

DBMS CASE STUDY

ART GALLERY MANAGEMENT SYSTEM

Presented By:

Himangi Aneja 21525

Priyam Gupta 21546

Yana Aggarwal 21569

CONTENTS TO BE DISCUSSED

- O1** Specification Sheet
- O2** ER
- O3** EER
- O4** Schema
- O5** SQL and Relational Queries
- O6** Normalisation

SPECIFICATION SHEET

ArtBase, which builds a product for art galleries. The core of this product is a database with a schema that captures all the information that galleries need to maintain.

- The gallery keeps the information about artists, their names (which are unique), birthplaces, age, and style of art.**
- For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, lithograph, sculpture, photograph), quantity (assuming each artwork can have multiple copies like limited edition sets which produce the same artwork in particular quantities) and its price must be stored. [Every artwork has to be made by an artist.]**
- Pieces of artwork are also classified into groups of various kinds, for example, portraits, still lifes, works by Picasso, or works of the 19th century; a given piece may belong to more than one group. Each group is identified by a name (like those above) that describes the group.**
- Finally, gallery keep information about customers. For each customer, gallery keep their unique name, address, total amount of rupees they have spent in the gallery, and the artists and groups of art that each customer tends to like. [Customer is only signified when he/she buys an artwork]**

SPECIFICATION SHEET

- **ArtBase Gallery has repository of information of all its exhibitions(each exhibition has its unique exhibition ID).It includes the time, date, place, and the artwork being displayed (artwork title). Each exhibition has its corresponding sale(s).**
- **The gallery keeps records of sales done. Each successful sale has its sales ID, the artwork sold(artwork title), the customer who bought it, its date,the quantity and the sale price.(exhibitions generate the major part of sales) (each artwork sold has a unique sale id)(each artwork has multiple copies available hence quantity is mentioned)(every sale has an artwork sold)**
- **The gallery has to ship the artwork for the various exhibitions. Each shipment has shipment_id, the exhibition to ship to (exhibition_id), the date, the quantity and the cost of shipment.(Every exhibition will have a shipment) (SHIPMENT(s) are only done for exhibitions and identified by unique shipment id)**

SPECIFICATION SHEET

For EER:

Artist can be of ONLY two types contract based and in-house as explained below:

- **Contract Artists:** Contract artists are hired to work on a specific project or for a specific period of time. Their contract amount details and contract start and end details are specifically mentioned.
- **In-house Artists:** In-house artists are employed by a company or organization and work exclusively for them. Their monthly salary is explicitly defined.

Sales are ONLY divided into the 3 following categories:

Sales would be through gallery walk in purchases, through exhibitions. However, our gallery offers a virtual tour as well and hence provides the opportunity to the customers to make online purchases.

ENTITY - RELATIONSHIP DIAGRAM

ENHANCED ENTITY - RELATIONSHIP DIAGRAM

SCHEMA

NORMALISATION

QUERIES

QUERIES

1. How many pieces of artwork have been sold during each exhibition, and what is the total revenue generated by each exhibition?

SQL QUERY:

```
SELECT EXHIBITION_ID, SUM(SALE_QUANTITY*SALE_PRICE), SUM(SALE_QUANTITY)
FROM EXHIBITION_SALES JOIN SALES
ON SALES.SALES_ID = EXHIBITION_SALES.SALES_ID
GROUP BY EXHIBITION_ID;
```

RELATIONAL QUERY:

$T1 \leftarrow \text{SALES} \bowtie_{\text{SALES.SALES_ID}=\text{EXHIBITION_SALES.SALES_ID}} \text{EXHIBITION_SALES}$

$T2 \leftarrow \text{EXHIBITION_ID} \bowtie_{\text{SUM(SALE_QUANTITY*SALE_PRICE), SUM(SALE_QUANTITY)}} T1$

$T3 \leftarrow \pi_{\text{EXHIBITION_ID, SUM(SALE_QUANTITY*SALE_PRICE), SUM(SALE_QUANTITY)}} T2$

QUERIES

2. Query to find the number of artworks in the gallery that have not been sold yet.

SQL QUERY:

```
SELECT COUNT(*)  
FROM ARTWORK  
WHERE TITLE NOT IN(  
    SELECT TITLE  
    FROM SALES  
);
```

RELATIONAL QUERY:

$$T1 \leftarrow \pi_{\text{TITLE}}(\text{SALES})$$
$$T2 \leftarrow \sigma_{\text{TITLE NOT IN } T1}(\text{ARTWORK})$$
$$T3 \leftarrow \pi_{\text{COUNT}(*)} T2$$

QUERIES

3. Find the titles and types of all artwork in the database, along with the names of the artists who created them, and the names of the customers who have bought them.

SQL QUERY:

```
SELECT Artwork.Title, Artwork.Type, Artwork.Name, Customer.customer_name  
FROM ((Artwork  
    JOIN Sales ON Artwork.Title = Sales.Title)  
    JOIN Customer ON sales.Customer_Name=Customer.Customer_Name);
```

RELATIONAL QUERY:

```
p ← artwork ⋈(artwork.title = sales.title) sales  
r ← p ⋈(sales.customer_name = customer.customer_name) customer  
s ← π(artwork.title, artwork.type, artwork.name, customer.customer_name) r;
```

QUERIES

4. Query to find the number of shipments associated with each exhibition.

SQL QUERY:

```
SELECT EXHIBITION_ID, COUNT(*)  
FROM EXHIBITION, SHIPMENT  
WHERE EXHIBITION.EXHIBITION_ID= SHIPMENT.EXHIBITION_ID  
GROUP BY EXHIBITION_ID;
```

RELATIONAL QUERY:

```
T1 ← EXHIBITION ⋈EXHIBITION.EXHIBITION_ID = SHIPMENT.EXHIBITION_ID SHIPMENT  
T2 ← EXHIBITION.EXHIBITION_ID ⋈COUNT(*) T1  
T3 ← πEXHIBITION.EXHIBITION_ID, COUNT(*) T2
```

QUERIES

5. Retrieve the names of all artists who have created more than 10 artworks.

SQL QUERY:

```
SELECT artist.name  
FROM artist JOIN artwork ON artist.name = artwork.name  
GROUP BY artist.name  
HAVING COUNT(*) > 10;
```

RELATIONAL QUERY:

```
T1 ← ARTIST ⋈artist.name = artwork.name ARTWORK  
T2 ← ARTIST.NAME ⋈COUNT(*) T1  
T3 ← σCOUNT(*)>10 T2
```