ART GALLERY MANAGEMENT SYSTEM

solving inventory and sales challenges

ARTIST MANAGEMENT

- Store artist profiles, biographies, and contact details
- track artist contributions and exhibition history

ARTWORK INVENTORY

- Mantain records of artwork titles, descriptions,,pricing and availability
- Track which pieces are on display, sold, or in storage

System Design

SALES MANAGEMENT

- Log customer purchases, transaction details, and payment history
- Generate receipts and track revenue trends

PL/SQL IMPLEMANTATION

- Stored Procedures: Automate tracking of sold and available artwork
- Triggers: Notify administrators when an artwork is sold or restocked
- Functions::Calculate total revenue, artiist contributions, and popular

Expected Benefits

EFFICIENCY

- Reduce manual work by automating inventory and sale tracking
- simplifies artist and customer management

ACCURARCY

- Ensures up-to-date records for artwork availability and transactions
- Reduces human errors in pricing and inventory updates.

AUTOMATION

- Sends notifications for upcoming exhibitions and low stock levels.
- Automates report generation on sales and popular artworks

SCALABILITY AND SECURITY

- Supports multiple galleries and large artwork collections
- Ensures secure access and role-based permissions for staff members

Phase II – Business Process Modeling

1. Swimlane Diagram: Art Gallery Sales & Exhibition Process

Here's a textual layout of the swimlane diagram you can recreate in **Lucidchart** or **draw.io**:

Actors (Swimlanes)

- Customer
- Gallery Admin
- System (Database)
- Curator

Process Flow:

Customer:

Starts → Browses artworks→ Selects artwork → Sends purchase request

Gallery Admin:

- Receives purchase request
- Checks availability in system
- Confirms payment
- Updates inventory and triggers sales record entry

System (Database)

- Validates request → Checks artwork status
- Records transaction
- Updates artwork as "Sold"
- Sends confirmation to admin and customer

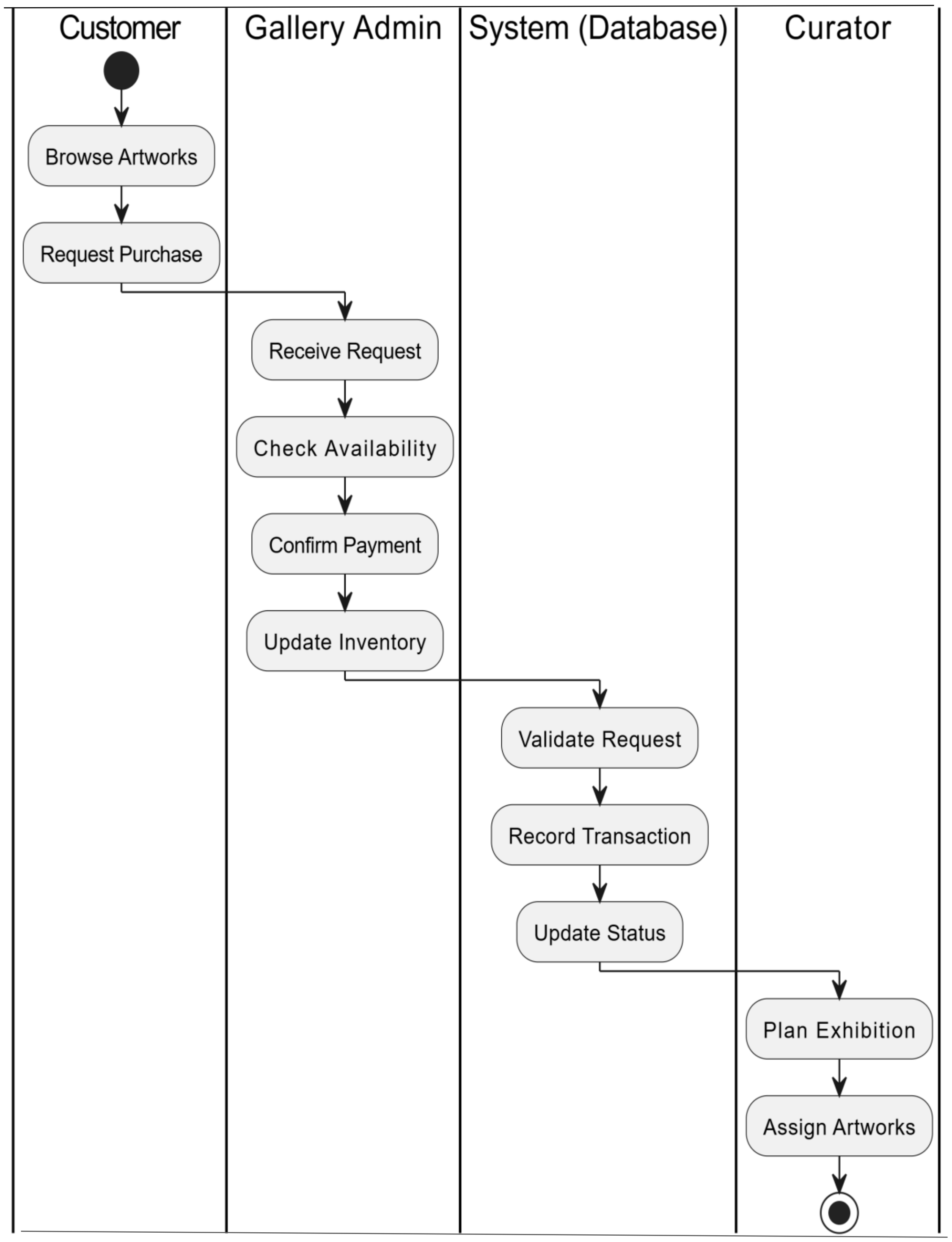
Curator

- Schedules upcoming exhibition
- Assigns available artworks
- Coordinates with admin to avoid overlaps

End Point

- Confirmation emails are sent
- Dashboard updates for sales & exhibition metrics

I can now **replicate this visually** using **draw.io** with UML/BPMN symbols



2. One-Page Explanation

Business Process Modeling: Art Gallery Management System

This business process models how the **Art Gallery Management System** handles artwork browsing, purchase requests, and exhibition scheduling—critical components of a gallery's operation. The process ensures smooth interaction among **customers**,

administrators, **curators**, and the **database system**, aligned with **Management Information Systems (MIS)** principles.

Main Components and Interactions

- **Customer** initiates the process by browsing available artworks and making purchase requests.
- **Gallery Administrator** verifies requests, confirms payments, and coordinates with the system to finalize sales.
- The **System (Oracle Database)** validates artwork status, updates records, and sends transaction logs.
- The **Curator** uses inventory data to plan future exhibitions and avoid scheduling conflicts.

MIS Support

- The process supports MIS by **automating decision-making**, **ensuring data accuracy**, and **improving workflow efficiency**.
- Real-time data from the database helps in **tracking popular artworks**, **managing inventory**, and **enhancing customer engagement**.

Organizational Efficiency

- Automating key tasks reduces manual errors, enables better resource planning, and improves the experience for both staff and customers.
- By integrating all roles into one unified process, the system provides timely information for better managerial decisions.

Perfect! Let’s proceed with **Phase III – Logical Model Design** for your **Art Gallery Management System**.

Phase III Deliverables

1. **Entity-Relationship (ER) Diagram**
2. **Attributes, Data Types, Keys**
3. **Relationships & Constraints**
4. **Normalization (up to 3NF)**
5. **Real-World Data Scenarios Support**

1. Main Entities and Attributes

Entity	Attributes
Artist	artist_id (PK), full_name, bio, birth_year, contact_email, specialization

Artwork	artwork_id (PK), title, artist_id (FK), category, price, creation_year, status
Exhibition	exhibition_id (PK), title, start_date, end_date, location, curator_id (FK)
Customer	customer_id (PK), full_name, email, phone, address
Sale	sale_id (PK), customer_id (FK), artwork_id (FK), sale_date, final_price
Curator	curator_id (PK), full_name, email, experience_years
Exhibition_Artwork	exhibition_id (FK), artwork_id (FK), *(Composite PK)*

2. Relationships and Cardinality

- One **Artist** → Many **Artworks** (1:N)
- One **Customer** → Many **Sales** (1:N)
- One **Artwork** → One or Zero **Sales** (1:0..1)
- One **Curator** → Many **Exhibitions** (1:N)
- **Exhibitions** ⇄ **Artworks** = Many-to-Many

3. Constraints

- **PK/ FK**: Defined as shown in entity list.
- **NOT NULL**: On all mandatory fields (like names, foreign keys).
- **CHECK**
    - price > 0
    - status IN (Available, Sold)
- **UNIQUE**: email for customers, curators, artists
- **DEFAULT**: status = 'Available' in Artwork

4. Normalization (Up to 3NF)

- No repeating groups (1NF)
- Full functional dependency (2NF)
- No transitive dependencies (3NF)

Each table stores **only related data** and avoids **duplication**:

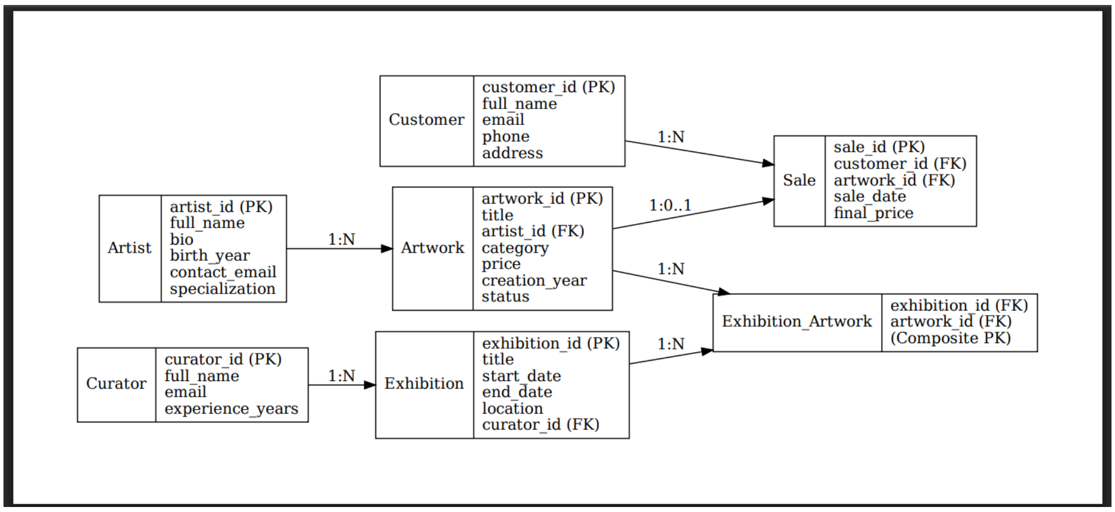
Customer and Artist details are **separated**

Sales and Exhibitions are **separate transaction entities**

Many-to-Many is resolved using a **junction table**

5. Supports Real-World Scenario

- Browsing and buying artworks
- Recording multiple exhibitions
- Tracking which artworks were featured in which events
- Preventing double sale of sold artworks
- Analytics: e.g., total sales per artist, popular exhibitions



Customer
customer_id (PK)
full_name
email
phone
address
1:N
Sale
sale_id (PK)
customer_id (FK)
artwork_id (FK)
sale_date
final_price
Artist
artist_id (PK)
full_name
bio
birth_year
contact_email
specialization
1:N
Artwork
artwork_id (PK)
title
artist_id (FK)
category
price
creation_year
status
1:0..1
1:N
Exhibition_Artwork
exhibition_id (FK)
artwork_id (FK)
(Composite PK)
Curator
curator_id (PK)
full_name
email
experience_years
1:N
Exhibition
exhibition_id (PK)
title
start_date
end_date
location
curator_id (FK)
1:N