

University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: February 27

Group Number: 8

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Kat Duangkham	31579329	i7g3b	katduangkham@yahoo.ca
Chanaldy Soenarjo	58643727	l1c0s	chsoenarjo@gmail.com
Glen Ren	76331917	u8u1x	glenrenzh@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 email address, and then let us assign you to a TA for your project supervisor.)

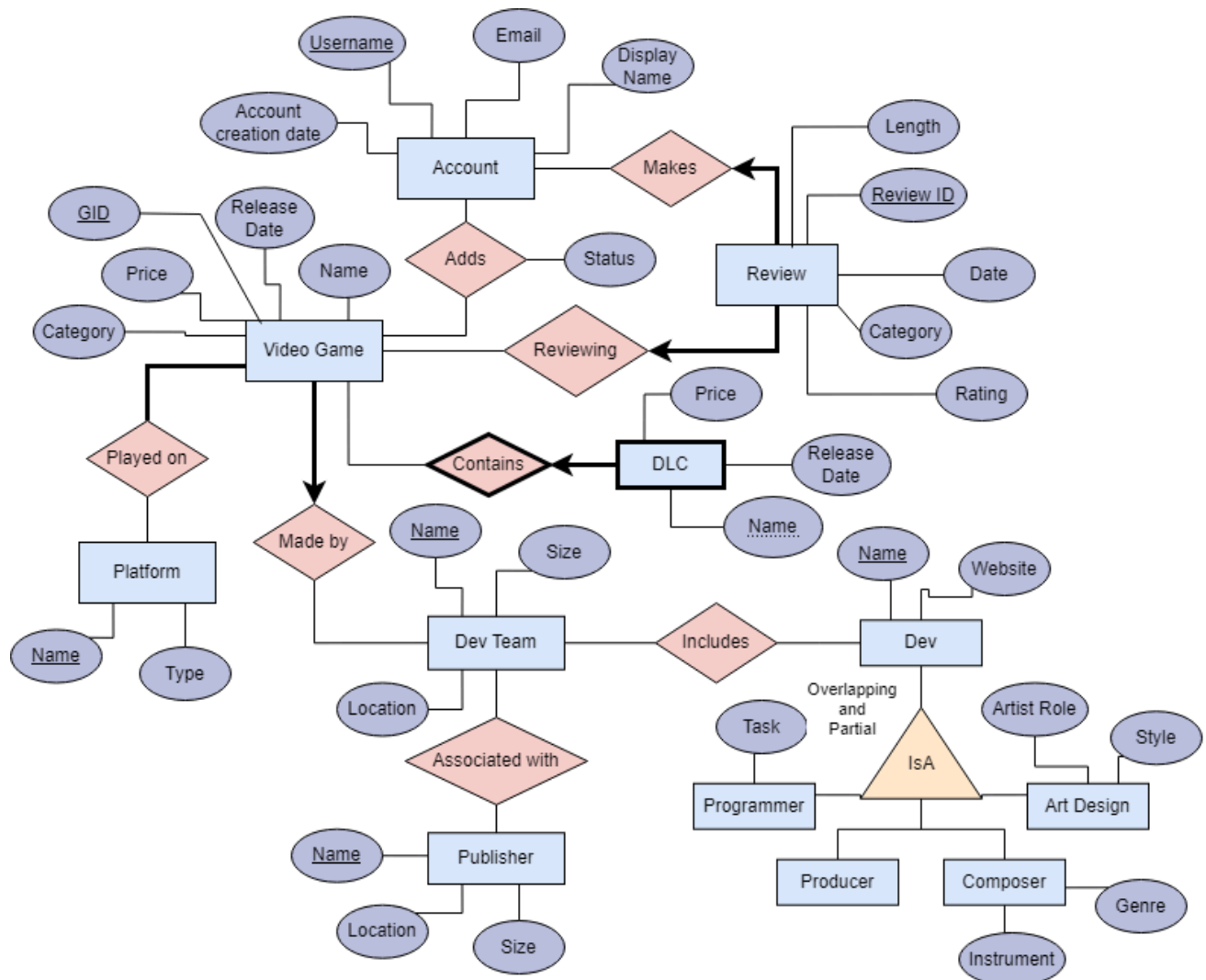
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Description

The domain of our application would be product information regarding video games, as well as personal user data, such as product review, completion status, and ownership. This is modelled by the use of having a user account, in which video games can be added. Information about the video game is also stored, such as the main developers, publisher, and any DLC (downloadable content) it has. A user may add a review to the video game as well. As such, this program would be ideal for tracking what video games a person has played, wants to play, or is playing, all in one spot regardless of what platform that video game is on.

ER Diagram

We added the attribute GID to Video Game and made it the primary key instead of Release Date and Name. We also added Length and Category to the Review Entity. The constraints on the IsA relationship is now labelled as overlapping and partial.



Relational Schema (Note: Underline is primary key, bold is foreign key)

Producer(**Name**: VARCHAR)

Composer(**Name**: VARCHAR, Genre: VARCHAR, Instrument: VARCHAR)

ArtDesign(**Name**: VARCHAR, ArtistRole: VARCHAR, Style: VARCHAR)

Programmer(**Name**: VARCHAR, Task: VARCHAR)
**Name is Dev.Name for all four schemas above*

Dev(**Name**: VARCHAR, Website: VARCHAR)

Includes(**DevTeamName**: VARCHAR, **DevName**: VARCHAR)
**DevTeamName is DevTeam.Name*
**DevName is Dev.Name*

Publisher(**Name**: VARCHAR, Location: VARCHAR, Size: INT)

AssociatedWith(**DevTeamName**: VARCHAR, **PublisherName**: VARCHAR)
**DevTeamName is DevTeam.Name*
**PublisherName is Publisher.Name*

DevTeam(**Name**: VARCHAR, Size: INT, Location: VARCHAR)

Platform(**Name**: VARCHAR, Type: VARCHAR)

PlayedOn(**GID**: INT, **Name**: VARCHAR)
**GID is Game.GID*
**Name is Platform.Name*

ContainsDLC(**GID**: INT, **Name**: VARCHAR, Price: FLOAT(24), ReleaseDate: DATE)
**GID is Game.GID*

VideoGameMadeBy(**GID**: INT, Name: VARCHAR, ReleaseDate: DATE, Price: FLOAT(24),
Category: VARCHAR, **DevTeamName**: VARCHAR NOT NULL)
**DevTeamName is DevTeam.Name*

Adds(**Username**: VARCHAR, **GID**: INT, Status: VARCHAR)
**Username is Account.Username*
**GID is Game.GID*

MakesReviewReviewing(ReviewID: INT, Date: DATE, Rating: INT, Category: VARCHAR, Length: INT, **Username**: VARCHAR NOT NULL, **GID**: INT NOT NULL)

**Username is Account.Username*

**GID is Game.GID*

Account(Username: VARCHAR, Email: VARCHAR UNIQUE NOT NULL, DisplayName: VARCHAR, CreationDate: DATE) **Note that Email is a candidate key for Account*

Functional Dependencies

Account.Username -> Account.Email, Account.DisplayName, Account.CreationDate

Account.Email -> Account.Username, Account.DisplayName, Account.CreationDate

MakesReviewReviewing.ReviewID -> MakesReviewReviewing.Date,
MakesReviewReviewing.Rating, MakesReviewReviewing.Category,
MakesReviewReviewing.Length, MakesReviewReviewing.Username,
MakesReviewReviewing.GID

MakesReviewReviewing.Length -> MakesReviewReviewing.Category

Adds.Username, Adds.GID -> Adds.Status

VideoGameMadeBy.GID -> VideoGameMadeBy.Name, VideoGameMadeBy.ReleaseDate,
VideoGameMadeBy.Price, VideoGameMadeBy.Category, VideoGameMadeBy.DevTeamName

ContainsDLC.GID, ContainsDLC.Name -> ContainsDLC.Price, ContainsDLC.ReleaseDate

Platform.Name -> Platform.Type

DevTeam.Name -> DevTeam.Size, DevTeam.Location

Publisher.Name -> Publisher.Location, Publisher.Size

Dev.Name -> Dev.Website

Programmer.Name -> Programmer.Task

ArtDesign.Name -> ArtDesign.ArtistRole, ArtDesign.Style

Composer.Name -> Composer.Instrument, Composer.Genre

Normalization

We decided to normalize to 3NF.

For our tables to be valid in 3NF, its non-trivial functional dependencies must either have the attributes on the left-hand side be a superkey, or have the attributes on the right-hand side be part of a candidate key.

Looking at our functional dependencies, the only violation of those rules is:

MakesReviewReviewing.Length -> MakesReviewReviewing.Category

as MakesReviewReviewing doesn't have Length as a superkey, and its only candidate key is ReviewID so it passes neither requirement.

All of our other schemas would already be in 3NF as their functional dependencies only have primary keys or candidate keys on the left-hand side. Thus, nothing needs to be changed except for MakesReviewReviewing.

This was our process of normalizing MakesReviewReviewing to be in 3NF:

MakesReviewReviewing(ReviewID, Date, Rating, Category, Length, Username, GID)

FD's: ReviewID \rightarrow Date, Rating, Category, Length, Username, GID

Length \rightarrow Category

In order to be in 3NF, the LHS of any non-trivial FD must be a superkey, or the RHS is part of a candidate key.

Length \rightarrow Category violates 3NF as Length isn't a superkey, nor is Category part of any candidate key (MakesReviewReviewing's only key is ReviewID).

So to normalize MakesReviewReviewing to 3NF, first we need to find its min. cover.

1. Decompose all FD's:

Before

ReviewID \rightarrow Date, Rating, Category, Length, Username, GID

Length \rightarrow Category

After

- ① ReviewID \rightarrow Date ③ ReviewID \rightarrow Category ⑤ ReviewID \rightarrow Username
② ReviewID \rightarrow Rating ④ ReviewID \rightarrow Length ⑥ ReviewID \rightarrow GID
⑦ Length \rightarrow Category

2. Minimize LHS of FD's:

Since all LHS of our FDs have only one attribute, no further minimization

3. Delete redundant FD's:

ReviewID⁺: {ReviewID, Date, Rating, Category, Length, Username, GID}

ReviewID⁺ without ①: {ReviewID, Rating, Category, Length, Username, GID} \times

ReviewID⁺ without ②: {ReviewID, Date, Category, Length, Username, GID} \times

ReviewID⁺ without ③: {ReviewID, Date, Rating, Length, Username, GID} \times

ReviewID⁺ without ④: {ReviewID, Date, Rating, Category, Username, GID} \times

ReviewID⁺ without ⑤: {ReviewID, Date, Rating, Category, Length, GID} \times

ReviewID⁺ without ⑥: {ReviewID, Date, Rating, Category, Length, Username} \times

Length⁺: {Length, Category}

Length⁺ without ⑦: {Length} \times

Since the closures without each FD are not the same, none are redundant.

So the minimal
cover of
MakesReviewReviewing
are these FD's:

ReviewID \rightarrow Date	ReviewID \rightarrow Category	ReviewID \rightarrow Username
ReviewID \rightarrow Rating	ReviewID \rightarrow Length	ReviewID \rightarrow GID
Length \rightarrow Category		

Now using these FD's, we can decompose MakesReviewReviewing.

First, Length \rightarrow Category violates 3NF, so we decompose to BCNF using it:



We end up with MakesReviewReviewing₁(ReviewID, Date, Rating, Length, Username, GID)

and MakesReviewReviewing₂(Length, Category).

MakesReviewReviewing₁ doesn't violate BCNF since its only FDs has ReviewID, the key, on LHS.

MakesReviewReviewing₂ doesn't violate BCNF since it only has 2 attributes.

So those are our final decomposition results.

ReviewID \rightarrow Category from our min. cover wasn't preserved by decomposition, so we add

MakesReviewReviewing₃(ReviewID, Category).

So to reiterate, our final schemas after normalizing to 3NF are:

Producer(**Name**: VARCHAR)

Composer(**Name**: VARCHAR, Genre: VARCHAR, Instrument: VARCHAR)

ArtDesign(**Name**: VARCHAR, ArtistRole: VARCHAR, Style: VARCHAR)

Programmer(**Name**: VARCHAR, Task: VARCHAR)

Dev(**Name**: VARCHAR, Website: VARCHAR)

Includes(**DevTeamName**: VARCHAR, **DevName**: VARCHAR)

Publisher(**Name**: VARCHAR, Location: VARCHAR, Size: INT)

AssociatedWith(**DevTeamName**: VARCHAR, **PublisherName**: VARCHAR)

DevTeam(**Name**: VARCHAR, Size: INT, Location: VARCHAR)

Platform(**Name**: VARCHAR, Type: VARCHAR)

PlayedOn(**GID**: INT, **Name**: VARCHAR)

ContainsDLC(**GID**: INT, **Name**: VARCHAR, Price: FLOAT(24), ReleaseDate: DATE)

VideoGameMadeBy(**GID**: INT, Name: VARCHAR, ReleaseDate: DATE, Price: FLOAT(24),
Category: VARCHAR, **DevTeamName**: VARCHAR NOT NULL)

Adds(**Username**: VARCHAR, **GID**: INT, Status: VARCHAR)

MakesReviewReviewing1(**ReviewID**: INT, Date: DATE, Rating: INT, **Length**: INT NOT NULL,
Username: VARCHAR NOT NULL, **GID**: INT)

**Length is MakesReviewReviewing2.Length*

**Username is Account.Username*

**GID is Game.GID*

MakesReviewReviewing2(**Length**: INT, Category: VARCHAR)

MakesReviewReviewing3(**ReviewID**: INT, Category: VARCHAR)

**ReviewID is MakesReviewReviewing1.ReviewID*

Account(**Username**: VARCHAR, Email: VARCHAR UNIQUE NOT NULL, DisplayName: VARCHAR,
CreationDate: DATE) **Note that Email is a candidate key for Account*

*Note: Underline is primary key, **bold** is foreign key, all foreign keys are the same as before (referencing the same attribute from the Relational Schema section) if unspecified.*

And here are our functional dependencies after normalization (*italicized* are the changes):

Account.Username -> Account.Email, Account.DisplayName, Account.CreationDate

Account.Email -> Account.Username, Account.DisplayName, Account.CreationDate

*MakesReviewReviewing1.ReviewID -> MakesReviewReviewing1.Date,
MakesReviewReviewing1.Rating, MakesReviewReviewing1.Length,
MakesReviewReviewing1.Username, MakesReviewReviewing1.GID*

MakesReviewReviewing2.Length -> MakesReviewReviewing2.Category

MakesReviewReviewing3.ReviewID -> MakesReviewReviewing3.Category

Adds.Username, Adds.GID -> Adds.Status

VideoGameMadeBy.GID -> VideoGameMadeBy.Name, VideoGameMadeBy.ReleaseDate,
VideoGameMadeBy.Price, VideoGameMadeBy.Category, VideoGameMadeBy.DevTeamName

ContainsDLC.GID, ContainsDLC.Name -> ContainsDLC.Price, ContainsDLC.ReleaseDate

Platform.Name -> Platform.Type

DevTeam.Name -> DevTeam.Size, DevTeam.Location

Publisher.Name -> Publisher.Location, Publisher.Size

Dev.Name -> Dev.Website

Programmer.Name -> Programmer.Task

ArtDesign.Name -> ArtDesign.ArtistRole, ArtDesign.Style

Composer.Name -> Composer.Instrument, Composer.Genre

SQL DDL Statements

```
CREATE TABLE Producer (  
    Name VARCHAR,  
    PRIMARY KEY (Name),  
    FOREIGN KEY (Name) REFERENCES Dev)
```

```
CREATE TABLE Composer (  
    Name VARCHAR,  
    PRIMARY KEY (Name),  
    FOREIGN KEY (Name) REFERENCES Dev,  
    Genre VARCHAR,  
    Instrument VARCHAR)
```

```
CREATE TABLE ArtDesign (  
    Name VARCHAR,  
    PRIMARY KEY (Name),  
    FOREIGN KEY (Name) REFERENCES Dev,  
    ArtistRole VARCHAR,  
    Style VARCHAR)
```

```
CREATE TABLE Programmer (  
    Name VARCHAR,  
    PRIMARY KEY (Name),  
    FOREIGN KEY (Name) REFERENCES Dev,  
    Task VARCHAR)
```

```
CREATE TABLE Dev (  
    Name VARCHAR,  
    Website VARCHAR,  
    PRIMARY KEY (Name))
```

```
CREATE TABLE Includes (  
    DevTeamName VARCHAR,  
    DevName VARCHAR,  
    PRIMARY KEY (DevTeamName, DevName),  
    FOREIGN KEY (DevTeamName) REFERENCES DevTeam(Name),  
    FOREIGN KEY (DevName) REFERENCES Dev (Name))
```

```
CREATE TABLE Publisher (  
    Name VARCHAR PRIMARY KEY,  
    Location VARCHAR,  
    Size INT)
```

```
CREATE TABLE AssociatedWith (  
    DevTeamName VARCHAR,  
    PublisherName VARCHAR,  
    PRIMARY KEY (DevTeamName, PublisherName),  
    FOREIGN KEY (DevTeamName) REFERENCES DevTeam(Name),  
    FOREIGN KEY (PublisherName) REFERENCES Publisher(Name))
```

```
CREATE TABLE DevTeam (  
    Name VARCHAR,  
    Size INT,  
    Location VARCHAR,  
    PRIMARY KEY (Name))
```

```
CREATE TABLE Platform (  
    Name VARCHAR,  
    Type VARCHAR,  
    PRIMARY KEY (Name))
```

```
CREATE TABLE PlayedOn (  
    GID INT,  
    Name VARCHAR,  
    PRIMARY KEY (GID, Name))  
    FOREIGN KEY (GID) REFERENCES VideoGameMadeBy,  
    FOREIGN KEY (Name) REFERENCES Platform)
```

```
CREATE TABLE ContainsDLC (  
    GID INT,  
    Name VARCHAR,  
    Price FLOAT(24)  
    Category VARCHAR,  
    DevTeamName VARCHAR NOT NULL,  
    FOREIGN KEY (DevTeamName) REFERENCES DevTeam(Name))
```

```
CREATE TABLE Adds (  
    Username VARCHAR,  
    GID INT,  
    Status VARCHAR,  
    PRIMARY KEY (Username, GID),  
    FOREIGN KEY (Username) REFERENCES Account)
```

```
CREATE TABLE MakesReviewReviewing1 (  
    ReviewID INT,  
    Date DATE,  
    Rating INT,  
    Length INT NOT NULL,  
    Username VARCHAR NOT NULL,  
    GID INT NOT NULL,  
    PRIMARY KEY (ReviewID),  
    FOREIGN KEY (Length) REFERENCES MakesReviewReviewing2,  
    FOREIGN KEY (Username) REFERENCES Account,  
    FOREIGN KEY (GID) REFERENCES VideoGame)
```

```
CREATE TABLE MakesReviewReviewing2 (  
    Length INT,  
    Category VARCHAR,  
    PRIMARY KEY (Length))
```

```
CREATE TABLE MakesReviewReviewing3 (  
    ReviewID INT,  
    Category VARCHAR,  
    PRIMARY KEY (ReviewID),  
    FOREIGN KEY (ReviewID) REFERENCES MakesReviewReviewing1)
```

```
CREATE TABLE Account (  
    Username VARCHAR,  
    Email VARCHAR UNIQUE NOT NULL,  
    DisplayName VARCHAR,  
    CreationDate DATE,  
    PRIMARY KEY (Username))
```

Insert Statements

Producer

```
INSERT  
INTO Producer(Name)  
VALUES ('Ryota Niitsuma')
```

```
INSERT  
INTO Producer(Name)  
VALUES ('Davor Hunski ')
```

```
INSERT  
INTO Producer(Name)  
VALUES ('Yohei Shimbori')
```

```
INSERT  
INTO Producer(Name)  
VALUES ('Toru Narihiro')
```

```
INSERT  
INTO Producer(Name)  
VALUES ('Hitoshi Yamagami')
```

Composer

```
INSERT  
INTO Composer(Name, Genre, Instrument)  
VALUES ('Atsushi Kitajoh','Jazz','Varied')
```

```
INSERT  
INTO Composer(Name, Genre, Instrument)  
VALUES ('Hiroki Morishita','Battle','Varied')
```

```
INSERT  
INTO Composer(Name, Genre, Instrument)  
VALUES ('Keiichi Okabe','Battle','Varied')
```

```
INSERT  
INTO Composer(Name, Genre, Instrument)  
VALUES ('Yoshinori Kawamoto','Battle','Varied')
```

```
INSERT
INTO Composer(Name, Genre, Instrument)
VALUES ('Keigo Hoashi','Battle','Varied')
```

ArtDesign

```
INSERT
INTO ArtDesign(Name, ArtistRole, Style)
VALUES ('Azusa Shimada','Lead 2D character design','anime')
```

```
INSERT
INTO ArtDesign(Name, ArtistRole, Style)
VALUES ('Takuji Kawano','Character Concept Artist','anime')
```

```
INSERT
INTO ArtDesign(Name, ArtistRole, Style)
VALUES ('Hiromi Sagara','Character Concept Artist','anime')
```

```
INSERT
INTO ArtDesign(Name, ArtistRole, Style)
VALUES ('Oguchi','Character Designer','anime')
```

```
INSERT
INTO ArtDesign(Name, ArtistRole, Style)
VALUES ('Yusuke Kozaki','Character Designer','anime')
```

Programmer

```
INSERT
INTO Programmer(Name, Task)
VALUES ('Tomohiko Imanishi','Chief Programmer')
```

```
INSERT
INTO Programmer(Name, Task)
VALUES ('Ryo Onishi','Lead Programmer')
```

```
INSERT
INTO Programmer(Name, Task)
VALUES ('Takashi Wagatsuma','Game Programmer')
```

```
INSERT
INTO Programmer(Name, Task)
VALUES ('Misa Yasui','System Programmer')
```

```
INSERT
INTO Programmer(Name, Task)
VALUES ('Takahiro Kitagawa','Sound Programmer')
```

Dev

```
INSERT
INTO Dev(Name, Website)
VALUES ('Ryota Niitsuma','https://www.mobygames.com/person/362265/ryota-niitsuma/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Tomohiko
Imanishi','https://www.mobygames.com/person/364171/tomohiko-imanishi/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Azusa Shimada','https://www.mobygames.com/person/1108482/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Davor Hunski','https://www.mobygames.com/person/41188/davor-hunski/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Yohei Shimbori','https://www.mobygames.com/person/509159/yohei-shimbori/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Atsushi Kitajoh','https://www.mobygames.com/person/364191/atsushi-kitajoh/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Takuji Kawano','https://www.mobygames.com/person/98565/takuji-kawano/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Hiromi Sagara','https://www.mobygames.com/person/484529/hiromi-sagara/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Oguchi','https://www.mobygames.com/person/1101660/oguchi/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Yusuke Kozaki','https://www.mobygames.com/person/406971/yusuke-kozaki/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Toru Narihiro','https://www.mobygames.com/person/534705/toru-narihiro/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Hitoshi Yamagami','https://www.mobygames.com/person/50693/hitoshi-yamagami/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Hiroki Morishita','https://www.mobygames.com/person/609076/hiroki-morishita/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Keiichi Okabe','https://www.mobygames.com/person/129821/keiichi-okabe/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Yoshinori
Kawamoto','https://www.mobygames.com/person/178444/yoshinori-kawamoto/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Keigo Hoashi','https://www.mobygames.com/person/600583/keigo-hoashi/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Ryo Onishi','https://www.mobygames.com/person/710385/ryo-onishi/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Takashi
Wagatsuma','https://www.mobygames.com/person/565395/takashi-wagatsuma/')
```

```
INSERT
INTO Dev(Name, Website)
VALUES ('Misa Yasui','https://www.mobygames.com/person/729004/misa-yasui/')
```



```
INSERT
INTO Dev(Name, Website)
VALUES ('Takahiro
Kitagawa','https://www.mobygames.com/person/1083697/takahiro-kitagawa/')
```

Includes

```
INSERT
INTO Includes(DevTeamName, DevName)
VALUES ('Atlus','Ryota Niitsuma')
```

```
INSERT
INTO Includes(DevTeamName, DevName)
VALUES ('Atlus','Tomohiko Imanishi')
```

```
INSERT
INTO Includes(DevTeamName, DevName)
VALUES ('Atlus','Azusa Shimada')
```

```
INSERT
INTO Includes(DevTeamName, DevName)
VALUES ('Croteam','Davor Hunski ')
```

```
INSERT
INTO Includes(DevTeamName, DevName)
VALUES ('Bandai Namco Studios Inc.','Yohei Shimbori')
```

Publisher

```
INSERT
INTO Publisher(Name, Location, Size)
VALUES ('Sega','Tokyo, Japan','3,459')
```

```
INSERT
INTO Publisher(Name, Location, Size)
VALUES ('Devolver Digital','Texas, US','235')
```

```
INSERT
INTO Publisher(Name, Location, Size)
VALUES ('Bandai Namco Entertainment','Tokyo, Japan','710')
```

```
INSERT
INTO Publisher(Name, Location, Size)
VALUES ('Square Enix','Tokyo, Japan','712')
```

```
INSERT
INTO Publisher(Name, Location, Size)
VALUES ('Annapurna Interactive','California, USA','25')
```

```
INSERT
INTO Publisher(Name, Location, Size)
VALUES ('Maddy Makes Games','null','10')
```

AssociatedWith

```
INSERT
INTO AssociatedWith(DevTeamName, PublisherName)
VALUES ('Atlus','Sega')
```

```
INSERT
INTO AssociatedWith(DevTeamName, PublisherName)
VALUES ('Croteam','Devolver Digital')
```

```
INSERT
INTO AssociatedWith(DevTeamName, PublisherName)
VALUES ('Bandai Namco Studios Inc.','Bandai Namco Entertainment')
```

```
INSERT
INTO AssociatedWith(DevTeamName, PublisherName)
VALUES ('Arc System Works','Bandai Namco Entertainment')
```

```
INSERT
INTO AssociatedWith(DevTeamName, PublisherName)
VALUES ('Maddy Makes Games','Null')
```

DevTeam

```
INSERT
INTO DevTeam(Name, Size, Location)
VALUES ('Atlus','338','Tokyo, Japan')
```

```
INSERT
INTO DevTeam(Name, Size, Location)
VALUES ('Croteam','42','Zagreb, Croatia')
```

```
INSERT
INTO DevTeam(Name, Size, Location)
VALUES ('Bandai Namco Studios Inc.',1219,'Tokyo, Japan')
```

```
INSERT
INTO DevTeam(Name, Size, Location)
VALUES ('Arc System Works',180,'Yokohama, Japan')
```

```
INSERT
INTO DevTeam(Name, Size, Location)
VALUES ('Maddy Makes Games',10,'null')
```

Platform

```
INSERT
INTO Platform(Name, Type)
VALUES ('Steam','PC')
```

```
INSERT
INTO Platform(Name, Type)
VALUES ('Nintendo Switch','Console')
```

```
INSERT
INTO Platform(Name, Type)
VALUES ('Play Store','Mobile')
```

```
INSERT
INTO Platform(Name, Type)
VALUES ('3DS','Portable')
```

```
INSERT
INTO Platform(Name, Type)
VALUES ('Playstation 5','Console')
```

PlayedOn

```
INSERT
INTO PlayedOn(GID, PlatformName)
VALUES ('216878','Steam')
```

```
INSERT
INTO PlayedOn(GID, PlatformName)
VALUES ('1382330','Nintendo Switch')
```

```
INSERT
INTO PlayedOn(GID, PlatformName)
VALUES ('257510','Steam')
```

```
INSERT
INTO PlayedOn(GID, PlatformName)
VALUES ('1778820','Steam')
```

```
INSERT
INTO PlayedOn(GID, PlatformName)
VALUES ('101945','Steam')
```

```
INSERT
INTO PlayedOn(GID, PlatformName)
VALUES ('504230','Nintendo Switch')
```

ContainsDLC

```
INSERT
INTO ContainsDLC(GID, Name, Price, ReleaseDate)
VALUES ('101945','Dragon Ball FighterZ: Broly','4.99','2018-03-28')
```

```
INSERT
INTO ContainsDLC(GID, Name, Price, ReleaseDate)
VALUES ('101945','Dragon Ball FighterZ: Android 21 (Lab Coat)','4.99','2022-02-23')
```

```
INSERT
INTO ContainsDLC(GID, Name, Price, ReleaseDate)
VALUES ('101945','Dragon Ball FighterZ: Vegito (SSGSS)','4.99','2018-05-31')
```

```
INSERT
INTO ContainsDLC(GID, Name, Price, ReleaseDate)
VALUES ('101945','Dragon Ball FighterZ: Bardock','4.99','2018-03-28')
```

```
INSERT
INTO ContainsDLC(GID, Name, Price, ReleaseDate)
VALUES ('101945','Dragon Ball FighterZ: Goku (Ultra Instinct)','4.99','2020-05-21')
```

VideoGameMadeBy

```
INSERT
INTO VideoGameMadeBy(GID, Name, ReleaseDate, Price, Category, DevTeamName)
VALUES ('216878','Persona 3: Reload','2024-02-02','69.99','RPG','Atlus')
```

```

INSERT
INTO VideoGameMadeBy(GID, Name, ReleaseDate, Price, Category, DevTeamName)
VALUES ('1382330','Persona® 5 Strikers','2021-02-22','79.99','RPG','Atlus')

INSERT
INTO VideoGameMadeBy(GID, Name, ReleaseDate, Price, Category, DevTeamName)
VALUES ('257510','The Talos Principle','2014-12-04','36.99','Puzzle','Croteam')

INSERT
INTO VideoGameMadeBy(GID, Name, ReleaseDate, Price, Category, DevTeamName)
VALUES ('1778820','TEKKEN 8','2024-01-25','93.49','Fighting','Bandai Namco Studios Inc.')

INSERT
INTO VideoGameMadeBy(GID, Name, ReleaseDate, Price, Category, DevTeamName)
VALUES ('101945','Dragon Ball','2018-01-26','59.99','Fighting','Arc System Works')

INSERT
INTO VideoGameMadeBy(GID, Name, ReleaseDate, Price, Category, DevTeamName)
VALUES ('504230','Celeste','2018-01-25','25.99','Platformer','Maddy Makes Games')

```

Adds

```

INSERT
INTO Adds(Username, GID, Status)
VALUES ('dkos','216878','playing')

INSERT
INTO Adds(Username, GID, Status)
VALUES ('popcornman','1382330','complete')

INSERT
INTO Adds(Username, GID, Status)
VALUES ('dreamindream','257510','backlog')

INSERT
INTO Adds(Username, GID, Status)
VALUES ('heronboy','1778820','playing')

INSERT
INTO Adds(Username, GID, Status)
VALUES ('mobius','101945','playing')

```

```
INSERT
INTO Adds(Username, GID, Status)
VALUES ('mobius','504230','complete')
```

MakesReviewReviewing1

```
INSERT
INTO MakesReviewReviewing1(ReviewID, Date, Rating, Length, Username, GID)
VALUES ('2341','2023-02-18','5','300','dkos','216878')
```

```
INSERT
INTO MakesReviewReviewing1(ReviewID, Date, Rating, Length, Username, GID)
VALUES ('1234','2023-09-05','7','200','popcornman','1382330')
```

```
INSERT
INTO MakesReviewReviewing1(ReviewID, Date, Rating, Length, Username, GID)
VALUES ('5756','2019-05-27','9','50','dreamindream','257510')
```

```
INSERT
INTO MakesReviewReviewing1(ReviewID, Date, Rating, Length, Username, GID)
VALUES ('1304','2020-08-13','8','1000','heronboy','1778820')
```

```
INSERT
INTO MakesReviewReviewing1(ReviewID, Date, Rating, Length, Username, GID)
VALUES ('8765','2021-10-14','10','400','mobius','101945')
```

MakesReviewReviewing2

```
INSERT
INTO MakesReviewReviewing2(Category, Length)
VALUES ('Medium','300')
```

```
INSERT
INTO MakesReviewReviewing2(Category, Length)
VALUES ('Medium','200')
```

```
INSERT
INTO MakesReviewReviewing2(Category, Length)
VALUES ('Short','50')
```

```
INSERT
INTO MakesReviewReviewing2(Category, Length)
VALUES ('Long','1000')
```

```
INSERT
INTO MakesReviewReviewing2(Category, Length)
VALUES ('Medium','400')
```

MakesReviewReviewing3

```
INSERT
INTO MakesReviewReviewing3(ReviewID, Category)
VALUES ('2341','Medium')
```

```
INSERT
INTO MakesReviewReviewing3(ReviewID, Category)
VALUES ('1234','Medium')
```

```
INSERT
INTO MakesReviewReviewing3(ReviewID, Category)
VALUES ('5756','Short')
```

```
INSERT
INTO MakesReviewReviewing3(ReviewID, Category)
VALUES ('1304','Long')
```

```
INSERT
INTO MakesReviewReviewing3(ReviewID, Category)
VALUES ('8765','Medium')
```

Account

```
INSERT
INTO Account(Username, Email, DisplayName, Creation Date)
VALUES ('dkos','dkos1884@email.com', 'Salvation', 2023-02-14 )
```

```
INSERT
INTO Account(Username, Email, DisplayName, Creation Date)
VALUES ('popcornman','ilovemovies@office.ca','SilentWatcher','2023-01-08')
```

```
INSERT
INTO Account(Username, Email, DisplayName, Creation Date)
VALUES ('dreamindream','inception@inception.com','LucidDreamer','2010-07-08')
```

```
INSERT
INTO Account(Username, Email, DisplayName, Creation Date)
VALUES ('heronboy','ghilblilover@live.com','Heron','2023-12-08')
```

```
INSERT  
INTO Account(Username, Email, DisplayName, Creation Date)  
VALUES ('mobius','ahoy@gmail.com','Polybius','2017-09-08')
```