

|  |
| --- |
| Museum  **SUBJECT AREAS** |
| **Logo / Image** |

**Contents**

[2 Museum Description 3](#_Toc196748426)

[2.1 Museum background 3](#_Toc196748427)

[2.2 Problems. Current Situation 3](#_Toc196748428)

[2.3 The Benefits of implementing a database. Project Vision 3](#_Toc196748429)

[3 Model description 4](#_Toc196748430)

[3.1 Definitions & Acronyms 4](#_Toc196748431)

[3.2 Logical Scheme 5](#_Toc196748432)

[3.3 Objects 5](#_Toc196748433)

# 

# Museum Description

## Museum background

Museums are more than just buildings full of artifacts — they are places where history, culture, and creativity come alive. They preserve important pieces of our shared heritage, offer spaces for learning, and inspire visitors of all ages. To do this important work, museums must carefully manage their collections, organize exhibitions and educational programs, and coordinate the efforts of many staff members.

In the past, much of this work was done using paper records or simple spreadsheets. While those methods served their purpose for a time, they often led to missing information, slow processes, and challenges in keeping track of everything. As museums grow and offer more to the public, they need better tools to stay organized and efficient.

This project was created to meet that need — by building a professional database that helps museums manage their collections, staff, visitors, events, and budgets all in one place.

With this system, museums can focus less on paperwork and more on what really matters: sharing knowledge, protecting history, and creating great experiences for every visitor.

## Problems. Current Situation

Today, many museums still rely on manual methods to manage their collections, events, and visitor information. Paper records, spreadsheets, and disconnected systems make it difficult to keep track of where artifacts are located, how exhibitions are organized, or who attended specific events. Important information can easily be lost, duplicated, or overlooked. Without a central database, museum staff often face challenges such as time-consuming record searches, inconsistent data, and limited ability to analyze visitor trends or financial performance. Managing loans, borrowed items, educational programs, and ticket sales separately adds even more complexity to daily operations. As museums grow in size and reach, the lack of an organized, efficient system puts strain on resources and limits their ability to plan ahead, collaborate with other institutions, and serve visitors effectively. There is a clear need for a unified solution that brings together all key areas of museum management into one reliable and easy-to-use platform.

## The Benefits of implementing a database. Project Vision

The museum database project aims to create a centralized, professional, and efficient system for managing all museum operations. By integrating collection management, visitor tracking, financial planning, and educational programming into a single digital platform, the museum will:

* Improve operational efficiency
* Enhance cultural preservation
* Expand educational outreach
* Strengthen visitor engagement
* Build stronger collaborations with other institutions
* Support long-term strategic growth with reliable data insights

The system is designed to be scalable, secure, and user-friendly, ensuring that museums can adapt and grow while maintaining a high standard of collection care and public service.

# Model description

## Definitions & Acronyms

Museum Organization that owns artifacts and organizes exhibitions.

Budget Financial plan or record for a museum.

Employee Staff member working for the museum.

Object Complete list of museum-owned items.

Storage Place where items are safely kept when not displayed.

Object\_loan Items the museum has loaned to other institutions.

Object\_borrow Items the museum has borrowed from other institutions.

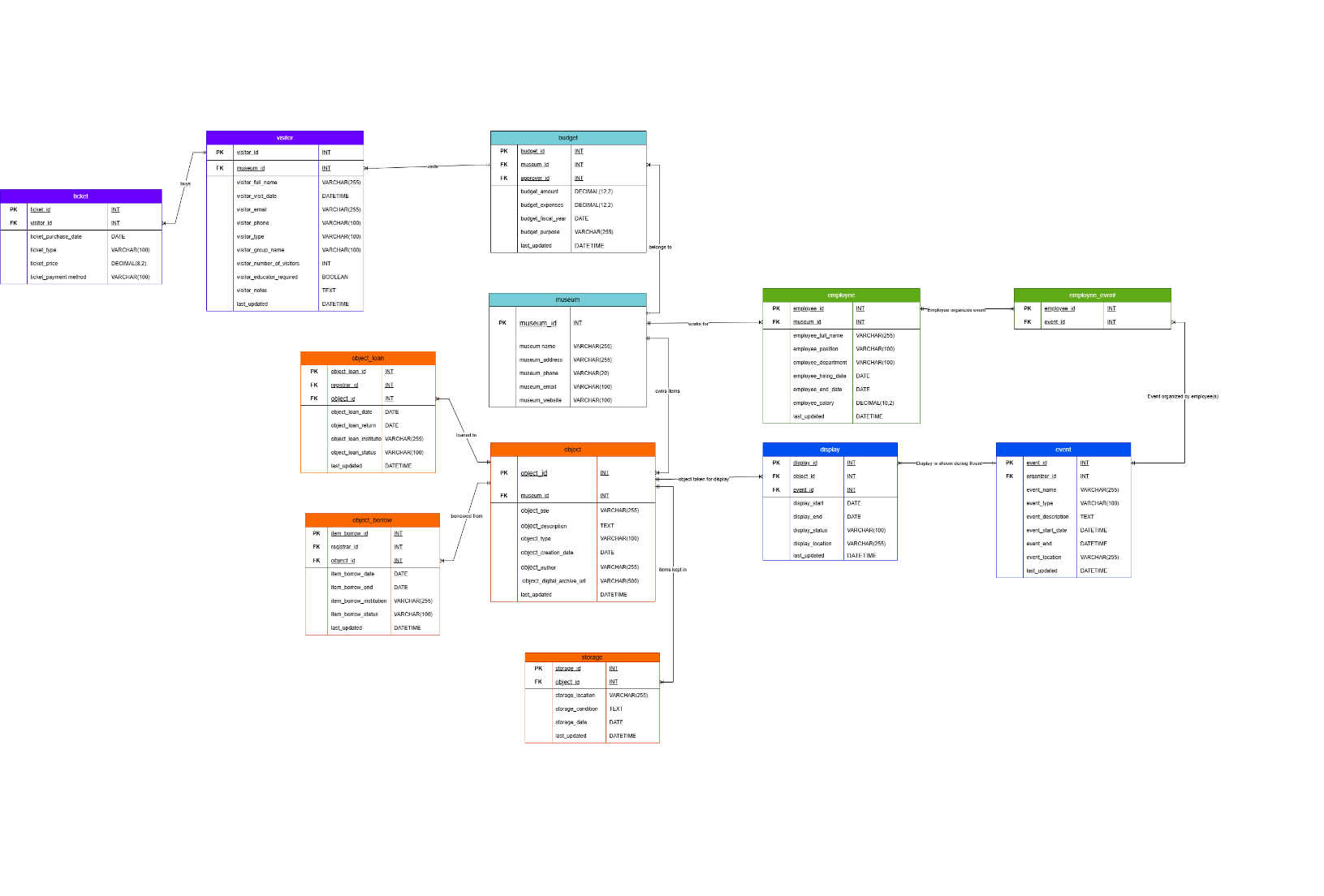
Display Record of when an item is exhibited publicly.

Event Activity or gathering held by the museum (e.g., tour, lecture).

Visitor Individual or group visiting the museum.

Ticket Purchase that allows visitor access to a visit or event.

## Logical Scheme



## Objects

The database schema supports museum operations through a structured set of tables, including: museum (core museum data), budget (financial tracking), employee (staff information), object (museum collection items), display (items on public view), storage (items kept in controlled conditions), item\_loan (artifacts loaned to other institutions), object\_borrow (artifacts borrowed from other institutions), exhibition (organized exhibits), event (museum events and activities), education (educational programs), visitor (visitor information), visit (visit records), ticket (ticket sales), and visitor\_visit (visitor attendance management).  
This comprehensive system efficiently tracks all aspects of museum activity — from artifact movements and financial planning to visitor engagement and educational programming — enabling museum administrators to coordinate operations, protect collections, monitor resources, and improve visitor experiences through data-driven insights.

The museum table keeps basic info about each museum like its name, location, phone, email, and website.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| museum | museum\_id | Unique identifier for the museum, cannot be NULL, PK | INT |
| museum\_name | The name of the museum, cannot be NULL | VARCHAR (255) |
| museum\_address | The physical address where the museum is located, cannot be NULL | VARCHAR (255) |
| museum\_phone | Museum’s contact phone number, cannot be NULL | VARCHAR (20) |
| museum\_email | Museum’s general email address, cannot be NULL | VARCHAR (100) |
| museum\_website | Museum’s official website URL, can be null (maybe museum does not have a website) | VARCHAR (100) |

Comments on table relationships

**One-to-many** relationship between:

museum (museum\_id) and budget (museum\_id) – many budget records can exist for one museum (e.g., different fiscal years, special project budgets). Each budget entry is linked directly to a specific museum.

museum (museum\_id) and employee (museum\_id) – many employees work for a specific museum. Each employee is assigned to one museum via the museum\_id foreign key.

museum (museum\_id) and inventory (museum\_id) – many inventory items (artifacts, artworks, historical objects) belong to a specific museum. Each item is tied directly to one museum through museum\_id.

Example with data

|  |  |  |
| --- | --- | --- |
| museum\_id | museum\_name | museum\_address |
| 1 | National Art Museum | 1 Main Street, Capital City |
| 2 | History and Heritage Center | 25 Heritage Road, Oldtown |

|  |  |  |
| --- | --- | --- |
| museum\_phone | museum\_email | museum\_website |
| +1-555-123-4567 | info@nationalartmuseum.org | www.nationalartmuseum.org |
| +1-555-987-6543 | contact@heritagecenter.com | www.heritagecenter.com |

The budget table keeps track of each museum’s financial information like total funds, expenses, and the purpose of the budget. It also records who approved it and links it back to the right museum for easy tracking.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| budget | budget\_id | Unique identifier for the budget record, cannot be NULL, PK | INT |
| museum\_id | FK, links the budget to a museum table | INT |
| approver\_id | FK, links to the employee who approved the budget | INT |
| budget\_amount | The total allocated amount for the museum's budget | DECIMAL (12,2) |
| budget\_expenses | The total expenses used from the allocated budget | DECIMAL (12,2) |
| budget\_fiscal\_year | Fiscal year the budget applies to (example: 2024) | DATE |
| budget\_purpose | Description of the purpose or project related to the budget | VARCHAR (255) |
| last\_updated | Budgets can be adjusted after approval. | DATETIME |

Comments on table relationships

**One-to-many** relationship between:

museum (museum\_id) and budget (museum\_id) – many budget records can exist for one museum (e.g., different fiscal years, special project budgets). Each budget entry is linked directly to a specific museum.

Example with data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| budget\_id | museum\_id | approver\_id | budget\_  amount | budget\_  expenses | budget\_  fiscal\_year | budget\_  purpose |
| 1 | 1 | 101 | 500000.00 | 320000.00 | 2024-01-01 | Historical preservation project |
| 2 | 1 | 101 | 200000.00 | 50000.00 | 2023-01-01 | New art acquisition program |

The object table records all museum items, including artworks, artifacts, and historical objects. It contains important details such as the item's title, type, creation date, author, and a link to its digital archive. Since museums can house a wide variety of artifacts, the name "object" was chosen as a more general term for all stored items.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| object | object\_id | Unique identifier for each item in the museum's collection, cannot be NULL, PK | INT |
| museum\_id | FK, links the item to the specific museum it belongs to | INT |
| object\_title | The title or name of the item | VARCHAR (255) |
| object\_description | Detailed description of the item, including history, features, etc. | TEXT |
| object\_type | The type or category of the item (e.g., painting, sculpture, artifact) | VARCHAR(100) |
| object\_creation\_date | Date when the item was originally created or made | DATE |
| object\_author | The author, creator, or artist of the item | VARCHAR (255) |
| object\_digital\_archive\_url | URL link to a digital record or scanned version of the item | VARCHAR (500) |
| last\_updated | Shows the most recent time any info in the record was changed. | DATETIME |

Comments on table relationships

**One-to-many** relationship between:

museum (museum\_id) and object (museum\_id) – one museum can have many items. Each item belongs to one museum through the museum\_id foreign key.

object (object\_id) and object\_loan (object\_id) – one object can be loaned out multiple times over its lifetime. Each loan record is linked to a specific item.

object (object\_id) and object\_borrow (object\_id) – one item from object table can be borrowed from another institution multiple times. Each borrowing record connects to a specific item.

object (object\_id) and display (object\_id) – one item can be displayed multiple times at different exhibitions or events. Each display record links back to a single item.

object (object\_id) and storage (object\_id) – one item can have multiple storage records if its location or storage condition changes over time. Each storage record is linked to one item.

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| object\_id | museum\_id | object\_title | object\_description |
| 1 | 1 | The Starry Night | Famous oil painting by Vincent van Gogh depicting a night sky over a quiet town |
| 2 | 2 | Egyptian Sarcophagus | Ancient wooden coffin decorated with hieroglyphics, from the 18th Dynasty |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| object\_type | object\_creation\_date | object\_author | object\_digital\_archive\_url | last\_updated |
| Painting | 1889-06-01 | Vincent van Gogh | https://example.com/starry-night | 2025-01-03 14:00:00 |
| Artifact | 1300-01-01 | Unknown | https://example.com/egyptian-sarcophagus | 2024-09-15 11:13:21 |

This table keeps track of where each museum item is stored and its condition over time. Some sensitive items, like artworks, historical documents, or delicate artifacts, need to "rest" under special conditions (like controlled temperature and humidity). This table helps the museum manage storage locations and monitor the state of these items while they are not on display.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| storage | storage\_id | Unique identifier for the storage record, cannot be NULL, PK | INT |
| object\_id | FK, links the storage record to a specific item from inventory, cannot be NULL | INT |
| storage\_location | Physical location where the item is stored within the museum, cannot be NULL | VARCHAR (255) |
| storage\_condition | Description of the item's condition during storage (e.g., "excellent", "minor damage") | TEXT |
| storage\_date | Date when the item was moved into storage or condition was recorded, cannot be NULL | DATE |
| last\_updated | Shows the most recent time any info in the record was changed. | DATETIME |

Comments on table relationships

**One-to-many** relationship between:

inventory (object\_id) and storage (object\_id) – one item can have many storage records over time. Each storage record tracks where and under what conditions an item was stored.

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| storage\_id | inventory\_id | storage\_location | storage\_condition | storage\_date | last\_updated |
| 1 | 45 | Temporary Storage C2 | Excellent condition, no visible damage | 2022-05-10 | 2022-05-10 |
| 2 | 74 | Main Storage Room A5 | Very fragile, handle with care | 2023-02-15 | 2023-02-15 |

This table tracks items that are temporarily loaned out to other institutions or museums. It records which item was loaned, the borrowing institution, key loan dates, and the current status of the loan. It helps the museum manage and monitor the movement of valuable artifacts and artworks outside the collection.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| object\_loan | object\_loan\_id | Unique identifier for the loan record, cannot be NULL, PK | INT |
| registrar\_id | FK, links to the employee responsible for managing the loan (registrar), cannot be NULL | INT |
| object\_id | FK, links to the specific item being loaned, cannot be NULL | INT |
| object\_loan\_date | Date when the item was officially loaned out, cannot be NULL | DATE |
| object\_loan\_return | Date when the item is expected to be returned or is returned, cannot be NULL | DATE |
| object\_loan\_institution | Name of the institution receiving the loaned item | VARCHAR (255) |
| object\_loan\_status | Status of the loan (e.g., active, returned, overdue), cannot be NULL | VARCHAR (100) |
| last\_updated | Shows the most recent time any info in the record was changed. | DATETIME |

Comments on table relationships

**One-to-many** relationship between:

object (object\_id) and object\_loan (object \_id) – one inventory item can have many loan records over time. Each loan record documents when and where an item was sent on loan.

employee (registrar\_id) and object\_loan (registrar\_id) – one employee (registrar or collection manager) can manage multiple loan transactions.

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| object\_loan\_id | registrar\_id | object\_id | object \_loan\_date | object\_loan\_return | |
| 5 | 202 | 3 | 2023-05-10 | | 2024-05-10 |
| 8 | 202 | 4 | 2022-09-15 | | 2023-01-10 |

|  |  |  |
| --- | --- | --- |
| object\_loan\_institution | object\_loan\_status | last\_updated |
| Louvre Museum, Paris | Active | 2024-05-01 15:41:02 |
| Metropolitan Museum of Art, New York | Returned | 2023-01-11 17:00:02 |

This table manages records of items that the museum has borrowed from other institutions. It tracks when items were received, from where, expected return dates, and their borrowing status. This helps the museum maintain control over borrowed artifacts and honor lending agreements.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| object\_borrow | object\_borrow\_id | Unique identifier for the borrowed record, cannot be NULL, PK | INT |
| registrar\_id | FK, links to the employee responsible for managing the borrowing (registrar), cannot be NULL | INT |
| object\_id | FK, links to the specific item borrowed from another institution, cannot be NULL | INT |
| object\_borrow\_date | Date when the borrowed item was received, cannot be NULL | DATE |
| object\_borrow\_end | Date when the borrowed item is expected to be returned to the museum or is returned, cannot be NULL | DATE |
| object\_borrow\_institution | Name of the institution receiving the loaned object, cannot be NULL | VARCHAR (255) |
| object\_borrow\_status | Status of the borrowing (e.g., active, returned, overdue), cannot be NULL | VARCHAR (100) |
| last\_updated | Shows the most recent time any info in the record was changed. | DATETIME |

Comments on table relationships

**One-to-many** relationship between:

object (object\_id) and object\_borrow (object\_id) – one item/ object can have many borrowing records if it is borrowed from different institutions at different times for different events like exhibitions.

employee (registrar\_id) and object\_borrow (registrar\_id) – one employee (registrar or collection manager) can manage multiple borrow transactions.

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| object \_borrow\_id | registrar\_id | object\_id | object\_borrow\_date | object\_borrow\_end |
| 2 | 4 | 6 | 2023-03-01 | 2023-08-17 |
| 6 | 4 | 85 | 2024-01-15 | 2024-12-01 |

|  |  |  |
| --- | --- | --- |
| object\_borrow\_institution | object\_borrow\_status | last\_updated |
| Rijksmuseum | Active | 2023-03-01 13:05:02 |
| Uffizi Gallery | Returned | 2024-12-01 10:17:41 |

This table keeps track of museum staff, including their names, positions, departments, hire dates, and salaries. It connects employees to the museum where they work and helps manage employment history and responsibilities.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| employee | employee\_id | Unique identifier for each employee, cannot be NULL, PK | INT |
| museum\_id | FK, links the employee to the museum where they work, cannot be NULL | INT |
| employee\_full\_name | Full name of the employee (first name and last name), cannot be NULL | VARCHAR (255) |
| employee\_position | Job title or role of the employee (e.g., Registrar, Curator, Guide), cannot be NULL | VARCHAR (100) |
| employee\_department | Department where the employee works (e.g., Collections, Education, Administration), cannot be NULL | VARCHAR (100) |
| employee\_hiring\_date | Date when the employee started working at the museum, cannot be NULL, date < end\_date if do not work anymore | DATE |
| employee\_end\_date | If the employee has left, end\_date must be after hiring date. If still working, end\_date is NULL | DATE |
| employee\_salary | Employee's salary amount, cannot be NULL | DECIMAL (10,2) |
| last\_updated | Shows the most recent time any info in the record was changed. | DATETIME |

Comments on table relationships

**One-to-many** relationship between:

museum (museum\_id) and employee (museum\_id) – one museum can have many employees. Each employee belongs to one museum.

employee (employee\_id) and object\_loan (registrar\_id) – one employee (registrar or collection manager) can manage multiple outgoing loan transactions.

employee (employee\_id) and borrow (registrar\_id) – one employee (registrar or collection manager) can manage multiple incomings borrow transactions.

**Many-to-many** relationship between:  
employee (employee\_id) via employee\_event (employee\_id) to event(organizer\_id) – one employee assist in organizing many events for the museum, but many employees can help to organize one event, it depends on event/ exhibition/ education program and ect.

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| employee\_id | museum\_id | employee\_full\_name | employee\_position | employee\_department |
| 5 | 8 | Anna Smith | Curator | Collections |
| 10 | 8 | Emily Johnson | |  | | --- | |  |  |  | | --- | | Education Manager | | Education |

|  |  |  |  |
| --- | --- | --- | --- |
| employee\_hiring\_date | employee\_end\_date | employee\_salary | last\_updated |
| 2019-07-01 | NULL | 1700 | 2020-01-03 11:14:33 |
| 2020-10-20 | NULL | 2500 | 2020-10-20 15:33:32 |

This table tracks which employees are involved in organizing specific museum events, allowing multiple employees to be linked to multiple events.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| employee\_event | employee\_id | FK, links to the employee organizing or helping organize an event, cannot be NULL | INT |
| event\_id | FK, links to the event organized or helped organized by the employee, cannot be NULL | INT |

Comments on table relationships

**Many-to-many** relationship between:  
employee (employee\_id) via employee\_event (employee\_id) to event(organizer\_id) – one employee can organize many events for the museum, but many employees can help to organize one event, it depends on event/ exhibition/ education program and ect.

Example with data

|  |  |  |
| --- | --- | --- |
| employee\_id | event\_id | employee\_event |
| 45 | 17 | 1 |
| 95 | 17 | 2 |

Objects from the museum's collection can be prepared for temporary displays for exhibitions, special events or celebrations. Each display represents the presence of a single object at a particular event. Events can showcase multiple displays at once, allowing several artifacts, artworks, or items to be exhibited together during the same occasion. For this reason, the display table is added to keep track of displayed objects.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| display | display\_id | Unique identifier for each display record, **cannot be NULL**, PK | INT |
| object\_id | FK, links the display to a specific inventory item being exhibited | INT |
| event\_id | FK, links the display to the exhibition where the item is shown | INT |
| display\_start | Date when the item was placed on display | DATE |
| display\_end | Date when the item was removed from display, can be null if object is still displayed | DATE |
| display\_status | Current status of the display (e.g., "active", "returned", "planned", "cancelled") | VARCHAR (100) |
| display\_location | Shows where object is displayed (e.g. The Dining Hall, Room A25 etc.) | VARCHAR (255) |
| last\_updated | Shows the most recent time any info in the record was changed. | DATETIME |

Comments on table relationships

**One-to-many** relationship between:

Object (object\_id) and display (object\_id). Each object from the collection can be displayed many times at different events over time. A display always refers to exactly one specific object.

Display (event\_id) and event (event\_id). Each display is shown during exactly one event. However, one event can have many displays (many different objects being exhibited together).

Example with data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| display\_id | object\_id | event\_id | display\_start | display\_end | | display\_status | |
| 44 | 10885 | 52 | 2023-06-01 | | 2023-09-01 | | returned | |
| 96 | 4128 | 3 | 2024-01-15 | | NULL | | active | |

|  |  |
| --- | --- |
| display\_location | last\_updated |
| The White Room | 2023-09-01 10:41:23 | |
|  | 2024-01-15 09:41:23 | |

The event table stores detailed information about events organized by the museum, such as workshops, exhibitions, and educational programs. It tracks key event details including title, type, description, schedule, location, linked through employee\_event and display tables shown during the event. This helps the museum manage and organize all events efficiently and ensures proper linkage between events, employees, and displayed objects if needed.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| event | event\_id | Unique identifier for each event, cannot be NULL, PK | INT |
| organizer\_id | FK, links to the employee who is responsible for organizing the event, cannot be NULL | INT |
| event\_name | Title or name of the event (e.g., "Art Workshop", "Lecture Series"), cannot be NULL | VARCHAR (255) |
| event\_type | Type of event (e.g. workshop, exhibition, educational program etc.), cannot be NULL | VARCHAR (100) |
| event\_description | Detailed description of the event content or purpose | TEXT |
| event\_start\_date | Date and time when the event begins, cannot be NULL | DATETIME |
| event\_end | Date and time when the event ends, cannot be NULL | DATETIME |
| event\_location | Location where the event will take place (room, hall, area), cannot be NULL | VARCHAR (255) |
| last\_updated | Shows the most recent time any info in the record was changed. | DATETIME |

\

Comments on table relationships

**Many-to-many** relationship between:  
Employee (employee\_id) via employee\_event (employee\_id) to event(organizer\_id) – one employee can organize many events for the museum. However, many employees can help to organize one event too, it depends on event/ exhibition/ education program and ect.

**One-to-many** relationship between:

Event(event\_id) and display(event\_id) - one event can have multiple displays (objects exhibited) but the same object cannot be displayed in several exhibitions at the same time.

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| event\_id | organizer\_id |  | event\_name | event\_type |
| 102 | 41 |  | Renaissance Art Lecture | educational program |
| 96 | 75 |  | Spring Art Workshop | workshop |
| event\_description | event\_start\_date | event\_end | event\_location | last\_updated |
| Lecture on Renaissance masterpieces and techniques | 2023-06-15 10:00:00 | 2023-06-15 12:00:00 | Auditorium | 2023-06-15 14:07:00 |
| Hands-on art activities for children aged 6–12 | 2023-07-10 14:00:00 | 2023-07-10 16:00:00 | Main Exhibition Hall | 2023-07-10 17:14:08 |

This table stores information about visitors to the museum, whether individuals, groups, or school groups. It tracks basic contact details, visitor type, group information (if applicable), special requests, and whether an educator is needed during their visit.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| visitor | visitor\_id | Unique identifier for each visitor, **cannot be NULL**, PK | INT |
| visitor\_full\_name | Full name of the visitor (optional for groups or anonymous visits) | VARCHAR (255) |
| visitor\_visit\_date |  | DATETIME |
| visitor\_email | Email address of the visitor (optional) | VARCHAR (255) |
| visitor\_phone | Phone number of the visitor (optional) | VARCHAR (100) |
| visitor\_type | Type of visitor (e.g., Individual, Group, School Group) | VARCHAR (100) |
| visitor\_group\_name | Shows the name of the visitor group (e.g. school groups), can be null | VARCHAR (100) |
| visitor\_number\_of\_visitors | Shows number of visitors | INT |
| visitor\_educator\_required | Indicates whether a museum educator is requested for the visit (Yes/No) | BOOLEAN |
| visitor\_notes | Additional notes about the visitor (special needs, requests, etc.) | TEXT |
|  | last\_updated | Shows the most recent time any info in the record was changed. | DATETIME |

Comments on table relationships

**One-to-many** relationship between:

visitor (visitor\_id) and ticket (visitor\_id) – one visitor can buy many tickets, but each ticket is linked to only one visitor.

visitor (museum\_id) and museum (museum\_id) - many visitors visit the same museum (in this case the diagram is designed for one museum. Museum\_id is kept in visitor table if there would be need to analyze visitor data from several museums)

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| visitor\_id | visitor\_full\_name | visitor\_email | visitor\_phone | visitor\_type |
| 1 | John Miller | john.miller@example.com | +1-555-789-1234 | Individual |
| 2 | Emily White | emily.white@example.com | +1-555-222-3333 | Individual |
| visitor\_group\_name | visitor\_number\_of\_visitors | visitor\_educator\_required | visitor\_notes | Visitor\_date |
| null | 1 | TRUE | Requires a special tour for students with disabilities | 2025-01-02 14:00:00 |
| null | 1 | FALSE | No special requests | 2025-03-05 10:10:00 |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| ticket | ticket\_id | Unique identifier for each ticket purchase, **cannot be NULL**, PK | INT |
| visitor\_id | FK, links the ticket to the visitor who made the purchase | INT |
| ticket\_purchase\_date | Date when the ticket was purchased | DATE |
| ticket\_type | Type of ticket (e.g., Standard, Student, Group, Senior) | VARCHAR (100) |
| ticket\_price | Price paid for the ticket | DECIMAL (8,2) |
| ticket\_payment\_method | Payment method used for the ticket purchase (e.g., Credit Card, Cash) | VARCHAR (100) |

The ticket table records purchase information for events or visits. It tracks the type of ticket, purchase date, price, and payment method, linking each ticket to a visitor.

Comments on table relationships

**One-to-many** relationship between:

visitor (visitor\_id) and ticket (visitor\_id) – one visitor can purchase multiple tickets over time for different visits or events.

Example with data

|  |  |  |
| --- | --- | --- |
| ticket\_id | visitor\_id | ticket\_purchase\_date |
| 1047 | 1047 | 2023-07-05 |
| 458 | 458 | 2023-08-01 |
| ticket\_type | ticket\_price | ticket\_payment\_method |
| Standard | 7.00 | Cash |
| School Group | 100.00 | Invoice |