

|  |
| --- |
| Business Template  **Subject areas** |
| **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 auction house specializes in selling antiques and artwork through auction. Seller submit their item to the company, which then decides the best auction to present them. With company employees, each item is assigned to a lot number before the auction. Buyer bid on item, and the final sale price is recorded with the starting price. Time of the sales is also recorded. Both buyer and seller can participate in multiple auction, and the same person or a company can participate both as a seller or as a buyer.

## Problems. Current Situation

Currently, auction records are managed manually, leading to inefficiencies, data loss, and errors. Tracking transactions, bid, and item histories is cumbersome. The absence of a centralized database makes it difficult to retrieve past auction details and verify sales records.

## the Benefits of implementing a database. Project Vision

A database system will improve efficiency, accuracy, and accessibility of auction data. It will enable easy tracking of item, bid, and final sale prices while ensuring data integrity. The system will allow auction house employees to access and update auction details quickly, improving the overall auction process.

# Model description

## Definitions & Acronyms

**Lot** **Number**: A unique identifier assigned to each auctioned item (*referenced in the Item table*).

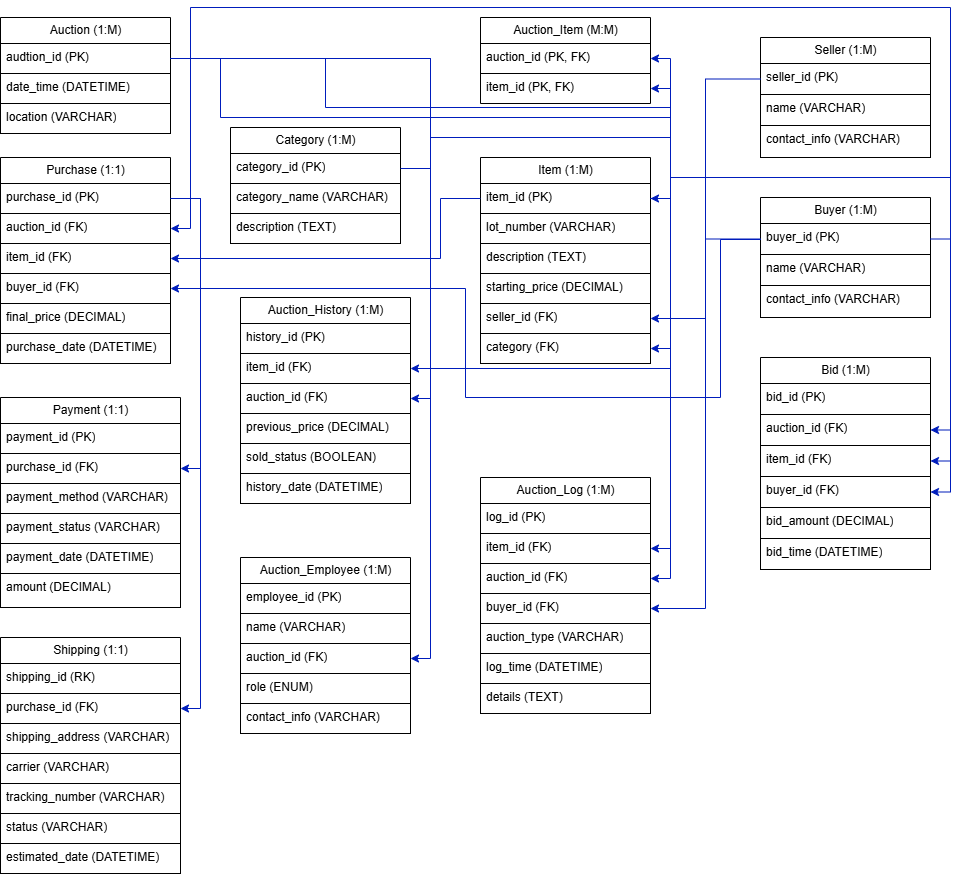
**Seller**: The person or entity providing an item for auction (*referenced in the Seller table*).

**Buyer**: The person or entity purchasing an item at auction (*referenced in the Buyer table*).

**Auction**: An event where item are bid on and sold to the highest bidder (*referenced in the Auction table*).

**Final Sale Price**: The price at which an item is sold after bidding (*stored in the Sales table*).

## Logical Scheme



## Objects

1. Auction

Table Auction contains all information for the auction date and location.

Relationship: One-to-Many (Auction <-> Auction\_Item) -> One auction can have multiple items.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Auction | auctionID | Public key, PK | INT |
| dateTime | Date and time for the auction | DATETIME, NOT NULL |
|  | location | Location of the auction | VARCHAR(255), NOT NULL |

1. Auction Item

Table Auction items contain item identification and auction identification.

Relationship: Many-to-Many (Auction\_Item <-> Item) -> One item can be in multiple auctions, and one auction can have multiple items.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Auction Item | auctionID | PK, FK | INT, Auction.auctionID, ON DELETE CASCADE |
| itemID | PK, FK | INT, Item.itemID, ON DELETE CASCADE |

1. Auction History

Table Auction History contains all information about date and time of auction and sold status.

Relationship: One-to-Many (Auction\_History <-> Auction, Auction\_History <-> Item)

-> One item can have multiple historical records.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Auction History | historyID | Public key, PK | INT |
| itemID | FK | INT, Item.itemID, ON DELETE CASCADE |
|  | auctionID | FK | INT, Auction.auctionID, ON DELETE CASCADE |
|  | previousPrice | Previous price of an item | DECIMAL, NOT NULL, CHECK (previousPrice > 0) |
|  | soldStatus | Status of an item, can be true or false | BOOLEAN, NOT NULL |
|  | historyDate | Date and time of arrival | DATETIME, NOT NULL |

1. Auction Employee

Table Auction Employee contain all information about employees who work in the auction, every auction they work is logged in database.

Relationship: One-to-Many (Auction\_Employee <-> Auction) -> Multiple employees can work on one auction.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Auction Employee | employeeID | Public key, PK | INT |
| name | Name of the employee | VARCHAR (100), NOT NULL |
|  | auctionID | FK | INT, Auction.auctionID, ON DELETE CASCADE |
|  | role | Role of the employee, can be “Manager”, “Auctioneer”, “Marketing”, “Legal Advisor”, “Valuer” | ENUM ('Manager', 'Auctioneer', 'Marketing', 'Legal Advisor', 'Valuer'), NOT NULL |
|  | contactInfo | Contact information about employees | VARCHAR (255), NOT NULL |

1. Auction Log

Table Auction Log contains all information about item, auction, buyer, and all the logs and details.

Relationship: One-to-Many (Auction\_Log<-> Auction, Item, Buyer) -> One log entry belongs to a single auction, but an auction can have multiple logs.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Auction Log | logID | Public key, PK | INT |
| itemID | FK | INT, Item.itemID, ON DELETE CASCADE |
|  | auctionID | FK | INT, Auction.auctionID, ON DELETE CASCADE |
|  | buyerID | FK | INT, Buyer.buyerID, ON DELETE CASCADE |
|  | auctionType | Type of auction | VARCHAR (50), NOT NULL |
|  | logTime | Date and time of log | DATETIME, NOT NULL |
|  | details | Description | TEXT |

1. Category

Table Category contains descriptions of the item category.

Relationship: One-to-Many (Category <-> Item) -> Each item belongs to one category, but one category can have multiple items.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Category | categoryID | PK | INT |
| categoryName | Item category name | VARCHAR (255), NOT NULL, UNIQUE |
|  | description | Category description | TEXT |

1. Item

Table Item contain all information about item, their seller, category and price.

Relationship: One-to-Many (Item <-> Seller) -> Multiple items can have a single seller.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Item | itemID | Public key, PK | INT |
| lotNumber | Number in the auction | VARCHAR (50), NOT NULL, UNIQUE |
|  | description | Description of the item | TEXT, NOT NULL |
|  | startingPrice | Starting price of the item | DECIMAL (10,2), NOT NULL, CHECK (startingPrice > 0) |
|  | sellerID | FK | INT, Seller.sellerID, ON DELETE CASCADE |
|  | categoryID | FK | INT, Category.categoryID, ON DELETE SET NULL |

1. Seller

Table Seller contain all contact info about seller.

Relationship: One-to-Many (Seller <-> Item) -> One seller can sell multiple items.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Seller | sellerID | PK | INT |
| name | Seller name information | VARCHAR (255), NOT NULL |
|  | sellerContactInfo | Contact of the seller | VARCHAR (255), NOT NULL |

1. Buyer

Table Buyer contain all contact info about buyer.

Relationship: One-to-Many (Buyer <-> Bid, Purchase) -> One buyer can place multiple bids and make multiple purchases.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Buyer | buyerID | PK | INT |
| name | Buyer name information | VARCHAR (255), NOT NULL |
|  | buyerContactInfo | Contact of the buyer | VARCHAR (255), NOT NULL |

1. Bid

Table Bid contains all the information about buyers and all the bid amount and time. Relationship: One-to-Many (Bid <-> Buyer, Auction, Item) -> One buyer can place multiple bids on different item.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Bid | bidID | Public key, PK | INT |
| auctionID | FK | INT, Auction.auctionID, ON DELETE CASCADE |
|  | itemID | FK | INT, Item.itemID, ON DELETE CASCADE |
|  | buyerID | FK | INT, Buyer.buyerID, ON DELETE CASCADE |
|  | bidAmount | Last price on the item | DECIMAL (10,2), NOT NULL, CHECK (bidAmount > 0) |
|  | bidTime | Time of the bidding | DATETIME, NOT NULL |

1. Purchase

Table Purchase contains all the information about prices and buyer and seller.

Relationship: One-to-One (Purchase <-> Auction, Item, Buyer) -> One item has one purchase record.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Purchase | purchaseID | Public key, PK | INT |
| auctionID | FK | INT, Auction.auctionID, ON DELETE CASCADE |
|  | itemID | FK | INT, Item.itemID, ON DELETE CASCADE |
|  | buyerID | FK | INT, Buyer.buyerID, ON DELETE CASCADE |
|  | finalPrice | Last price on the item | DECIMAL (10,2), NOT NULL, CHECK (finalPrice > 0) |
|  | purchaseDate | Date of the purchase | DATETIME, NOT NULL |

1. Payment

Table Payment contain all the information about payment history, method etc.

Relationship: One-to-One (Payment <-> Purchase) -> One purchase has one payment.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Payment | payment\_id | Public key, PK | INT |
| purchaseID | FK | INT, Purchase.purchaseID, ON DELETE CASCADE |
|  | paymentMethod | Method of the payment | VARCHAR (50), NOT NULL |
|  | paymenttatus | Status of the payment | VARCHAR (50), NOT NULL |
|  | amount | Payment amount | DECIMAL (10,2), NOT NULL, CHECK (amount > 0) |
|  | paymentDate | Date of the payment | DATETIME, NOT NULL |

1. Shipment

Table Shipment contains all the information about shipment addresses etc.

Relationship: One-to-One (Shipment <-> Purchase) -> One purchase has one shipment.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Shipment | shipmentID | Public key, PK | INT |
| purchaseID | FK | INT, Purchase.purchaseID, ON DELETE CASCADE |
|  | shipmentAddress | Address on the shipment label | VARCHAR (255), NOT NULL |
|  | carrier | Who carries the shipment | VARCHAR (100), NOT NULL |
|  | trackingNumber | Number for the tracking of the package | VARCHAR (255), UNIQUE |
|  | status | Status of the shipment | VARCHAR (50), NOT NULL |
|  | estimatedDate | Estimated date of arrival | DATETIME, NOT NULL |