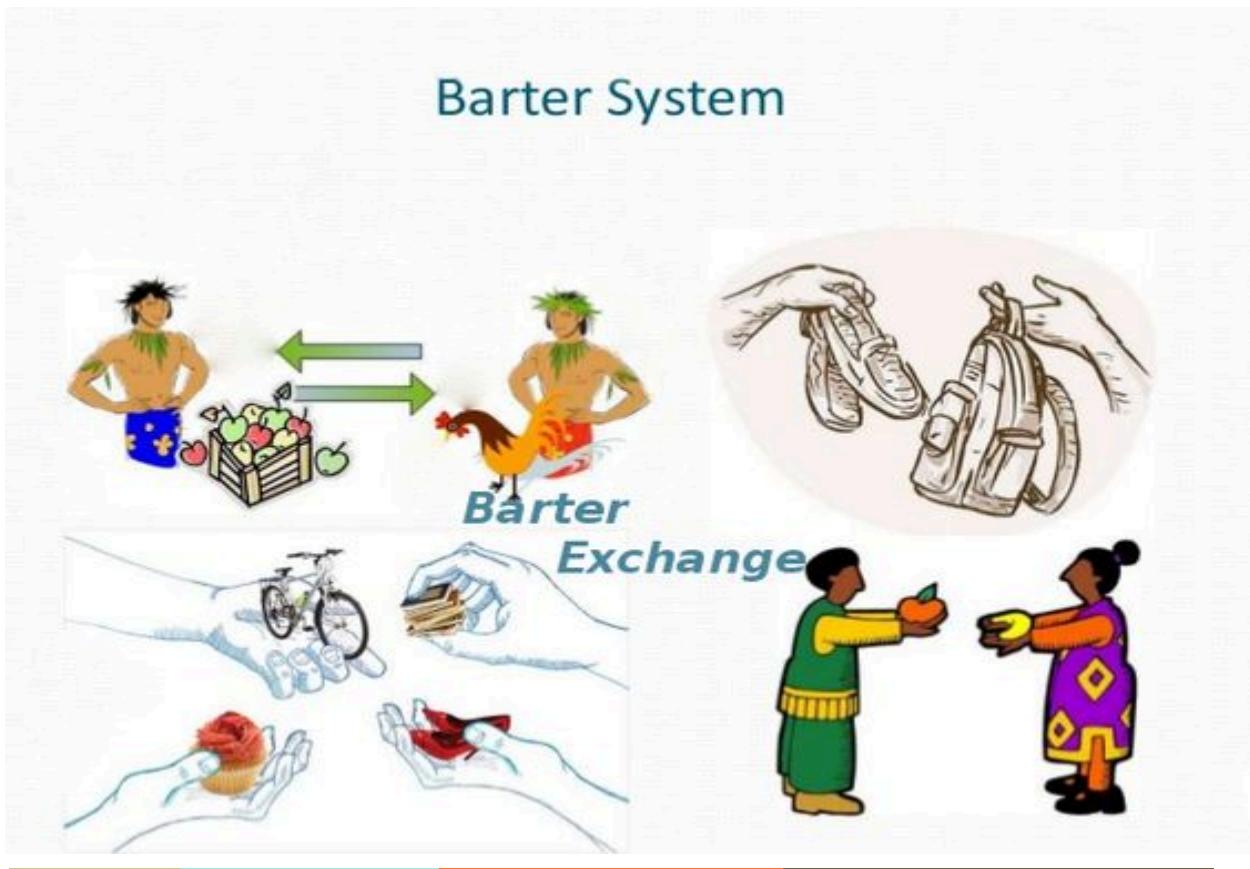


## Barter System



# BARTERING

I - SAFE 2022

By

Cecile Darelle MATSIMBO FOTSING &  
Line Melissa ONDAYE EYENGUI

<b>I. Glossary.....</b>	<b>2</b>
<b>II. Introduction.....</b>	<b>2</b>
<b>III. System Purpose.....</b>	<b>2</b>
<b>IV. Technical implementation.....</b>	<b>2</b>
A. Set of Data Used for Schema Validation.....	2
B. Complete Relational Schema Validation.....	3
<b>V. System Overview.....</b>	<b>4</b>
A. User guide.....	4
Rule 1: Explanation of Table 5 in a Form.....	4
Rule 2: Historical Overview of Bartering.....	5
Rule 3: Navigation Form “Barter_App” .....	5
Rule 4: Modifying Information in the Barter_App.....	6
B. Application Limits.....	8
C. Data Dictionary.....	8
D. The Data Models.....	9
<b>VI. Conclusion of the Project.....</b>	<b>10</b>
A. Technical Conclusion.....	10
B. Personal Conclusion.....	10

## I. Glossary

**ERD:** Entity-relationship Diagram

**LDM:** Logical Data Model

**CDM:** Conceptual Data Model

**SVG:** Scalable Vector Graphic

## II. Introduction

The subject of our work was inspired by a bundle of questions that we had in mind regarding the world's trade market. We were wondering, for example : how did people in the Ancient ages exchange products or services without money or even a credit card? Through researches on the internet, some websites satisfied our curiosity, but, most importantly, presented in depth the oldest trade technique that ever existed. The name “bartering” was given to it by English people while “troc” seems more used in the French language. In fact, its principle is very simple: two

barterers must only estimate the value of their products and/or services and agree to exchange them, with nothing in return.

### III. System Purpose

As the aim of this exercise relies on our ability to build an access file, we have decided to present the different relationships between each seller and customer, and between the items exchanged and their owners. Moreover, we also want to emphasize on the importance of “barter” as the only exchange means and the only way of obtaining something you wanted/needed during this time.

### IV. Technical implementation

#### A. Set of Data Used for Schema Validation

	<a href="#">idProduct</a>	<a href="#">ProductName</a>	<a href="#">CurrentValue</a>	<a href="#">ProductType</a>	<a href="#">Click to Add</a>
1	Wheat	1,83 - 4,57 \$	Food		
2	Cacao beans	2,89 - 3,24 \$	Food		
3	Services	Varies	Service		
4	Salt	0,34 - 1,68€	Spices		
5	Gold	0	Good		
6	Rice	1,25 - 6,09 \$	Food		
7	Ginger	2,18 - 10,17 €	Spices		
8	Cinnamon	2,16 - 5,59 \$	Spices		
9	Horses	2300 - 3000 €	Animal		
10	Wax	0	Good		
12	Corn	0,67 - 1,82 \$	Food		

Table 1: Historical Products

[idProduct](#)

	<a href="#">idActors</a>	<a href="#">IdProduct</a>	<a href="#">Click to Add</a>
1	2	7	
2	2	8	
3	3	4	
3	3	5	
4	4	10	
5	5	1	
6	6	9	
7	7	3	
*	0	0	

Table 2: Products owned by Actors

[idActor](#)

	<a href="#">IdExchange</a>	<a href="#">IdActors</a>	<a href="#">Click to Add</a>
1	1	7	
2	2	3	
2	2	6	
3	3	2	
3	3	5	
4	4	3	
4	4	4	
*	0	0	

Table 4: Actors involved in the trading

[idActor](#)

	<a href="#">idActors</a>	<a href="#">ActorsNames</a>	<a href="#">HistoricalTradingPeriod</a>	<a href="#">PossessedProducts</a>	<a href="#">Click to Add</a>
1	Aztecs	XIII century	Cacao		
2	India	III - II century BC	Ginger, Cinnamon		
3	Mali Empire	XIII - XVI century	Salt, Gold		
4	Holland	XIX century	Wax		
5	Roman Empire	I century	Wheat		
6	Portugal	XV century	Horses		
7	(?) militars	XIV	services		
8	Tunisia	XII Century	Corn		

Table 3: Historical Actors

[idExchange](#)

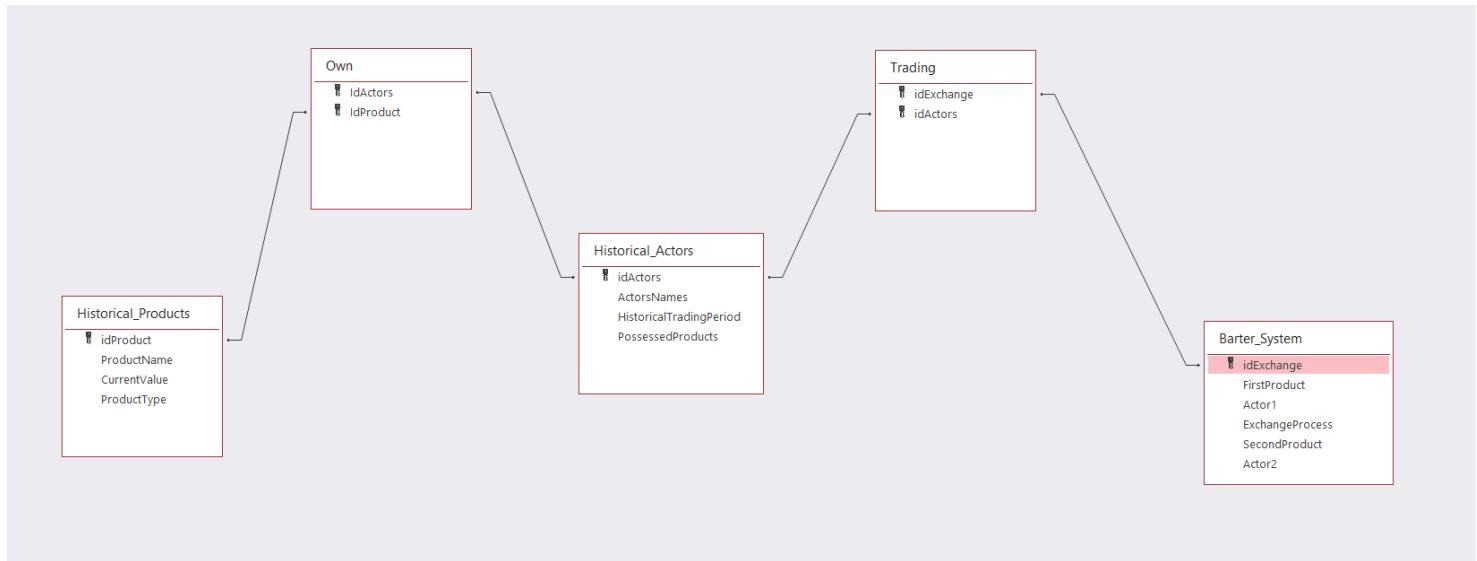
	<a href="#">idExchange</a>	<a href="#">FirstProduct</a>	<a href="#">Actor1</a>	<a href="#">ExchangeProcess</a>	<a href="#">SecondProduct</a>	<a href="#">Actor2</a>	<a href="#">Click to Add</a>
1	1	Cacao	Aztecs	Food - Service	Service	Militaries	
2	2	Horses	Portugal	Animal - Good	Gold	Mali Empire	
3	3	Ginger	India	Spices - Food	Wheat	Roman Empire	
4	4	Wax	Holland	Good - Food	Salt	Mali Empire	
5	5	Corn	Tunisia	Food - Animal	Horses	Portugal	

Table 5: Barter System

The schema above might be hard to pick up, but it's quite easy to understand. So, as you can see, the [idProduct](#) from **Table 1** is related to the [idActor](#) from **Table 3** in **Table 2**. This is to show which Product was owned by each Actor.

The same goes with the [idActor](#) from **Table 3** which is once more related with the [idExchange](#) from **Table 5** in **Table 4**. Here, we intend to explain how trading between the two parties took place, by showing who brought what and what he/she exchanged it for. The details of this process is explained in **Table 5**.

## B. Complete Relational Schema Validation



**Figure 1:** Relational Schema of the Database

The relationship above explains in detail how the tables above are related to each, by clearly defining the Primary Keys in each table.

## V. System Overview

### A. User guide

As explained above, bartering mainly involves the exchange (goods or services) for other goods or services without using money. We all know the pain of reading a manual that's lengthy, wordy, and smothered in text. So, in the following lines, we will try to briefly detail every step.

#### **Rule 1: Explanation of Table 5 in a Form**

*Figure 2* below captures what we tried to illustrate in **Table 5** above. As you can see, it has the same name as the table in question ([Barter\\_System](#)). This form will help you navigate easily in **Table 5** and it gives you information about the two parties (named as [Actor1](#) and [Actor2](#))

involved and the product they trucked. The product owned by each of them is mentioned on the next row. In our database, this form is named “S”.

idExchange	1	Change_id	<input type="button" value="▼"/>								
Actor1	Aztecs										
Product_Own	Cacao										
Actor2	Militaries										
Product_Own	Service										
Exchange_Process	Food - Service										
<table border="1"> <thead> <tr> <th>idActors</th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Karo</td> </tr> <tr> <td>7</td> <td>11</td> </tr> <tr> <td>*</td> <td></td> </tr> </tbody> </table>				idActors		1	Karo	7	11	*	
idActors											
1	Karo										
7	11										
*											

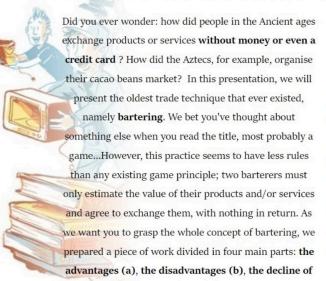
Record: 1 of 2 | Previous | Next | No Filter | Search

Figure 2: Barter System Form, “S”

The Previous and Next buttons will allow you to navigate easily across the Records without problem, or you can still use the “Change\_id” Combo box. However, it might be more useful when you are looking for an exact exchange. That is, when you know which actors and products are involved. You can also notice that the exact id’s of the Actors actually appears on the *little table* in the form. It might not be useful, but it gives you more information about the Actor.

## Rule 2: Historical Overview of Bartering

**WHAT IS BARTERING ?**



Do you ever wonder: how did people in the Ancient ages exchange products or services **without money or even a credit card**? How did the Aztecs, for example, organise their cacao beans market? In this presentation, we will present the oldest trade technique that ever existed, namely **bartering**. We bet you've thought about something else when you read the title, most probably a game... However, this practice seems to have less rules than any existing game principle: two barterers must only estimate the value of their products and/or services and agree to exchange them, with nothing in return. As we want you to grasp the whole concept of bartering, we prepared a piece of work divided in four main parts: the advantages (a), the disadvantages (b), the decline of the system (c), and lastly, its limits(d).

**Advantages of Bartering**

- **Flexibility:** Bartering made it possible for people to trade one products or even services rather than just material items.
- **Cash savings:** One of the main benefits of the barter trade is that they receive it without spending a penny. This then allowed to reserve, which is significant for new businesses.
- **Simplicity:** Barter trade is a simple system as compared to the nowadays such as the adverse balance of payments and foreign
- **No overexploitation of natural resources:** In a barter economy, the needs of the customer for the exchange, thereby meeting the needs of the producer.



further advantages include;

- **No concentration:** is not focused on storing common pressure (use
- **Build a strong relationship:** the exchange of businesses. So, time into strong would have a

Figure 3: Definition, Illustration & Advantages of Bartering

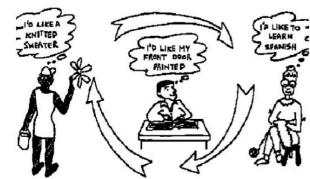
**Disadvantages of Bartering**



Just as any other type of trade, it has both advantages as well as disadvantages. While barter trade has immediate benefits, as we have seen above there are also certain disadvantages. Some of the cons of barter include;

- **Lack of divisibility:** In barter system, goods and services are not always equal, that is; there is no way to ensure that both parties' exchange of value fair. This can arise in issues especially when one party feels that what they giving is more valuable than what the other one is exchanging.
- **Lack of double coincidence of wants:** Bartering occurs only when two people desire an exchange of commodities that are mutually needed by each other. But in the absence of such coincidences of wants, there will be no transaction. For example: A farmer may not agree to trade his cow for a sack of rice. This is the major problem faced in barter system.
- **Problems of transportation and storage:** It's not easy to transport good or services in this system, especially when the traded good is big. Lack of storage space and transportation inconveniences can only be avoided with the use of money.

**Importance of Bartering**



Bartering enables individuals to trade items that they own but are not using for items that they need. In an **economical crunch**, this trading system can be a great way to get the goods and services you need without having to pull money.

On a broader level of view, bartering also results in the optimal allocation of resources by exchange with quantities that represent similar values. Bartering also help economies achieve **economical equilibrium** (occurs when quantity demanded = quantity supply).

**The decline of the System**

The invention of **money** led to the end of bartering. Nevertheless, the initiation of money as an exchange means business was not the only reason behind the failure of the barter system. Other reason was the impossibility of producing large quantities of expensive goods and services.

Although most countries nowadays uses the monetary system, individuals are still involved in barter system.

\* *simply because can be too efficient exchanging a service for a good with*

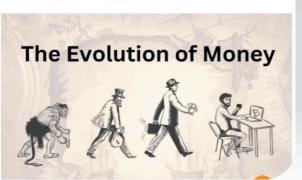
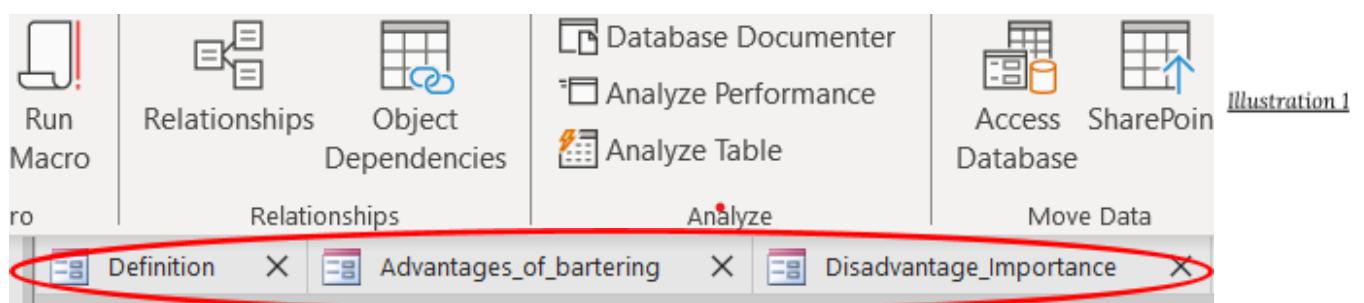


Figure 4: Disadvantage, Importance & End of Bartering

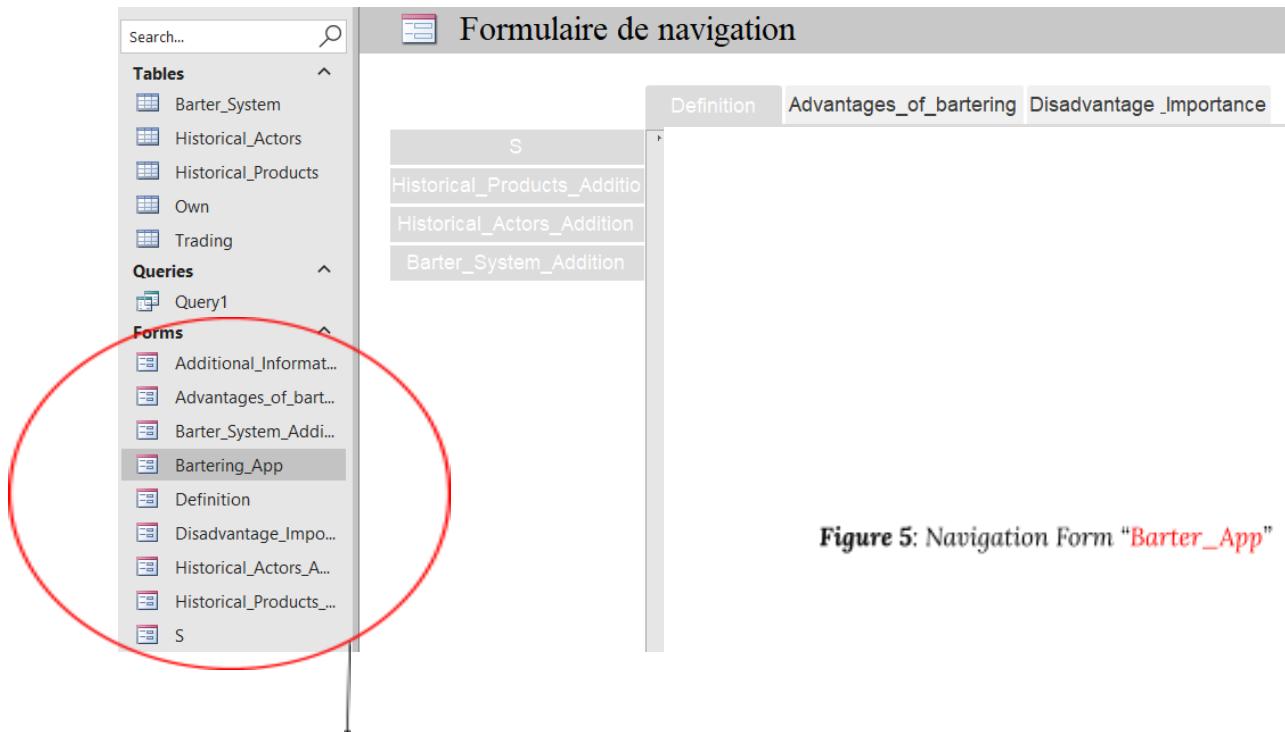


The figures above (Figure 3 & Figure 4), elaborate some historical information. On the database, they are divided into three forms (**Definition**, **Advantages\_of\_bartering** & **Disadvantage\_Importance**) as shown on the *Illustration 1* above.

This information will assist you in a better understanding of the term “Barter”. They can't be modified since they are the creators' personal research. Hence, if you want more knowledge about the subject, you can czech on the internet.

### **Rule 3: Navigation Form “Barter\_App”**

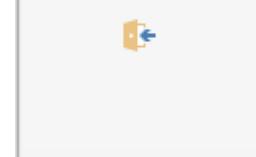
A Navigation Form was created to aid and abet a better access of our App for the user. This form permits the user to not only approach the uploaded information but also allows him/her to add, modify and save more data on it.



**Figure 5: Navigation Form “Barter\_App”**

The following circled forms appear on the **Barter\_App** as you may notice. You just have to click on any form in this App for it to open and display what it contains.

This little icon on *Illustration 2* beside the Navigation form is simply to close the form after saving the changes you have brought to.



**Illustration 2**

## Rule 4: Modifying Information in the Barter\_App

The screenshot shows the 'Historical\_Products' additional form. At the top, there are tabs for 'Definition', 'Advantages\_of\_bartering', and 'Disadvantage\_Importance'. The main area contains fields for 'idProduct' (set to 1), 'Change\_id' (a dropdown menu), 'ProductName' (set to 'Wheat'), 'CurrentValue' (set to '1.83 - 4.57 \$'), and 'ProductType' (set to 'Food'). Below these fields is a dropdown menu labeled 'IdActors' with options 5 and 0. A red circle highlights this dropdown. To the right of the form are buttons for 'Edit', 'Save', and 'Close Form'. At the bottom, there are navigation buttons for 'Previous' and 'Next Record'.

**Figure 6:** Historical Products Additional Form

This Form (**Figure 6**) accredit you to add new Goods and Services into the system. These Products must be one of those traded during this time (Barter period). You also have the possibility to go through the products already present in the Form. This is done by clicking on the Combo box beside **Change\_id**. This will immidiately change the **ProductName**, **CurrentValue** and **ProductType** to the one corresponding to the chosen **id**.

The screenshot shows the 'Historical\_Actors' additional form. At the top, there are tabs for 'Definition', 'Advantages\_of\_bartering', and 'Disadvantage\_Importance'. The main area contains fields for 'idActors' (set to 1), 'Change\_id' (a dropdown menu), 'ActorsNames' (set to 'Aztecs'), 'HistoricalTradingPeriod' (set to 'XIII century'), and 'PossessedProducts' (set to 'Cacao'). Below these fields is a dropdown menu labeled 'Change\_id' with options 1 and 2. A red circle highlights this dropdown. To the right of the form are buttons for 'Edit', 'Save', and 'Close Form'. At the bottom, there are navigation buttons for 'Previous' and 'Next Record'.

**Figure 7:** Historical Actors Additional Form

This Form (**Figure 7**) accredit you to add new Actors into the system. They must be one of those parties who took place in trading during this time (Barter period). You also have the possibility to go through the different Actors already present in the Form by clicking on the Combo box beside **Change\_id**. This will immidiately change the **ActorName**, **HistoricalTradingPeriod** and **PossessedProduct** to the one corresponding to the chosen **id**.

The screenshot shows a Microsoft Access form titled "Barter\_System". The form has several input fields and a table. At the top, there are fields for "IdExchange" (with a dropdown arrow) and "Actor1" (set to "Aztecs"). Below that is "Actor2" (set to "Militaries") and "Possesed\_Product" (with two entries: "Cacao" and "Service"). Underneath these is an "ExchangeProcess" field containing "Food - Service". At the bottom of the form is a table with columns "idActors" and "ActorsNames". The data in the table is circled in red and shows rows for 1 (Aztecs), 7 (India), and 0. There are also buttons for "Save", "Edit", "Undo", and "Close Form". On the left side of the form, there is a vertical bar with buttons for "Edit" and "Historical\_Products\_Addition", "Historical\_Actors\_Addition", and "Barter\_System\_Addition".

**Figure 8:** Barter System Additional Form

Beyond is **Figure 8**, which as his name states, is a form that ingresses the user to the Barter system and ensures a perfect addition of data. The ‘**Undo**’ button will help you cancel the whole record in case of any insufficient, uncertainty or lack of information.

## B. Application Limits

Introducing the shortage of your project can sometimes be more difficult than the conception. This is simply because as the conceptor you would prefer to have a 100% app. However, limits are areas of improvements. Talking about the shortcomings of our App, we will classify them from the most pertinent to the least.

Our main restraint of the **Barter\_App** is the lack of “*Request*” that will perform an automatic addition of the additional information in both **Table 2 & Table 4** respectively.

Explanation: As you can notice, when a new Product or Actor is added, normally they should also appear on “*Products Owned by actors*” (**Table 2**). But, this isn’t possible because the information already present on this table was manually entered and in this case the creation of a “*Request*” that can automatically solve this problem will be of great use. For example:

Illustration 3 displays three tables illustrating the relationship between products, actors, and their possessions:

- Products Table:**

idProduct	ProductName	CurrentValue	ProductType
1	Wheat	1,83 - 4,57 \$	Food
2	Cacao beans	2,89 - 3,24 \$	Food
3	Services	Varies	Service
- Actors Table:**

idActors	ActorsNames	HistoricalTradingPeriod	PossessedProducts
1	Aztecs	XIII century	Cacao
2	India	III - II century BC	Ginger, Cinnamon
- Relationship Table:**

IdActors	IdProduct	Click to Add
1	2	
2	7	
2	8	

**Illustration 3**

From *illustration 3*, you can have a view of what we tried to explain. So Aztecs i.e. idActor ‘1’ possess Cacao/Cacao beans i.e. idProduct ‘2’, So they are linked by their id’s on the little extract of **Table 2** above. Same goes for **Table 4**, ‘*Actors involved in trading*’

## C. Data Dictionary

C:\Users\33779\Documents\Database presentation _ Franck.accdb		mardi 16 janvier 2024																					
Table: Barter_System		Page: 1																					
<b>Properties</b>																							
AlternateBackShade: 95 AlternateBackTint: 100 BackTint: 100 DatasheetGridlinesThemeCol 3 DefaultView: 2 FilterOnLoad: False  HideNewField: False NameMap: Long binary data OrderByOnLoad: True PublishToWeb: 1 RecordCount: 5 TotalsRow: False																							
AlternateBackThemeColorInd 1 BackShade: 100 DatasheetForeColorInd 0 DateCreated: 05/12/2023 17:23:34 DisplayViewsOnSharePointSit 1  GUID: {guid {3FF1B419-21F9-40C9-8933-59F476631988}}  LastUpdated: 16/01/2024 08:17:59 OrderByOn: False Orientation: Left-to-Right ReadOnlyWhenDisconnected False ThemeFontIndex: 1 Updatable: True																							
<b>Columns</b>																							
<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>idExchange</td> <td>Long Integer</td> <td>4</td> </tr> <tr> <td>FirstProduct</td> <td>Short Text</td> <td>255</td> </tr> <tr> <td>Actor1</td> <td>Short Text</td> <td>255</td> </tr> <tr> <td>ExchangeProcess</td> <td>Short Text</td> <td>255</td> </tr> <tr> <td>SecondProduct</td> <td>Short Text</td> <td>255</td> </tr> <tr> <td>Actor2</td> <td>Short Text</td> <td>255</td> </tr> </tbody> </table>			Name	Type	Size	idExchange	Long Integer	4	FirstProduct	Short Text	255	Actor1	Short Text	255	ExchangeProcess	Short Text	255	SecondProduct	Short Text	255	Actor2	Short Text	255
Name	Type	Size																					
idExchange	Long Integer	4																					
FirstProduct	Short Text	255																					
Actor1	Short Text	255																					
ExchangeProcess	Short Text	255																					
SecondProduct	Short Text	255																					
Actor2	Short Text	255																					

**Figure 9:** Data Dictionary

“*You can have a better view of it at the end of the report.*”

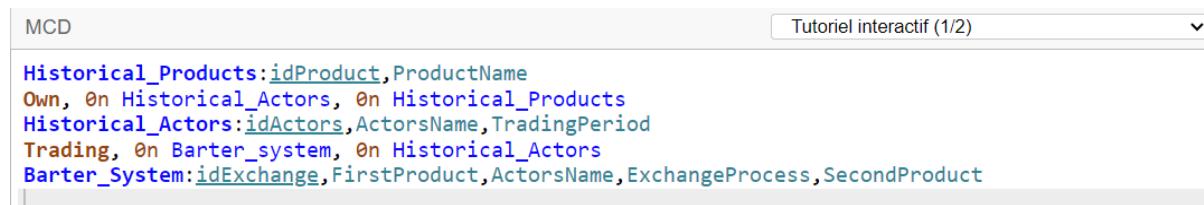
The Data dictionary stores and communicates metadata about data in our database. It helps us to catalog and communicate the structure and content of data, and provides meaningful descriptions for individually named data objects. The one below is just an overview of what a data dictionary looks like.

## D. The Data Models

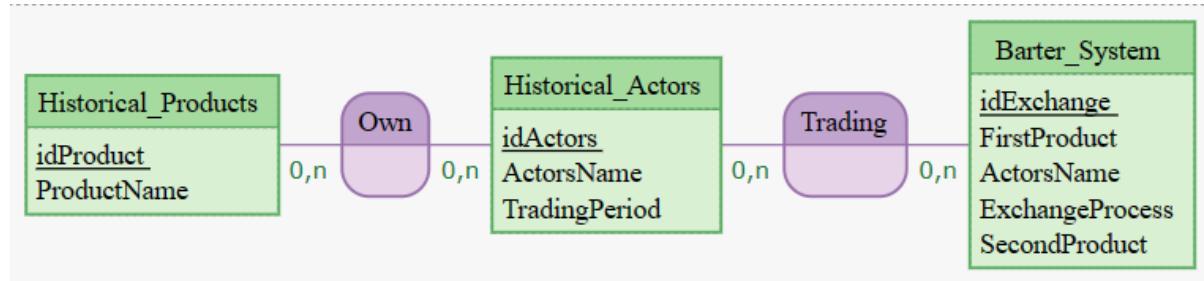
### Mocodo Concept:

Mocodo is an open-source tool for designing and teaching relational databases. It takes as an input a textual description of both entities and associations of an entity-relationship diagram and outputs a vectorial drawing in SVG and a relational schema.

As you can see on **Figure 10**, which is the input data, each line ("box" in the following) constitutes the definition of an entity or an association.

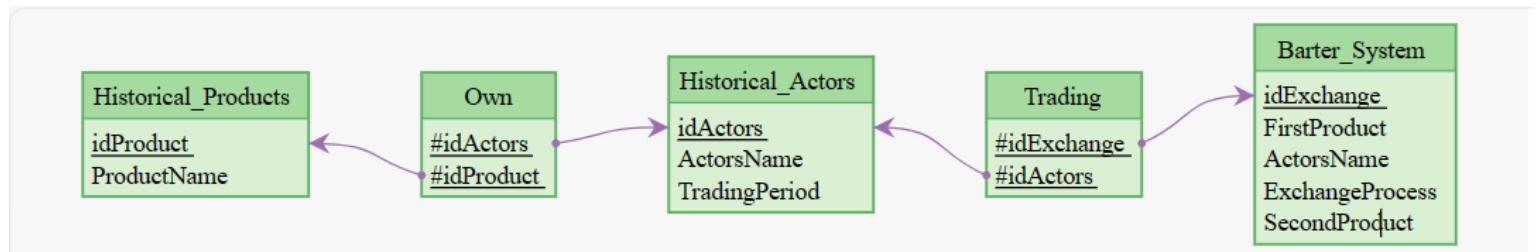


**Figure 10:** Mocodo Concept of Database



**Figure 11:** Data Conceptual Model (CDM) of the Database

The preliminary step in order to model our data is to collect information. This collected information was then treated for us to obtain our CDM and LDM, as you can see on **Figure 11** and **Figure 12** respectively. The **0,n** represents a “*One-Many relation*”.



**Figure 12:** Logical Data Model (LDM) of our Database

## VI. Conclusion of the Project

### A. Technical Conclusion

The implementation of this barter-based database project was to demonstrate the viability and potential of leveraging a decentralized and collaborative exchange system. Through the development and deployment of this database, several key technical insights and outcomes have been achieved.

Firstly, the design and implementation of the database infrastructure have successfully facilitated the recording and tracking of barter transactions. The database schema efficiently

captures essential details such as item descriptions, participant details, and transaction history, enabling seamless retrieval and analysis of barter activities. The relational structure of the database has proven effective in maintaining data integrity and ensuring the accuracy of recorded exchanges. Secondly, the scalability of the database has been a focal point, anticipating the potential growth and increasing complexity of the barter network. The project has laid a foundation for scalability by adopting efficient Forms and employing database clustering techniques. This ensures that the system can handle a growing volume of transactions and users without compromising on responsiveness.

Furthermore, the user interface and user guide have been carefully crafted to facilitate user engagement and ease of use. A user-friendly environment is put in place to encourage broader adoption of the barter system. While the technical implementation has been successful, it is crucial to acknowledge the ongoing nature of database projects. Continuous monitoring, performance optimization, and periodic updates will be essential to adapt to evolving user needs, technological advancements, and potential security threats.

## B. Personal Conclusion

In a nutshell, without knowing, we built a project very similar to a day-to-day activity that few enterprises still do today: an Access file with customers (barterers) on one hand and purchased items (bartering products) on another. It helped us to understand the complexity to track commercial operations and where it lies. Illustrating the connection between two entities may seem effortless while, in truth, more than one relationship may be necessary to explain the elements that link one to the other.

At the beginning, we were able to depict our whole idea with a simple drawing. Nonetheless, once we started working on Access, we needed to create a bunch of tables, relationships, sections, etc. It involved a lot of thinking and doubts. We sought help by asking a lot of questions (on the Internet, to the teacher in charge of the course, ...), especially regarding the ideal way we could connect two distinct barterers with their respective products. One of us was more at ease at understanding the database aspect; therefore, the group dynamic slowly changed from two students working together to one student learning from her fellow classmate performing the toughest tasks on Access.

It was difficult at first to experience this change, but with time, Cecile explained the benefits of this organization and I, Line, agreed to work this way. In fine, the activity held a potential for students, like us, to get more acquainted with the historical and scientific aspects of the course.

### **Properties**

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorIn	0
DatasheetGridlinesThemeCol	3	DateCreated:	05/12/2023 17:23:34
DefaultView:	2	DisplayViewsOnSharePointSit	1
FilterOnLoad:	False	GUID:	{guid {3FF1B419-21F9-40C9-8933-59F4766319B8}}
HideNewField:	False	LastUpdated:	16/01/2024 08:17:59
NameMap:	Long binary data	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Left-to-Right
PublishToWeb:	1	ReadOnlyWhenDisconnected	False
RecordCount:	5	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

### **Columns**

Name	Type	Size
idExchange	Long Integer	4
FirstProduct	Short Text	255
Actor1	Short Text	255
ExchangeProcess	Short Text	255
SecondProduct	Short Text	255
Actor2	Short Text	255

### **Properties**

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetFontHeight:	11
DatasheetFontItalic:	False	DatasheetFontName:	Calibri
DatasheetFontUnderline:	False	DatasheetFontWeight:	Normal
DatasheetForeColor:	0	DatasheetForeColor12:	0
DatasheetForeThemeColorIndex:	0	DatasheetGridlinesThemeColorIndex:	3
DateCreated:	05/12/2023 17:23:32	DefaultView:	2
DisplayViewsOnSharePointSite:	1	FilterOnLoad:	False
GUID:	{guid {5B121675-23DD-41DF-9AC6-6F3DF364D246}}	HideNewField:	False
LastUpdated:	13/01/2024 15:56:31	NameMap:	Long binary data
OrderByOn:	False	OrderByOnLoad:	True
Orientation:	Left-to-Right	PublishToWeb:	1
ReadOnlyWhenDisconnected:	False	RecordCount:	8
TabularCharSet:	0	TabularFamily:	34
ThemeFontIndex:	-1	TotalsRow:	False
Updatable:	True		

### **Columns**

Name	Type	Size
idActors	Long Integer	4
ActorsNames	Short Text	255
HistoricalTradingPeriod	Short Text	255
PossessedProducts	Short Text	255

### **Properties**

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorIn	0
DatasheetGridlinesThemeCol	3	DateCreated:	05/12/2023 17:23:28
DefaultView:	2	DisplayViewsOnSharePointSit	1
FilterOnLoad:	False	GUID:	{guid {795CE2B8-3602-42A6-874C-511CDFD19328}}
HideNewField:	False	LastUpdated:	13/01/2024 15:56:31
NameMap:	Long binary data	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Left-to-Right
PublishToWeb:	1	ReadOnlyWhenDisconnected	False
RecordCount:	10	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

### **Columns**

Name	Type	Size
idProduct	Long Integer	4
ProductName	Short Text	255
CurrentValue	Short Text	255
ProductType	Short Text	255

### **Properties**

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorIn	0
DatasheetGridlinesThemeCol	3	DateCreated:	12/12/2023 17:13:32
DefaultView:	2	DisplayViewsOnSharePointSit	1
FilterOnLoad:	False	GUID:	{guid {8F7420C1-F01B-44BF-8321-7B56699BAD22}}
HideNewField:	False	LastUpdated:	13/01/2024 15:56:48
NameMap:	Long binary data	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Left-to-Right
PublishToWeb:	1	ReadOnlyWhenDisconnected	False
RecordCount:	9	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

### **Columns**

Name	Type	Size
IdActors	Long Integer	4
IdProduct	Long Integer	4

### **Properties**

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorIn	0
DatasheetGridlinesThemeCol	3	DateCreated:	12/12/2023 17:13:36
DefaultView:	2	DisplayViewsOnSharePointSit	1
FilterOnLoad:	False	GUID:	{guid {5006B63F-D0BE-4818-A880-1F71FCC68C60}}
HideNewField:	False	LastUpdated:	16/01/2024 16:38:04
NameMap:	Long binary data	OrderBy:	[Trading].[idActors]
OrderByOn:	True	OrderByOnLoad:	True
Orientation:	Left-to-Right	PublishToWeb:	1
ReadOnlyWhenDisconnected	False	RecordCount:	8
ThemeFontIndex:	1	TotalsRow:	False
Updatable:	True		

### **Columns**

Name	Type	Size
idExchange	Long Integer	4
idActors	Long Integer	4