

|  |
| --- |
| 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 Auction House specializes in the sale of antiques and artwork through a well-organized auction system. Established in [year], the company has grown to become one of the premier auction houses in the region, known for its attention to detail, commitment to customer satisfaction, and ability to bring the finest art and antiques to the auction block.

Our professional staff has extensive knowledge of valuable items and experience in the auction process. They provide exceptional service to both sellers and buyers. The company has cultivated a network of relationships with art collectors, aficionados, and reputable sellers, allowing it to consistently provide high-quality items for auction.

With a focus on transparency and integrity, the Auction House is a trusted partner for those interested in selling or purchasing valuable antiques and artwork. Our auctions are meticulously planned and organized, ensuring a seamless experience for both sellers and buyers.

The Auction House's database plays a crucial role in the organization and management of our operations. It allows us to effectively catalog and track all items, auctions, sellers, buyers, and other relevant information. This centralized system helps us maintain the high level of organization and professionalism that our clients have come to expect.

As the company continues to grow and expand, we remain committed to our core values – providing exceptional service to our clients and maintaining the highest standards in the auction industry. The Auction House looks forward to many more years of success and innovation, providing excellent service to the world of art and antiques.

## Problems. Current Situation

The Auction House has been experiencing growing pains as the business expands. This growth has exposed several challenges and issues that need to be addressed to maintain smooth operations and continue delivering exceptional services to our clients. Some of the critical problems and the current situation include:

1. Inefficient Cataloging and Tracking: As the volume of items and auctions increases, the current manual approach to cataloging and tracking antiques and artwork is becoming increasingly time-consuming and error-prone.
2. Limited Visibility and Data Analysis: The lack of an integrated database system makes it difficult to gain insights into auction performance, sales trends, and buyer/seller preferences, restricting the company's ability to effectively strategize and make data-driven decisions.
3. Seller and Buyer Management: Manually tracking and managing the growing number of sellers and buyers is becoming increasingly challenging. The current approach lacks efficiency, leading to communication gaps and delays in addressing the needs of our clients.
4. Inadequate Inventory Management: Without a centralized database, managing the acquired items and ensuring their proper allocation to the appropriate auctions can be cumbersome and disorganized.
5. Fragmented Information: The current record-keeping system is disjointed, making it difficult for employees to easily access the information they need, leading to increased risk of errors and miscommunications between departments.

To overcome these challenges and maintain the company's reputation for excellence, implementing an efficient and organized database system is crucial. The new system should streamline cataloging and tracking processes, improve visibility and data analysis capabilities, optimize inventory management, and enhance overall communication and collaboration between departments. This will ultimately result in better auction experiences for both sellers and buyers and help to secure the Auction House's leading position in the industry.

## the Benefits of implementing a database. Project Vision

Implementing a well-organized and efficient database system is essential for modernizing the Auction House operations and addressing the existing challenges. Our vision for the project is to create a comprehensive and scalable database that streamlines essential business processes, improves overall efficiency, and enhances the auction experience for both sellers and buyers.

Here are the key benefits of implementing a database for the Auction House:

1. Streamlined Cataloging and Tracking: An integrated database system will automate and simplify the process of cataloging and tracking antiques and artwork, reducing the time and effort required while minimizing the risk of errors.
2. Centralized Information Management: A centralized database will ensure easy access to critical information for employees across all departments. This improves collaboration, communication, and decision-making and reduces the risk of errors arising from fragmented information.
3. Improved Inventory Management: The database system will enable better organization and allocation of acquired items to appropriate auctions, making inventory management more efficient and organized.
4. Enhanced Data Analysis and Reporting: With a cohesive database system, the Auction House can generate insightful reports and analyze sales trends, auction performance, and client preferences. This information drives strategic business planning and helps refine marketing efforts.
5. Better Client Management: A database-driven approach to managing sellers and buyers will allow the Auction House to efficiently monitor and address client needs, improving satisfaction levels while forming long-lasting relationships.
6. Scalability and Flexibility: A well-designed database system can grow and adapt to the evolving needs of the Auction House, ensuring its continued effectiveness even as the volume of items, auctions, and clients expands.

By realizing this project vision, the Auction House will be better equipped to handle its growing operations, maintain its reputation for excellence in the art and antiques market, and continue delivering outstanding service to its growing client base. The new database system will not only enable the company to overcome current challenges but also provide a robust foundation for future growth and success.

# Model description

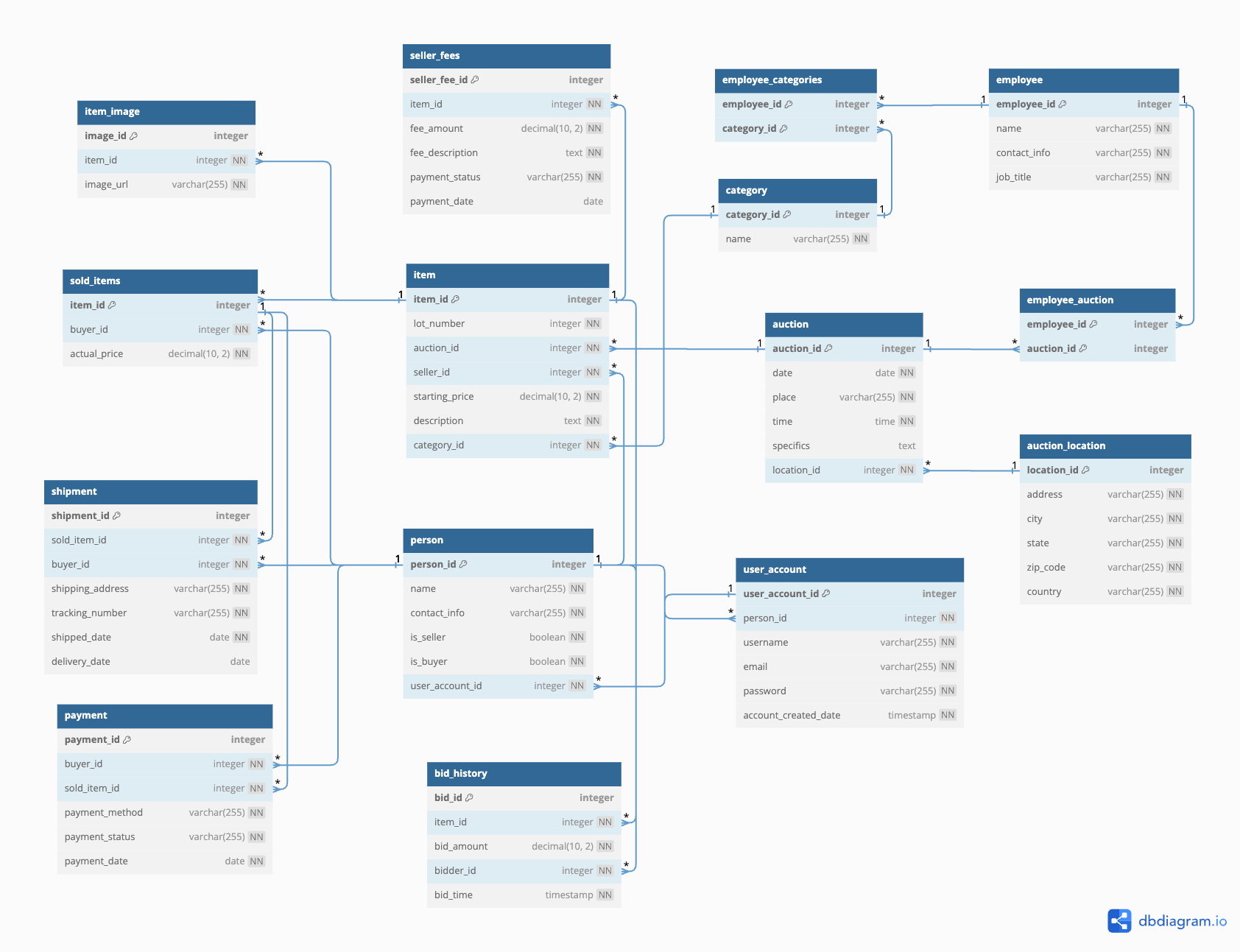
## Definitions & Acronyms

* ERD (Entity-Relationship Diagram): A visual representation of the database model, displaying its entities, attributes, and relationships between entities.
* PK (Primary Key): A unique identifier for a record in a table, used to establish relationships between tables and ensure data integrity.
* FK (Foreign Key): A column or set of columns in a table that refers to the primary key of another table, used to link records and establish relationships between tables.
* 3NF (Third Normal Form): A database normalization method that serves to eliminate redundancy and maintain data integrity, ensuring that non-prime attributes in a table are directly dependent on the primary key and do not have transitive dependencies.
* DDL (Data Definition Language): A subset of SQL (Structured Query Language) used for defining the structure of a database, including creating, altering, or deleting tables and other database objects.

Entities:

1. Auction: Represents an auction event held by the Auction House.
2. Auction Location: Represents the location where the auction is held.
3. Person: Represents an individual or firm acting as a seller, buyer, or both.
4. User Account: Represents the account information of a person associated with the Auction House platform.
5. Item: Represents antiques or artwork put up for auction.
6. Category: Represents the category of the item, such as antiques or artwork.
7. Sold Items: Represents the items sold during the auctions and their actual prices.
8. Bid History: Represents the bidding history for each item, including the bid amount, bidder, and time of the bid.
9. Employee: Represents an employee working for the Auction House.
10. Employee Auction: Represents a many-to-many relationship between employees and auctions, indicating which employees are responsible for managing specific auctions.
11. Employee Categories: Represents a many-to-many relationship between employees and categories, indicating the categories in which employees are experts.
12. Payment: Represents payment details made by a buyer for a sold item.
13. Shipment: Represents shipping information for a sold item from the Auction House to the buyer.
14. Item Image: Represents the images associated with items displayed at the auction.
15. Seller Fees: Represents the fees charged by the Auction House to the seller for services provided.

## Logical Scheme



## Objects

### Person

Table Description

The person table stores information about individuals or firms acting in the capacity of buyers, sellers, or both.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| person | person\_id | Unique identifier, PK | Integer |
| name | Full name of the person | Varchar |
| contact\_info | Contact information | Varchar |
| is\_seller | Whether the person is a seller | Boolean |
| is\_buyer | Whether the person is a buyer | Boolean |
| user\_account\_id | The associated user account, FK | Integer |

Comments on table relationships

The person table establishes relationships with multiple other tables like user\_account (one-to-one), item (one-to-many), sold\_items (one-to-many), bid\_history (one-to-many), payment (one-to-many), and shipment (one-to-many).

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| person\_id | name | contact\_info | is\_seller | is\_buyer | user\_account\_id |
| 1 | John Doe | [johndoe@email.com](mailto:johndoe@email.com) | True | True | 1 |
| 2 | Jane Smith | [janesmith@email.com](mailto:janesmith@email.com) | True | False | 2 |
| 3 | Alice Brown | [alice@email.com](mailto:alice@email.com) | False | True | 3 |

### Item

Table Description

The item table stores information pertaining to antiques and artwork up for auction.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| item | item\_id | Unique identifier, PK | Integer |
| lot\_number | The lot number for the item | Integer |
| auction\_id | The associated auction, FK | Integer |
| seller\_id | The associated seller, FK | Integer |
| starting\_price | The starting price of the item | Decimal |
| description | A description of the item | Text |
| category\_id | The item's category, FK | Integer |

Comments on table relationships

The item table establishes relationships with person (one-to-many), auction (one-to-many), and category (one-to-many). It also connects to sold\_items (one-to-one) and bid\_history (one-to-many).

Example with data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| item\_id | lot\_number | auction\_id | seller\_id | starting\_price | description | category\_id |
| 1 | 1001 | 1 | 1 | 500.00 | Antique Vase | 1 |
| 2 | 1002 | 1 | 2 | 2500.00 | Oil Painting (19th century) | 2 |
| 3 | 1001 | 2 | 1 | 1200.00 | Vintage Clock | 1 |

### Auction

Table Description

The auction table stores information about auction events held by the Auction House.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| auction | auction\_id | Unique identifier, PK | Integer |
| date | Date of the auction | Date |
| place | Venue of the auction | Varchar |
| time | Time of the auction | Time |
| specifics | Specific details about the auction | Text |
| location\_id | Associated auction location, FK | Integer |

Comments on table relationships

The auction table is connected to auction\_location (one-to-many), item (one-to-many), and employee\_auction (many-to-many).

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **auction\_id** | **date** | **place** | **time** | **specifics** | **location\_id** |
| 1 | 2023-06-01 | Gallery A | 18:00 | 19th-century paintings | 1 |
| 2 | 2023-06-15 | Gallery B | 19:00 | Vintage furniture and antiques | 2 |

### Auction location

Table Description

The auction\_location table stores information about locations where auctions take place.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| auction\_location | location\_id | Unique identifier, PK | Integer |
| address | Street address of the location | Varchar |
| city | City of the location | Varchar |
| state | State of the location | Varchar |
| zip\_code | Zip code of the location | Varchar |
| country | Country of the location | Varchar |

Comments on table relationships

The auction\_location table is connected to auction (one-to-many).

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **location\_id** | **address** | **city** | **state** | **zip\_code** | **country** |
| 1 | 123 Main St | New York | NY | 10001 | USA |
| 2 | 456 Market St | San Francisco | CA | 94105 | USA |

### Category

Table Description

The category table stores information about item categories (e.g., antiques or artwork).

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| category | category\_id | Unique identifier, PK | Integer |
| name | Name of the category | Varchar |

Comments on table relationships

The category table is connected to item (one-to-many) and employee\_categories (many-to-many).

Example with data

|  |  |
| --- | --- |
| **category\_id** | **name** |
| 1 | Antiques |
| 2 | Artwork |

### Sold items

Table Description

The sold\_items table stores information about items sold during the auctions and their actual prices.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| sold\_items | item\_id | Unique identifier, PK, FK | Integer |
| buyer\_id | The associated buyer, FK | Integer |
| actual\_price | Price paid for the item | Decimal |

Comments on table relationships

The sold\_items table is connected to person (many-to-one), item (one-to-one), payment (one-to-one), and shipment (one-to-one).

Example with data

|  |  |  |
| --- | --- | --- |
| **item\_id** | **buyer\_id** | **actual\_price** |
| 1 | 3 | 600.00 |
| 2 | 1 | 3200.00 |

### Bid history

Table Description

The bid\_history table stores the bidding history for each item, including the bid amount, bidder, and time of the bid.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| bid\_history | bid\_id | Unique identifier, PK | Integer |
| item\_id | The associated item, FK | Integer |
| bid\_amount | Amount of the bid | Decimal |
| bidder\_id | The associated bidder, FK | Integer |
| bid\_time | Time of the bid | Timestamp |

Comments on table relationships

The bid\_history table is connected to person (many-to-one) and item (many-to-one).

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bid\_id** | **item\_id** | **bid\_amount** | **bidder\_id** | **bid\_time** |
| 1 | 1 | 500.00 | 3 | 2023-06-01 18:05:00 |
| 2 | 1 | 600.00 | 3 | 2023-06-01 18:10:00 |
| 3 | 2 | 2500.00 | 1 | 2023-06-01 18:15:00 |
| 4 | 2 | 3200.00 | 1 | 2023-06-01 18:20:00 |

### Employee

Table Description

The employee table stores information about employees working for the Auction House.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| employee | employee\_id | Unique identifier, PK | Integer |
| name | Full name of the employee | Varchar |
| contact\_info | Contact information | Varchar |
| job\_title | Job title of the employee | Varchar |

Comments on table relationships

The employee table is connected to employee\_auction (many-to-many) and employee\_categories (many-to-many).

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| **employee\_id** | **name** | **contact\_info** | **job\_title** |
| 1 | Mark Smith | [mark@email.com](mailto:mark@email.com) | Auctioneer |
| 2 | Susan White | [susan@email.com](mailto:susan@email.com) | Art Specialist |

### Employee auction

Table Description

The employee\_auction table represents the many-to-many relationship between employees and auctions, indicating which employees are responsible for managing specific auctions.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| employee\_auction | employee\_id | Employee responsible for auction, FK, PK | Integer |
| auction\_id | The associated auction, FK, PK | Integer |

Comments on table relationships

The employee\_auction table connects employee and auction in a many-to-many relationship.

Example with data

|  |  |
| --- | --- |
| **employee\_id** | **auction\_id** |
| 1 | 1 |
| 2 | 1 |

### Employee categories

Table Description

The employee\_categories table represents the many-to-many relationship between employees and categories, indicating the areas of expertise for each employee.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| employee\_categories | employee\_id | Employee identifier, FK, PK | Integer |
| category\_id | Category of expertise, FK, PK | Integer |

Comments on table relationships

The employee\_categories table connects employee and category in a many-to-many relationship.

Example with data

|  |  |
| --- | --- |
| **employee\_id** | **category\_id** |
| 2 | 2 |
| 3 | 1 |

### Payment

Table Description

The payment table stores information about payments made by buyers for sold items.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| payment | payment\_id | Unique identifier, PK | Integer |
| buyer\_id | The associated buyer, FK | Integer |
| sold\_item\_id | The associated sold item, FK | Integer |
| payment\_method | Method used for payment | Varchar |
| payment\_status | Status of the payment | Varchar |
| payment\_date | Date of the payment | Date |

Comments on table relationships

The payment table is connected to person (many-to-one) and sold\_items (one-to-one).

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **payment\_id** | **buyer\_id** | **sold\_item\_id** | **payment\_method** | **payment\_status** | **payment\_date** |
| 1 | 3 | 1 | Credit Card | Paid | 2023-06-02 |
| 2 | 1 | 2 | Wire Transfer | Paid | 2023-06-03 |

### Shipment

Table Description

The shipment table stores information about shipping for sold items from the Auction House to the buyer.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| shipment | shipment\_id | Unique identifier, PK | Integer |
| sold\_item\_id | The associated sold item, FK | Integer |
| buyer\_id | The associated buyer, FK | Integer |
| shipping\_address | Address for shipping | Varchar |
| tracking\_number | Tracking number | Varchar |
| shipped\_date | Date shipped | Date |
| delivery\_date | Date delivered | Date |

Comments on table relationships

The shipment table is connected to person (many-to-one) and sold\_items (one-to-one).

Example with data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **shipment\_id** | **sold\_item\_id** | **buyer\_id** | **shipping\_address** | **tracking\_number** | **shipped\_date** | **delivery\_date** |
| 1 | 1 | 3 | 555 Elm St | TRK123 | 2023-06-03 | 2023-06-06 |
| 2 | 2 | 1 | 786 Oak St | TRK456 | 2023-06-04 | 2023-06-08 |

### Item image

Table Description

The item\_image table stores information about images associated with items displayed at the auction.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| item\_image | image\_id | Unique identifier, PK | Integer |
| item\_id | The associated item, FK | Integer |
| image\_url | URL of the item image | Varchar |

Comments on table relationships

The item\_image table is connected to item (many-to-one).

Example with data

|  |  |  |
| --- | --- | --- |
| **image\_id** | **item\_id** | **image\_url** |
| 1 | 1 | [www.example.com/image1.jpg](http://www.example.com/image1.jpg) |
| 2 | 1 | [www.example.com/image2.jpg](http://www.example.com/image2.jpg) |
| 3 | 2 | [www.example.com/image3.jpg](http://www.example.com/image3.jpg) |

### Seller fees

Table Description

The seller\_fees table stores information about the fees charged by the Auction House to the seller for services provided.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| seller\_fees | seller\_fee\_id | Unique identifier, PK | Integer |
| item\_id | The associated item, FK | Integer |
| fee\_amount | Amount of the fee | Decimal |
| fee\_description | Description of the fee | Text |
| payment\_status | Status of the fee payment | Varchar |
| payment\_date | Date of the fee payment | Date |

Comments on table relationships

The seller\_fees table is connected to item (many-to-one).

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **seller\_fee\_id** | **item\_id** | **fee\_amount** | **fee\_description** | **payment\_status** | **payment\_date** |
| 1 | 1 | 50.00 | Commission Fee | Paid | 2023-06-10 |
| 2 | 2 | 320.00 | Commission Fee | Paid | 2023-06-10 |
| 3 | 3 | 120.00 | Commission Fee | Unpaid |  |