

|  |
| --- |
| Business Template  **An Auction House** |
| C:\Users\natal\Downloads\ChatGPT Image Oct 24, 2025, 04_41_43 PM.png |

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 artworks through organized auctions.

Sellers (individuals or companies) register their items to be auctioned, while buyers participate to purchase these items. After receiving items from sellers, the auction house evaluates them and decides at which upcoming auction it will be most profitable to present each one. If an item does not have an owner, the auction company lists it on its own behalf.

Before an auction, every item is assigned a lot number, which functions similarly to a product code. Different auctions may have the same lot numbers since numbering restarts per event. During the auction, buyers (registered participants) can place bids on any number of lots. A single participant may act as both a seller and a buyer in different transactions.

The company manages all auction events, including item listings, lot assignments, and sales tracking.

Currently, most records are managed manually or across separate spreadsheets, which complicates data consistency, reporting, and traceability.

## Problems. Current Situation

Before the implementation of a structured database system, the auction house faced several major problems:

* Scattered data storage: Information about sellers, buyers, lots, payments, and auctions was stored in separate spreadsheets or even on paper.Difficult to track item ownership and sales history.
* No data integration: It was difficult to trace a complete history of an item, from its seller to the final sale.
* Inconsistent lot statuses: Tracking the state of lots (pending, sold, shipped, cancelled) required manual updates.
* Risk of errors and fraud: Without centralized validation, participant registration and payment tracking were prone to inconsistencies.
* Limited analytics: Evaluating profitability, buyer activity, or auction efficiency required manual data compilation and calculations.

## the Benefits of implementing a database. Project Vision

The goal of this project is to design and implement a centralized, relational database that automates all key auction house processes — from item registration to sale completion.

Expected benefits:

* Data integrity: Primary and foreign key constraints ensure that all records remain consistent and accurate.
* Transparency: Every auction, payment, and transaction is recorded in a unified system.
* Efficiency and reporting: Quick generation of performance reports, including profit analysis, buyer activity, and item history.
* Security and traceability: Each transaction and payment can be audited through unique transaction IDs.
* Scalability: The database supports adding new payment methods, auction types, and categories without redesign.
* The system provides a foundation for digital transformation of the auction house, improving operational efficiency, data accuracy, and overall business transparency.

# Model description

## Definitions & Acronyms

|  |  |
| --- | --- |
| Term / Acronym | Definition |
| member | A participant in the auction system. Can be either a company or an individual person. |
| member\_company | Contains data about legal entities — registration number, trade name, tax code, etc. |
| member\_person | Contains data about individual participants — name, surname, date of birth, contact information. |
| auction | An event where lots are offered for sale. Includes date, time, location, and description. |
| lot | A specific item presented at an auction. Each lot has a unique number and starting price. |
| item | The actual object offered for sale. It normally belongs to a seller, but if no owner is registered, the auction company itself becomes the default seller. |
| item\_seller | A linking table defining which member owns which item, with share percentage and ownership role. It also supports cases where the seller is the auction company. |
| bid | A bid made by a buyer for a particular lot, including amount, date, and time. |
| order | A completed sale record. Contains information about the buyer, price, delivery status, and timestamps. |
| payment\_details | Information about payment transactions — method ('credit\_card', 'paypal', 'cash'), date, and status. |
| auction\_members | A table recording which members are registered to participate in specific auctions. |
| lot\_status | A reference table of allowed lot statuses ('pending', 'sold', 'shipped', 'cancelled'). |
| lot\_category | A reference table for classifying items by type ('painting', 'sculpture', 'antique', etc.). |
| PK (Primary Key) | A unique identifier for each record in a table. |
| PFK (Primary - Foreign Key) | A unique identifier for each record in a table which is also A reference to a primary key in another table, ensuring relational integrity. |
| FK (Foreign Key) | A reference to a primary key in another table, ensuring relational integrity. |
| NN (Not Null) | A field that must contain a value (cannot be left empty). |
| UC (Unique Constraint) | A restriction that prevents duplicate values in a column. |
| CHECK() | A constraint ensuring valid values (for example, percentage range 0–100 or minimum age 18). |
| INT | Integer numeric type used for identifiers, codes, or counts. Cannot contain decimals. |
| VARCHAR(n) | Variable-length string up to n characters. Used for textual attributes such as names, titles, or cities. |
| TEXT | Long text field used for detailed descriptions or comments. |
| DATE | Stores only a date (year-month-day). |
| TIMESTAMP | Stores both date and time (with optional time zone). Used for tracking creation or modification. |
| DECIMAL(p,s) | Fixed-point numeric type with precision (p) and scale (s). Ideal for monetary values. |
| BOOLEAN | Logical true/false field. Used to indicate binary states. |
| TINYINT | Small integer (0–255) used for flags or percentages. |
| 1 : 1 | One-to-One. Each record in Table A is related to only one record in Table B, and vice versa |
| 1 : N (1 : 0..) | One-to-Many. One record in Table A can be linked to multiple records in Table B, but each record in Table B belongs to only one record in Table A. |
| N : 1 (0.. : 1) | Many-to-One. The inverse of 1:N — many records in Table A are linked to one record in Table B. |
| M : N | Many-to-Many. Records in Table A can relate to multiple records in Table B, and vice versa. Usually implemented through a join table (associative entity). |

## 

## Logical Scheme

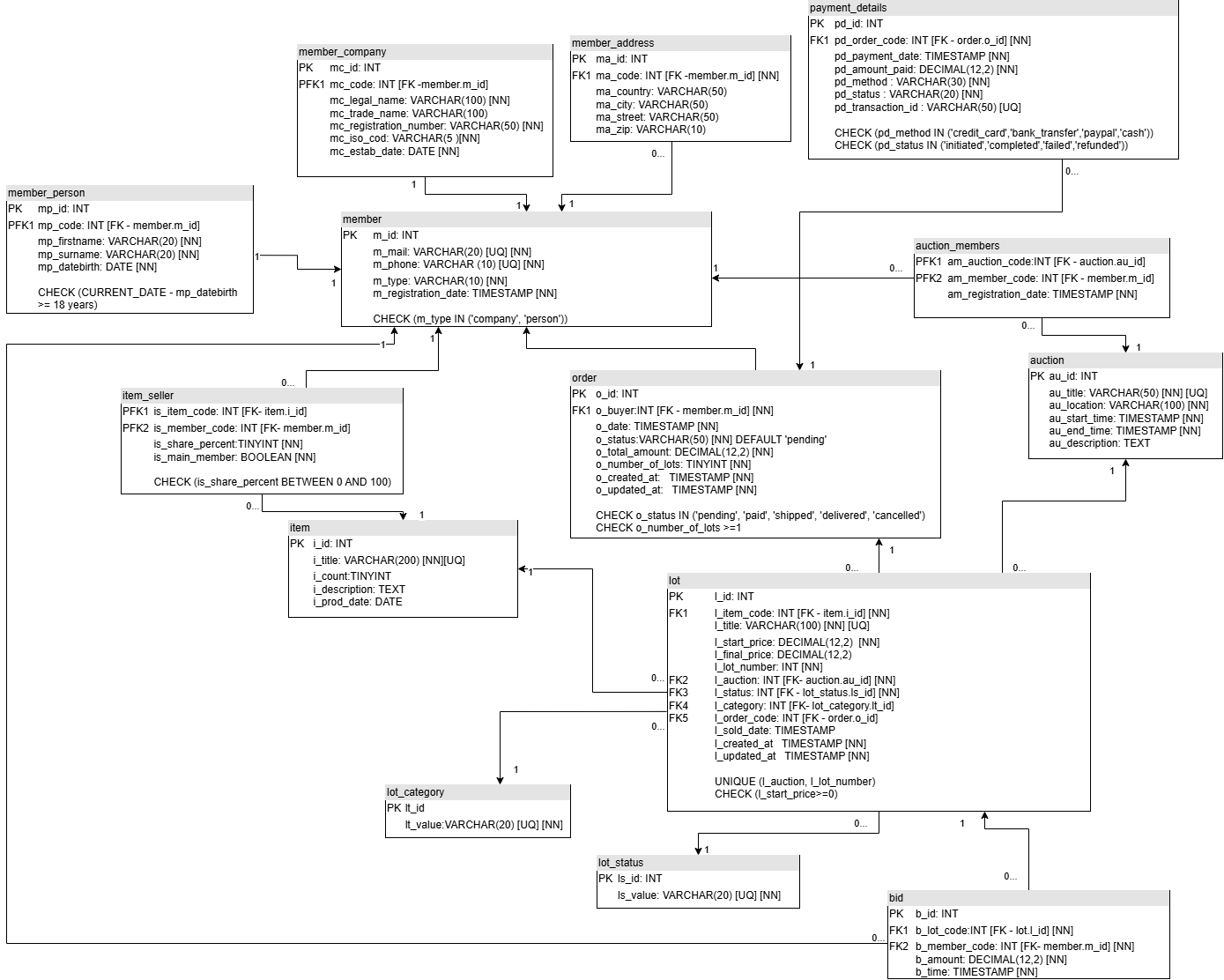


Figure 1. Datalogical Schema of the Auction House Database

## **Objects**

### Table "member"

Description

The member table is a universal registry that stores information about all participants of the auction house.

Each participant can be either an individual or a company.

Members register in the system using their email address and phone number, both of which are mandatory and must be unique to prevent duplication.

These contact details are required to ensure efficient communication with clients and to maintain reliable correspondence between the auction administration and participants.

A registered member can act as both a buyer and a seller within the auction platform.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| member | m\_id | Primary Key (unique member identifier). | INT |
| m\_mail | Email address; unique for each member, UQ, NN. | VARCHAR(20) |
| m\_phone | Contact phone number, unique for each member, NN, UQ | VARCHAR(10) |
| m\_type | Member type, restricted to 'company' or 'person'; NN | VARCHAR(10) |
| m\_registration\_date | Member registration timestamp; NN. | TIMESTAMP |

Comments on table relationships

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| member\_person | 1 : 1 | Each record in member can be linked to one member\_person record representing an individual participant. The relationship is established through member\_person.mp\_code → member.m\_id. |
| member\_company | 1 : 1 | Each record in member can be linked to one member\_company record representing a company participant. The relationship is established through member\_company.mc\_code → member.m\_id. |
| member\_address | 1 : N | A single member can have multiple addresses stored in the member\_address table (member\_address.ma\_code → member.m\_id). |
| item\_seller | M : N (through association) | Each member can be linked to multiple items through item\_seller and each item may have multiple co-sellers. This represents seller participation in listed items (item\_seller.is\_member\_code → member.m\_id). |
| bid | 1 : N | Each member can place many bids on various lots (bid.b\_member\_code → member.m\_id). |
| order | 1 : N | A member can have multiple orders as a buyer (order.o\_buyer → member.m\_id). |
| auction\_members | M : N (via join table) | The auction\_members table implements a many-to-many relationship between member and auction, allowing a member to register for multiple auctions and each auction to include many members (auction\_members.am\_member\_code → member.m\_id). |
| lot | (Indirect) | member relates to lot indirectly through bid, order, and item\_seller, reflecting participation as buyer, seller, or bidder. |
| payment\_details | (Indirect) | Linked indirectly through order (a member is the buyer associated with payments). |

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| m\_id | m\_mail | m\_phone | m\_type | m\_registration\_date |
| 1 | person@gmail.com | +380898782739 | person | 2023-03-15T13:00:00 |
| 2 | company@gmail.com | +380898782730 | company | 2023-03-15T13:00:00 |

### Table "member\_person"

Description

The member\_person table stores attributes of natural persons (individual members) linked to the member table. Each person participating in the auction must be at least 18 years old.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| member\_person | mp\_id | Primary Key (unique person identifier). | INT |
| mp\_code | FK → member.m\_id. | INT |
| mp\_firstname | First name; NN. | VARCHAR(20) |
| mp\_surname | Surname; NN. | VARCHAR(20) |
| mp\_datebirth | Date of birth; NN; CHECK age ≥ 18 years | DATE |

Comments on table relationships

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| member | 1 : 1 | Each member\_person belongs to exactly one parent member (member\_person.mp\_code → member.m\_id). |

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mp\_id | mp\_code | mp\_firstname | mp\_surname | mp\_datebirth |
| 101 | 1 | Alice | Brown | 1990-05-14 |

### Table "member\_company"

Description

The member\_company table stores data about companies participating in the auction, linked to the member table.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| member\_company | mc\_id | Primary Key (unique company identifier). | INT |
| mc\_code | FK → member.m\_id. | INT |
| mc\_legal\_name | Legal name; NN. | VARCHAR(100) |
| mc\_trade\_name | Trade name (optional). | VARCHAR(100) |
| mc\_registration\_number | Registration number; NN. | VARCHAR(50) |
| mc\_iso\_cod | Country ISO code; NN. | VARCHAR(5) |
| mc\_estab\_date | Establishment date; NN. | DATE |

Comments on table relationships

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| member | 1 : 1 | Each member\_company belongs to exactly one parent member (member\_company.mc\_code → member.m\_id). |

Example with data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| mc\_id | mc\_code | mc\_legal\_name | mc\_trade\_name | mc\_registration\_number | mc\_iso\_cod | mc\_estab\_date |
| 201 | 2 | Northwind LLC | Northwind | REG-2025-0001 | UA | 2005-03-20 |

### Table "member\_address"

Description

The `member\_address` table stores addresses for members.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| member\_address | ma\_id | Primary Key (unique address identifier). | INT |
| ma\_code | FK → member.m\_id; NN. | INT |
| ma\_country | Country. | VARCHAR(50) |
| ma\_city | City. | VARCHAR(50) |
| ma\_street | Street. | VARCHAR(50) |
| ma\_zip | Postal/ZIP code. | VARCHAR(10) |

Comments on table relationships

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| member | N : 1 | One member can have many addresses. (member\_address.ma\_code → member.m\_id). |

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ma\_id | ma\_code | ma\_country | ma\_city | ma\_street | ma\_zip |
| 301 | 1 | Ukraine | Kyiv | Khreshchatyk 1 | 01001 |

### Table "item"

Description

The item table stores objects offered for sale within the auction system.

If a seller wishes to list a set of items — for example, twelve identical antique chairs — they are registered under a common title, such as “Living Room Chairs of Dracula’s Mansion”, with the total quantity specified.

After it is determined how many chairs will be presented as lots, each unit receives its own specific title in the lot table, for instance: “Living Room Chairs of Dracula’s Mansion 1”, “Living Room Chairs of Dracula’s Mansion 2”, and so on.

The auction house reserves the right not to list items that do not meet the auction’s criteria and may also limit the number of items displayed, providing an explanation. Therefore, not every registered item is necessarily converted into a lot.

An item may have one or multiple owners who act as sellers; this relationship is represented through the intermediate table item\_seller.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| item | i\_id | Primary Key (unique item identifier). | INT |
| i\_title | Item title; UQ; NN. | VARCHAR(200) |
| i\_count | Quantity (optional). | TINYINT |
| i\_description | Description. | TEXT |
| i\_prod\_date | Production date. | DATE |

Comments on table relationships

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| item\_seller | 1 : N | One item can be linked to multiple sellers/owners via item\_seller. |
| lot | 1 : N | An item can appear in multiple lots (example with chairs in the description) (lot.l\_item\_code → item.i\_id). |

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i\_id | i\_title | i\_count | i\_description | i\_prod\_date |
| 401 | Antique Chairs | 1 | Living Room Chairs of Dracula’s Mansion | 1890-01-01 |

### Table "item\_seller"

Description

The `item\_seller` table links items to their owners.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| item\_seller | is\_item\_code | PFK; FK → item.i\_id. | INT |
| is\_member\_code | PFK; FK → member.m\_id. | INT |
| is\_share\_percent | Ownership share percent; NN; CHECK (0–100). | TINYINT |
| is\_main\_member | Primary owner flag; NN. | BOOLEAN |

Comments on table relationships - Many-to-many between items and members.

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| item | N : 1 | Each link references one item (item\_seller.is\_item\_code → item.i\_id). |
| member | N : 1 | Each link references one member (item\_seller.is\_member\_code → member.m\_id). |

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| is\_item\_code | is\_member\_code | is\_share\_percent | is\_main\_member |
| 401 | 2 | 100 | TRUE |

### Table "lot"

Description

The lot table represents an item in a specific auction.

Once an item is selected for participation in a particular auction, it is assigned a lot number.

This number, combined with the auction code, ensures the unique identification of the item within that auction.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| lot | l\_id | Primary Key (unique lot identifier). | INT |
| l\_item\_code | FK → item.i\_id; NN. | INT |
| l\_title | Lot title; UQ; NN. | VARCHAR(100) |
| l\_start\_price | Starting price; NN; CHECK (>= 0). | DECIMAL(12,2) |
| l\_final\_price | Final price (nullable). | DECIMAL(12,2) |
| l\_lot\_number | Sequential number; NN; UNIQUE with l\_auction. | INT |
| l\_auction | FK → auction.au\_id; NN. | INT |
| l\_status | FK → lot\_status.ls\_id; NN. | INT |
| l\_category | FK → lot\_category.lt\_id; NN. |  |
| l\_order\_code | FK → order.o\_id. | INT |
| l\_sold\_date | Sold timestamp (nullable). | TIMESTAMP |
| l\_created\_at | Creation timestamp; NN. | TIMESTAMP |
| l\_updated\_at | Update timestamp; NN. | TIMESTAMP |

Comments on table relationships - Each lot belongs to one auction; an auction has many lots.

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| item | 1 : N | Each lot references one item (lot.l\_item\_code → item.i\_id). |
| auction | N : 1 | Each lot belongs to exactly one auction (lot.l\_auction → auction.au\_id). |
| lot\_status | N : 1 | Each lot has exactly one status (lot.l\_status → lot\_status.ls\_id). |
| l\_category | N : 1 | Each lot has exactly one category (lot.l\_category → lot\_category.lt\_id). |
| order | N : 1 | A lot may create an order upon sale (lot.l\_order\_code → order.o\_id). |
| bid | 1 : N | A lot can receive many bids (bid.b\_lot\_code → lot.l\_id). |
| member | (Indirect) | Member participation as seller via item\_seller and as buyer via bid/order. |

Example with data

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| l\_id | l\_item\_code | l\_title | l\_start\_price | l\_final\_price | l\_lot\_number | l\_auction | l\_status | l\_category | l\_order\_code | l\_sold\_date | l\_created\_at | l\_updated\_at |
| 501 | 401 | Antique Vase Lot | 500.00 | 1250.00 | 12 | 601 | 1 | 2 | 701 | 2025-09-01 19:30:00 | 2025-08-15 10:00:00 | 2025-09-01 19:31:00 |

### Table "lot\_status"

Description

Reference table for lot statuses (for example: draft, active, sold, unsold, cancelled)

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| lot\_status | ls\_id | Primary Key (status identifier). | INT |
| ls\_value | Status value; UQ; NN. | VARCHAR(20) |

Comments on table relationships

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| lot | 1 : N | A single status value can be used by many lots (lot.l\_status → lot\_status.ls\_id). |

Example with data

|  |  |
| --- | --- |
| ls\_id | ls\_value |
| 1 | sold |

### Table "lot\_category"

Description

Reference table for item categories (for example: antiques, art)

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| lot\_category | lt\_id | Primary Key (category identifier). | INT |
| lt\_value | Category value; UQ; NN. | VARCHAR(20) |

Comments on table relationships

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| lot | 1 : N | A single category can be used by many lots (lot.l\_category → lot\_category.lt\_id). |

Example with data

|  |  |
| --- | --- |
| lt\_id | lt\_value |
| 1 | antiques |

### Table "bid"

Description

The `bid` table records bids from members on lots.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| bid | b\_id | Primary Key (bid identifier). | INT |
| b\_lot\_code | FK → lot.l\_id; NN. | INT |
| b\_member\_code | FK → member.m\_id; NN. | INT |
| b\_amount | Bid amount; NN. | DECIMAL(12,2) |
| b\_time | Timestamp; NN. | TIMESTAMP |

Comments on table relationships - Each bid references one lot and one member.

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| lot | N : 1 | Each bid targets one lot (bid.b\_lot\_code → lot.l\_id). |
| member | N : 1 | Each bid is placed by one member (bid.b\_member\_code → member.m\_id). |
| auction | (Indirect) | Bids are tied to an auction through their lot. |

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| b\_id | b\_lot\_code | b\_member\_code | b\_amount | b\_time |
| 801 | 501 | 3 | 1200.00 | 2025-09-01 19:29:30 |

### Table "auction"

Description

The `auction` table stores auction events.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| auction | au\_id | Primary Key (auction identifier). | INT |
| au\_title | Title; UQ; NN. | VARCHAR(50) |
| au\_location | Location; NN. | VARCHAR(100) |
| au\_start\_time | Start time; NN. | TIMESTAMP |
| au\_end\_time | End time; NN. | TIMESTAMP |
| au\_description | Description. | TEXT |

Comments on table relationships

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| lot | 1 : N | One auction lists many lots (lot.l\_auction → auction.au\_id). |
| auction\_members | 1 : N | One auction can have many registered members (auction\_members.am\_auction\_code → auction.au\_id). |

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| au\_id | au\_title | au\_location | au\_start\_time | au\_end\_time | au\_description |
| 601 | Autumn Fine Art Auction | Kyiv Hall A | 2025-09-01 18:00:00 | 2025-09-01 21:00:00 | Evening sale of fine art and antiques. |

### Table "auction\_members"

Description

Members registered for auctions.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| auction\_members | am\_auction\_code | PFK; FK → auction.au\_id. | INT |
| am\_member\_code | PFK; FK → member.m\_id. | INT |
| am\_registration\_date | Registration timestamp; NN. | TIMESTAMP |

Comments on table relationships - Many-to-many between auction and member.

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| auction | N : 1 | Each registration links to one auction (auction\_members.am\_auction\_code → auction.au\_id). |
| member | N : 1 | Each registration links to one member (auction\_members.am\_member\_code → member.m\_id). |

Example with data

|  |  |  |
| --- | --- | --- |
| am\_auction\_code | am\_member\_code | am\_registration\_date |
| 601 | 3 | 2025-08-28 09:15:00 |

### Table "order"

Description

The order table represents order data.

One member can have multiple separate orders, and each order must contain at least one lot in order to be processed.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| order | o\_id | Primary Key (order identifier). | INT |
| o\_buyer | FK → member.m\_id; NN. | INT |
| o\_date | Order date; NN | TIMESTAMP |
| o\_status | Status; NN; DEFAULT 'pending'; CHECK IN ('pending','paid','shipped','delivered','cancelled'). | VARCHAR(50) |
| o\_total\_amount | Total amount; NN. | DECIMAL(12,2) |
| o\_number\_of\_lots | Number of lots won; NN CHECK number >=1 | TINYINT |
| o\_created\_at | Creation timestamp; NN. | TIMESTAMP |
| o\_updated\_at | Update timestamp; NN. | TIMESTAMP |

Comments on table relationships

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| member | N : 1 | Each order belongs to one buyer member (order.o\_buyer → member.m\_id). |
| payment\_details | 1 : N | An order can have many payment records (payment\_details.pd\_order\_code → order.o\_id). |
| lot | 1 : N | A sold lot references the created order (lot.l\_order\_code → order.o\_id). |

Example with data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| o\_id | o\_buyer | o\_date | o\_status | o\_total\_amount | o\_number\_of\_lots | o\_created\_at | o\_updated\_at |
| 701 | 3 | 2025-09-01 19:31:05 | paid | 1250.00 | 3 | 2025-09-01 19:31:05 | 2025-09-01 20:00:00 |

### Table "payment\_details"

Description

Payment transactions associated with orders.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| payment\_details | pd\_id | Primary Key (payment identifier). | INT |
| pd\_order\_code | FK → order.o\_id; NN. | INT |
| pd\_payment\_date | Payment timestamp; NN. | TIMESTAMP |
| pd\_amount\_paid | Paid amount; NN. | DECIMAL(12,2) |
| pd\_method | Payment method; NN; CHECK IN ('credit\_card','bank\_transfer','paypal','cash'). | VARCHAR(30) |
| pd\_status | Payment status; NN; CHECK IN ('initiated','completed','failed','refunded'). | VARCHAR(20) |
| pd\_transaction\_id | External transaction id; UQ. | VARCHAR(50) |

Comments on table relationships - Each payment references one order; an order may have many payments.

|  |  |  |
| --- | --- | --- |
| Related Table | Relationship Type | Description |
| order | N : 1 | Each payment record references one order (payment\_details.pd\_order\_code → order.o\_id). |
| member | (Indirect) | The buyer is the member linked via order. |

Example with data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| pd\_id | pd\_order\_code | pd\_payment\_date | pd\_amount\_paid | pd\_method | pd\_status | pd\_transaction\_id |
| 901 | 701 | 2025-09-01 19:32:00 | 1250.00 | credit\_card | completed | TXN-9F3A-20250901 |