SRS Documentation

Artbase Management System

Prepared by,

DEBOPRIYO GHOSH	Roll: 35
SAIKAT JANA	Roll: 50
MOUSUMI MONDAL	Roll: 40
SARADINDU RANA	Roll: 28

B.Tech., Sem - V, Group- 1

Dated: 07/04/2021

Contents

Draw the ER diagram; Design the relational schema with minimum redundancy.

Your old love is still there, however, so you set up a database company, ArtBase that 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. Galleries keep 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), and its price must be stored. Pieces of artwork are also classified into groups of various kinds, for example, portraits, still life, 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, galleries keep information about customers. For each customer, galleries keep their unique name, address, total amount of dollars they have spent in the gallery (very important!), and the artists and groups of art that each customer tends to like. Each transaction (for the procurement of art-work) needs to be recorded. You can assume more requirement(s) to make your solution more realistic

1.Data Requirement

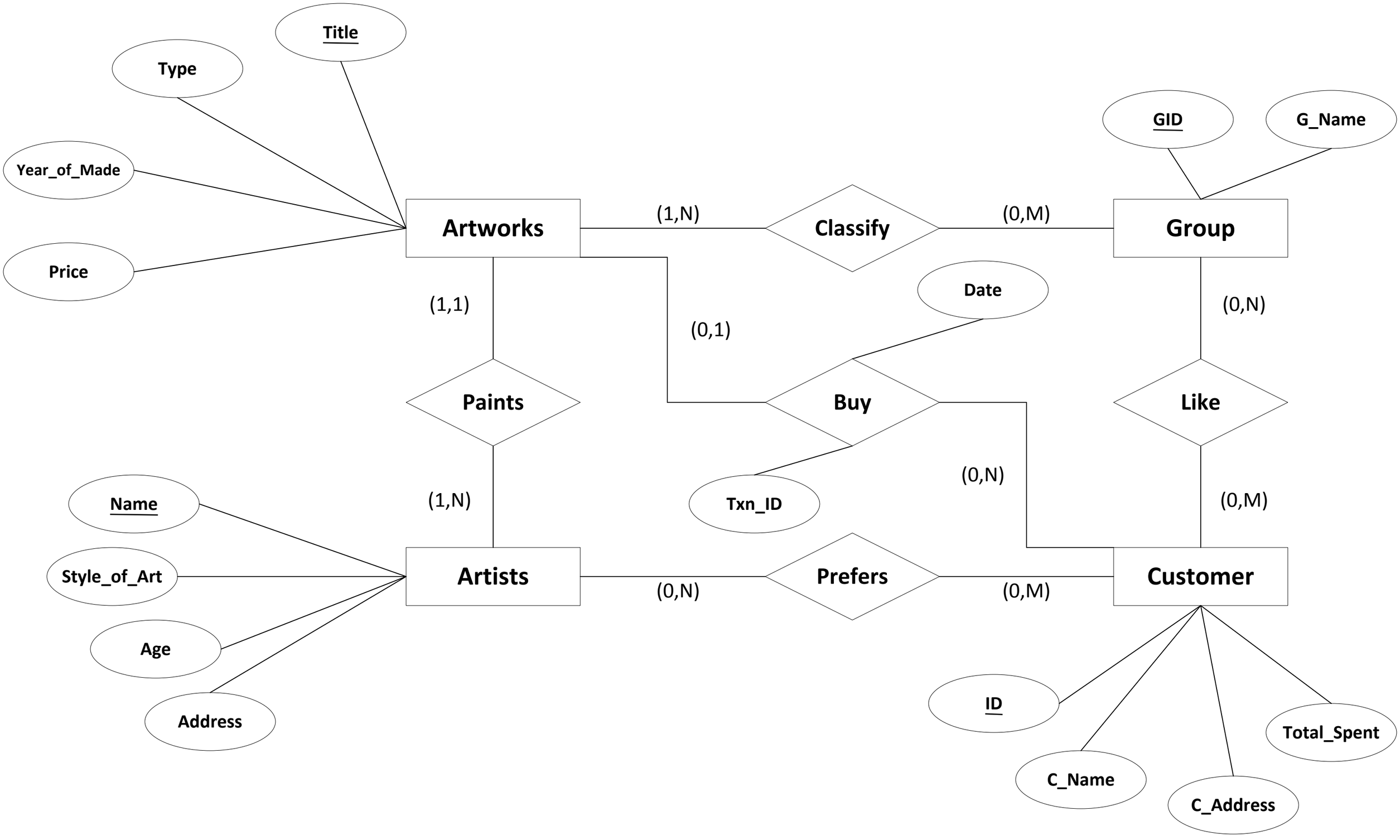
- **Artworks:** Every artwork has a unique title. Artwork has a specific type (e.g. painting, lithograph, sculpture, and photograph) and the year in which it is made. It has a fixed price to sell.
- **Artists:** Every artist has an unique name , age and address. Everyone has specific style of the art.
- **Customer:** Every customer has an unique id and their respective name , address . Total dollers spent in the gallery by the customer will be recorded.
- **Group:** Each group is identified by an unique id .It has name that describes the group.

2. Assumption

1) An artist can create multiple artworks. But the recorded artists has atleast one artwork.
2) A customer may or may not like or prefer a special kind of artists and special type of artwork group. Customer can prefer more than one group and artist. Customer may buy zero,one or more than one artwork. When a customer buy a artwork transaction id and date will be recorded.
3) A artwork is classified into one or more than one group.
4) An artist can be preferred by zero ,one or more than one customer.
5) A group can be liked by zero ,one or more than one customer.
6) An artwork can not be brought by more than one customer.

3. Entity Relationship Diagram

A. Diagram


Title
Type
Year_of_Made
Price
Artworks
(1,N)
Classify
(0,M)
GID
G_Name
Group
(1,1)
(0,1)
Date
(0,N)
Paints
Buy
Like
(0,N)
(1,N)
Txn_ID
(0,M)
Name
Style_of_Art
Age
Address
Artists
(0,N)
Prefers
(0,M)
Customer
ID
C_Name
C_Address
Total_Spent

B. Description

a) In this diagram the entities are Artworks, Artists, Customer and Group.
b) Artist paints artworks so they are connected by the relationship 'Paints'.
c) Artworks are classified in groups. They are connected by the relationship 'Classify'.
d) Customer buys artworks so they are connected by the relationship 'Buy'. 'Buy' has two attributes 'Date' and 'Txn_ID'.
e) A customer prefers some artists. So customer and artists are connected by the relationship 'Prefers'.
f) Customer also likes some artwork groups so, group and customer is connected by the relationship 'Like'.

C. Attributes

Artwork: {Title, Type, Year_of_Made, Price}
Artists: {Name , Style_of_Art, Age, Addrress}
Customer: {ID, C_Name, C_Address, Total_spent}
Group: {GID ,G_Name}
Buy: {Txn_ID , Date}

4.Relationships

a. Artists-Artworks (1:N) comprises.
b. Customer-Artists (N:M) comprises.
c. Customer-Artworks (1:N) comprises.
d. Customer-Group (N:M) comprises.
e. Artworks-group (M:N) holds.

5. Relational Schema

Artworks:

Artworks

- <Title>
  - Type of art : <Type>
  - Year of art was made : <Year_of_Made>
  - Price of artwork : <Price>
  - Artist name : <Name> (Forgien key from ‘Artists’ table)
  - Customer id : <ID> (Forgien key from ‘Customer’ table)
  - Date of sold : <Date>
  - Transaction id : <Txn_ID>

Artists:

Artists

- <Name>
  - Artist age : <Age>
  - Artist address : <Address>
  - Style of art : <Style_of_Art>

Customer:

Customer

- <ID>
  - Customer name : <C_Name>
  - Customer address : <C_Address>
  - Amount spent in gallery : <Total_Spent>

Group:

Group

- <GID>
  - Group name : <G_Name>

Prefers:

Prefers

- Artist name : <Name> (Forgien key from ‘Artists’ table)
- Customer id : <ID> (Forgien key from ‘Customer’ table)

Like:

Like

- Group id : <GID> (Forgien key from ‘Group’ table)
- Customer id : <ID> (Forgien key from ‘Customer’ table)

Classify:

Classify

- Group id :<GID> (Forgien key from ‘Group’ table)
- Artwork title : <Title> (Forgien key from ‘Artworks’ table)

6. Normalization of Relational Schema

Artwork {Title, Type, Year_of_Made, Price , Name, ID, Date, Txn_ID}

FD->

Title-> {Type, Year_of_Made, Price, Name, ID, Date, Txn_ID}
Txn_ID-> {Date, ID}

Candidate key -> Title

Here no attributes are multi valued .So, table is in **1NF**.

Here Title is the candidate key and Txn_ID is non prime attribute ,so there are no partial dependency present. So, table is in **2NF**.

Here Txn_ID is a non prime attribute and Date , ID is depend on that, so table is **not** in **3NF.**

Now the decompose table will be Artworks{Title, Type, Year_of_Made, Price, Name,Txn_ID} And another table will be Transaction{ Txn_ID, Date, ID}.

Now from the new table Artworks candidate key will remain same “Title” and Transaction table candidate key will be “Txn_ID”. Now both table Artworks and Transaction is in **3NF**.

In Artworks table Title is the candidate key so the table is in **BCNF,** and in Transaction table candidate key is Txn_ID so, table is also in **BCNF**.

Artists{Name, Age, Address, Style_of_Art}

FD->

Name->Age, Address, Style_of_Art

Candidate key -> Name

Here no attributes are multi valued ,so table is in **1NF**.

Here all attributes dependent on “Name” that is candidate key and no partial dependencies are present ,so table is in **2NF**.

Here all attributes dependent on “Name” and no transitive dependencies are present, so table is in **3NF**.

Here all attributes dependent on candidate key ,so table also in **BCNF**.

Customer {ID, C_Name, C_Address, Total_Spent}

FD->

ID-> C_Name, C_address, Total_Spent

Candidate Key-> ID

Here no attributes are multi valued ,so table is in **1NF**.

Here all attributes dependent on “ID” that is candidate key and no partial dependencies are present ,so table is in **2NF.**

Here all attributes dependent on “ID” and no transitive dependencies are present. So, table is in **3NF**.

Here all attributes dependent on candidate key, so table also in **BCNF.**

Group{GID, G_Name}

FD:

GID-> G_Name

Candidate Key-> GID

Here no attributes are multi valued ,so table is in **1NF**.

Here all attributes dependent on “GID” that is candidate key and no partial dependencies are present ,so table is in **2NF.**

Here all attributes dependent on “GID” and no transitive dependencies are present. So, table is in **3NF**.

Here all attributes dependent on candidate key, so table also in **BCNF.**

Prefers {Name, ID}

Candidate Key-> {Name, ID}

Here no attributes are multi valued. So, table is in **1NF**.

Here no attributes are dependent on any other attribute, all the attributes together form a candidate key. So, table is in **2NF**, **3NF** and also is in **BCNF**.

Like {GID, ID}

Candidate Key-> {GID,ID}

Here no attributes are multi valued. So, table is in **1NF**.

Here no attributes are dependent on any other attribute, all the attributes together form a candidate key. So, table is in **2NF**, **3NF** and also is in **BCNF**.

Classify{GID, Title}

Candidate Key-> {GID, Title}

Here no attributes are multi valued. So, table is in **1NF**.

Here no attributes are dependent on any other attribute, all the attributes together form a candidate key. So, table is in **2NF**, **3NF** and also is in **BCNF**.