

|  |
| --- |
| Business Template  **Subject areas** |
|  |

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 4](#_Toc62212634)

[2.1 Logical Scheme 4](#_Toc62212636)

[2.2 Objects 4](#_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

**Disparate Information Sources:** Before implementing a unified database, auction details, item information, seller and buyer data, and auction results were managed in separate systems or manually.

**Data Redundancy:** Without a centralized database, there was significant duplication in data entry.

**Tracking Challenges:** Tracking individual items across multiple auctions and managing the

many-to-many relationships between parties (as sellers and buyers) was complex.

**Reporting Issues:** Generating comprehensive reports regarding auction performance, item histories, and seller/buyer activity was time-consuming and prone to errors.

## the Benefits of implementing a database. Project Vision

Implementing 3rd form Normalizing will:

**Centralize Data Management**: Consolidate all auction-related data into a single repository.

**Reduce Data Redundancy:** Maintain consistent and up-to-date information with proper normalization.

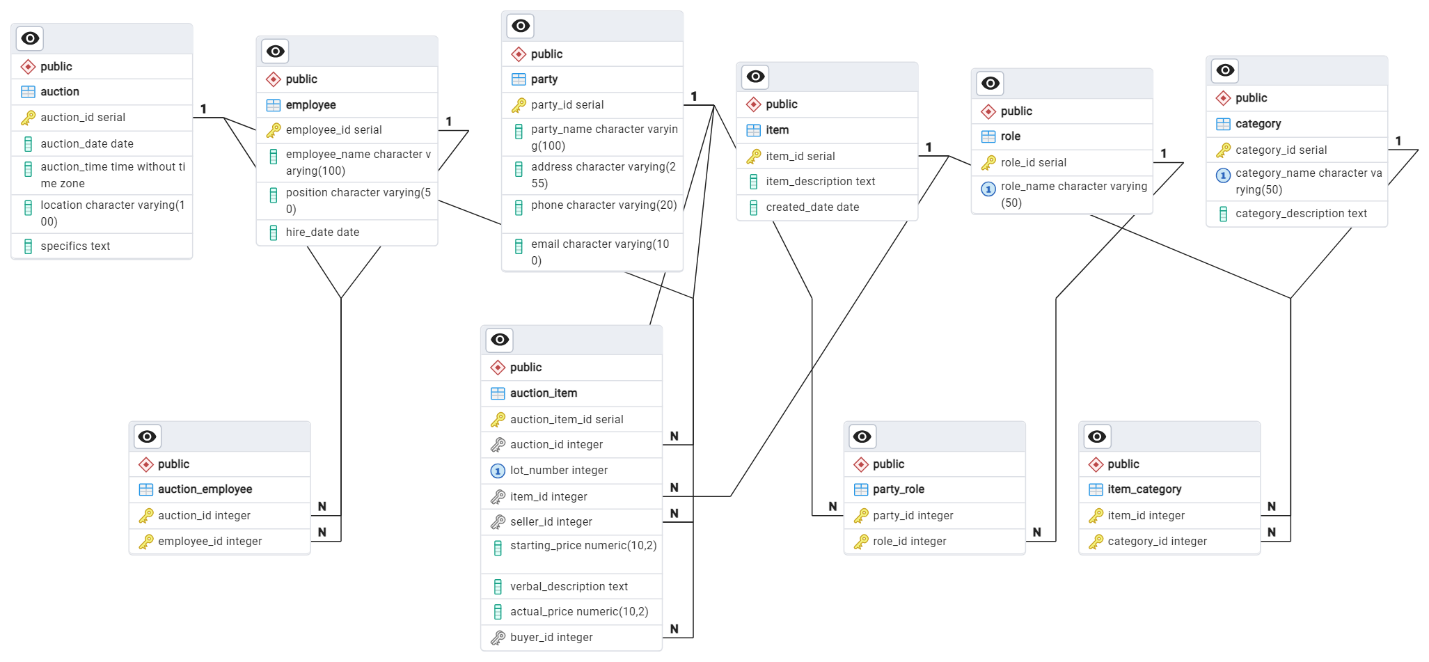
**Improve Data Integrity:** Enforce relationships through foreign keys and constraints to ensure accurate and reliable data.

**Enhance Reporting:** Simplify the generation of detailed reports and analytics to support business decision-making.

**Support Complex Relationships:** Effectively manage many-to-many relationships (e.g., a person can be both a seller and a buyer, and items may fall under multiple categories).

# Model description

## Logical Scheme



## Objects







**1. Party**

* **party\_id**
  + Data Type: SERIAL
  + Properties: Primary Key
  + Description: Unique identifier for each party.
* **party\_name**
  + Data Type: VARCHAR(100)
  + Properties: NOT NULL
  + Description: The name of the party.
* **address**
  + Data Type: VARCHAR(255)
  + Properties: (Optional)
  + Description: Mailing or physical address of the party.
* **phone**
  + Data Type: VARCHAR(20)
  + Properties: (Optional)
  + Description: Contact phone number.
* **email**
  + Data Type: VARCHAR(100)
  + Properties: (Optional)
  + Description: Email address.

**2. Role**

* **role\_id**
  + Data Type: SERIAL
  + Properties: Primary Key
  + Description: Unique identifier for each role.
* **role\_name**
  + Data Type: VARCHAR(50)
  + Properties: NOT NULL, UNIQUE
  + Description: Name of the role (e.g., Seller, Buyer).

**3. Party\_Role**

* **party\_id**
  + Data Type: INT
  + Properties: NOT NULL, Part of Composite Primary Key, Foreign Key referencing Party(party\_id)
  + Description: Identifier of the party.
* **role\_id**
  + Data Type: INT
  + Properties: NOT NULL, Part of Composite Primary Key, Foreign Key referencing Role(role\_id)
  + Description: Identifier of the role assigned to the party.

**4. Auction**

* **auction\_id**
  + Data Type: SERIAL
  + Properties: Primary Key
  + Description: Unique identifier for the auction.
* **auction\_date**
  + Data Type: DATE
  + Properties: NOT NULL
  + Description: The date on which the auction occurs.
* **auction\_time**
  + Data Type: TIME
  + Properties: NOT NULL
  + Description: The time when the auction takes place.
* **location**
  + Data Type: VARCHAR(100)
  + Properties: NOT NULL
  + Description: The venue or location of the auction.
* **specifics**
  + Data Type: TEXT
  + Properties: (Optional)
  + Description: Additional auction details (e.g., special notes).

**5. Item**

* **item\_id**
  + Data Type: SERIAL
  + Properties: Primary Key
  + Description: Unique identifier for the item.
* **item\_description**
  + Data Type: TEXT
  + Properties: NOT NULL
  + Description: Description of the item.
* **created\_date**
  + Data Type: DATE
  + Properties: NOT NULL, DEFAULT CURRENT\_DATE
  + Description: Date the item record was created.

**6. Auction\_Item**

* **auction\_item\_id**
  + Data Type: SERIAL
  + Properties: Primary Key
  + Description: Unique identifier for the auction-item record.
* **auction\_id**
  + Data Type: INT
  + Properties: NOT NULL, Foreign Key referencing Auction(auction\_id)
  + Description: Identifier for the auction in which the item is included.
* **lot\_number**
  + Data Type: INT
  + Properties: NOT NULL, Unique constraint per auction (i.e., unique together with auction\_id)
  + Description: Lot number assigned to the item within the auction.
* **item\_id**
  + Data Type: INT
  + Properties: NOT NULL, Foreign Key referencing Item(item\_id)
  + Description: Identifier for the item being auctioned.
* **seller\_id**
  + Data Type: INT
  + Properties: NOT NULL, Foreign Key referencing Party(party\_id)
  + Description: Identifier for the seller (must be a party with a seller role).
* **starting\_price**
  + Data Type: NUMERIC(10,2)
  + Properties: NOT NULL
  + Description: The starting price of the item.
* **verbal\_description**
  + Data Type: TEXT
  + Properties: (Optional)
  + Description: A brief description of the item provided at the auction.
* **actual\_price**
  + Data Type: NUMERIC(10,2)
  + Properties: (Optional)
  + Description: The final price paid by the buyer after the auction.
* **buyer\_id**
  + Data Type: INT
  + Properties: (Optional), Foreign Key referencing Party(party\_id)
  + Description: Identifier for the buyer (if the item is sold).

**7. Employee**

* **employee\_id**
  + Data Type: SERIAL
  + Properties: Primary Key
  + Description: Unique identifier for each employee.
* **employee\_name**
  + Data Type: VARCHAR(100)
  + Properties: NOT NULL
  + Description: The name of the employee.
* **position**
  + Data Type: VARCHAR(50)
  + Properties: (Optional)
  + Description: The employee’s position or job title.
* **hire\_date**
  + Data Type: DATE
  + Properties: (Optional)
  + Description: The date when the employee was hired.

**8. Auction\_Employee**

* **auction\_id**
  + Data Type: INT
  + Properties: NOT NULL, Part of Composite Primary Key, Foreign Key referencing Auction(auction\_id)
  + Description: Identifier for the auction.
* **employee\_id**
  + Data Type: INT
  + Properties: NOT NULL, Part of Composite Primary Key, Foreign Key referencing Employee(employee\_id)
  + Description: Identifier for the employee assigned to the auction.

**9. Category**

* **category\_id**
  + Data Type: SERIAL
  + Properties: Primary Key
  + Description: Unique identifier for each category.
* **category\_name**
  + Data Type: VARCHAR(50)
  + Properties: NOT NULL, UNIQUE
  + Description: Name of the category (e.g., Antique, Artwork).
* **category\_description**
  + Data Type: TEXT
  + Properties: (Optional)
  + Description: Description of the category.

**10. Item\_Category**

* **item\_id**
  + Data Type: INT
  + Properties: NOT NULL, Part of Composite Primary Key, Foreign Key referencing Item(item\_id)
  + Description: Identifier for the item.
* **category\_id**
  + Data Type: INT
  + Properties: NOT NULL, Part of Composite Primary Key, Foreign Key referencing Category(category\_id)
  + Description: Identifier for the category.