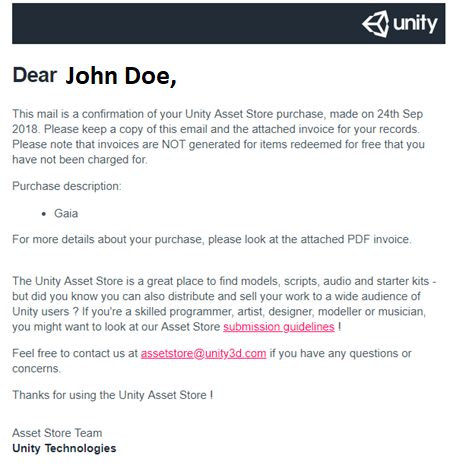
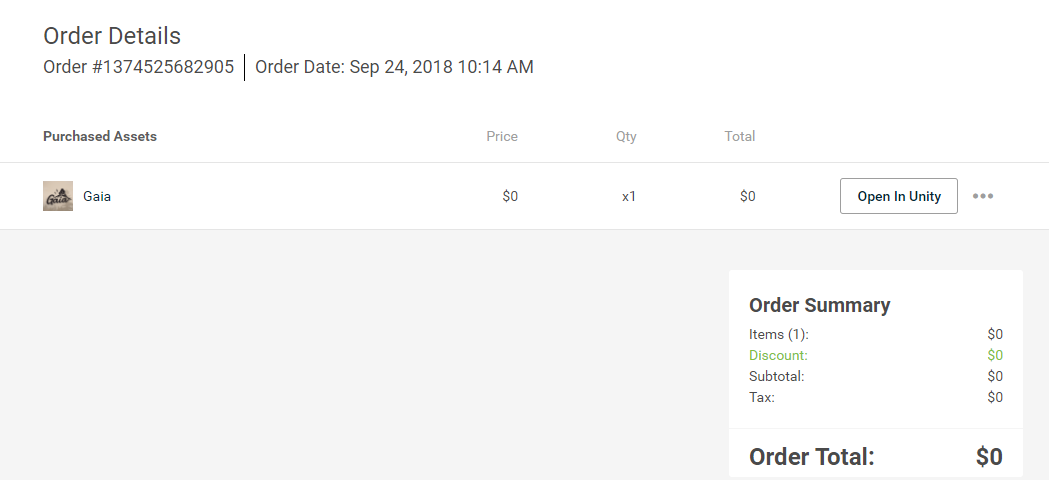
# Unity Store Example

## Invoice





### Attributes (0NF)

CustomerID, CustomerName, CustomerEmail, OrderNumber, Order Date, {AssetID, AssetName, Price, ItemQuantity, ItemTotal}, OrderItemAmount, Discount, Subtotal, Tax, OrderTotal

### 1NF

OrderNumber, CustomerID, CustomerFirstName, CustomerLastName, CustomerEmail, Order Date, OrderItemAmount, Discount, Subtotal, Tax, OrderTotal

OrderNumber(fk), AssetID, AssetName, Price, ItemQuantity, ItemTotal

### 2NF

OrderNumber, CustomerID, CustomerFirstName, CustomerLastName, CustomerEmail, Order Date, OrderItemAmount, Discount, Subtotal, Tax, OrderTotal

OrderNumber(fk), AssetID(fk), AssetHistoricalPrice, ItemQuantity, ItemTotal

AssetID, AssetName, AssetCurrentPrice

\*decided to store historical price and current price

### 3NF

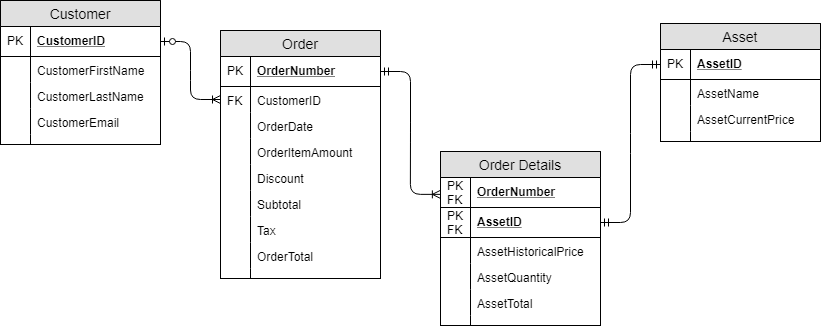
OrderNumber, CustomerID(fk), Order Date, OrderItemAmount, Discount, Subtotal, Tax, OrderTotal

CustomerID, CustomerFirstName, CustomerLastName, CustomerEmail

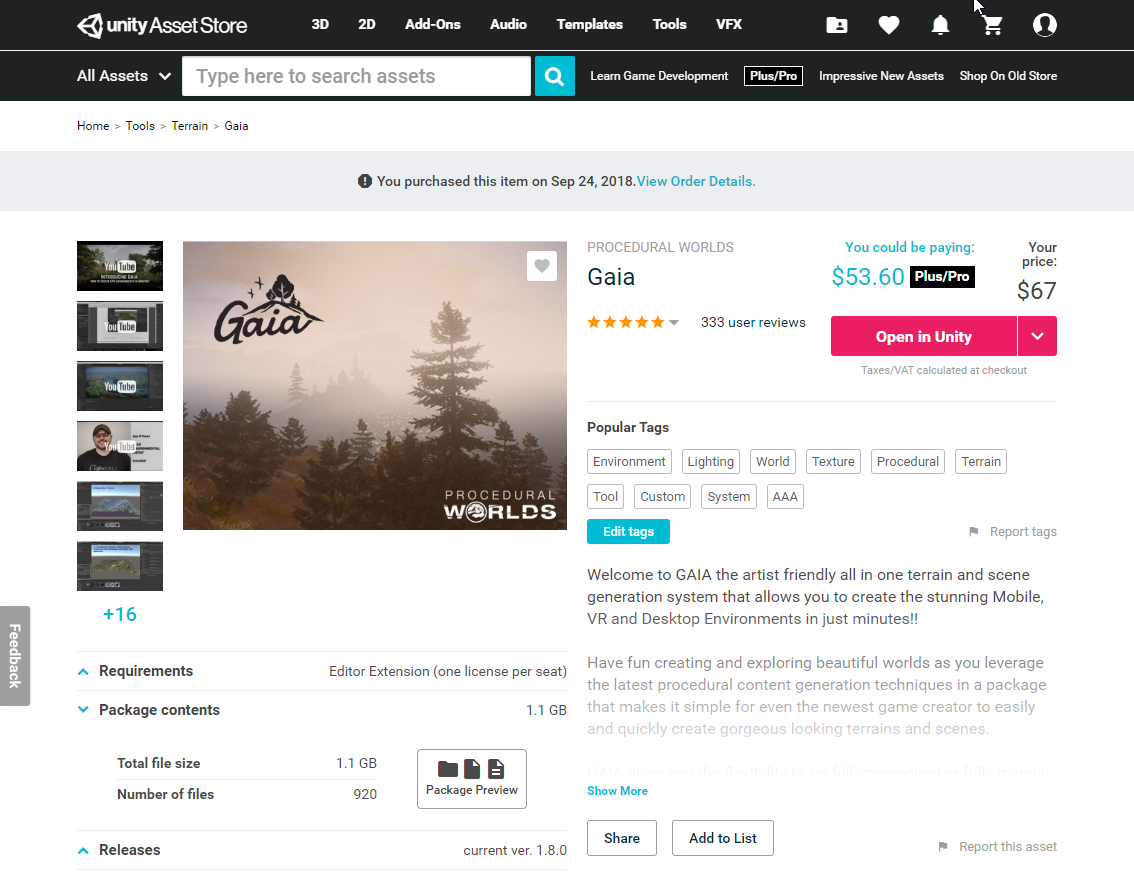
OrderNumber(fk), AssetID(fk), AssetHistoricalPrice, ItemQuantity, ItemTotal

AssetID, AssetName, AssetCurrentPrice

### ERD



## Product View



### Attributes (0NF)

PublisherID, PublisherName, ProductID, ProductName, ProductProPrice, ProductPrice, Rating, {TagID, TagName}, FileSize, NumberOfFiles, LicenseType, CurrentVersion

\*license type will be asset or editor extension

\*\*Assume users don’t create tags, just pick from a list if they “Edit tags”

### 1NF

ProductID, ProductName, ProductProPrice, ProductPrice, Rating, FileSize, NumberOfFiles, LicenseType, CurrentVersion, PublisherID, PublisherName

ProductID(fk), TagID, TagName

### 2NF

ProductID, ProductName, ProductProPrice, ProductPrice, Rating, FileSize, NumberOfFiles, LicenseType, CurrentVersion, PublisherID, PublisherName

ProductID(fk), TagID(fk)

TagID, TagName

### 3NF

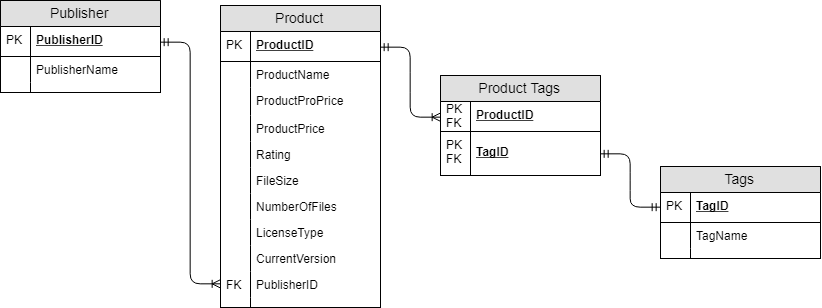
ProductID, ProductName, ProductProPrice, ProductPrice, Rating, FileSize, NumberOfFiles, LicenseType, CurrentVersion, PublisherID(fk)

PublisherID, PublisherName

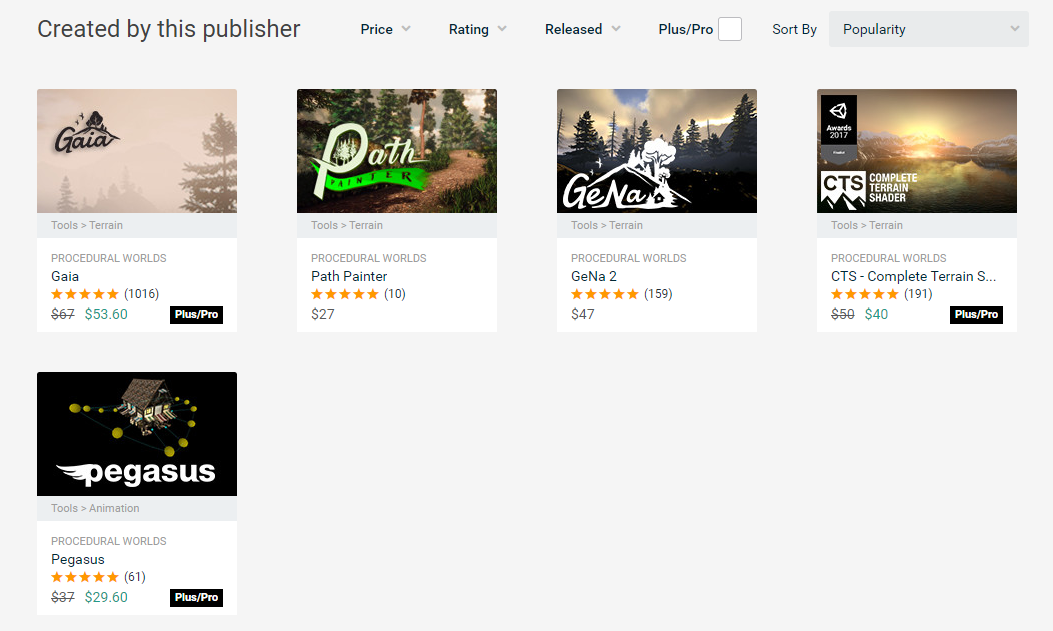
ProductID(fk), TagID(fk)

TagID, TagName

### ERD



## Publisher Products View



### Attributes (0NF)

PublisherID, AssetID, AssetCategory, AssetName, Rating, ReleaseDate, Price, AssetType

* Type will be Plus/Pro

### 1NF

AssetID, AssetCategory, AssetName, Rating, ReleaseDate, Price, AssetType, PublisherID

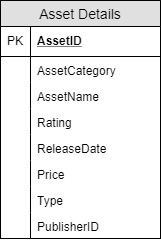
### 2NF

AssetID, AssetCategory, AssetName, Rating, ReleaseDate, Price, AssetType, PublisherID

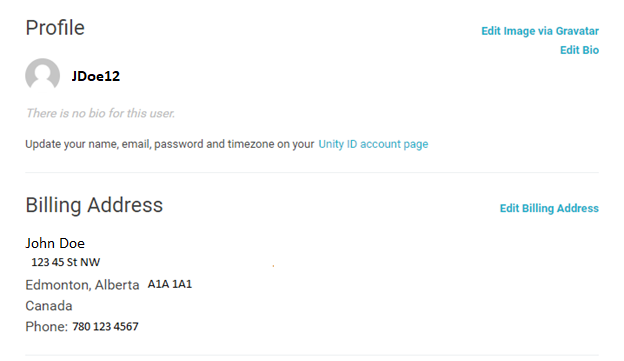
### 3NF

AssetID, AssetCategory, AssetName, Rating, ReleaseDate, Price, AssetType, PublisherID

### ERD

****

## User Profile View



### Attributes (0NF)

ProfileID, ProfileName, CustomerName, Street Address, City, Province, Postal Code, Country, Phone

\*Profile name is based off email

### 1NF

ProfileID, ProfileName, CustomerFirstName, CustomerLastName, Street Address, City, Province, Postal Code, Country, Phone

### 2NF

ProfileID, ProfileName, CustomerFirstName, CustomerLastName, Street Address, City, Province, Postal Code, Country, Phone

### 3NF

ProfileID, ProfileName, CustomerFirstName, CustomerLastName, Street Address, City, Province, Postal Code, Country, Phone

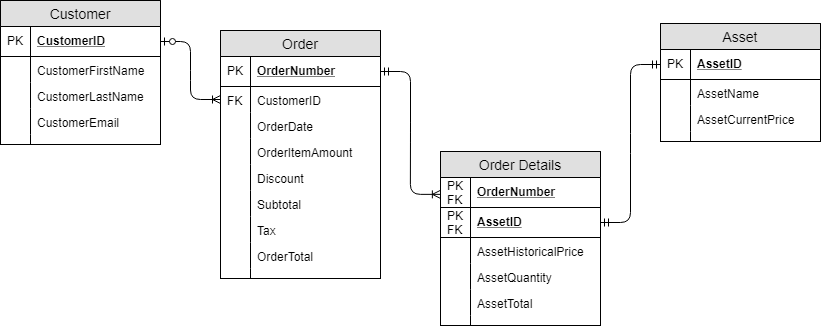
### ERD



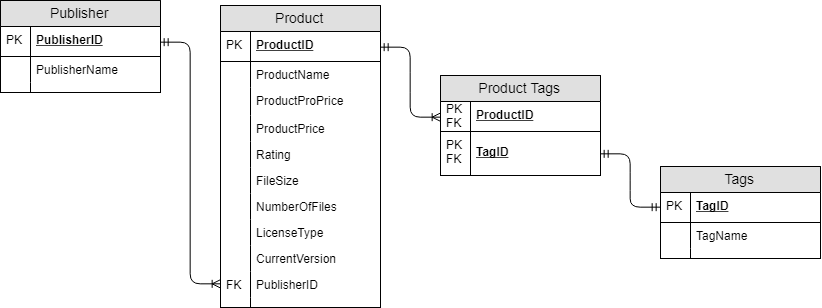
## Merged ERD

### Original ERD’s

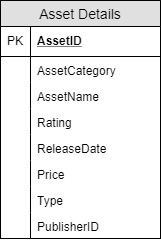
#### Invoice View ERD



#### Product View ERD



#### Publisher Products View ERD

****

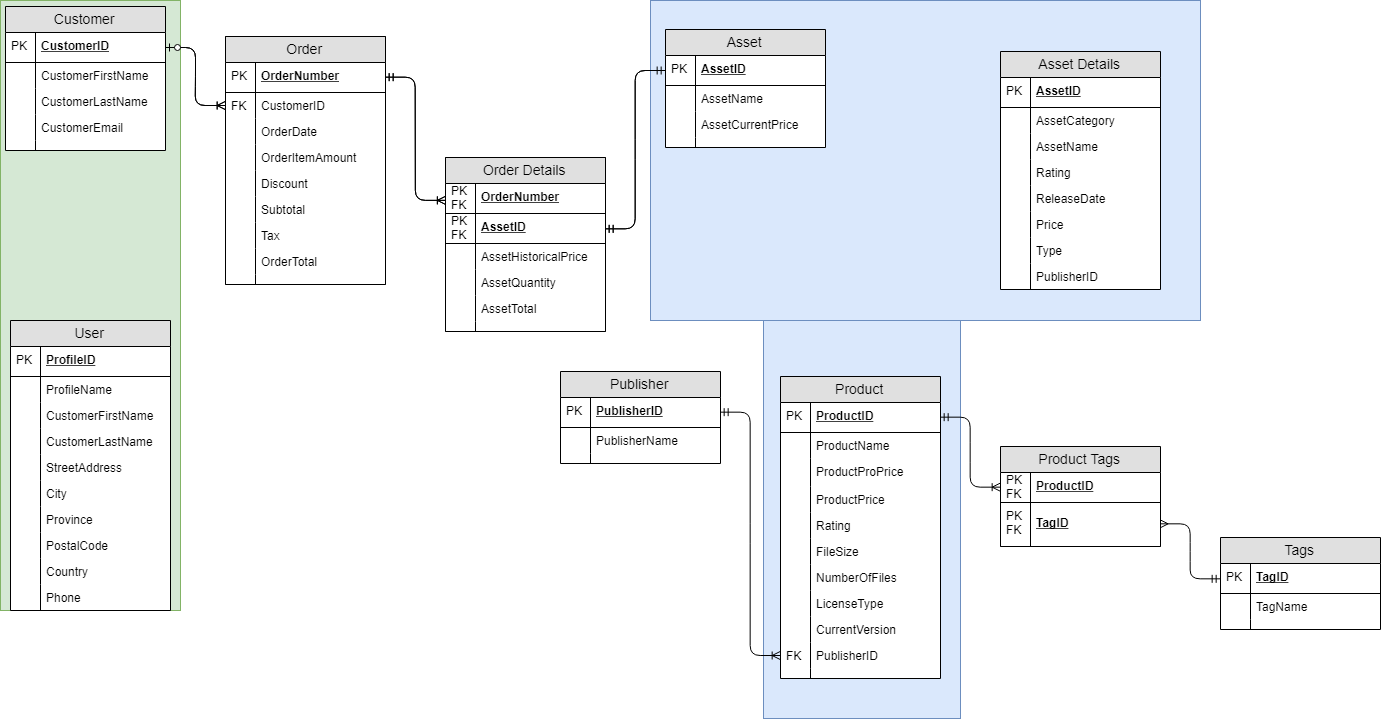
#### User Profile ERD



### Merged ERD

#### Step 1

Identify the entities that will merge together based on primary keys



#### Step 2

Merge the entities together, preserving fk’s. Make sure all the tables are connected, and any appropriate name changes are made

