

|  |
| --- |
| Business Template  **AN AUCTION HOUSE DATABASE** |
| **Logo / Image** |

Contents

[1 Business Description 3](#_Toc62212630)

[1.1 Business background 3](#_Toc62212631)

[1.2 Problems. Current Situation 3](#_Toc62212632)

[1.3 The benefits of implementing a database. Project Vision 3](#_Toc62212633)

[2 Model description 3](#_Toc62212634)

[2.1 Definitions & Acronyms 3](#_Toc62212635)

[2.2 Logical Scheme 3](#_Toc62212636)

[2.3 Objects 3](#_Toc62212637)

# 

# Business Description

## Business background

The company sells antiques and artwork at auction. The owners of items put up for auction by the company are legal sellers. The people who purchase these items are referred to as buyers. After receiving a batch of items from the sellers, the firm decides at which auction it will be more profitable to present a particular item. Before the next auction, each item displayed is assigned a separate lot number, which plays the same role as the product code entered before. Two items sold at different auctions may have the same lot numbers. The details about each auction are recorded by the company. The date, place, and time are noted, as well as any other specifics (for example, oil paintings from before 1900). Information about each item sold is also entered: the auction for which it is claimed, the lot number, the seller, the starting price, and a brief verbal description. The seller is allowed to display any number of items, and the buyer can purchase any number of items. The same person or firm can act as both a seller and a buyer. After the auction, employees of the auction house record the actual price paid for an item and the buyer's data.

## Problems. Current Situation

# The problem we're talking about is when people want to join auctions, but it's hard because they have to be there at a certain time. Sometimes, they can't make it because they live far away. This makes them miss out on the auctions a lot. Also, it's expensive and takes a long time to travel far just to buy one thing. Not everyone can go to these auction places — only a few chosen ones with special tickets. So, many people who want to buy things can't. These auctions are also usually in one area, so if there's an auction in another faraway place, people don't even know. This means only a few people go to these auctions, and the seller might not get the price they were hoping for their stuff.

## the Benefits of implementing a database. Project Vision

To make things better and easier, buyers can share their opinions about the sellers. They can talk about how accurate the item descriptions were, how happy they were with the seller's messages, and how fast the seller sent the products.

This special computer system is like an 'Online Auction.' Anyone, no matter where they are, can use it anytime. It helps people buy things at lower prices, choose from many different items, and it's faster than traditional markets. It's made to be simple for everyone. People who want to sell things can easily join and start selling without much trouble. The main idea of this website is to create a great online way of buying things that saves time and money for everyone.

# Model description

## Definitions & Acronyms

# Entities

# 1. User: This entity captures information about the buyer or seller who signed up on the online auction web portal.

# 2. Administrator: This entity captures information about all the individuals responsible for the functioning of the web portal.

# 3. Product: This entity captures information about the products available for auction.

# 4. Auction: This entity captures information about products currently being auctioned.

# 5. Feedback: This entity captures information about the reactions/feedback from any buyer or seller regarding a product auction.

# 6. Bid: This entity captures information about the bids placed by buyers for a product, i.e., the price a buyer is willing to pay in an auction.

# 7. Product\_Category: This entity captures information about the category to which a specific product belongs.

# 8. Shipment: This entity captures information about the shipment details of a sold product following an auction.

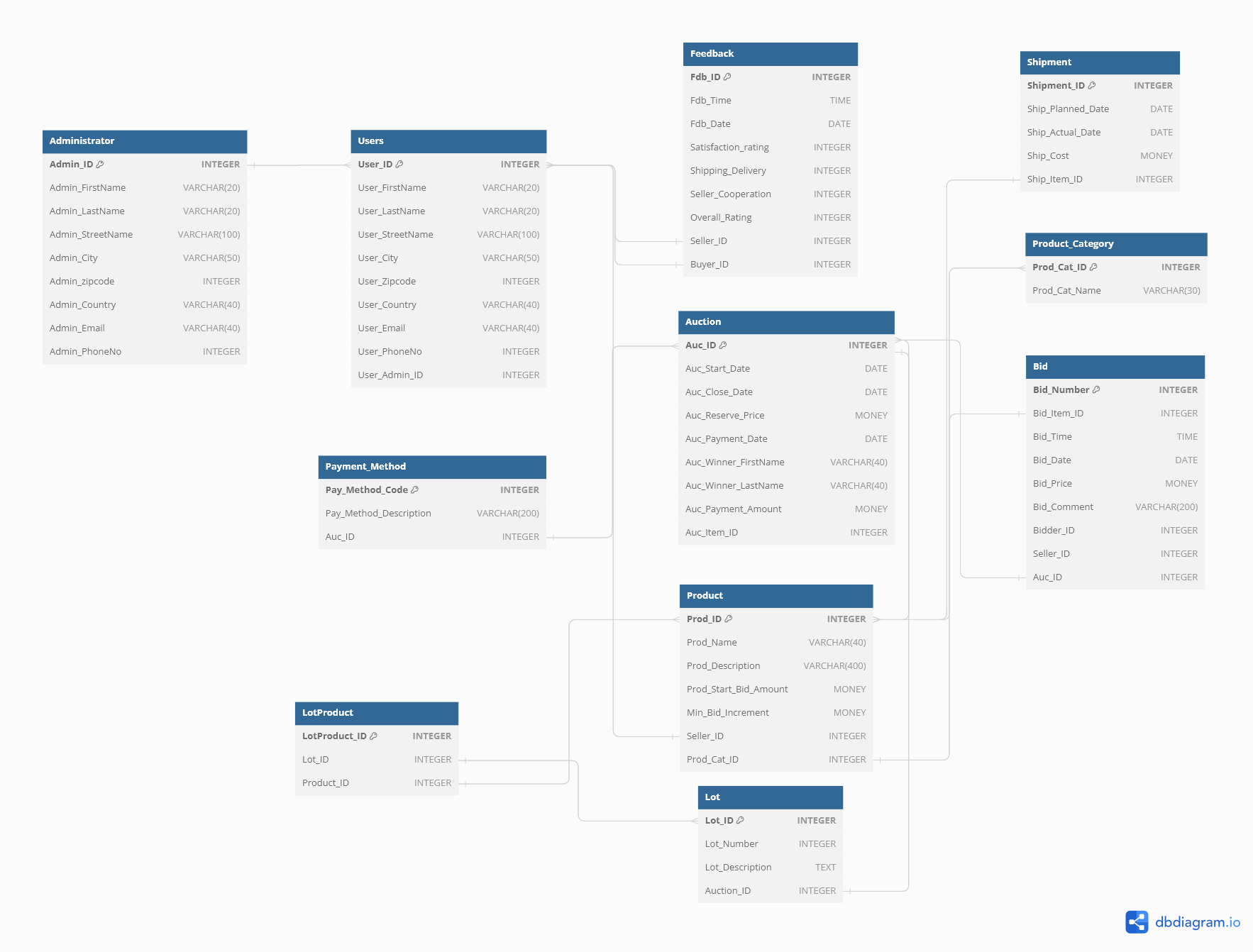
# 9. Payment\_Method: This entity captures information about the chosen payment method for any sold product by a buyer or accepted payment methods by the seller for any product.

# 10. Lot: This entity captures information about a group of products auctioned together, also known as a lot.

11. THE LOTPRODUCT: TABLE REPRESENTS THE RELATIONSHIP BETWEEN LOTS AND PRODUCTS IN A DATABASE.

## Logical Scheme

<image>



## Objects

**Table: Administrator**

The Administrator table represents a collection of information about administrators within a system. Administrators typically manage and oversee various aspects of the system, and this table is designed to store their personal and contact details.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **Administrator** | Admin\_ID | INTEGER [PK] |
| Admin\_FirstName | VARCHAR(20) |
| Admin\_LastName | VARCHAR(20) |
| Admin\_StreetName | VARCHAR(100) |
| Admin\_City | VARCHAR(50) |
| Admin\_zipcode | INTEGER |
| Admin\_Country | VARCHAR(40) |
| Admin\_Email | VARCHAR(40) |
| Admin\_PhoneNo | INTEGER |

**Table: Users**

The Users table represents a collection of information about individuals or users within a system. It is designed to store personal and contact details of the system's users.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **Users** | User\_ID | INTEGER [PK] |
| User\_FirstName | VARCHAR(20) |
| User\_LastName | VARCHAR(20) |
| User\_StreetName | VARCHAR(100) |
| User\_City | VARCHAR(50) |
| User\_zipcode | INTEGER |
| User\_Country | VARCHAR(40) |
| User\_Email | VARCHAR(40) |
| User\_PhoneNo | INTEGER |

**Table: Product\_Category**

The Product\_Category table is designed to represent various categories or types of products within a system. Each product category is identified by a unique identifier and a descriptive name.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **Product Category** | Prod\_Cat\_ID | INTEGER [PK] |
| Prod\_Cat\_Name | VARCHAR(30) |

**Table: Product**

The Product table represents individual products available within a system, facilitating detailed tracking and management of each product's information, pricing, and categorization.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **Product** | Prod\_ID | INTEGER [PK] |
| Prod\_Name | VARCHAR(40) |
| Prod\_Description | VARCHAR(400) |
| Prod\_Start\_Bid\_Amount | MONEY |
| Min\_Bid\_Increment | MONEY |
| Seller\_ID | INTEGER |
| Prod\_Cat\_ID | INTEGER |

**Table: Auction**

The Auction table represents information related to auctions for products within a system, including auction details, payment information, and winner details.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **Auction** | Auc\_ID | INTEGER [PK] |
| Auc\_Start\_Date | DATE |
| Auc\_Close\_Date | DATE |
| Auc\_Reserve\_Price | MONEY |
| Auc\_Payment\_Date | DATE |
| Auc\_Winner\_FirstName | VARCHAR(40) |
| Auc\_Winner\_LastName | VARCHAR(40) |
| Auc\_Payment\_Amount | MONEY |
| Auc\_Item\_ID | INTEGER |

**Table: Feedback**

The Feedback table is designed to capture feedback and ratings provided by buyers regarding their experience with a specific seller. It records various aspects of the transaction and interaction to evaluate the satisfaction of the buyer.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **Feedback** | Fdb\_ID | INTEGER [PK] |
| Fdb\_Time | TIME |
| Fdb\_Date | DATE |
| Satisfaction\_rating | INTEGER |
| Shipping\_Delivery | INTEGER |
| Seller\_Cooperation | INTEGER |
| Overall\_Rating | INTEGER |
| Seller\_ID | INTEGER |
| Buyer\_ID | INTEGER |

**Table: Bid**

The Bid table represents bids made by users in an auction for specific items. It records important details about each bid, including bid number, bid time, bid date, bid price, comments, and the associated auction and users (bidders and sellers).

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **Bid** | Bid\_Number | INTEGER [PK] |
| Bid\_Item\_ID | INTEGER |
| Bid\_Time | TIME |
| Bid\_Date | DATE |
| Bid\_Price | MONEY |
| Bid\_Comment | VARCHAR(200) |
| Bidder\_ID | INTEGER |
| Seller\_ID | INTEGER |
| Auc\_ID | INTEGER |

**Table: Shipment**

The Shipment table represents information related to the shipping of items, including planned and actual shipping dates, shipping cost, and the associated item being shipped.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **SHIPMENT** | Shipment\_ID | INTEGER [PK] |
| Ship\_Planned\_Date | DATE |
| Ship\_Actual\_Date | DATE |
| Ship\_Cost | MONEY |
| Ship\_Item\_ID | INTEGER |

**Table: Payment\_Method**

The Payment\_Method table represents various payment methods that can be used for transactions in the system. It provides descriptions and identifiers for different payment options available to users during auctions or transactions.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **Payment\_Method** | Pay\_Method\_Code | INTEGER [PK] |
| Pay\_Method\_Description | VARCHAR(200) |
| Auc\_ID | INTEGER |

**Table: Lot**

The Lot table represents distinct groups of items or products that are auctioned as a single unit. Each lot is assigned a unique identifier and includes relevant information such as lot number, description, and the associated auction.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **Lot** | Lot\_ID | INTEGER [PK] |
| Lot\_Number | INTEGER |
| Lot\_Description | TEXT |
| Auction\_ID | INTEGER |

**Table: LotProduct**

The LotProduct table represents the relationship between lots and products in a database. It serves to associate specific products with certain lots. This relationship is crucial for managing and tracking inventory within a system.

|  |  |  |
| --- | --- | --- |
| Table Name | Field Name | Data type |
| **LotProduct** | LotProduct\_ID | INTEGER [PK] |
| Lot\_ID | INTEGER |
| Product\_ID | INTEGER |

In the provided project structure, the **many-to-many** relationship can be observed between the *Lot table* and the *Product table* through the *LotProduct* table.