



Politechnika
Wrocławska



1ST REPORT

Online Videogames Store Database

Database System Engineering – Computer Science

The project will consists on a Online Videogames Store Database

PROFESSOR: KRYSTIAN WOJTKIEWICZ

Alba Márquez Rodríguez

Contents

Scope of the Project	2
Requirements	3
Conceptual Diagram	4
Person.....	4
Game	5
Sales	5
[Logical Diagram].....	7
Physical Diagram	8
Tables	9
Game Tables.....	9
Sales Tables	10
Person Tables	10
DDL	12
Create	12
Alter	14
Drop.....	16
Bibliography	17

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.

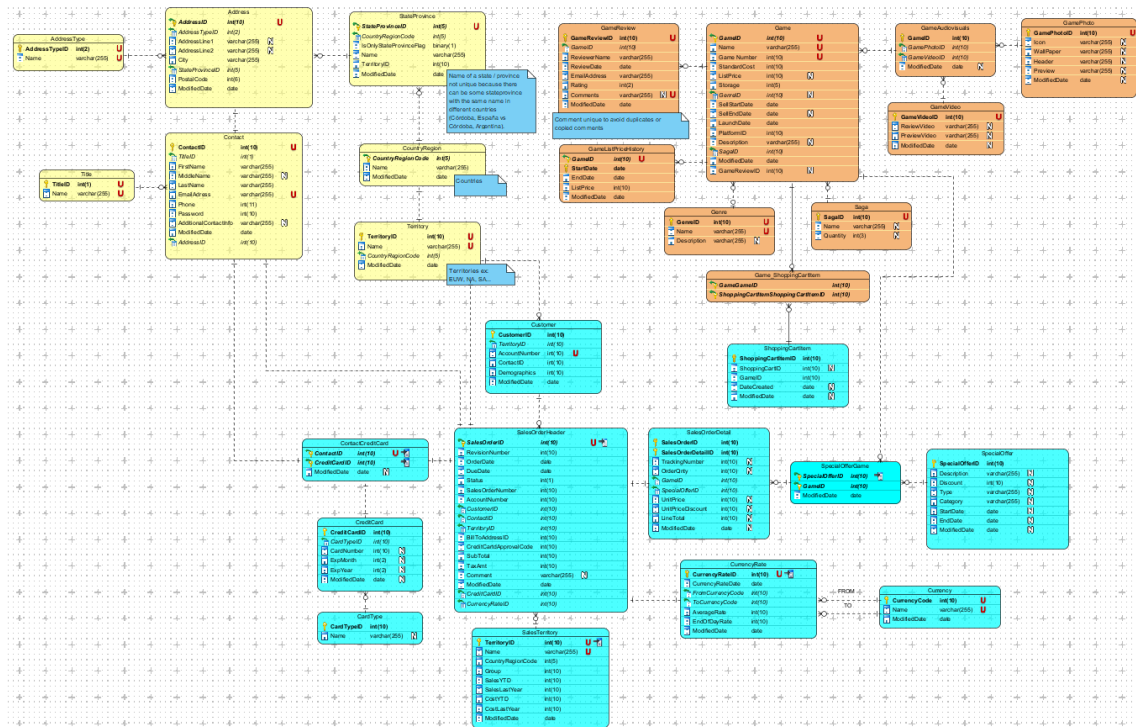
Requirements

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 ShoppingCartItem where the games that will be purchased will be stored until the purchase is completed.

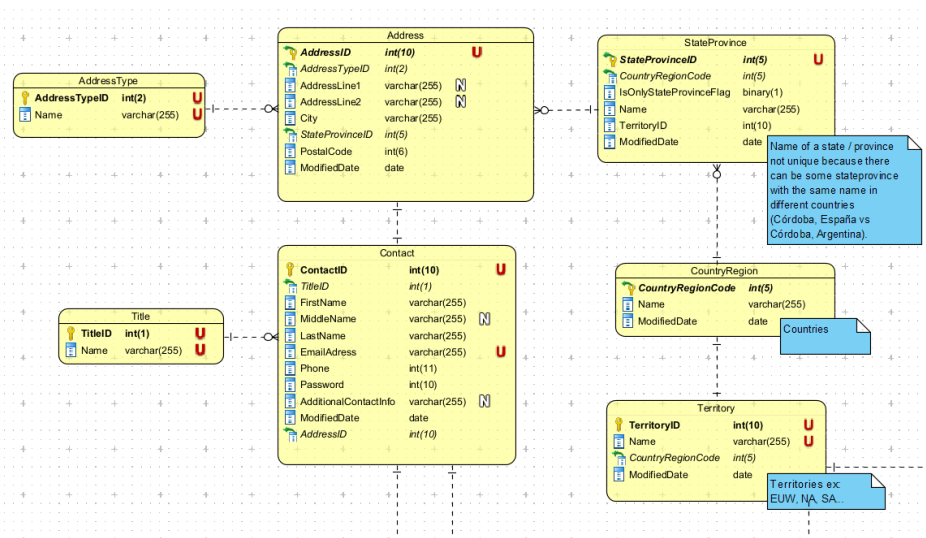
Conceptual Diagram

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



Person

In the Contact (yellow) part:

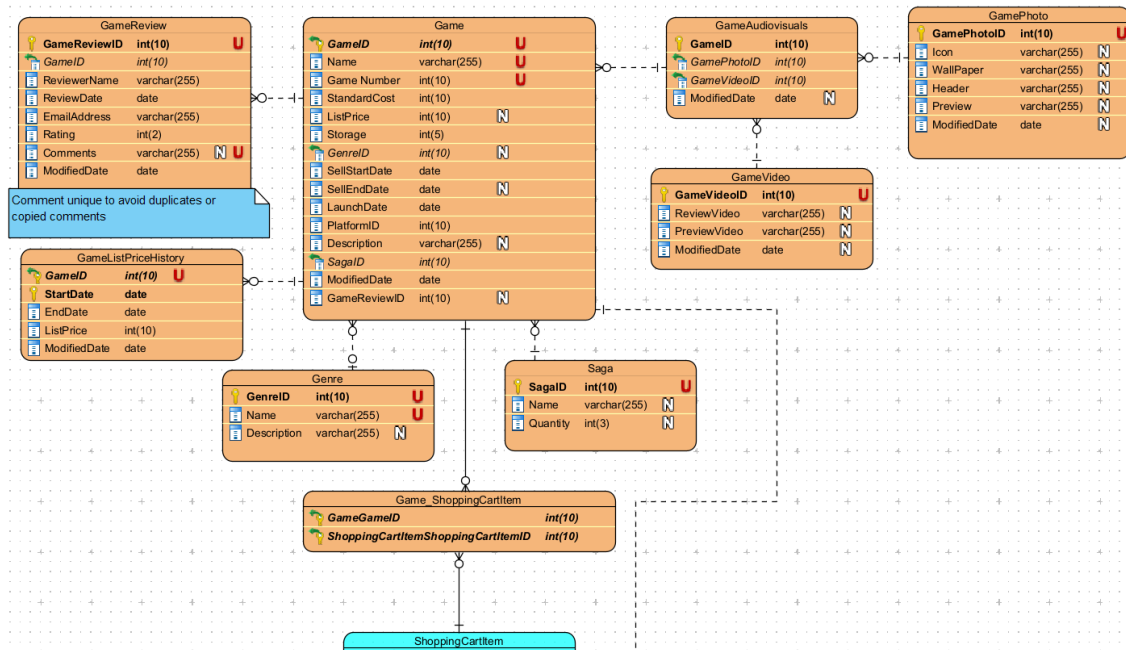


The main table is Contact where all the main data from the person will be stored. Address is other important table that will be related with contact by the AddressID. AddressType and Title are auxiliary tables to find if the address is a street, avenue... and the title of the person (Mr, Mss...).

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

Game

In the Game (red) part:

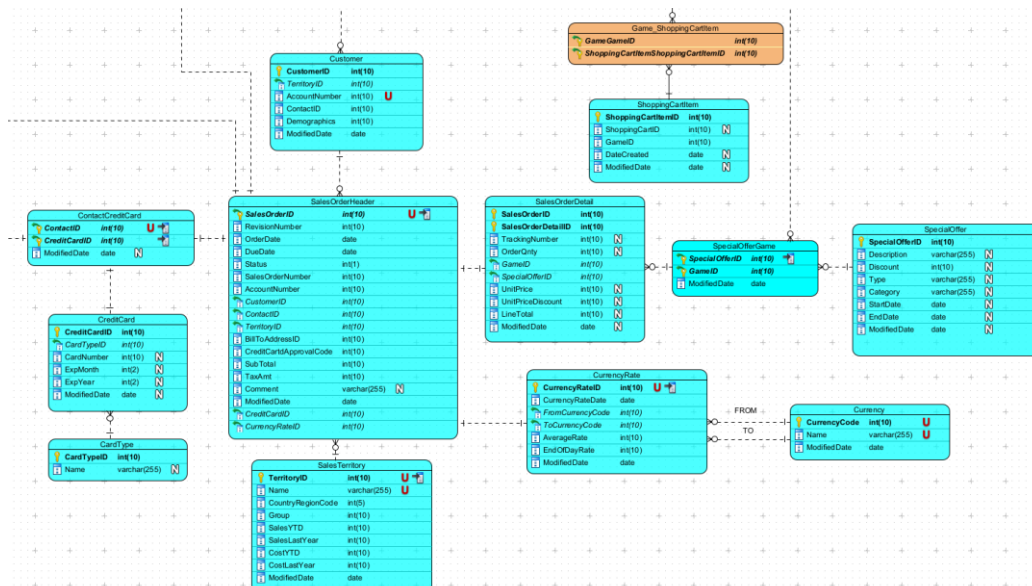


The main table here is Game where the essential information from the game will be stored. Game Audio-visuals is an auxiliary table to relate the game photo and game video to the game. Genre and Saga tables are for add more information and stock the data from those fields separately. Game Review table 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:



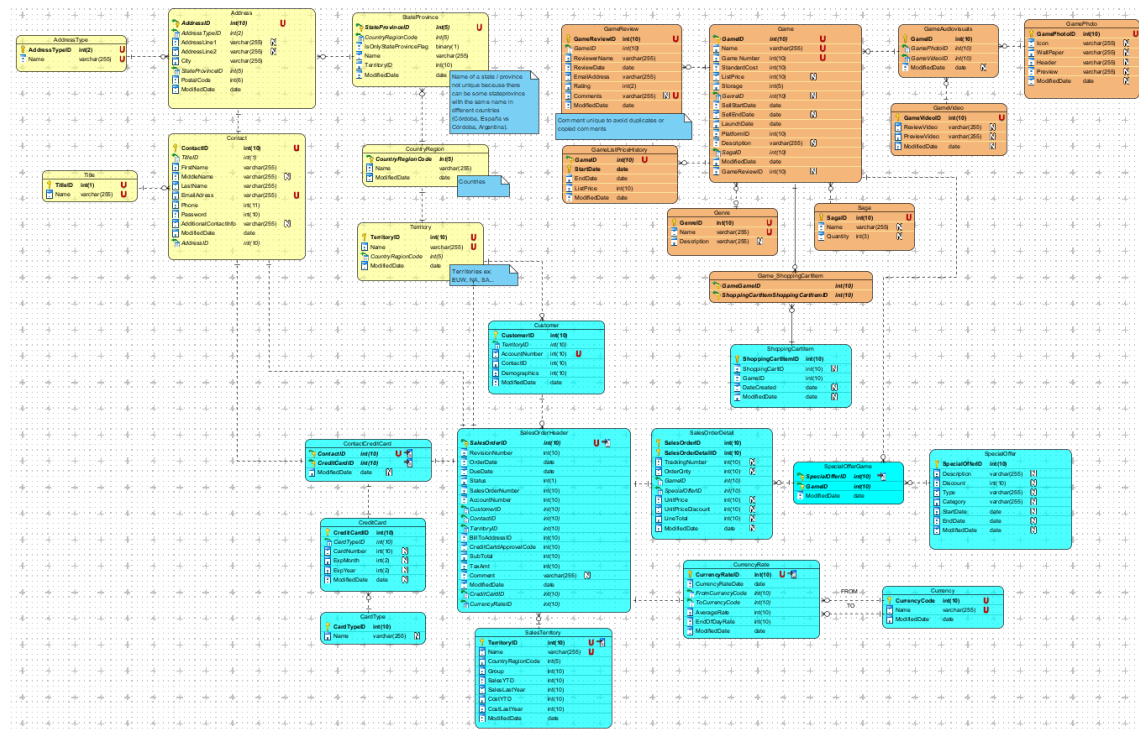
The main table of this part is SalesOrderHeader, with the leading information from a sale, there is additional information in SalesOrderDetail which contains the SpecialOfferGame and SpecialOffer. In the ShoppingCartItem will be stored the game that is going to be ordered. Customer will store the data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

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

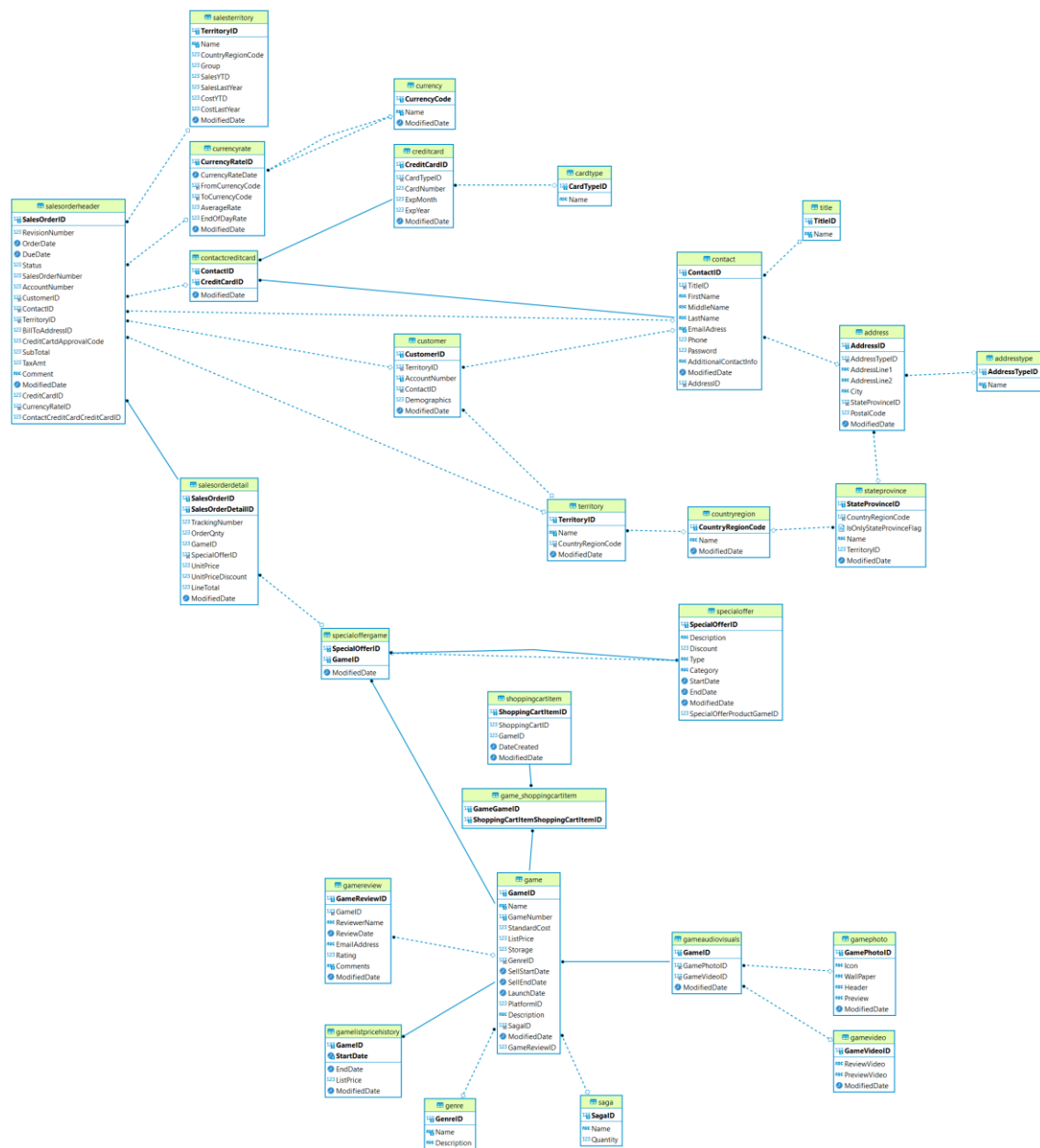
There are tables related to the creditcard (contactcreditcard, CreditCard and CardType).

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

Logical Diagram



Physical Diagram



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:

Game Tables

Game	
GameID	int(10) U
Name	varchar(255) U
Game Number	int(10)
StandardCost	int(10) N
ListPrice	int(10) N
Storage	int(5)
GenreID	int(10) N
SellStartDate	date
SellEndDate	date
LaunchDate	date
PlatformID	int(10)
Description	varchar(255) N
SagaID	int(10) N
ModifiedDate	date
GameReviewID	int(10) N

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.

Genre 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.

Genre	
GenreID	int(10) U
Name	varchar(255) U
Description	varchar(255) N

GameID	Name	GameNumber	StandardCost	ListPrice	Storage	GenreID	SellStartDate	SellEndDate	LaunchDate	PlatformID	Description	SagaID	ModifiedDate	GameReviewID
1	Example	1	10	1	10	1	2021-11-11	2021-11-12	2021-11-11	1	description	1	2021-11-11	1
* NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

GenreID	Name	Description
1	genre	description
* NULL	NULL	NULL

SagaID	Name	Quantity
1	example	1
* NULL	NULL	NULL

Saga	
SagaID	int(10) U
Name	varchar(255) N
Quantity	int(3) N

Saga 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.

GameListPriceHistory 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.

GameListPriceHistory	
GameID	int(10) U
StartDate	date
EndDate	date
ListPrice	int(10)
ModifiedDate	date

GameID	StartDate	EndDate	ListPrice	ModifiedDate
1	2021-11-11	2021-11-12	10	2021-11-11
* NULL	NULL	NULL	NULL	NULL

GameReview	
GameReviewID	int(10) U
GameID	int(10)
ReviewerName	varchar(255)
ReviewDate	date
EmailAddress	varchar(255)
Rating	int(2)
Comments	varchar(255) N
ModifiedDate	date

GameReview 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.

GameReviewID	GameID	ReviewerName	ReviewDate	EmailAddress	Rating	Comments	ModifiedDate
1	1	Alba	2021-11-11	268941@student.pwr.edu.pl	10	comment	2021-11-11
* NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

GameAudiovisuals	
GameID	int(10) U
GamePhotoID	int(10)
GameVideoID	int(10)
ModifiedDate	date N

GameVideo	
GameVideoID	int(10) U
ReviewVideo	varchar(255) N
PreviewVideo	varchar(255) N
ModifiedDate	date N

GameVideo and GamePhoto Tables will contain images or video links from a game. The images and video will be official so there will be a limited number of Audiovisuals per game.

GamePhoto	
GamePhotoID	int(10) U
Icon	varchar(255) N
WallPaper	varchar(255) N
Header	varchar(255) N
Preview	varchar(255) N
ModifiedDate	date N

GameID	GamePhotoID	GameVideoID	ModifiedDate
1	1	1	2021-11-11
* NULL	NULL	NULL	NULL

GamePhotoID	Icon	WallPaper	Header	Preview	ModifiedDate
1	icon	wallpaper	header	preview	2021-11-11
* NULL	NULL	NULL	NULL	NULL	NULL

GameVideoID	ReviewVideo	PreviewVideo	ModifiedDate
1	review	preview	2021-11-11
* NULL	NULL	NULL	NULL

Sales Tables

SalesOrderID	int(10)	U
RevisionNumber	int(10)	
OrderDate	date	
DueDate	date	
Status	int(1)	
SalesOrderNumber	int(10)	
AccountNumber	int(10)	
CustomerID	int(10)	
ContactID	int(10)	
TerritoryID	int(10)	
BIToAddressID	int(10)	
CreditCardApprovalCode	int(10)	
SubTotal	int(10)	
TaxAmt	int(10)	
Comment	varchar(255)	N
ModifiedDate	date	
CreditCardID	int(10)	
CurrencyRateID	int(10)	

SalesOrderHeader 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.

Game_ShoppingCartItem table is an intermediary table to create a many to many relation between Game and ShoppingCartItem Tables.

GameGameID	int(10)
ShoppingCartItemShoppingCartItemID	int(10)

ShoppingCartItemID	int(10)
ShoppingCartID	int(10)
GameID	int(10)
DateCreated	date
ModifiedDate	date

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

SalesOrderID	int(10)
SalesOrderDetailID	int(10)
TrackingNumber	int(10)
OrderQty	int(10)
GameID	int(10)
SpecialOfferID	int(10)
UnitPrice	int(10)
UnitPriceDiscount	int(10)
LineTotal	int(10)
ModifiedDate	date

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

SpecialOfferID	int(10)
GameID	int(10)
ModifiedDate	date

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

SpecialOfferID	int(10)
Description	varchar(255)
Discount	int(10)
Type	varchar(255)
Category	varchar(255)
StartDate	date
EndDate	date
ModifiedDate	date

SpecialOffer Table contains all the information regarding that special offer.

CurrencyRate Table contains the information about a currency rate with different from and to currencies. There will be two fields with

CurrencyRateID	int(10)
CurrencyRateDate	date
FromCurrencyCode	int(10)
ToCurrencyCode	int(10)
AverageRate	int(10)
EndOfDayRate	int(10)
ModifiedDate	date

CurrencyCodes so two relations will link that table with Currency Table.

CurrencyCode	int(10)
Name	varchar(255)
ModifiedDate	date

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

SalesTerritory 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.

TerritoryID	int(10)
Name	varchar(255)
CountryRegionCode	int(5)
Group	int(10)
SalesYTD	int(10)
SalesLastYear	int(10)
CostYTD	int(10)
CostLastYear	int(10)
ModifiedDate	date

ContactID	int(10)
CreditCardID	int(10)
ModifiedDate	date

ContactCreditCard 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.

CreditCardID	int(10)
CardTypeID	int(10)
CardNumber	int(10)
ExpMonth	int(2)
ExpYear	int(2)
ModifiedDate	date

CardTypeID	int(10)
Name	varchar(255)

CardType Table store the type of card that is stored in CreditCard Table.

Person Tables

ContactID	int(10)
TitleID	int(1)
FirstName	varchar(255)
MiddleName	varchar(255)
LastName	varchar(255)
EmailAddress	varchar(255)
Phone	int(11)
Password	int(10)
AdditionalContactInfo	varchar(255)
ModifiedDate	date
AddressID	int(10)

Contact 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.

TitleID	int(1)
Name	varchar(255)

Address	
AddressID	int(10) U
AddressTypeID	int(2)
AddressLine1	varchar(255)
AddressLine2	varchar(255)
City	varchar(255)
StateProvinceID	int(5)
PostalCode	int(6)
ModifiedDate	date

Although 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.

AddressType	
AddressTypeID	int(2) U
Name	varchar(255) U

StateProvince	
StateProvinceID	int(5) U
CountryRegionCode	int(5)
IsOnlyStateProvinceFlag	binary(1)
Name	varchar(255)
TerritoryID	int(10)
ModifiedDate	date

StateProvince Table contains the data regarding the States or Provinces that an address can belong to.

CountryRegion contains the countries where the states or provinces of the past table belong to.

CountryRegion	
CountryRegionCode	int(5) U
Name	varchar(255)
ModifiedDate	date

Territory	
TerritoryID	int(10) U
Name	varchar(255)
CountryRegionCode	int(5)
ModifiedDate	date

Territory Table contains the data of which territory belongs the Country or Region from the past table.

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), OrderQty 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, EmailAddress 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, CreditCardApprovalCode 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`.`currencyrates`, `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