**Northeastern Art Gallery Management**

**Team Members: -**

|  |  |
| --- | --- |
| **NAME** | **NUID** |
| Anish Surti | 001814243 |
| Nilank Sharma | 001279669 |
| Sidharth Upadhyay | 001389944 |

**Purpose: -**

Northeastern Art Gallery Management is a database system that allows buyers to purchase different art styles online/offline. In addition to this, buyers also have the option to buy arts online. It provides various categories too, in buyers’ accordance. It gives privileges to buyers to create their own portfolio.

**Business Problems Addressed: -**

* Northeastern University Art Management, marketing, and sales staff to generate descriptive reports.
* Buyers don’t have to go to museums and buy from selected art stuff. Also, to give thorough art information to enhance or aware buyers regarding it.
* Deliver shrewdness to artists and management staff to concentrate on art work by thorough analysis of report and effectively covering age group of users.
* Permit sales staff to analyze and refine sales quarterly objectives.
* It will eliminate the delays in the generation of reports that which item has sold to whom and also   
  Searching will become more efficient and faster.
* It will also provide assurance to the customer that they can buy the art they like there would be no pressure.

**Business Rules: -**

* Each Artist may participate in zero or more Art Exhibitions.
* Each Artist may have zero or more Art Work.
* Each Art will have at most one buyer.
* Each Exhibition will have start and end date.
* Each Gallery will have one or more Employees.
* Each Employee, Customer will have one Address.
* Each Customer will have one or more Bills.
* Each Employee will have one fixed Wage
* Each Customer can place one or more orders and each order will have order details.

**Design Requirements: -**

* Use Crow’s Foot Notation.
* Specify the Primary Key in each table by PK besides the table.
* Draw a line between the fields of each table to show the relationships between each table. This line should be pointed directly to the fields in each table that are used to form the relationship.
* Specify which table is on the one side of the relationships by placing a one next to the field where the line starts.
* Specify which table is on the many sides of the relationship by placing a crow’s feet symbol next to the field where the line ends.

**Design Decisions: -**

|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Why Entity Included** | **Entity Relationships** |
| **Payment\_Method** | It records all payment related data of customers for an art/arts. A customer can make payment through credit/debit card. The entity also stores related data about all payments. | Identifying One-to-one relationship exist with entity Bills and Order as payment would not exist without the Bills. |
| **Customers** | Customer is a subtype of User entity. He/she can buy new artworks of different artists from various exhibitions. Exhibition of various arts are conducted to attract customers. | This is a distinct subtype under entity User. It forms One-to-One relationship with entity Addresses and One-to-Many relationship with entity Bills and Order. |
| **Artists** | Artist will be responsible for creating arts which will be showcased in Exhibitions from where customer can buy, so that they can earn money. | This will form One-to-Many non-identifying relationship with entity Arts, as an Artist can exist without an Art. |
| **Arts** | The entity stores records of all Arts and their corresponding Artists. An exhibition contains multiple Arts. It also contains information of Gallery where it is held. | Arts and Exhibition will have many-to-many relationships which is established by bridge table between them. Arts also forms Many-to-One relationship with Gallery and One-to-One relationship with Order Details. |
| **Bills** | Bills are generated when a Customer buys an Art from an Exhibition. It also contains relationship with Payment\_Method which stores Debit/Credit Card information made for the Payment. | It forms One-to-One identifying relationship with Payment Method, as Payment\_Method cannot exist without Bills. It forms Many-to-One relationship with Customers and Exhibitions. |
| **Addresses** | Address contains records of all Users. It contains information about street, state, zip code, country etc. | It forms One-to-One relationship with entity Customers and Employees, as both can contain only one address. It has also same relationship with Gallery entity. |
| **Exhibitions** | Exhibitions is held in a Gallery which displays multiple Arts. A gallery can hold multiple exhibitions at different time. | It forms Many-to-One identifying relationship with entity Gallery. It also forms One-to-Many relationship with Bills and Many-to-Many relationship with Arts which is established by Bridge table between them. |
| **Wages** | Stores all records related to wages of the employees working in the Gallery. Like their hourly wage, shift time etc. | It has just One-to-One relationship with Employees entity, as each employee can have only one corresponding record in the Wages table. |
| **Gallery** | Gallery is the location in University where multiple exhibitions are conducted. Each exhibition displays various Arts made by various Artists. Customers visits Gallery and based on their interest they may or may not buy it. | It has One-to-Many relationship with Arts and Employees respectively. The address of the Gallery is stored in the Entity Address which also stores addresses of entity Users. |
| **Employees** | Employees are the people who works in the Gallery. They work on shift basis and they are paid based on number of hours they work. | It has Many-to-One relationship with entity Gallery and One-to-One relationship with entity Wages as each employee must have only record in the Wages entity. |
| **Order** | This table stores details of an order with attributes like Order\_ID, Order\_Status, Customer\_ID, Order\_Total, Exhibition\_ID to correlate Order details with specific customer with status and total amount on an order. | It has Many-to-One relationship with Customers, One-to-Many relationship with Order Details and One-to-One relationship with Payment Method. |
| **Order Details** | This is to contain information on order details with respect to the specific art the order relates to and attributes like line\_total and art\_id. | It has Many-to-One relationship with Order and One-to-One relationship with Arts. |
| **Arts\_has\_exhibitions** | This is to establish many-to-many relationship between two entities Arts and Exhibitions. It basically acts as a Bridge table between the two entities. | It forms Many-to-One relationship with both Arts and Exhibitions. |