiceProvider) VALUES ('Spotify');
INSERT INTO OnlineService (ServiceProvider) VALUES ('Amazon Prime Music');
INSERT INTO OnlineService (ServiceProvider) VALUES ('Tidal');
INSERT INTO OnlineService (ServiceProvider) VALUES ('Deezer');
INSERT INTO OnlineService (ServiceProvider) VALUES ('Pandora');

-- Music and Movie Service Providers
INSERT INTO OnlineService (ServiceProvider) VALUES ('Amazon Prime');

-- Movie Service Providers
INSERT INTO OnlineService (ServiceProvider) VALUES ('Netflix');
INSERT INTO OnlineService (ServiceProvider) VALUES ('Hulu');
INSERT INTO OnlineService (ServiceProvider) VALUES ('Disney+');
INSERT INTO OnlineService (ServiceProvider) VALUES ('HBO Max');

-- Game Service Providers
INSERT INTO OnlineService (ServiceProvider) VALUES ('Steam');
INSERT INTO OnlineService (ServiceProvider) VALUES ('Epic Games');
INSERT INTO OnlineService (ServiceProvider) VALUES ('Origin');
INSERT INTO OnlineService (ServiceProvider) VALUES ('Xbox Game Pass');
INSERT INTO OnlineService (ServiceProvider) VALUES ('PlayStation Now');

INSERT INTO MusicService (ServiceProvider, MusicServiceName) VALUES
('Spotify', 'Spotify Music');
INSERT INTO MusicService (ServiceProvider, MusicServiceName) VALUES ('Amazon
Prime', 'Amazon Prime Music');
INSERT INTO MusicService (ServiceProvider, MusicServiceName) VALUES
('Tidal', 'Tidal Music');
```

```
INSERT INTO MusicService (ServiceProvider, MusicServiceName) VALUES
('Deezer', 'Deezer Music');
INSERT INTO MusicService (ServiceProvider, MusicServiceName) VALUES
('Pandora', 'Pandora Music');

INSERT INTO MovieService (ServiceProvider, MovieServiceName) VALUES
('Netflix', 'Netflix Movies');
INSERT INTO MovieService (ServiceProvider, MovieServiceName) VALUES ('Amazon
Prime', 'Amazon Prime Video');
INSERT INTO MovieService (ServiceProvider, MovieServiceName) VALUES ('Hulu',
'Hulu Movies');
INSERT INTO MovieService (ServiceProvider, MovieServiceName) VALUES
('Disney+', 'Disney+ Movies');
INSERT INTO MovieService (ServiceProvider, MovieServiceName) VALUES ('HBO
Max', 'HBO Max Movies');

INSERT INTO GameService (ServiceProvider, GameServiceName) VALUES ('Steam',
'Steam Games');
INSERT INTO GameService (ServiceProvider, GameServiceName) VALUES ('Epic
Games', 'Epic Games Store');
INSERT INTO GameService (ServiceProvider, GameServiceName) VALUES ('Origin',
'Origin Games');
INSERT INTO GameService (ServiceProvider, GameServiceName) VALUES ('Xbox
Game Pass', 'Xbox Game Pass');
INSERT INTO GameService (ServiceProvider, GameServiceName) VALUES
('PlayStation Now', 'PlayStation Now Games');

INSERT INTO Song (SongName, SongYear, SongGenre) VALUES ('Song 1', 2020,
'Pop');
INSERT INTO Song (SongName, SongYear, SongGenre) VALUES ('Song 2', 2018,
'Rock');
INSERT INTO Song (SongName, SongYear, SongGenre) VALUES ('Song 3', 2021,
'Hip Hop');
INSERT INTO Song (SongName, SongYear, SongGenre) VALUES ('Song 4', 2021,
'Hip Hop');
INSERT INTO Song (SongName, SongYear, SongGenre) VALUES ('Song 5', 2021,
'Hip Hop');

INSERT INTO Movie (MovieName, MovieYear, MovieGenre, MCompanyName) VALUES
('Ready Player One', 2020, 'Sci-fi', 'Warner Bros.');
INSERT INTO Movie (MovieName, MovieYear, MovieGenre, MCompanyName) VALUES
('Jurrasic World', 2019, 'Action', 'Universal Pictures');
```

```
INSERT INTO Movie (MovieName, MovieYear, MovieGenre, MCompanyName) VALUES
('Skyfall', 2021, 'Drama', 'Columbia Pictures');
INSERT INTO Movie (MovieName, MovieYear, MovieGenre, MCompanyName) VALUES
('Pokémon Detective Pikachu', 2018, 'Adventure', 'Warner Bros.');
INSERT INTO Movie (MovieName, MovieYear, MovieGenre, MCompanyName) VALUES
('Furious 7', 2022, 'Science Fiction', 'Universal Pictures');

INSERT INTO VideoGame (GameName, GameYear, GameGenre) VALUES ('Game 1',
2020, 'Action');
INSERT INTO VideoGame (GameName, GameYear, GameGenre) VALUES ('Game 2',
2019, 'Adventure');
INSERT INTO VideoGame (GameName, GameYear, GameGenre) VALUES ('Game 3',
2021, 'RPG');
INSERT INTO VideoGame (GameName, GameYear, GameGenre) VALUES ('Game 4',
2018, 'Shooter');
INSERT INTO VideoGame (GameName, GameYear, GameGenre) VALUES ('Game 5',
2022, 'Sports');

INSERT INTO Artist (ArtistName) VALUES ('Artist 1');
INSERT INTO Artist (ArtistName) VALUES ('Artist 2');
INSERT INTO Artist (ArtistName) VALUES ('Artist 3');
INSERT INTO Artist (ArtistName) VALUES ('Artist 4');
INSERT INTO Artist (ArtistName) VALUES ('Artist 5');

INSERT INTO MovieCompany (MCompanyName) VALUES ('Warner Bros.');
INSERT INTO MovieCompany (MCompanyName) VALUES ('Universal Pictures');
INSERT INTO MovieCompany (MCompanyName) VALUES ('Columbia Pictures');
INSERT INTO MovieCompany (MCompanyName) VALUES ('Walt Disney Pictures');
INSERT INTO MovieCompany (MCompanyName) VALUES ('Marvel Studios');

INSERT INTO Developer (DevName) VALUES ('Developer 1');
INSERT INTO Developer (DevName) VALUES ('Developer 2');
INSERT INTO Developer (DevName) VALUES ('Developer 3');
INSERT INTO Developer (DevName) VALUES ('Developer 4');
INSERT INTO Developer (DevName) VALUES ('Developer 5');

INSERT INTO Uses (Email, ModelNumber, Brand) VALUES ('JohnDoe@gmail.com',
12345, 'Samsung');
INSERT INTO Uses (Email, ModelNumber, Brand) VALUES
('JaneSmith@outlook.com', 67890, 'Apple');
INSERT INTO Uses (Email, ModelNumber, Brand) VALUES
('MichaelJohnson@hotmail.com', 54321, 'Google');
```

```sql
INSERT INTO Uses (Email, ModelNumber, Brand) VALUES ('EmilyBrown@yahoo.com',
98765, 'Microsoft');
INSERT INTO Uses (Email, ModelNumber, Brand) VALUES
('DavidWilson@webmail.com', 23456, 'Sony');

INSERT INTO SubscribesTo (Email, ServiceProvider, BillingFrequency,
CostPerPayment, YearlyCost) VALUES ('JohnDoe@gmail.com', 'Netflix',
'Monthly', 9.99, 119.88);
INSERT INTO SubscribesTo (Email, ServiceProvider, BillingFrequency,
CostPerPayment, YearlyCost) VALUES ('JaneSmith@outlook.com', 'Hulu',
'Monthly', 11.99, 143.88);
INSERT INTO SubscribesTo (Email, ServiceProvider, BillingFrequency,
CostPerPayment, YearlyCost) VALUES ('MichaelJohnson@hotmail.com', 'Amazon
Prime', 'Annual', 119.00, 119.00);
INSERT INTO SubscribesTo (Email, ServiceProvider, BillingFrequency,
CostPerPayment, YearlyCost) VALUES ('EmilyBrown@yahoo.com', 'Disney+',
'Yearly', 79.99, 79.99);
INSERT INTO SubscribesTo (Email, ServiceProvider, BillingFrequency,
CostPerPayment, YearlyCost) VALUES ('DavidWilson@webmail.com', 'Spotify',
'Monthly', 9.99, 119.88);

INSERT INTO Accesses (ModelNumber, Brand, ServiceProvider) VALUES (12345,
'Samsung', 'Netflix');
INSERT INTO Accesses (ModelNumber, Brand, ServiceProvider) VALUES (67890,
'Apple', 'Hulu');
INSERT INTO Accesses (ModelNumber, Brand, ServiceProvider) VALUES (54321,
'Google', 'Amazon Prime');
INSERT INTO Accesses (ModelNumber, Brand, ServiceProvider) VALUES (98765,
'Microsoft', 'Disney+');
INSERT INTO Accesses (ModelNumber, Brand, ServiceProvider) VALUES (23456,
'Sony', 'Spotify');

INSERT INTO Plays (ServiceProvider, SongName, SongYear, NumPlays) VALUES
('Spotify', 'Song 1', 2020, 100);
INSERT INTO Plays (ServiceProvider, SongName, SongYear, NumPlays) VALUES
('Amazon Prime Music', 'Song 2', 2019, 50);
INSERT INTO Plays (ServiceProvider, SongName, SongYear, NumPlays) VALUES
('Tidal', 'Song 3', 2021, 75);
INSERT INTO Plays (ServiceProvider, SongName, SongYear, NumPlays) VALUES
('Deezer', 'Song 4', 2018, 120);
INSERT INTO Plays (ServiceProvider, SongName, SongYear, NumPlays) VALUES
('Pandora', 'Song 5', 2022, 90);
```

```
INSERT INTO Streams (ServiceProvider, MovieName, MovieYear, NumStreams)
VALUES ('Amazon Prime', 'Ready Player One', 2020, 200);
INSERT INTO Streams (ServiceProvider, MovieName, MovieYear, NumStreams)
VALUES ('Netflix', 'Jurrasic World', 2019, 150);
INSERT INTO Streams (ServiceProvider, MovieName, MovieYear, NumStreams)
VALUES ('Hulu', 'Skyfall', 2021, 180);
INSERT INTO Streams (ServiceProvider, MovieName, MovieYear, NumStreams)
VALUES ('Disney+', 'Pokémon Detective Pikachu', 2018, 220);
INSERT INTO Streams (ServiceProvider, MovieName, MovieYear, NumStreams)
VALUES ('HBO Max', 'Furious 7', 2022, 160);

INSERT INTO Provides (ServiceProvider, GameName, GameYear, TimePlayed)
VALUES ('Steam', 'Game 1', 2020, 10);
INSERT INTO Provides (ServiceProvider, GameName, GameYear, TimePlayed)
VALUES ('Epic Games', 'Game 2', 2019, 8);
INSERT INTO Provides (ServiceProvider, GameName, GameYear, TimePlayed)
VALUES ('Origin', 'Game 3', 2021, 12);
INSERT INTO Provides (ServiceProvider, GameName, GameYear, TimePlayed)
VALUES ('Xbox Game Pass', 'Game 4', 2018, 15);
INSERT INTO Provides (ServiceProvider, GameName, GameYear, TimePlayed)
VALUES ('PlayStation Now', 'Game 5', 2022, 20);

INSERT INTO Writes (SongName, SongYear, ArtistName) VALUES ('Song 1', 2020,
'Artist 1');
INSERT INTO Writes (SongName, SongYear, ArtistName) VALUES ('Song 2', 2019,
'Artist 2');
INSERT INTO Writes (SongName, SongYear, ArtistName) VALUES ('Song 3', 2021,
'Artist 3');
INSERT INTO Writes (SongName, SongYear, ArtistName) VALUES ('Song 4', 2018,
'Artist 4');
INSERT INTO Writes (SongName, SongYear, ArtistName) VALUES ('Song 5', 2018,
'Artist 5');

INSERT INTO Makes (GameName, GameYear, DevName) VALUES ('Game 1', 2020,
'Developer 1');
INSERT INTO Makes (GameName, GameYear, DevName) VALUES ('Game 2', 2019,
'Developer 2');
INSERT INTO Makes (GameName, GameYear, DevName) VALUES ('Game 3', 2021,
'Developer 3');
INSERT INTO Makes (GameName, GameYear, DevName) VALUES ('Game 4', 2018,
'Developer 4');
INSERT INTO Makes (GameName, GameYear, DevName) VALUES ('Game 5', 2022,
'Developer 5');
```