

Alba Márquez Rodríguez

1st Report

Online Videogames Store Database

Database System Engineering – Computer Science

The project will consists on a Online Videogames Store Database

Professor: Krystian Wojtkiewicz



Contents

[Scope of the Project 2](#_Toc87545849)

[Requirements 3](#_Toc87545850)

[Conceptual Diagram 4](#_Toc87545851)

[[Logical Diagram] 7](#_Toc87545852)

[Physical Diagram 8](#_Toc87545853)

[Tables 9](#_Toc87545854)

[Game Tables 9](#_Toc87545855)

[Sales Tables 9](#_Toc87545856)

[Person Tables 10](#_Toc87545857)

[DDL 12](#_Toc87545858)

[Create 12](#_Toc87545859)

[Alter 14](#_Toc87545860)

[Drop 16](#_Toc87545861)

[Bibliography 17](#_Toc87545862)

# Scope of the Project

The Online Videogame Store Database Design is being undertaken for the purpose of creating a database to be managed by a user working on an Online Videogame Store. It is composed of definitions for database objects derived by mapping entities to tables, attributes to columns, unique identifiers to unique keys and relationships to foreign keys.

This project will include different tables containing the information that establishes the database. The most important elements of this database will be the information of the persons which can be a customer or a worker. Information about the Game, that will be the entity that the store will sell. And the information about Sales.

Description of the project scope and boundaries - about 1 paragraph. The longer, the better. Try to focus on the elements that impact the data storage needs. This part will be the base for database requirements. I would suggest writing it at the start and then supplementing it after the database model is finished.

# Requirements

The main requirements of the database as stated in the previous point will be:

* **Person**: The most important information from the person will be stored in the tables Contact and Territory. In Contact table there will be the person id, email and password which are crucial to identify a person. On the one hand, as the videogames will be online address is not important, on the other hand the territory is also important due to videogames servers.
* **Game**: The most important information about the videogames will be stored in the game table with the game id, standard\_cost (it will be the videogame price if there is no special offer available), platform, start\_sell\_date and the end\_sell\_date.
* **Sales**: It will be the main part of the database since it is a store and is based on sales. The main tables will be SalesOrderHeader with the SalesOrderID, CustomerID, Subtotal, TaxAmt, CreditCardID and the CreditCardApprovalCode. Other important table will be CreditCard with all the information from the credit card of the order payment. Other essential table from the sales will be the ShoppingChartItem where the games that will be purchased will be stored until the purchase is completed.

A synthetic requirements list based on the description provided in the previous point.

# Conceptual Diagram

The diagram could be divided in three parts related to person (yellow), game (red) and sales (blue) information.

Timeline

Description automatically generated with medium confidence

## Person

In the Contact (yellow) part:

Diagram

Description automatically generated

The main table is Contact where all the main data from the person will be stored. Address is other important table that will be related with contact by the AddressID. AddressType and Title are auxiliary tables to find if the address is a street, avenue… and the title of the person (Mr, Mss…).

StateProvince, CountryRegion and Territory are related to the localization of the address in a greater aspect.

## Game

In the Game (red) part:

Diagram

Description automatically generated

The main table here is Game where the essential information from the game will be stored. Game Audio-visuals is an auxiliary table to relate the game photo and game video to the game. Genre and Saga tables are for add more information and stock the data from those fields separately. Game Review table will have a review from the game. GameListPriceHistory will stored the data of the prices that the game had, so every time the standardprice field changes it will be stored there.

Game\_ShoppingCartItem is an intermediary table to create a many to many relationship with ShoppingCartItem.

## Sales

The blue part is referred to the Sales information:

Diagram

Description automatically generated

The main table of this part is SalesOrderHeader, with the leading information from a sale, there is additional information in SalesOrderDetail which contains the SpecialOfferGame and SpecialOffer. In the ShoppingCartItem will be stored the game that is going to be ordered. Customer will store the data of the customer ordering an order.

CurrencyRate and Currency store the data related to the rate of each currency.

There are tables related to the creditcard (contactcreditcard, CreditCard and CardType).

SalesTerritory is related to the territory where the order has been done.

**Conceptual diagram - it should focus on the primary storage elements (entities) relevant to the project’s scope. Each of the primitives indicated on the diagram should be shortly described.**

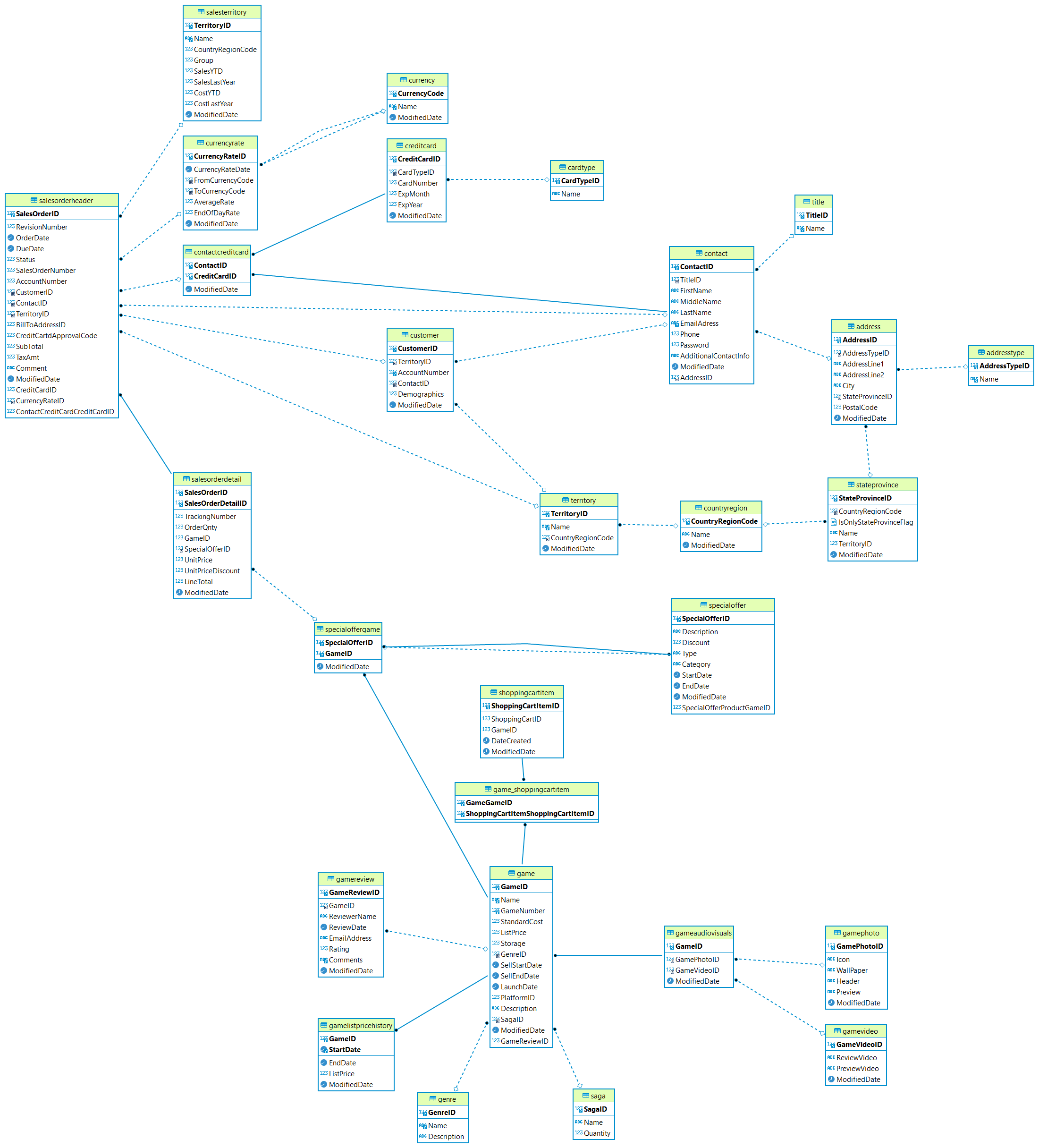
# [Logical Diagram]

Timeline

Description automatically generated with medium confidence

Logical diagram - more detailed than conceptual but still built a part of the implementation technology

# Physical Diagram



Physical diagram - preferably done in the DBeaver or similar software that puts all elements of the database existing in the DBMS on the diagram, NOTE: Physical diagram made in the Visual Paradigm is not acceptable

# Tables

This part of the document will be focussed on the tables that are going to take part on the database, their attributes and functions. As in previous chapters we will divide the tables in three:

## Table Description automatically generatedGame Tables

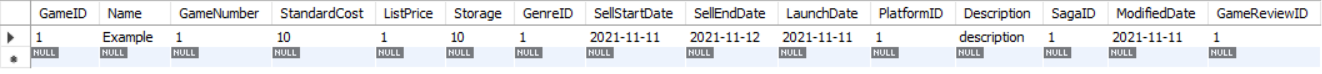
Game Table is the main table of that section, it contains the most relevant data from every game. Its Primary Key is the GameID and the Foreign keys it contains are GenreID and SagaID as well as GameID.

Table

Description automatically generatedGenre Table contains all genres of which a game can be. Its Primary Key is GenreID and contains the Genre Name and a brief description about the genre.

Graphical user interface, application, Teams

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated

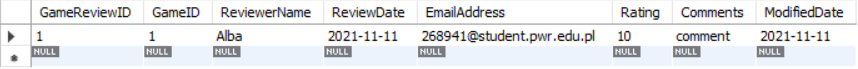
Table

Description automatically generatedTable

Description automatically generatedSaga Table contains all the sagas to which the videogames that are stored in the database belong. Its Primary Key is SagaID and contains the Saga Name and a brief description about the saga.

Text

Description automatically generatedGameListPriceHistory Table contains changes in the list price of a game over time. It also contains the StartDate of the price, the EndDate, the Price and the ModifiedDate. The Primary Key are the GameID and the StartDate.

Table

Description automatically generatedGameReview Table contains customer reviews of games they have purchased. Its PrimaryKey is GameReviewID and contains the GameID, ReviewerName, ReviewDate, EmailAddress, Rating, Comments and ModifiedDate. Comment is nullable due to its not essential function on a review. A review could consist on a rating without a comment.

Table

Description automatically generated

GameAudiovisuals Table is a table to link the GamePhoto and GameImage so its Foreign Keys are related to this tables. Its Primary Key is GameID.

Graphical user interface

Description automatically generated with medium confidenceTable

Description automatically generatedGraphical user interface, application, table

Description automatically generatedGraphical user interface, application

Description automatically generatedTable

Description automatically generatedGameVideo and GamePhoto Tables will contain images or video links from a game. The images and video will be officials so there will be a limited number of Audiovisuals per game.

## Sales Tables

Table

Description automatically generatedSalesOrderHeader Table will be the main table of the Sales Tables section. It will contain the main information and data from a Sale. Its PrimaryKey is SalesOrderID.

Diagram

Description automatically generatedGame\_ShoppingCartItem table is an intermediary table to create a many to many relation between Game and ShoppingCartItem Tables.

ShoppingCartItemTable is the table that contains the items that a customer will order.

Text, table

Description automatically generatedTable

Description automatically generatedSalesOrderDetail Table contain extra information about an order. Its Primary Keys are SalesOrderID and SalesOrderDetailID. If there is an SpecialOffer an SpecialOfferID will be provided.

Table

Description automatically generatedSpecialOfferGame contains the game regarding the special offer. Is an intermediary table to link SalesOrderDetail, Game and SpecialOffer table

Table

Description automatically generatedSpecialOffer Table contains all the information regarding that special offer.

Graphical user interface, application, table, Excel

Description automatically generatedCurrencyRate Table contains the information about a currency rate with different from and to currencies. There will be two fields with CurrencyCodes so two relations will link that table with Currency Table.

Table

Description automatically generatedCurrency Table contains all the currencies available to pay in the store, with their code (id) and name.

Text, table

Description automatically generatedSalesTerritory Table contains the territory where the sales have been done and information regarding the sales that have been done on that territory. The main purpose of this table will be provide information for analysis.

Table, Excel

Description automatically generatedContactCreditCard Table is an intermediary table to link Contact, SalesOrderHeader and CreditCard Tables.

CreditCard Table contains the data related to a CreditCard Stored in the database.

Table, Excel

Description automatically generated with medium confidenceCardType Table store the type of card that is stored in CreditCard Table.

## Table Description automatically generatedPerson Tables

Graphical user interface, application, table, Excel

Description automatically generatedContact Table is the main table of the Person Tables section. It contains the main information regarding a Person (Contact).

Title Table contains all the possible titles a person can have.

Table

Description automatically generatedTable

Description automatically generatedAlthough the store is online the address of the person is required so there is an Address Table which is linked with AddressType.

AddressType Store all the types of address possible.

Diagram

Description automatically generated with low confidenceStateProvince Table contains the data regarding the States or Provinces that an address can belong to.

Table

Description automatically generatedCountryRegion contains the countries where the states or provinces of the past table belong to.

Table, Excel

Description automatically generatedTerritory Table contains the data of which territory belongs the Country or Region from the past table.

**Description of tables relevant attributes with exemplary records**

# DDL

## Create

To create the tables:

CREATE TABLE Game\_ShoppingCartItem (GameGameID int(10) NOT NULL, ShoppingCartItemShoppingCartItemID int(10) NOT NULL, PRIMARY KEY (GameGameID, ShoppingCartItemShoppingCartItemID));

CREATE TABLE Game (GameID int(10) NOT NULL, Name varchar(255) NOT NULL UNIQUE, GameNumber int(10) NOT NULL UNIQUE, StandardCost int(10) NOT NULL, ListPrice int(10), Storage int(5) NOT NULL, GenreID int(10), SellStartDate date NOT NULL, SellEndDate date, LaunchDate date NOT NULL, PlatformID int(10) NOT NULL, Description varchar(255), SagaID int(10) NOT NULL, ModifiedDate date NOT NULL, GameReviewID int(10), PRIMARY KEY (GameID));

CREATE TABLE Genre (GenreID int(10) NOT NULL AUTO\_INCREMENT, Name varchar(255) NOT NULL UNIQUE, Description varchar(255), PRIMARY KEY (GenreID));

CREATE TABLE Saga (SagaID int(10) NOT NULL AUTO\_INCREMENT, Name varchar(255), Quantity int(3), PRIMARY KEY (SagaID));

CREATE TABLE ShoppingCartItem (ShoppingCartItemID int(10) NOT NULL AUTO\_INCREMENT, ShoppingCartID int(10), GameID int(10) NOT NULL, DateCreated date, ModifiedDate date, PRIMARY KEY (ShoppingCartItemID));

CREATE TABLE SpecialOffer (SpecialOfferID int(10) NOT NULL, Description varchar(255), Discount int(10), Type varchar(255), Category varchar(255), StartDate date, EndDate date, ModifiedDate date, SpecialOfferProductGameID int(10) NOT NULL, PRIMARY KEY (SpecialOfferID));

CREATE TABLE SpecialOfferGame (SpecialOfferID int(10) NOT NULL, GameID int(10) NOT NULL, ModifiedDate date NOT NULL, PRIMARY KEY (SpecialOfferID, GameID), INDEX (SpecialOfferID));

CREATE TABLE SalesOrderDetail (SalesOrderID int(10) NOT NULL, SalesOrderDetailID int(10) NOT NULL, TrackingNumber int(10), OrderQnty int(10), GameID int(10) NOT NULL, SpecialOfferID int(10) NOT NULL, UnitPrice int(10), UnitPriceDiscount int(10), LineTotal int(10), ModifiedDate date, PRIMARY KEY (SalesOrderID, SalesOrderDetailID));

CREATE TABLE CardType (CardTypeID int(10) NOT NULL AUTO\_INCREMENT, Name varchar(255), PRIMARY KEY (CardTypeID));

CREATE TABLE CreditCard (CreditCardID int(10) NOT NULL AUTO\_INCREMENT, CardTypeID int(10) NOT NULL, CardNumber int(10), ExpMonth int(2), ExpYear int(2), ModifiedDate date, PRIMARY KEY (CreditCardID));

CREATE TABLE Currency (CurrencyCode int(10) NOT NULL AUTO\_INCREMENT, Name varchar(255) NOT NULL UNIQUE, ModifiedDate date NOT NULL, PRIMARY KEY (CurrencyCode));

CREATE TABLE CurrencyRate (CurrencyRateID int(10) NOT NULL AUTO\_INCREMENT, CurrencyRateDate date NOT NULL, FromCurrencyCode int(10) NOT NULL, ToCurrencyCode int(10) NOT NULL, AverageRate int(10) NOT NULL, EndOfDayRate int(10) NOT NULL, ModifiedDate date NOT NULL, PRIMARY KEY (CurrencyRateID), UNIQUE INDEX (CurrencyRateID));

CREATE TABLE SalesTerritory (TerritoryID int(10) NOT NULL AUTO\_INCREMENT, Name varchar(255) NOT NULL UNIQUE, CountryRegionCode int(5) NOT NULL, `Group` int(10) NOT NULL, SalesYTD int(10) NOT NULL, SalesLastYear int(10) NOT NULL, CostYTD int(10) NOT NULL, CostLastYear int(10) NOT NULL, ModifiedDate date NOT NULL, PRIMARY KEY (TerritoryID), UNIQUE INDEX (TerritoryID));

CREATE TABLE ContactCreditCard (ContactID int(10) NOT NULL, CreditCardID int(10) NOT NULL, ModifiedDate date, PRIMARY KEY (ContactID, CreditCardID), UNIQUE INDEX (ContactID), INDEX (CreditCardID));

CREATE TABLE Customer (CustomerID int(10) NOT NULL AUTO\_INCREMENT, TerritoryID int(10) NOT NULL, AccountNumber int(10) NOT NULL UNIQUE, ContactID int(10) NOT NULL, Demographics int(10) NOT NULL, ModifiedDate date NOT NULL, PRIMARY KEY (CustomerID));

CREATE TABLE Territory (TerritoryID int(10) NOT NULL AUTO\_INCREMENT, Name varchar(255) NOT NULL UNIQUE, CountryRegionCode int(5) NOT NULL, ModifiedDate date NOT NULL, PRIMARY KEY (TerritoryID));

CREATE TABLE CountryRegion (CountryRegionCode int(5) NOT NULL, Name varchar(255) NOT NULL, ModifiedDate date NOT NULL, PRIMARY KEY (CountryRegionCode));

CREATE TABLE Contact (ContactID int(10) NOT NULL AUTO\_INCREMENT, TitleID int(1) NOT NULL, FirstName varchar(255) NOT NULL, MiddleName varchar(255), LastName varchar(255) NOT NULL, EmailAdress varchar(255) NOT NULL UNIQUE, Phone int(11) NOT NULL, Password int(10) NOT NULL, AdditionalContactInfo varchar(255), ModifiedDate date NOT NULL, AddressID int(10) NOT NULL, PRIMARY KEY (ContactID));

CREATE TABLE Title (TitleID int(1) NOT NULL AUTO\_INCREMENT, Name varchar(255) NOT NULL UNIQUE, PRIMARY KEY (TitleID));

CREATE TABLE Address (AddressID int(10) NOT NULL, AddressTypeID int(2) NOT NULL, AddressLine1 varchar(255), AddressLine2 varchar(255), City varchar(255) NOT NULL, StateProvinceID int(5) NOT NULL, PostalCode int(6) NOT NULL, ModifiedDate date NOT NULL, PRIMARY KEY (AddressID));

CREATE TABLE AddressType (AddressTypeID int(2) NOT NULL AUTO\_INCREMENT, Name varchar(255) NOT NULL UNIQUE, PRIMARY KEY (AddressTypeID));

CREATE TABLE StateProvince (StateProvinceID int(5) NOT NULL, CountryRegionCode int(5) NOT NULL, IsOnlyStateProvinceFlag binary(1) NOT NULL, Name varchar(255) NOT NULL, TerritoryID int(10) NOT NULL, ModifiedDate date NOT NULL, PRIMARY KEY (StateProvinceID));

CREATE TABLE SalesOrderHeader (SalesOrderID int(10) NOT NULL, RevisionNumber int(10) NOT NULL, OrderDate date NOT NULL, DueDate date NOT NULL, Status int(1) NOT NULL, SalesOrderNumber int(10) NOT NULL, AccountNumber int(10) NOT NULL, CustomerID int(10) NOT NULL, ContactID int(10) NOT NULL, TerritoryID int(10) NOT NULL, BillToAddressID int(10) NOT NULL, CreditCartdApprovalCode int(10) NOT NULL, SubTotal int(10) NOT NULL, TaxAmt int(10) NOT NULL, Comment varchar(255), ModifiedDate date NOT NULL, CreditCardID int(10) NOT NULL, CurrencyRateID int(10) NOT NULL, ContactCreditCardCreditCardID int(10) NOT NULL, PRIMARY KEY (SalesOrderID), UNIQUE INDEX (SalesOrderID));

CREATE TABLE GamePhoto (GamePhotoID int(10) NOT NULL AUTO\_INCREMENT, Icon varchar(255), WallPaper varchar(255), Header varchar(255), Preview varchar(255), ModifiedDate date, PRIMARY KEY (GamePhotoID));

CREATE TABLE GameVideo (GameVideoID int(10) NOT NULL AUTO\_INCREMENT, ReviewVideo varchar(255), PreviewVideo varchar(255), ModifiedDate date, PRIMARY KEY (GameVideoID));

CREATE TABLE GameAudiovisuals (GameID int(10) NOT NULL AUTO\_INCREMENT, GamePhotoID int(10) NOT NULL, GameVideoID int(10) NOT NULL, ModifiedDate date, PRIMARY KEY (GameID));

CREATE TABLE GameListPriceHistory (GameID int(10) NOT NULL UNIQUE, StartDate date NOT NULL, EndDate date NOT NULL, ListPrice int(10) NOT NULL, ModifiedDate date NOT NULL, PRIMARY KEY (GameID, StartDate));

CREATE TABLE GameReview (GameReviewID int(10) NOT NULL AUTO\_INCREMENT, GameID int(10) NOT NULL, ReviewerName varchar(255) NOT NULL, ReviewDate date NOT NULL, EmailAddress varchar(255) NOT NULL, Rating int(2) NOT NULL, Comments varchar(255) UNIQUE, ModifiedDate date NOT NULL, PRIMARY KEY (GameReviewID));

## Alter

To create the relations between tables:

ALTER TABLE Game\_ShoppingCartItem ADD CONSTRAINT FKGame\_ShoppingCartItem\_Game FOREIGN KEY (GameGameID) REFERENCES Game (GameID);

ALTER TABLE Game ADD CONSTRAINT FKGame\_Genre FOREIGN KEY (GenreID) REFERENCES Genre (GenreID);

ALTER TABLE Game ADD CONSTRAINT FKGame\_Saga FOREIGN KEY (SagaID) REFERENCES Saga (SagaID);

ALTER TABLE Game\_ShoppingCartItem ADD CONSTRAINT FKGame\_ShoppingCartItem FOREIGN KEY (ShoppingCartItemShoppingCartItemID) REFERENCES ShoppingCartItem (ShoppingCartItemID);

ALTER TABLE SpecialOffer ADD CONSTRAINT FKSpecialOffer FOREIGN KEY (SpecialOfferID, SpecialOfferProductGameID) REFERENCES SpecialOfferGame (SpecialOfferID, GameID);

ALTER TABLE SalesOrderDetail ADD CONSTRAINT FKSalesOrderDetail\_SpecialOfferGame FOREIGN KEY (SpecialOfferID, GameID) REFERENCES SpecialOfferGame (SpecialOfferID, GameID);

ALTER TABLE CreditCard ADD CONSTRAINT FKCreditCard\_CardType FOREIGN KEY (CardTypeID) REFERENCES CardType (CardTypeID);

ALTER TABLE CurrencyRate ADD CONSTRAINT FKCurrencyRate\_CurrencyFROM FOREIGN KEY (FromCurrencyCode) REFERENCES Currency (CurrencyCode);

ALTER TABLE CurrencyRate ADD CONSTRAINT FKCurrencyRate\_CurrencyTO FOREIGN KEY (ToCurrencyCode) REFERENCES Currency (CurrencyCode);

ALTER TABLE ContactCreditCard ADD CONSTRAINT FKContactCreditCard\_CreditCard FOREIGN KEY (CreditCardID) REFERENCES CreditCard (CreditCardID);

ALTER TABLE Customer ADD CONSTRAINT FKCustomer\_Territory FOREIGN KEY (TerritoryID) REFERENCES Territory (TerritoryID);

ALTER TABLE Territory ADD CONSTRAINT FKTerritory\_CountryRegion FOREIGN KEY (CountryRegionCode) REFERENCES CountryRegion (CountryRegionCode);

ALTER TABLE Customer ADD CONSTRAINT FKCustomer\_Contact FOREIGN KEY (ContactID) REFERENCES Contact (ContactID);

ALTER TABLE Contact ADD CONSTRAINT FKContact\_Title FOREIGN KEY (TitleID) REFERENCES Title (TitleID);

ALTER TABLE Contact ADD CONSTRAINT FKContact\_Address FOREIGN KEY (AddressID) REFERENCES Address (AddressID);

ALTER TABLE Address ADD CONSTRAINT FKAddress\_AddressType FOREIGN KEY (AddressTypeID) REFERENCES AddressType (AddressTypeID);

ALTER TABLE Address ADD CONSTRAINT FKAddress\_StateProvince FOREIGN KEY (StateProvinceID) REFERENCES StateProvince (StateProvinceID);

ALTER TABLE StateProvince ADD CONSTRAINT FKStateProvince\_CountryRegion FOREIGN KEY (CountryRegionCode) REFERENCES CountryRegion (CountryRegionCode);

ALTER TABLE SalesOrderHeader ADD CONSTRAINT FKSalesOrderHeader\_SalesOrderDetail FOREIGN KEY (SalesOrderID) REFERENCES SalesOrderDetail (SalesOrderID);

ALTER TABLE SalesOrderHeader ADD CONSTRAINT FKSalesOrderHeader\_Contact FOREIGN KEY (ContactID) REFERENCES Contact (ContactID);

ALTER TABLE SalesOrderHeader ADD CONSTRAINT FKSalesOrderHeader\_ContactCreditCard2 FOREIGN KEY (ContactID, CreditCardID) REFERENCES ContactCreditCard (ContactID, CreditCardID);

ALTER TABLE SalesOrderHeader ADD CONSTRAINT FKSalesOrderHeader\_Territory FOREIGN KEY (TerritoryID) REFERENCES Territory (TerritoryID);

ALTER TABLE SalesOrderHeader ADD CONSTRAINT FKSalesOrderHeader\_SalesTerritory FOREIGN KEY (TerritoryID) REFERENCES SalesTerritory (TerritoryID);

ALTER TABLE SalesOrderHeader ADD CONSTRAINT FKSalesOrderHeader\_CurrencyRate FOREIGN KEY (CurrencyRateID) REFERENCES CurrencyRate (CurrencyRateID);

ALTER TABLE GameListPriceHistory ADD CONSTRAINT FKGameListPriceHistory\_Game FOREIGN KEY (GameID) REFERENCES Game (GameID);

ALTER TABLE GameReview ADD CONSTRAINT FKGameReview\_Game FOREIGN KEY (GameID) REFERENCES Game (GameID);

ALTER TABLE Game ADD CONSTRAINT FKGame\_GameAudiovisuals FOREIGN KEY (GameID) REFERENCES GameAudiovisuals (GameID);

ALTER TABLE GameAudiovisuals ADD CONSTRAINT FKGameAudiovisuals\_GamePhoto FOREIGN KEY (GamePhotoID) REFERENCES GamePhoto (GamePhotoID);

ALTER TABLE GameAudiovisuals ADD CONSTRAINT FKGameAudiovisuals\_GameVideo FOREIGN KEY (GameVideoID) REFERENCES GameVideo (GameVideoID);

ALTER TABLE SalesOrderHeader ADD CONSTRAINT FKSalesOrderHeader\_Customer FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID);

ALTER TABLE SpecialOfferGame ADD CONSTRAINT FKSpecialOfferGame\_Game FOREIGN KEY (GameID) REFERENCES Game (GameID);

ALTER TABLE SpecialOfferGame ADD CONSTRAINT FKSpecialOfferGame\_SpecialOffer FOREIGN KEY (SpecialOfferID) REFERENCES SpecialOffer (SpecialOfferID);

ALTER TABLE ContactCreditCard ADD CONSTRAINT FKContactCreditCard FOREIGN KEY (ContactID) REFERENCES Contact (ContactID);

## Drop

To eliminate all the tables:

DROP TABLE `onlinevideogamestore`.`address`, `onlinevideogamestore`.`addresstype`, `onlinevideogamestore`.`cardtype`, `onlinevideogamestore`.`contact`, `onlinevideogamestore`.`contactcreditcard`, `onlinevideogamestore`.`countryregion`, `onlinevideogamestore`.`creditcard`, `onlinevideogamestore`.`currency`, `onlinevideogamestore`.`currencyrate`, `onlinevideogamestore`.`customer`, `onlinevideogamestore`.`game`, `onlinevideogamestore`.`game\_shoppingcartitem`, `onlinevideogamestore`.`gameaudiovisuals`, `onlinevideogamestore`.`gamelistpricehistory`, `onlinevideogamestore`.`gamephoto`, `onlinevideogamestore`.`gamereview`, `onlinevideogamestore`.`gamevideo`, `onlinevideogamestore`.`genre`, `onlinevideogamestore`.`saga`, `onlinevideogamestore`.`salesorderdetail`, `onlinevideogamestore`.`salesorderheader`, `onlinevideogamestore`.`salesterritory`, `onlinevideogamestore`.`shoppingcartitem`, `onlinevideogamestore`.`specialoffer`, `onlinevideogamestore`.`specialoffergame`, `onlinevideogamestore`.`stateprovince`, `onlinevideogamestore`.`territory`, `onlinevideogamestore`.`title`;

# Bibliography