Works

The works table stores non-temporal basic information about works/items that are in the museum database. This includes information about the

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| wor-WorkCharID  (Primary Key) | WorkCharID | Not NULL | Stores the character identifier of a museum work  Not null because it is a primary key used in identifying the piece. |
| wor-WorkNumID  (Primary Key) | WorkNumID | Not NULL | Stores the numerical identifier of a museum work  Not null because it is a primary key used in identifying the piece. |
| wor-DatabaseEntryMuseum  (Primary Key)  Foreign Key=> references **Owner** table. | DatabaseEntrylocation | Not NULL | Stores the location in which the work originally existed before it’s inserted into the new database.  Not null because it is a primary key used in identifying the piece, was added because the museums involved in the merge may have items with the same WorkCharID and WorkNumID and this makes each item unique. |
|  |  |  |  |
| wor-WorkDescription | WorkDescription | None | Records a brief description about the work. This may includes information about how the work physically looks like, what the work is trying to describe, how different material/medium are used to build the work |
| wor-WorkPhysicalProperty | WorkPhysicalProperty | None | Sort museum items/works by storing what type of work the piece physically is. A work physical property can be: Metal Work, Furniture  Carving, Ceramics  Painting, Textile, Electrical Work, or Mechanical Work. This is useful because these values do not describe conceptual themes about the work which are stored elsewhere. |
| wor-WorkClassification | WorkClassification | None | Categorizes works which share the same physical property by classifying them based on more specific characteristics (for example, a painting may be a portrait, a landscape etc.). This attribute was chosen in order to maintain information on subtypes of some of the museums which served a similar purpose |
|  |  |  |  |
| wor-WorkCreator | WorkCreator | None | Stores the original creator of the museum work |
| wor-WorkCreationDate | WorkCreationDate | None | Stores the date that the work was created |
| wor-WorkBorrowable | WorkBorrowable | None | This attribute records whether a work in our database is a ‘potentially borrowed’ (i.e. isn’t owned by any of our five partner museums) This is required because whether a work locations, works, and be borrowed is not temporal data. Some works have been sold and will most likely not be borrowable and items owned by the museum do not need to be borrowed but items not owned by the museum must indicate if they can be borrowed. |
| wor-GeographicRegion | Varchar | Can be null | Add it later |
| wor-FieldofScience | Varchar | Can be null | Add it later |
| wor-Name | WorkName |  |  |

WorkMediums

This table stores information about the materials and media used in creating each of the museum’s works. This table must be present because many of the works from each of the museums have multiple stolen, destroyed, orAsian, Mediterranean, and they have been made of.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| wme-WorkCharID  (Primary Key)  Foreign Key=> references **Works** table. | WorkCharID | Not Null;  foreign key constraints | Needed because it is a unique character identifier of a work |
| wme-WorkNumID  (Primary Key)  Foreign Key=> references **Works** table. | WorkNumID | Not Null;  foreign key constraints | Needed because it is a unique numerical identifier of a work |
| wme-DatabaseEntrymuseum  (Primary Key)  Foreign Key=> references **Owner**  table. | DatabaseEntryLocation | Not Null;  foreign key constraints | Stores the location in which the work was originally entered into the new database. Must be added to all medium table to because it is a primary key from the works table that uniquely identifies works which may have the same WorkNumID and WorkCharID |
|  |  |  |  |
| wme-WorkMedium | WorkMedium | Not Null (each work must at least have one medium) | The physical materials and media the work are created from. |

Locations

This table stores non-temporal information about each of the locations that physically exist in each museum. This information includes location name, dimensions, suggested maximum and minimum works capacity, and the partner museum where the location exists.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| loc-RoomName  (Primary Key) | LocationName | Not Null | Stores the name of the museum room or location (ex: museum galleries) |
| loc-BuildingName  (Primary Key)  Foreign Key=> references **Owner** table. | MuseumName | Not Null;  foreign key constraints | Stores the name of the building that the location exists in. |
| loc-travelingExhibtionlocation | boolean | Not null | Check the building is for travelling |
| loc-LocationSuggestedCapacityMin | LocationSuggestedCapacityMin | Value Must be greater than or equal to zero. | Stores the suggested minimum number of works that can exist in a location |
| loc-LocationSuggestedCapacityMax | LocationSuggestedCapacityMax | Value Must be greater than or equal to zero. | Stores the suggested maximum number of works that can fit in a location |
| loc-LocationWidth | LocationWidth | Value Must be greater than or equal to zero. | Stores the width of the location measured in meters. |
| loc-LocationLength | LocationLength | Value Must be greater than or equal to zero. | Stores the length of the location measured in meters |

WorkLocations

This is table records temporal information regarding the works’ location. This includes recording past, current, and future locations of a work. It provides a full location history for all works in the five partner museums. The table is necessary in order to obtain information on where a work was located at different times and that is why the location of a work must be temporal data. The table mainly has attributes for foreign keys which reference a location and a work and also possesses the time of arrival and time of departure for a work.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| wol-WorkCharID  (Primary Key)  Foreign Key=> references **Works** tabl | WorkCharID | Not Null;  foreign key constraints | Stores the character identifier of a museum work |
| wol-WorkNumID  (Primary Key)  Foreign Key=> references **Works** tabl | WorkNumID | Not Null;  foreign key constraints | Stores the numerical identifier of a museum work |
| wol-DatabaseEntryMuseum  (Primary Key)  Foreign Key=> references **works** table. | DatabaseEntrylocation | Not Null;  foreign key constraints | Stores the location from where the work originally entered into the new database. This was added because a DatabaseEntryLocation is primary key in the works table. |
|  |  |  |  |
| wol-LocationName  (Primary Key)  Foreign Key=> references **Locations** table. | LocationName | Not NULL | Stores the name of the museum room or location (ex: museum galleries) that the work is stored or was stored in |
| wol-MuseumName  (Primary Key)  Foreign Key=> references **Locations** table. | MuseumName | Not Null;  foreign key constraints | Name of the museum that has the location where the work was or is being stored in |
| wol-WorkArrivalTime  (Primary Key) | WorkArrivalTime | Not Null | Stores the time and date that a museum work arrived at a specific location, required as a primary key to differentiate occasions of the same work showing up in the same location. |
| wol-WorkDepartureTime | WorkDepartureTime | None | Stores the time and date that a museum work departed from a specific location in order to know the extent that an item was in a location. |

Doors

This table stores non-temporal information about the pathways between rooms and/r gallery in each of the five partner museums. This table allows a location (i.e. room or gallery) to be connected to one or more other locations in the same museum. This table helps in planning exhibitions that need to use more than one location (i.e. gallery) in a museum. For every door physically in the museum between two rooms, room a and room b for this example, 2 entries are made into the database showing that there is a door leading from room a to room b and a door leading from room b to room a excluding cases of a one way door. This decision was made so that it could be specified if a door is only way because there would only be one record for the door in the database and the original location would not be reachable by the end location.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| dor-MuseumNameOrigin  (Primary Key)  Foreign Key=> references **Locations** table. | MuseumName | Not Null;  foreign key constraints | Stores the name of one of the five partner museums in the database. This attribute is not need for both locations because locations in separate museums would not be connected by a door. |
| dor-MuseumNameDestination  (Primary Key)   * Memo from Sam * Changed from museum name end   Foreign Key=> references **Locations** table. | MuseumName | Not Null;  foreign key constraints |  |
| dor-LocationNameOrigin  (Primary Key)  Foreign Key=> references **Locations** table. | LocationName | Not Null;  foreign key constraints | A museum’ location name that represents the origin or start of the connection (pathway). |
| dor-LocationNameEnd  (Primary Key)  Foreign Key=> references **Locations** table. | LocationName | Not Null;  foreign key constraints | A museum’s location name that represents the end or final point of the connection (pathway). |

Exhibitions

This table stores basic and temporal information about all current and past exhibitions that has or will be displayed in any of the five partner museums. This information include the name of exhibitions, a brief description of the exhibition and its collection, the museum in which the exhibition has occurred, currently occurring or will occur in the future and most important the date and time in which each exhibition starts and ends. It was chosen to be temporal data because exhibitions with the same name may happen more than once so a startdate is needed to differentiate them

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| exh-ExhibitName  (**Primary Key**) | ExhibitName | Not Null; | Stores the name of an exhibition that is either held in one of the five museums or as a traveling exhibition abroad. |
| exh-MuseumName  (**Primary Key**)  **Foreign Key**=> references Owner table. | MuseumName | Not Null;  foreign key constraints | Stores the name of one of the five partner museums in which the exhibition is or will be held. Was added as a primary key in order the differentiate exhibitions which may occur or have occurred at separate museums on the same date. |
| exh-ExhibitStartDate  (**Primary Key**) | ExhibitStartDate | Not Null; | The start date that an exhibition is displayed to the public. |
| exh-ExhibitEndDate | ExhibitEndDate | Not Null;  Value > StartDate   * Memo from sam. Why is this not null, what if we don’t know when it will end | The end date that an exhibition is displayed to the public. |
| exh-ExhibitDescription | ExhibitDescription | None | Stores a brief description of the exhibition and the exhibition works’ collection. |
| exh-SecurityPersonName   * Memo from sam, changed name from Security because security does not describe enough |  | Could be null | Copy and paste travelling exhibition table |
| exh-IsTravelingExhibtion   * Memo from sam: Changed from just TravellingExhibition | * Memo from sam, creating domain for this attribute | Not null |  |
| exh-ExhibitDepartureDate   * Memo from sam, this should be identical to end date, remains not added to works file on nov28 | ExhibitDepartureDate | **Could be null**  **If not null, travelling exhibition all has to be true** | The date the works departed from the original museum and were sent out to be displayed in a traveling exhibition. |

ExhibitionLocations

This table stores temporal information about the locations that exhibits are, have, or will be occupying. It is temporal because it is accessing temporal data from the exhibitions table which is needed to differentiate exhibitions by the same name in the same museum. All attributes are needed to reference the exhibition and location for which an Exhibition\_Location record will refer to.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description- |
| elo-ExhibitName  (Primary Key)  Foreign Key=> references Exhibitions table. | ExhibitName | Not Null; | Stores the name of an exhibit that is either displayed in one of the five museums or as a traveling exhibition abroad. |
| elo-ExhibitStartDate  (Primary Key)  Foreign Key=> references **Exhibitions** table. | ExhibitStartDate | Not Null;  foreign key constraints | The start date that the exhibit was displayed at a certain location. Sets apart exhibitions at the same museum by the same name and also gives a timeline for when an exhibition will be run. |
| elo-LocationName  (Primary Key)  Foreign Key=> references **Locations** table. | LocationName | Not Null;  foreign key constraints | Name of a location within one of the five partner museums that an exhibit is displayed. |
| elo-MuseumOfLocation  (Primary Key)  Foreign Key=> references **Locations** table.   * Memo from sam: Changed from just MuseumName after added the MuseumOfExhbiition attribute to reduce confusion | MuseumName | Not Null;  foreign key constraints | Stores the name of one of the five partner museums in which the exhibition will be held.. |
| elo-LocationStartDate | PK |  |  |
| elo-LocationEndDate |  |  |  |
| elo-MuseumOfExhibition   * Memo from sam: This must be here for reasons of the foregin key which reference the exhbiition | PK |  |  |

ExhibitionWorks

This table stores non-temporal information about the works contained in each Exhibition. Note that: Exhibitions are identified by their name, starting date and the museum they are located at. The date an item entered the exhibition is not needed because only the location is needed for security and insurance purposes so it was decided a start date for the item would not be included. All attributes are needed as they are needed to uniquely identify exhibitions and works for which the table refers to.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| exw-ExhibitName  (Primary Key)  Foreign Key=> references **Exhibitions** table. | ExhibitName | Not Null; | Stores the name of an exhibit that is either displayed in one of the five museums or as a traveling exhibition abroad. |
| exw-ExhibitStartDate  (Primary Key)  Foreign Key=> references **Exhibitions** table. | ExhibitStartDate | Not Null;  foreign key constraints | The start date that the exhibit was displayed at a certain location. |
| exw-  (Primary Key)  Foreign Key=> references **Exhibitions** table. | MuseumName | Not Null;  foreign key constraints | Name of the museum that the exhibit was contained in. |
|  |  |  |  |
| exw-WorkCharID  (Primary Key)  Foreign Key=> references **Works** table. | WorkCharID | Not Null;  foreign key constraints | Stores the character identifier of a museum work |
| exw-WorkNumID  (Primary Key)  Foreign Key=> references **Works** table. | WorkNumID | Not Null;  foreign key constraints | Stores the numerical identifier of a museum work |
| exw-WorkDatabaseEntryMuseum  (Primary Key)  Foreign Key=> references **Works** table.   * Memo from sam, changed from just DatabaseEntryMuseum | DatabaseEntrylocation | Not Null;  foreign key constraints | Stores the location in which the work originally existed before it’s inserted into the new database. |
| exw-DateWorkAdded   * Memo from sam, changed from DateAdded | PK   * Memo from sam created domain DateWorkAdded |  | NOTE: added description |
| exw-DateWorkRemoved   * Memo from sam, changed from DateRemoved | * Memo from sam created domain DateWorkRemoved |  | NOTE: added description |

Owners

**:** This table stores non-temporal information about the organizations or people who currently possess or have owned a work/item in the museum’s database. This was chosen to be one of the tables for the sake of a lack of redundancy of information. A work can have multiple owners over its existence so it is not sufficient to enter an owner attribute into the works table. An owner can also have multiple works so if there is an owner's table that a Works\_Owners table must reference then this will make finding information owned by an owner easier. If, for example, a Works\_Owners table did not have to reference a record from an owner’s table then the owner may be misspelled sometimes when being entered into the database and this would create problems when trying to access all works by the same owner.

Owner(Combine museum and owner together into one table)

|  |  |  |
| --- | --- | --- |
| own-Name | PK | Make it to be unique |
| own-Email |  | Could be null |
| own-City |  | Could be null |
| own-State |  | Could be null |
| own-Streetname |  | Could be null |
| own-BuildingNumber |  | Could be null |
| own-isPartnerMuseum |  | Different with museum and person  True or false  Conld not be null |

WorkOwners

This table stores temporal information about the organizations or people who have owned or currently own a specific museum work/item in the museum database. All attributes except for OwnershipStartDate and OwnershipEndDate are required because they reference the primary keys of other tables. OwnershipStartDate is still required as a primary key however in the event that an owner purchases and sells a work multiple times.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| won-WorkCharID  (Primary Key)  Foreign Key=> references **Works** table. | WorkCharID | Not Null;  foreign key constraints | References the character identifier of a museum work. |
| won-WorkNumID  (Primary Key)  Foreign Key=> references **Works** table. | WorkNumID | Not Null;  foreign key constraints | References the numerical identifier of a museum work. |
| won-DatabaseEntryLocation  (Primary Key)  Foreign Key=> references **Works** table.   * Memo from sam, changed to WorkDatabaseEntryLocation | DatabaseEntryLocation | Not Null;  foreign key constraints | stores the name of the museum that first introduced the work into the system |
|  |  |  |  |
| won-OwnerName  (primary key)  Foreign Key=> references **Owners** table. | OwnerName | Not Null;  foreign key constraints | The name of the person or organization who possesses the work at a given point in time. |
|  |  |  |  |
| won-OwnershipStartDate  (Primary Key) | OwnershipStartDate | Not Null | The initial date that a work came into a person or organization’s possession. |
| won-OwnershipEndDate | OwnershipEndDate | None | The last date that a work was in a people possession. Helpful in determing how long a museum owned a work. Will not have values for works sold from the museum most of the time because it is not the museums business to know how long an outside owner has owned a work. It can therefore be null. |

WorkTransactions

This table stores temporal information about transactions involving museums’ works. Works’ transaction include purchasing a new work, loaning, borrowing, and selling a work as well as incidence of a work being donated, damaged, or when a work has gone missing.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| wtr-WorkCharID  (Primary Key)  Foreign Key=> references **Works** table. | WorkCharID | Not Null;  foreign key constraints | References the character identifier of a museum work which was involved in the transaction. |
| wtr-WorkNumID  (Primary Key)  Foreign Key=> references **Works** table. | WorkNumID | Not Null;  foreign key constraints | References the numerical identifier of a museum work which was involved in the transaction. |
| wtr-WorkDatabaseEntryMuseum  (Primary Key)  Foreign Key=> references **Works** table  -memo from sam changed from just DatabaseEntryMuseum | OwnerName | Not Null;  foreign key constraints | references the location that the museum work which was involved in the transaction initially entered the database. |
| wtr-MuseumInvolved  (Primary key)  FK to owner table   * Memo from sam, changed from MuseumAffected * I don’t know if this must be part of the primary key but I’m putting it in for now | OwnerName | NOT null |  |
| wtr-Client  FK to owner table | OwnerName | Could be null   * Clients may not want to provide information when involved in a transaction and not required to | The party which we are working with |
| wtr-TransactionType  (Primary Key)   * Memo from sam: I don’t know if this must be part of the primary key but I’m putting it in for now | TransactionType | Not Null | Stores the type of transaction happened on the work. A transaction can be either purchased, loaned, borrowed, sold, gone missing, donated, or damaged |
| wtr-TransactionTime  (Primary Key) | TransactionTime | Not Null | Stores the date and time that the transaction took place. Required as a primary key in order to differentiate items which have the same kind of transaction more than once. |

Sponsors

* Memo from sam, i think it would be good to have a sponsor and sponsors exhibitions table

This table stores temporal information about the people or organizations who sponsor specific traveling exhibits. It has it’s own table because there can be multiple sponsors for a travelling exhibition so it is not appropriate to have a sponsor attribute in the travelling exhibition table. All attributes except sponsorAmount are required because they are either foreign keys which refer to other tables primary keys for the travelling exhibition or in the case of sponsor name, it is a unique identifier for

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| spo-SponsorName  (Primary Key) | SponsorName | Not Null | Holds the name of a person or organization sponsoring a travelling exhibition. |
| spo-ExhibitName  (Primary Key)  Foreign Key=> references **exhibition**  table | ExhibitName | Not Null;  foreign key constraints | The name of the travelling exhibit the sponsor is sponsoring. |
| spo-MuseumName  (Primary Key)  Foreign Key=> references **exhibition** table | MuseumName | Not Null;  foreign key constraints | The name of the museum that is responsible for the traveling exhibition that the person or organization is sponsoring. |
| spo-ExhibitStartDate  (Primary Key) | ExhibitStartDate | Not Null; | The start date that an exhibition is displayed to the public. |
| spo-SponsorAmount | SponsorAmount | None | The amount of money that the sponsor paid to sponsor the exhibit. Included because it is helpful to know some of the financial benefits of the travelling exhibitions. |

WorkThemes

* Memo from sam, maybe topic? ConceptualCategory?
* I think there should definately be a seperate themes table, I think that might be a reason why carter thinks this table belongs in the works table, if there’s a seperate themes table it’s more clear that it’s not 1 to 1

This table stores non-temporal information about keywords and concepts related to museum works/items that helps in the organization and planning of future exhibitions. A work may fit into multiple themes so this table was created and the themes are based on what conceptual qualities a piece may have such as it being “fine brush be commemoratedwork”, a theme may describe a region or era a piece comes from. Themes do not fall under the same category as Physical property or classification because they are describing things about the work which are more relative terms and do not describe anything about what type of work the piece is.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| wth-WorkCharID  (Primary Key)  Foreign Key=> references **Works** table | WorkCharID | Not Null;  foreign key constraints | stores the reference to the character identifier of a museum work. |
| wth-WorkNumID  (Primary Key)  Foreign Key=> references **Works** table | WorkNumID | Not Null;  foreign key constraints | stores the reference to the numerical identifier |
| wth-WorkDatabaseEntryLocation  (Primary Key)  Foreign Key=> references **Works** table | OwnerName | Not Null;  foreign key constraints | stores the reference to the location where the work initially entered the database. |
| wth-Theme  (Primary Key) | Theme | None | stores the theme that is linked to a particular work |

WorksInsurance

This table stores temporal information about the insurance value of a particular museum work/item, including its insurance value when traveling abroad. It shows the changes the item may experience over time and this reason it is stored as temporal data.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Domain Name | Attribute constraints | Attribute description |
| win-WorkCharID  (Primary Key)  Foreign Key=> references **Works** table | WorkCharID | Not Null;  foreign key constraints | references the character identifier of a museum work. |
| win-WorkNumID  (Primary Key)  Foreign Key=> references **Works** table | WorkNumID | Not Null;  foreign key constraints | References the numerical identifier of a museum work. |
| win-DatabaseEntryLocation  (Primary Key)  Foreign Key=> references **Works** table | DatabaseEntryLocation | Not Null;  foreign key constraints | references the museum which entered the work into the database originally. |
|  |  |  |  |
| win-WorkInsureValue | WorkInsureValue | Monetary insurance based so it must be greater than or equal to 0. | The insurance value that a work held at a particular point in time. |
| win-InsureStartDate  (Primary Key) | InsureStartDate | Part of the primary key so it cannot be null. | The initial date that a work held a particular insurance value. |
| win-InsureEndDate | InsureEndDate | NONE | The last date that a work held a particular insurance value. |

Primary Key win-WorkCharID, win-WorkNumID, win-DatabaseEntryLocation, win-DatabaseEntryLocationAddress, win-InsureStartDate

Foreign key win-WorkCharID, win-WorkNumID, win-DatabaseEntryLocation, win-DatabaseEntryLocationAddress references Works.