DIGITAL COLLECTIONS DATABASE

FINAL PROJECT

For

IST659 – Database Admin Concepts & Management

By

Daniel Hanks

A report submitted in partial fulfillment of the requirements
for IST659

Submitted to

Professor Thompson

On

May 9, 2016

Daniel Hanks Jr
2014 Kingsdown Drive
Liverpool, NY
315-944-7574
dphanksj@syr.edu

Professor Thompson
Syracuse Unviersity School of Information Studies
Syracuse, NY 13224

Dear Mr. Thompson:

I am writing to inform you about the Digital Collections Database project proposal presented here. This proposal presents the plan for an information system that will bring this library to the forefront in digital preservation among its peers. This report will break down the plan to create a database driven system to store objects within a given collections, and allow the creation and updating of metadata within a given object.

If you have any questions or concerns regarding the project or the report please feel free to contact me by email at dphanksj@syr.edu or by phone at 315-944-7574. I would be happy to assist you with any further information you require in regards to the project.

Sincerely,

Daniel Hanks Jr

TABLE OF CONTENTS

Executive Summary

Digital Collections in the library contains collections or digitized parts of collections that are referred to as objects. This project will allow staff to enter the metadata associated with these objects into a central database which makes up an entire collection. To clarify, think of the database itself as the collection and the tables storing the data as objects.

Currently this library stores the metadata about the objects on spreadsheets, submits them to the IT staff for bulk exports into XML documents using a PHP script, which are then stored on a network drive. Records can then be displayed and preserved using that XML format. This method has numerous drawbacks. Examples include using IT staff resources for exports, inadequate method for digital preservation and simply an outdated approach to data input and output.

This project proposes to create a normalized database that allows staff to enter metadata associated with an object into a central database. This database can then be easily searched from and used to display the metadata to the public through a website. The database would be created using a prototype model allowing us to build a model of the database, show it to stakeholders, get feedback and modify the prototype accordingly. The database would support the creating and updating of objects, the generation of reports and the creation and maintenance of staff accounts to update the database.

Switching to this database driven system creates a change in the daily work flow for the stakeholders of this project. The elimination of spreadsheets and the removal of the IT staff from exporting data into XML allows the library staff to enter the metadata in themselves. Instead of multiple staff members working on separate sheets that need to be combined or multiple staff working on a shared spreadsheet one at a time they can now enter the data through a web based interface.

This also creates numerous benefits. This frees up IT staff resources now that they've been removed from the process. Library staff can now all work on one centralized database rather than using spreadsheets. Spreadsheets are prone to data errors and redundancy in data, which a properly designed database will eliminate. Having a centralized database also creates a more modern method for digital preservation that can be backed up and restored properly. This also makes it easier to create a user friendly interface that is searchable and can be displayed to the public on the web as well.

As a library, we continue to push to offer more digital collections for public consumption. It makes sense to modernize our methods of digital preservation and deliver a more efficient method to enter and display these objects and their metadata. We believe this a worthwhile investment that can only lead to more opportunities with future collections.

1 Planning

1.1 Description of the Business

This project is for a library that needs an internal database to store metadata associated with digitized objects that are part of a large, single collection. The purpose of this database is for digital preservation purposes along with offering a public display through websites if needed. The industry category for this should fall under Education. Currently this library stores the metadata about the objects on spreadsheets, submits them to the IT staff for bulk exports into XML documents using a PHP script, which are then stored on a network drive. Records can then be displayed and preserved using that XML format. Decisions are made by the librarians and IT staff involved in this process.

1.2 Business Problem

This project will attempt to eliminate the need of using the IT staff's resources for these exports, while creating a more modern approach to digital preservation, eliminating spreadsheets and using a database. This also gives us an opportunity to display collections to the public on the Internet by pulling the metadata from objects straight from the database and displaying it through a website. This also allows staff to enter, modify or delete object data using forms to connect directly to the database which will increase the quality of data being input and ultimately, output as well.

1.3 Decisions not made

Without having access to accurate data, the following decisions and issues cannot be addressed:

1. Are data integrity practices in place to ensure the current data is being recorded as intended?
2. Are unique identifiers being used to reference an object?
3. Does the current system address the sustainability that is necessary for digital information?
4. Are the essential attributes of a digital object being preserved?
5. Is the quality of data being output to the public a hindrance to future donors?

1.4 Project Scope

Digital Collections in the library contains collections or digitized parts of collections also referred to as objects. This project will allow staff to enter the metadata associated with these collections into a central database, which can then be easily searched from and ultimately displayed to the public through a website if necessary. Forms will be used to add, delete and modify these collections. A function considered for this project was giving the ability for certain staff to import spreadsheets of data themselves. This would be a modified script that the IT staff had been using to export data into XML that would check the spreadsheets for errors and then dump the data into the database. That data could then be checked and modified if necessary within the new database. Due to time constraints with this project, this function will not be implemented.

1.5 Assumptions and Constraints

The assumptions and constraints for this project are tied together for this project, in that there is one main programmer doing the work. The assumption is that the programmer's availability remains the same. This also assumes the programmer has access to the necessary technologies needed to create this project. Having one programmer is also a constraint since the project timelines do depend on his abilities.

2 Analysis

2.1 Methodology

This project will use the prototyping methodology for development. This methodology is described as follows:

There are several steps in the Prototyping Model:

1. The new system requirements are defined in as much detail as possible.
2. A preliminary design is created for the new system.
3. A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
4. The first prototype is analyzed, the developer notes its strengths and weaknesses, what needs to be added, and what should to be removed. The developer collects and analyzes the remarks from the users.
5. The first prototype is modified and a second prototype of the new system is constructed.
6. The second prototype is evaluated in the same manner as was the first prototype.
7. The preceding steps are iterated as many times as necessary, until the developer is satisfied that the prototype represents the final product desired.
8. The final system is constructed, based on the final prototype.
9. The final system is thoroughly evaluated and tested.

(Rouse, n.d.)

2.2 Processes Supported

Some processes supported in this project will include the following:

- Creating and updating objects.
- Generation of reports.
- Creation and maintenance of staff accounts to update the database.

2.3 Key Personnel

The key personnel that would be critical to this project include the IT Director, digital preservation staff at the library and the developer. The IT Director would ultimately green light this project and give the developer the time and resources necessary to complete this project. The library staff involved with digitizing collections are important to the developer since they give important feedback on the design of the database and forms, along with explaining the metadata to the developer. The developer not only has to create the database and UI, but they have to understand the needs of the library staff, along with understanding the metadata associated with the objects within a collection.

2.4 How Work Changes

The elimination of spreadsheets and the removal of the IT staff from exporting data into XML allows the library staff to enter the metadata in themselves. Instead of multiple staff members working on separate sheets that need to be combined or multiple staff working on a shared spreadsheet one at a time they can now enter the data through a web based interface. Obviously the change here for them is they lose the comfort of their spreadsheet which most find easy and intuitive and will have to become familiar with the web form and it's differences in that and a spreadsheet.

2.5 Benefits

There are numerous benefits to switching to a database-driven system. As touched on before, this frees up IT staff resources now that they've been removed from the process. Library staff can now all work on one centralized database rather than using spreadsheets. Spreadsheets are prone to data errors and redundancy in data, which a properly designed database will eliminate. Having a centralized database also creates a more modern method for digital preservation that can be backed up and restored properly. This also makes it easier to create a user friendly interface that is searchable and can be displayed to the public on the web as well.

2.6 Management Problems

Management problems could be created by the fear of a fairly drastic change in process. Some management might be more comfortable with the spreadsheet process since IT is the one that is processing the final data. Library staff couldn't accidently delete or modify XML files that are stored away for example. Library staff having more control of the data in that regard may be worrisome. There's also training of staff and possibly new hardware or software that may have to be implemented that would come at a cost.

3 Design

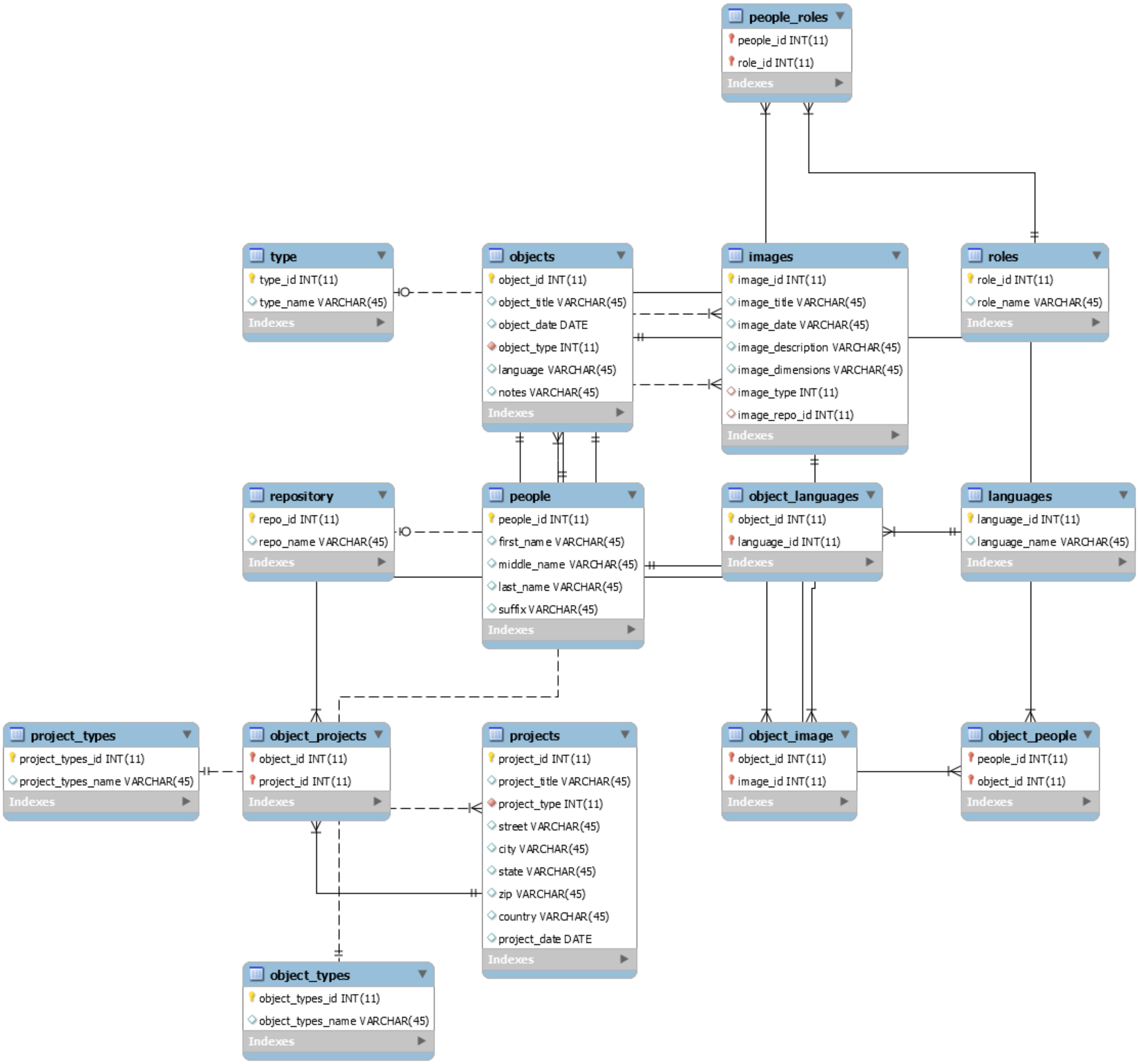
This information system was built with business rules in mind to help shape the creation of the database itself. These rules will help ensure data integrity, data quality in regards to input and output and ultimately the quality of preservation for each object. This section lays out 10 such business rules and shows how it they are implemented into the database.

In order to accomplish these design goals, a normalized database, containing 15 tables was created. These tables are normalize to the 3rd form which is a standard method that offers many benefits, such as a reduction is necessary disk storage and inconsistencies in data.

3.1 Business Rules

	Business Rule	Implementation
1	**Each Object has an Object Type.**	**1 to Many relationship between ObjectType table and Objects table.**
2	**Many Projects belong to each Object.**	**ObjectProject is a bridge table between the Projects table and the Object table.**
3	**Each Project has a Project Type**	**1 to Many relationship between ProjectType table and Projects table.**
4	**Many Images belong to each Object**	**ObjectImage is a bridge table between the Images table and the Object table.**
5	**Many Languages belong to each Object**	**ObjectLanguage is a bridge table between the Languages table and Object table.**
6	**Many People belong to each Object**	**ObjectPeople is a bridge table between the People table and Object table.**
7	**Each person in the People table has Roles**	**Roles are assigned to People in People table**
8	**Each Image has a Type.**	**ImageType is a bridge table between the Type**
9	**Each Image has a Repository.**	**1 to Many relationship between Image table and Repository table.**
10	**An Object can be Active or Inactive.**	**A check constraint on the Object table requires that an Object be Active or Inactive (0-Active, 1-Inactive) to indicate completion of object for display on web or more metadata still needs to be added.**

3.2 Data Model



4 Implementation

This project will be implemented through the use of tables, forms, queries, and reports. The tables will make use of primary and foreign keys to protect data integrity. Forms will be used for staff to add, modify and delete objects and metadata within the objects. Queries will be used to show options on dropdowns on forms to help enforce data integrity. Reports will be used to provide easy-to-read and printable views of the data.

4.1 Input Forms

This form allows library staff to maintain information about images. Each image has a unique ID, and has it's metadata stored for preservation purposes. This also makes these images easy to search for in future web interfaces and within the database itself.

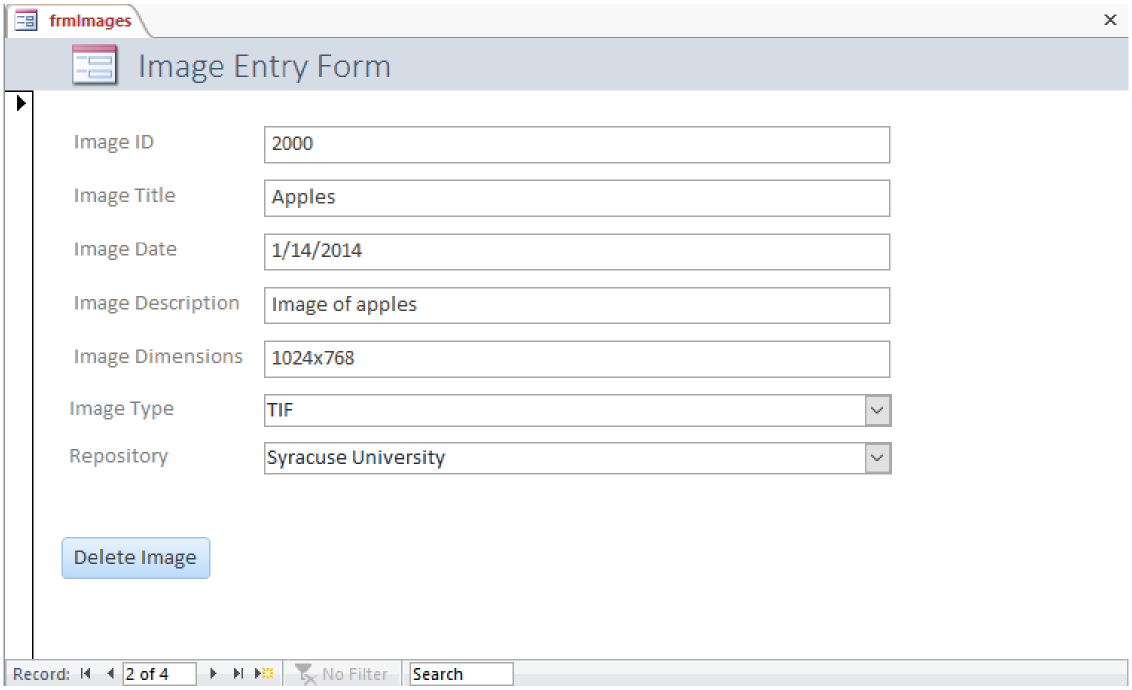
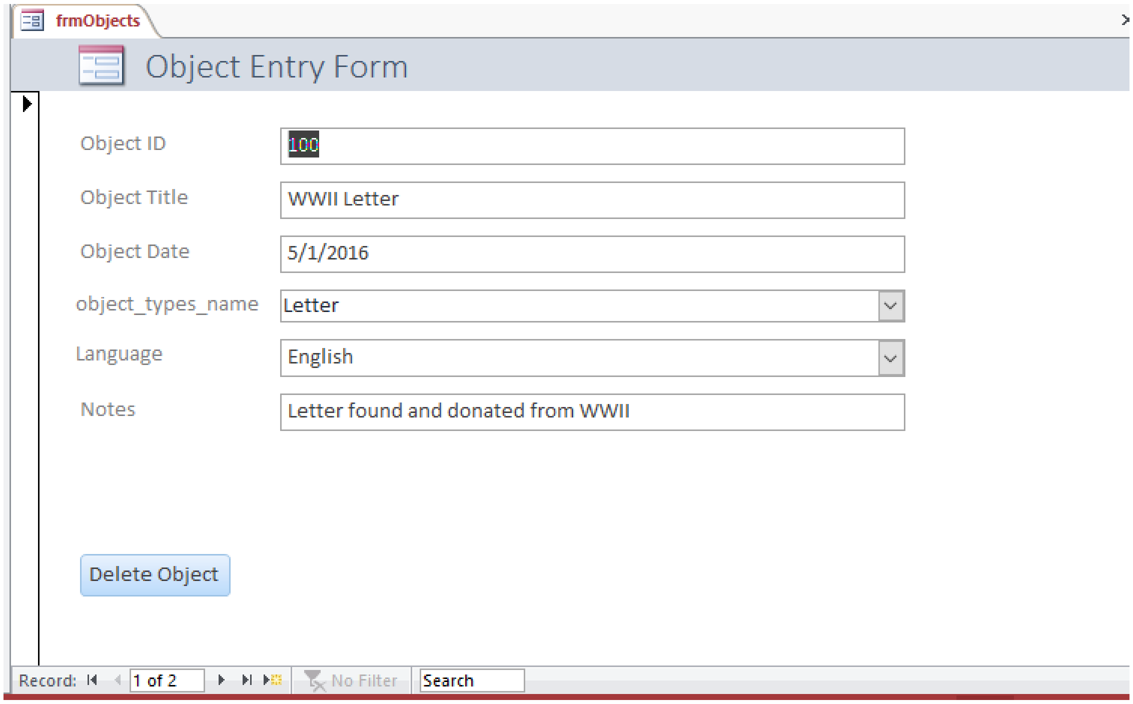


Image Entry Form

Image ID	2000
Image Title	Apples
Image Date	1/14/2014
Image Description	Image of apples
Image Dimensions	1024x768
Image Type	TIF
Repository	Syracuse University

Figure 1 – Image Entry Form

This form allows library staff to maintain information about objects. Similar to images,each object has a unique ID, and has it's metadata stored for preservation purposes and for searchability.

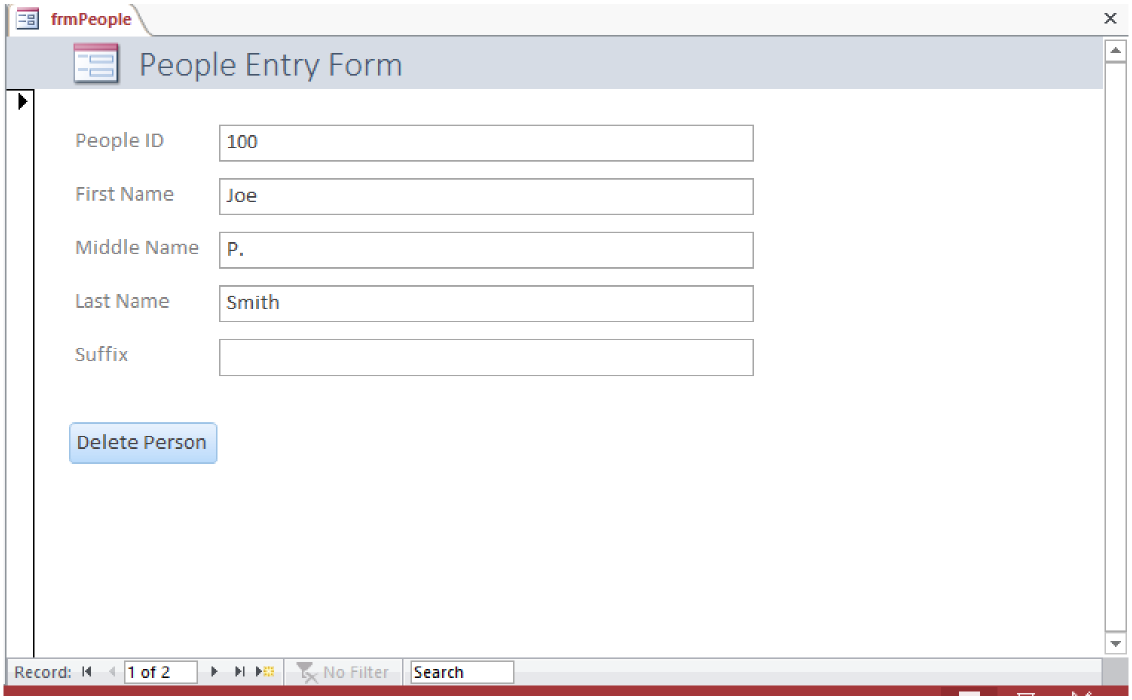


Object Entry Form

Object ID	100
Object Title	WWII Letter
Object Date	5/1/2016
object_types_name	Letter
Language	English
Notes	Letter found and donated from WWII

Figure 2-Object Entry Form

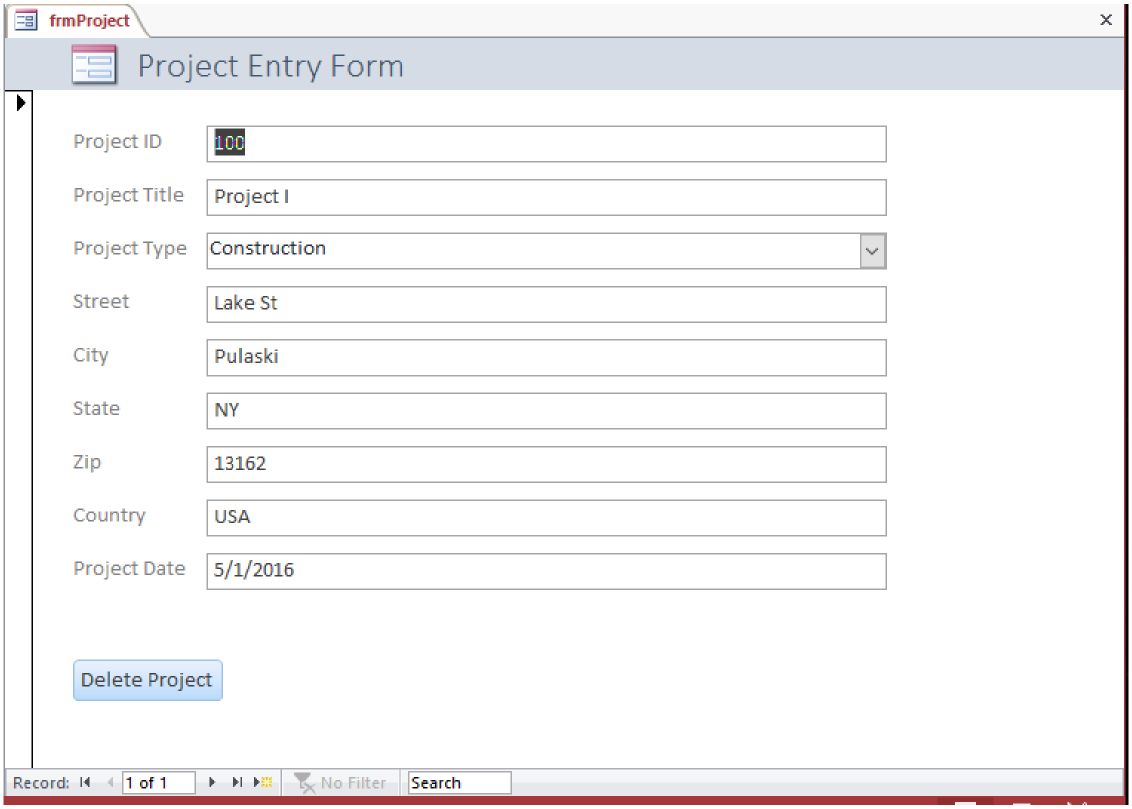
This form allows library staff to maintain information about objects. Similar to images, each object has a unique ID, and has its metadata stored for preservation purposes and for search ability.



People ID	100
First Name	Joe
Middle Name	P.
Last Name	Smith
Suffix	

Figure 3-People Entry Form

This form allows library staff to maintain information about projects. Each object has a unique ID, and has its metadata stored for preservation purposes and for search ability.



Project Entry Form

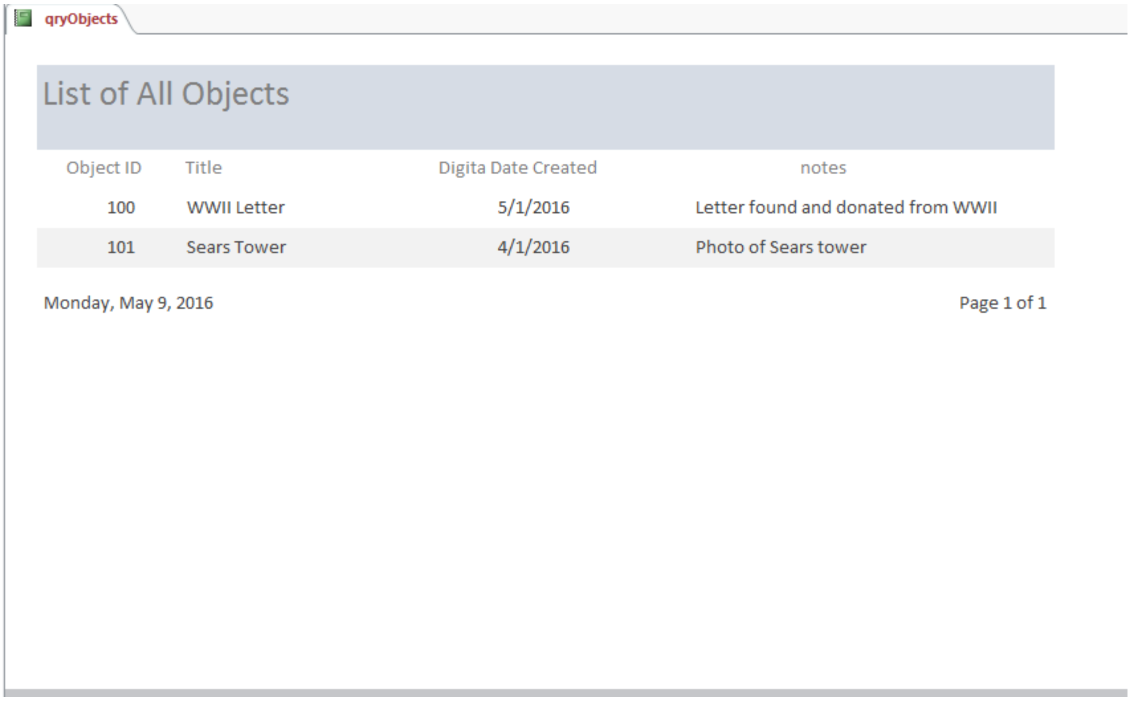
Project ID	100
Project Title	Project I
Project Type	Construction
Street	Lake St
City	Pulaski
State	NY
Zip	13162
Country	USA
Project Date	5/1/2016

Figure 4- Project Entry Form

4.2 Reports

The following show the reports to view objects, images and people in report format.

This shows the object report, which shows all objects and the required metadata for this report.



List of All Objects

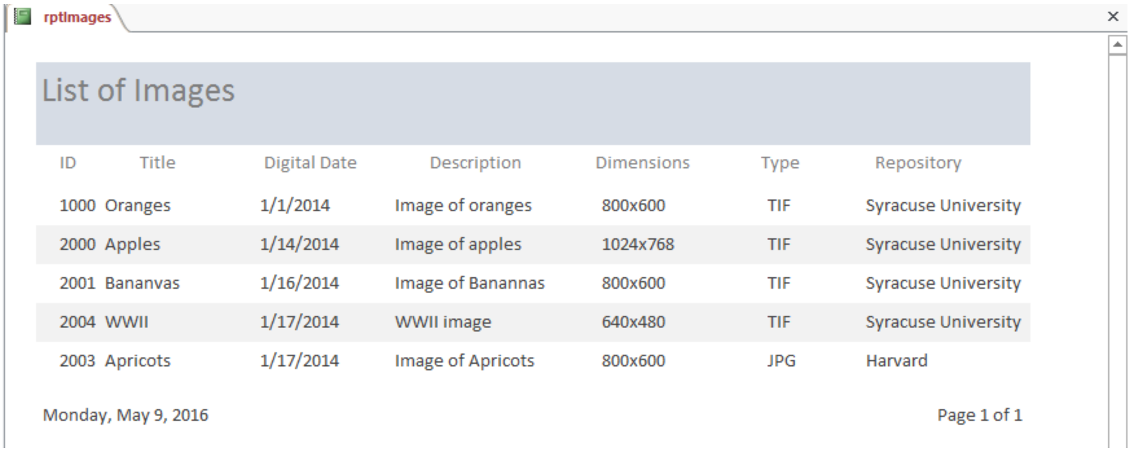
Object ID	Title	Digita Date Created	notes
100	WWII Letter	5/1/2016	Letter found and donated from WWII
101	Sears Tower	4/1/2016	Photo of Sears tower

Monday, May 9, 2016

Page 1 of 1

Figure 5-Object Report

This shows the image report, which shows all images and the required metadata for this report.



List of Images

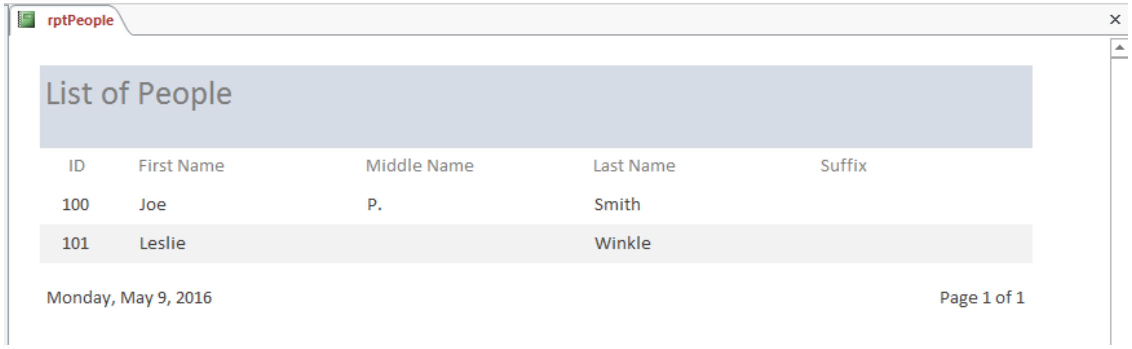
ID	Title	Digital Date	Description	Dimensions	Type	Repository
1000	Oranges	1/1/2014	Image of oranges	800x600	TIF	Syracuse University
2000	Apples	1/14/2014	Image of apples	1024x768	TIF	Syracuse University
2001	Bananvas	1/16/2014	Image of Banannas	800x600	TIF	Syracuse University
2004	WWII	1/17/2014	WWII image	640x480	TIF	Syracuse University
2003	Apricots	1/17/2014	Image of Apricots	800x600	JPG	Harvard

Monday, May 9, 2016

Page 1 of 1

Figure 6-Image Report

This shows the people report, which shows all people and the required metadata for this report.



List of People

ID	First Name	Middle Name	Last Name	Suffix
100	Joe	P.	Smith	
101	Leslie		Winkle	

Monday, May 9, 2016

Page 1 of 1

Figure 7-People Report

4.3 Additional Database Objects

The qryImages shows a list of all the images in the images table. It's linked to the type table and repository table to show type name and repository name rather than the id number for them. This makes the report much more useful to the user. It is used in rptImages:

```
SELECT images.*, type.type_name, repository.repo_name
FROM (images INNER JOIN type ON images.image_type = type.type_id) INNER JOIN
repository ON images.image_repo_id = repository.repo_id;
```

The qryObjects shows a list of all the objects in the objects table. It is used in the rptObjects:

```
SELECT objects.object_id, objects.object_title, objects.object_date, objects.notes FROM objects;
```

The qryPeople shows a list of all the people in the people table. It is used in rptPeople:

```
SELECT people.*
FROM people;
```

5 Support Requirements

Information systems have numerous supporting requirements. The ability to backup and restore data, securing the data and database audits and controls are some examples of these. The library emphasizes digital preservation making these requirements even more important. This is outlined below.

Objects that are digitized follow a specific process that begins with a physical object like a photo, for example. This photo is scanned into a digital format like a TIF, which are used because they maintain a very high image quality and preserve metadata within the image. The scanned TIF is considered the original copy, which we want nothing to do with. The TIF will be moved to a network drive, and a copy of it be stored elsewhere for use in the database. This copy will most likely be converted into a jpeg to reduce file size and is easier to display on a website. Both sets of images, along with the databases themselves will follow the same backup and recovery protocol. Nightly, incremental backups will be used for this data on their respective servers (server for image storage and a server for database storage). This means whatever changes are made that day will be backed up. A weekly full backup will be done as well, backing up all servers data. Daily backups will be rotated among a set of 10 tapes, while weekly backup tapes will be rotated as well. Tapes not being used for that week will be stored in a fire proof safe in an alternate location for disaster recovery purposes.

The databases themselves will feature auditing controls such as incident logging that logs all updates to data files and databases. The DBA will have full control of the database while users that simply update the database will only have rights to the objects themselves in the database. The DBA won't have the ability to de-activate the log in order to keep their power in check. A clone of the database server will be made with a VM to test software patches before implementing on the production server.

6 Documentation

This section contains the data dictionary and system catalog for the Digital Collections Database.

6.1 Data Dictionary

The data dictionary contains the information about the attributes and keys of a given entity. Each attribute has an attribute name, description, type/size, format, whether it is required (not null), the domain of the data (data range), default values for the attribute, whether it is a key, and any foreign keys listed .

Table Name: images

Table Description: The images table is used to store metadata about the images.

Name	Description	Type/Size	Format	Req'd	Domain	Key	Ref. Table
image_id	Image ID	INT		Y	Numeric only	PK	
image_title	Image Name	VARCHAR (45)		Y			
image_date	Image digital creation date	DATE	01/01/1900	Y			
image_description	Image description	VARCHAR (45)		N			
image_dimensions	Image dimensions	VARCHAR (45)		N			
image_type	Image type	INT		N	Numeric only	FK	Object_ projects
image_repo_id	Image repo ID	INT		Y	Numeric only	FK	repo-sitory

Table Name: objects

Table Description: The objects table is used to store metadata about the objects.

Name	Description	Type/Size	Format	Req'd	Domain	Key	Ref. Table
object_id	Object ID	INT		Y	Numeric only	PK	
object_title	Object title	VARCHAR (45)		Y			
object_date	Object date	DATE	01/01/1900	Y			
object_type	Object type	INT		N	Numeric only	FK	Object_ type
language	Analyst zip code	VARCHAR (45)		N			
notes		VARCHAR (45)		N			

6.2 System Catalog

A system catalog is the compilataion of objects within the database such as tables, queries, views, forms and reports. The purpose of this catalog is to document each object, expected data volumes, and links from this object to others in the system. This will help educate the people that weren't part of the creation of this system in the future. This system catalog contains 8 objects, 2 each of the following 4 types of objects: tables, forms, queries, and reports.

Object Name: images

Object Type: Table

Description: The images table is used to store metadata about the image.

Object Description:

Source: frmImages.

Destination: The images table will be used in all queries that retrieve image information.

Object Name: objects

Object Type: Table

Description: The objects table is used to store metadata about the objects in the database.

Object Description:

Source: frmObjects.

Destination: The objects table will be used in all queries that retrieve objects information.

Object Name: frmPeople

Object Type: Form

Description: Form that displays all people in the people table.

Object

Description:

Source: people table, qryPeople
Destination: This form (frmPeople) is used to maintain information about people in the database.

Object Name: frmImages

Object Type: Form

Description: Form that displays metadata with images in the image table.

Object Description:

Source: frmImages,qryImages
Destination: This form (frmImages) is used to maintain information about images in the database.

Object Name: rptImages

Object Type: Report

Description: Reports that displays a list of images and their associated metadata.

Object Description:

Source: frmImages,qryImages
Destination: This report (rptImages) is used to display a list of all the images found in the database. This allows staff to pull reports to check on metadata.

Object Name: rptObjects

Object Type: Report

Description: Reports that displays a list of objects and their associated metadata.

Object Description:

Source: frmObject,qryObjects
Destination: This report (rptObjects) is used to display a list of all the objects found in the database. This allows staff to pull reports to check on metadata.

Object Name: qryObjects

Object Type: Query

Description: Query that displays a list of objects and their associated metadata.

Object Description:

Source: frmObject,
Destination: This query(qryObjects) is used to display a list of all the objects found in the database. This query can be used to build reports with (rtpObjects).

Object Name:qryPeople

Object Type: Query

Description: Query that displays a list of people and their associated metadata.

Object Description:

Source: frmPeople
Destination: This query (qryPeople) is used to display a list of all the people found in the database. This query can be

used to build reports based on the people table (rptPeople).

7 Conclusion

In conclusion, the Digital Collections Database offers numerous upgrades to the older, outdated system currently in use. As discussed before, it offers benefits such as freeing up IT staff from doing bulk updates, allows staff to eliminate the use of spreadsheets and do the data entry into the database itself and also creates a more modern method for digital preservation that can be backed up and restored properly. The hope is that this project will put the library at the forefront of digital preservation, and encourage more donations to our collections.

8 Lessons Learned

The creation of this project allowed me to learn many valuable lessons that I hope to use in future projects. The 5 lessons learned from the Digital Collections Database were:

1. It is important recognize your project goals. I started off planning an all-encompassing collections database that would work for multiple collections. I later found I would have to scale down what I wanted for this database to complete the project on time. Ultimately I think the database might have hit performance issues by being a central database for all collections.

2. It is important to map out the database design through an ER diagram first. I had a rough diagram built, but I'll admit I started building tables without thoroughly analyzing it. This caused some mapping issues between fields types that were PK/FK's.

3. Piggy backing on #2, it is important to complete a project's phase before finishing the first one. Thoroughly finish your objective for that phase, and then start the next phase.

4. It is important to understand what constraints you want to place on data entry. The quality of data is so important to the database. Bad data can cause all sorts of issues in query building and reports. For example, if I didn't require a first name in my People table then a librarian could add J. Smith, John Smith or some other variation creating a new record for a person that already exists.

5. It is important to understand your own limitations when taking on a project such as this. I initially felt just doing an Access database would be the safe choice and offer the least resistance for the requirements I faced. Instead, I went MySQL and was unable to get my forms built in time. SQL Server I could have used Visual Studio to produce some forms rather quickly, Ascess has everything built in, while MySQL requires a little more programming. I had prototype forms built early but there was simply too much coding

involved for the time I had. I was able to compromise and link the MySQL tables to Access and build the forms and reports that way but, lesson learned.

9 Sources

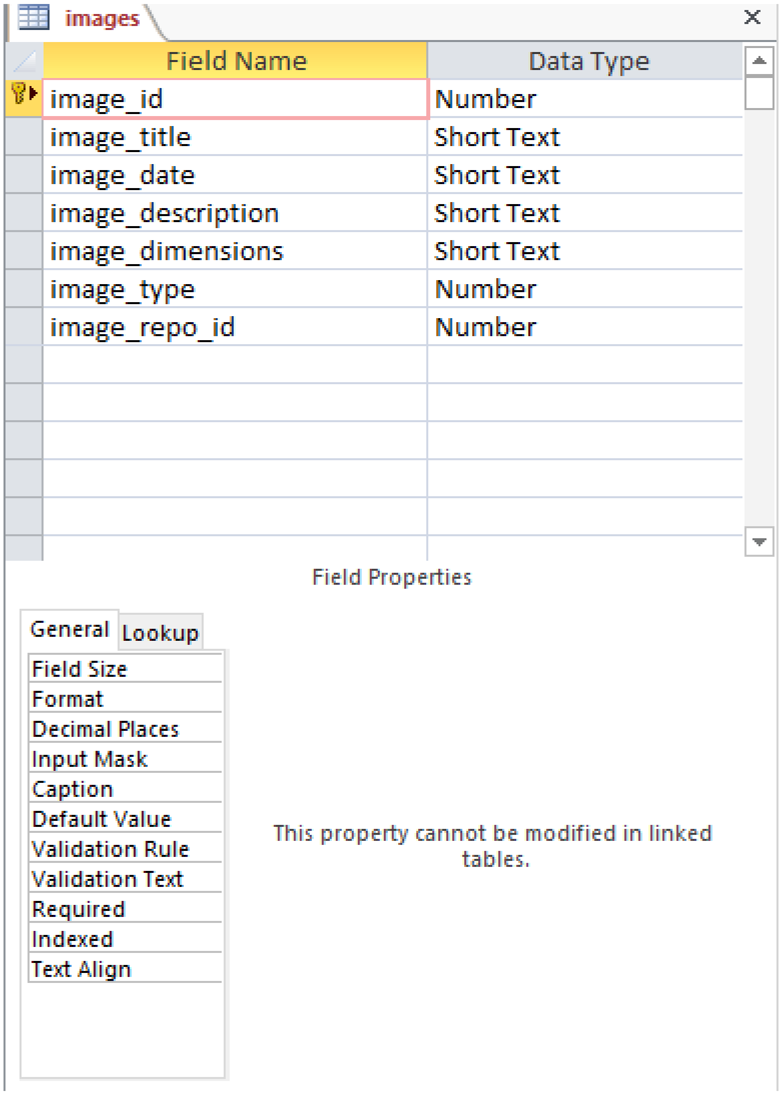
Rouse,Margaret (n.d.). In SearchCIO. Retrieved from http://searchcio.techtarget.com/definition/Prototyping-Model

10 Appendix

This Appendix shows a design view of all the tables in the Digital Collections Database and a description of their purpose:

image table

This table holds all of the metadata regarding images being stored.



images

Field Name	Data Type
image_id	Number
image_title	Short Text
image_date	Short Text
image_description	Short Text
image_dimensions	Short Text
image_type	Number
image_repo_id	Number

Field Properties

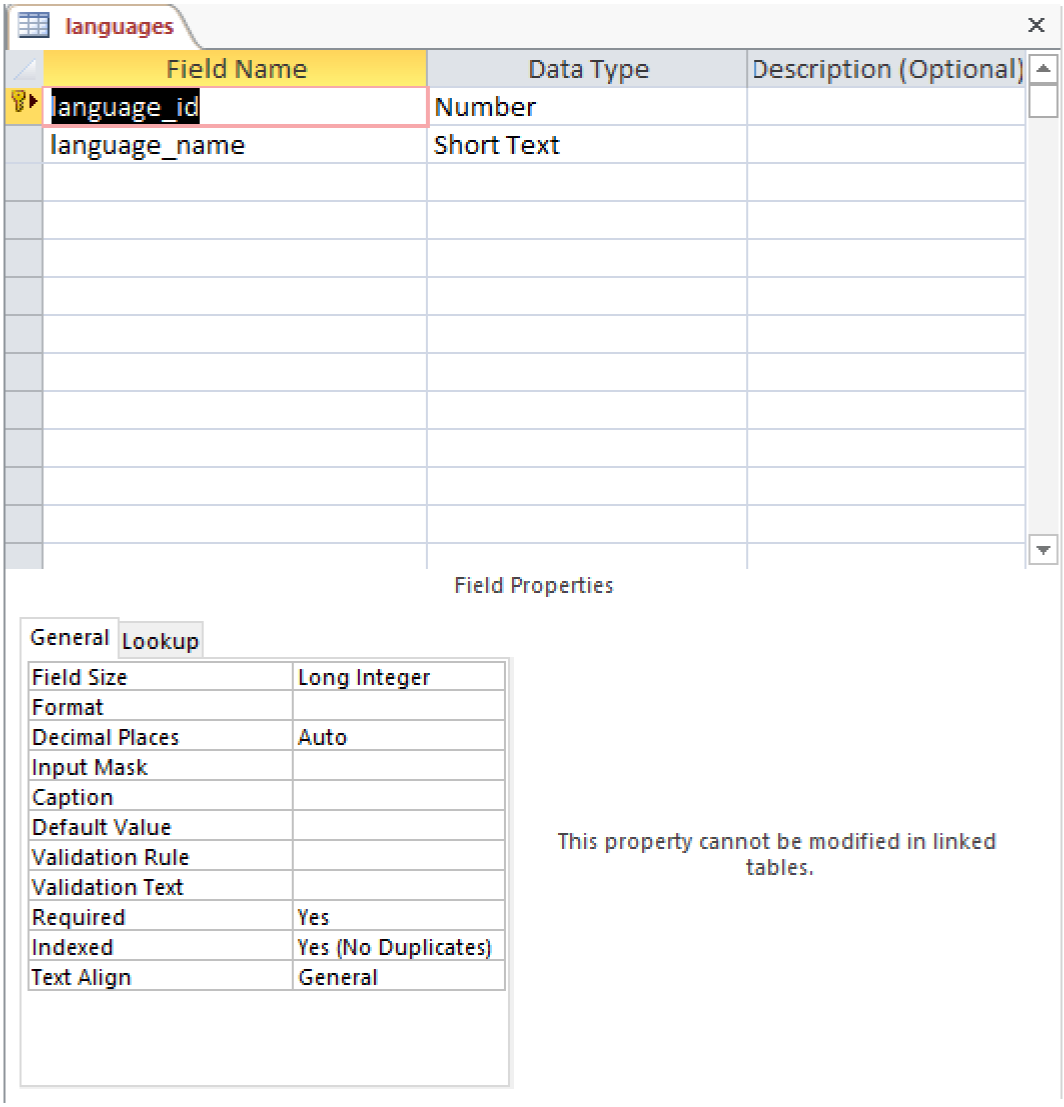
General Lookup

Field Size
Format
Decimal Places
Input Mask
Caption
Default Value
Validation Rule
Validation Text
Required
Indexed
Text Align

This property cannot be modified in linked tables.

languages table

This table holds the languages currently supported by this system. It's linked to the objects table to make it easier for user's to enter the languages.



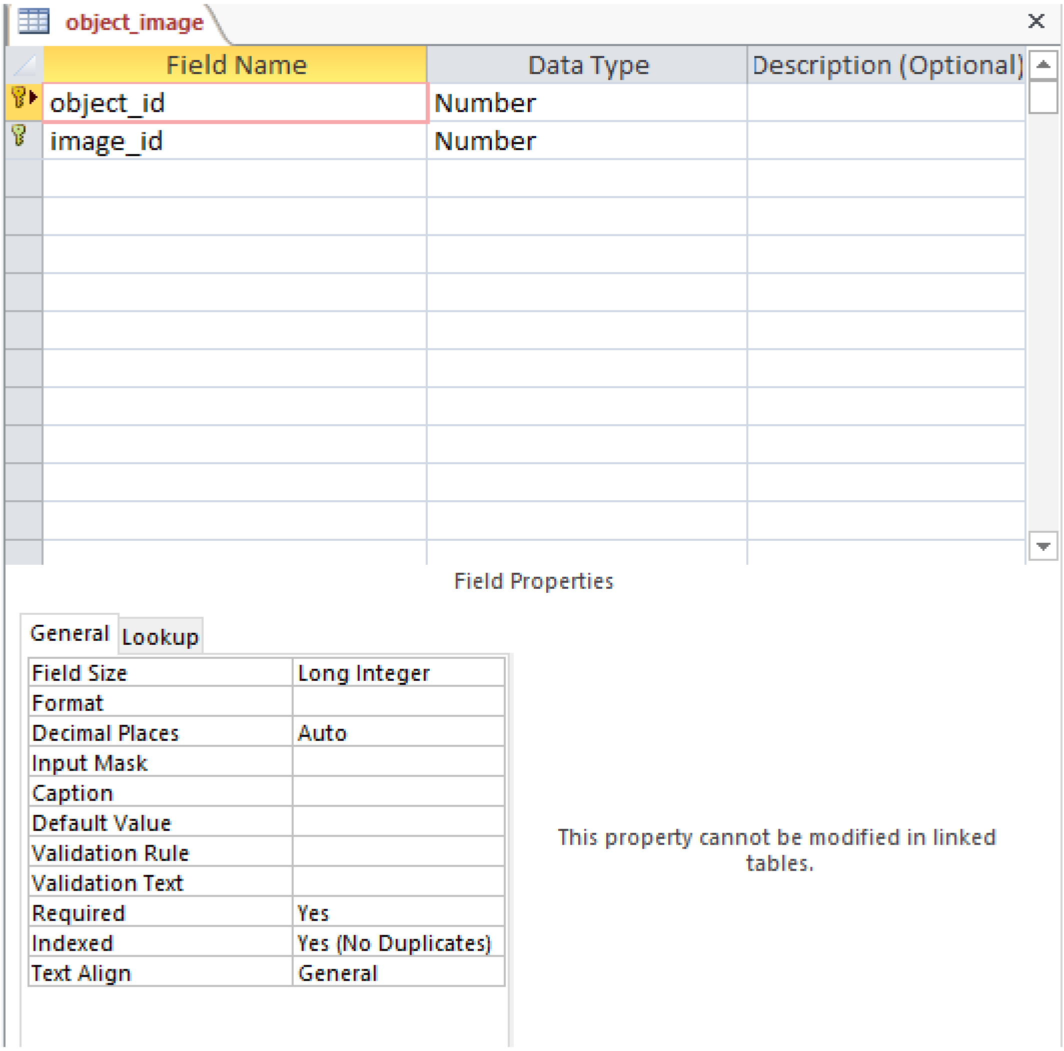
Field Name	Data Type	Description (Optional)
language_id	Number	
language_name	Short Text	

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

object_image table

This table links the object and image table, if an image happens to be part of a larger object, for example.



Field Name	Data Type	Description (Optional)
object_id	Number	
image_id	Number	

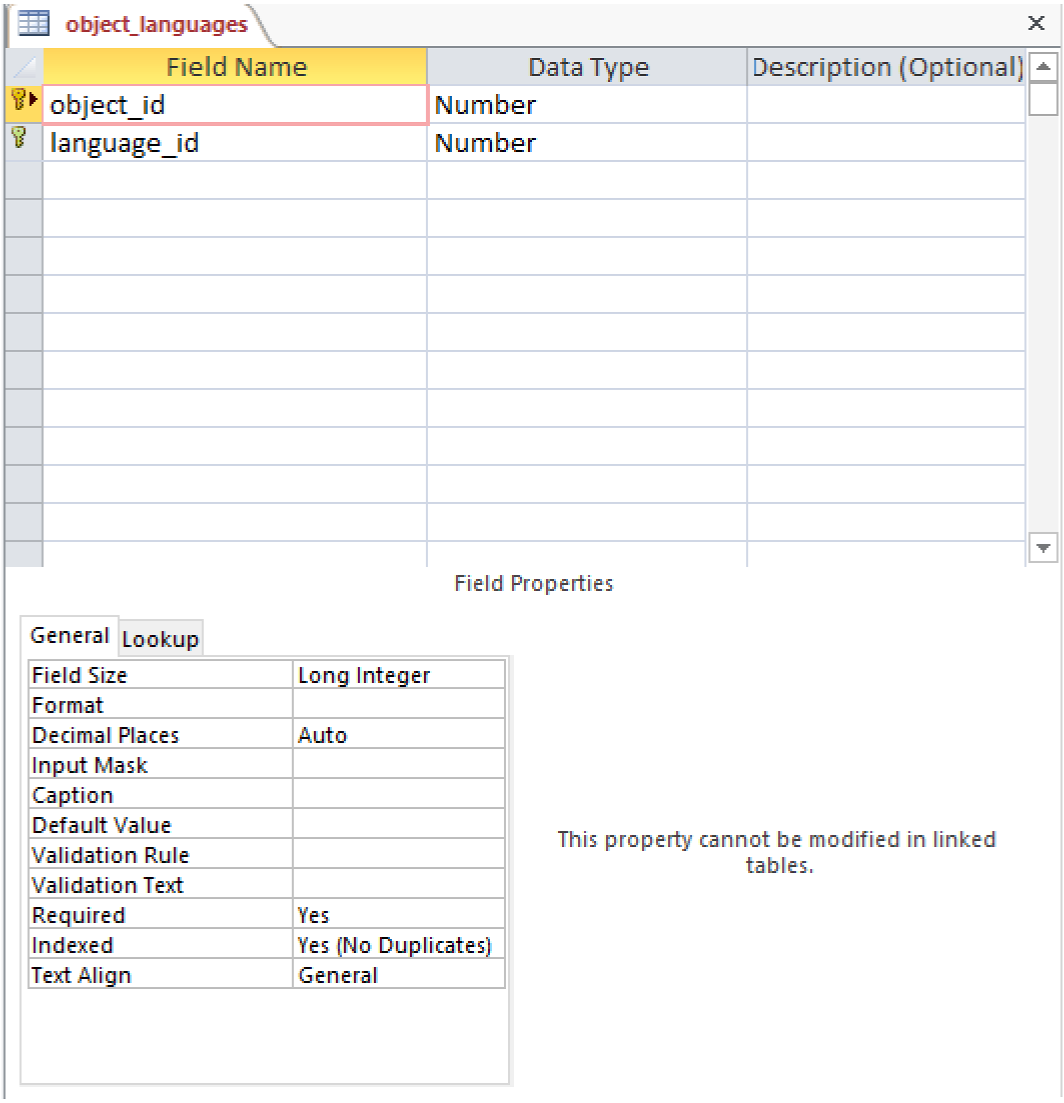
Field Properties

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

object_languages

This table links the objects and language table together.



Field Name	Data Type	Description (Optional)
object_id	Number	
language_id	Number	

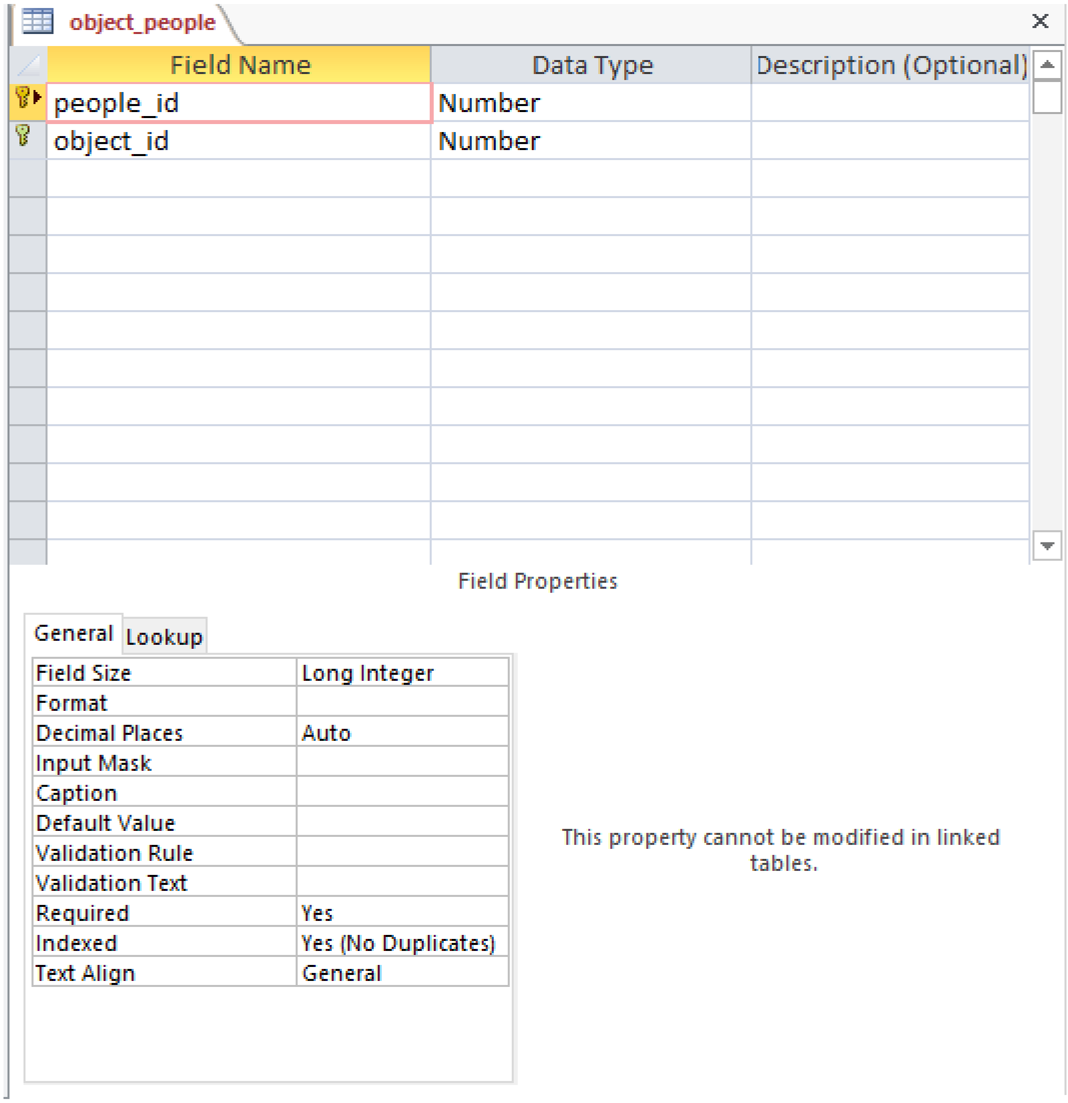
Field Properties

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

object_people

This table would link an people that might be associated with an object.



Field Name	Data Type	Description (Optional)
people_id	Number	
object_id	Number	

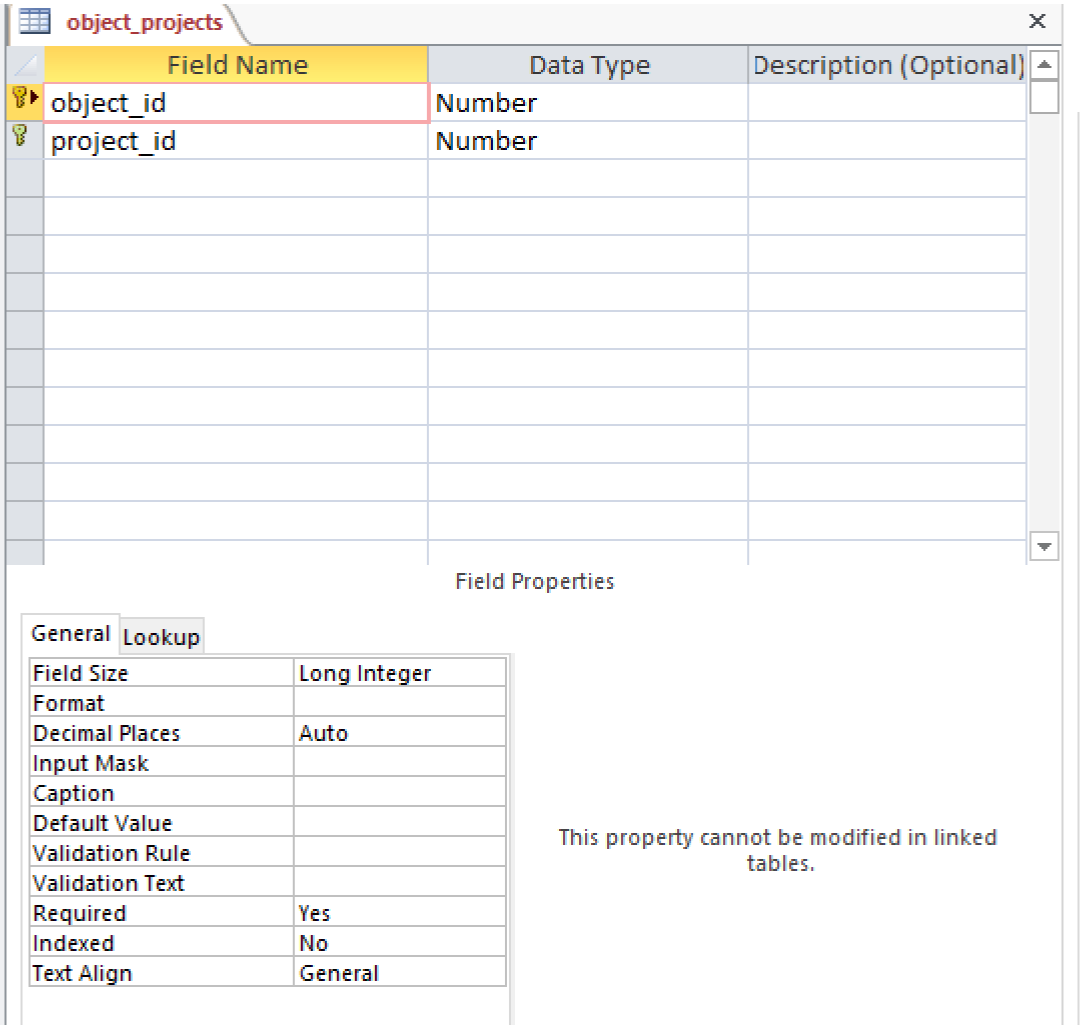
Field Properties

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

object_projects

This table would link any projects that belong to an object.



object_projects

Field Name	Data Type	Description (Optional)
object_id	Number	
project_id	Number	

Field Properties

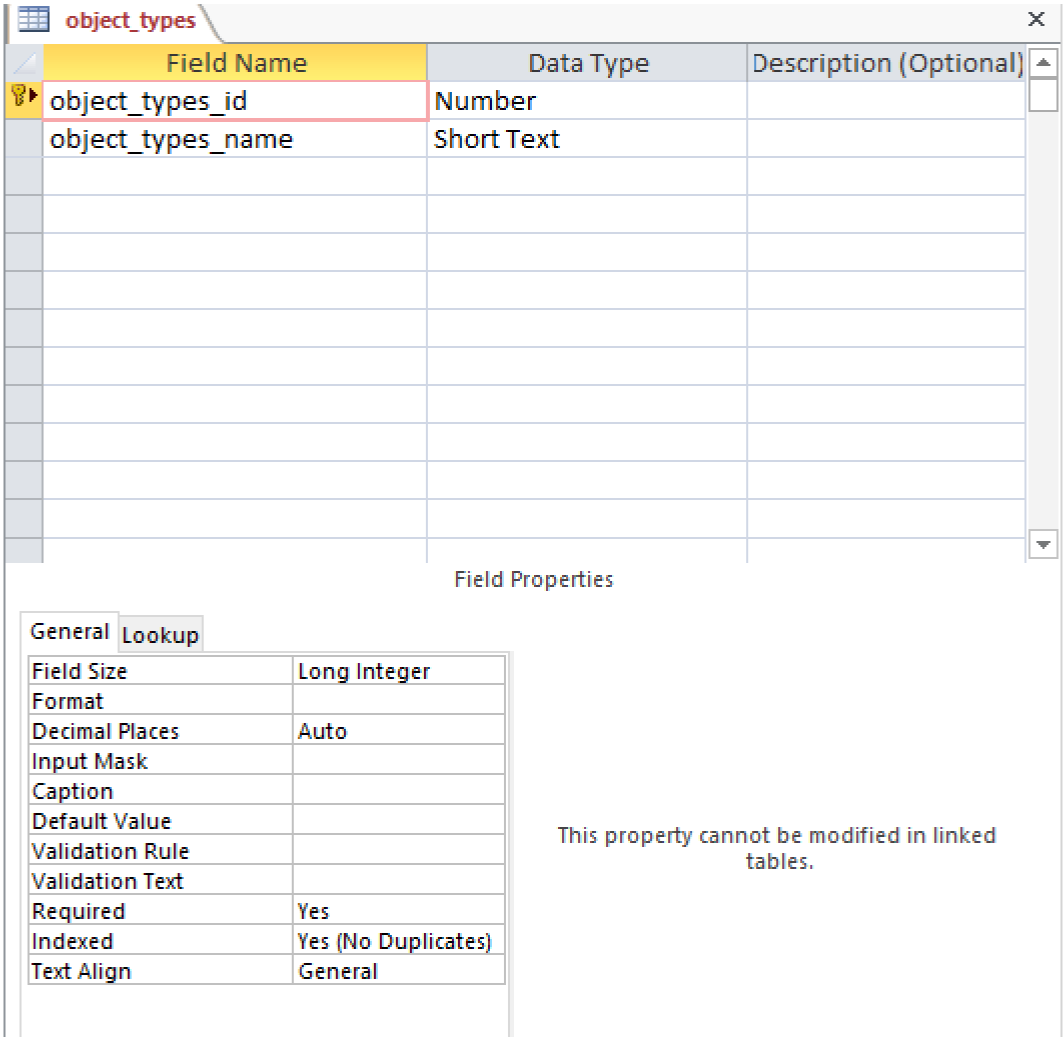
General Lookup

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	No
Text Align	General

This property cannot be modified in linked tables.

object_types

This table links objects to the different types of objects.



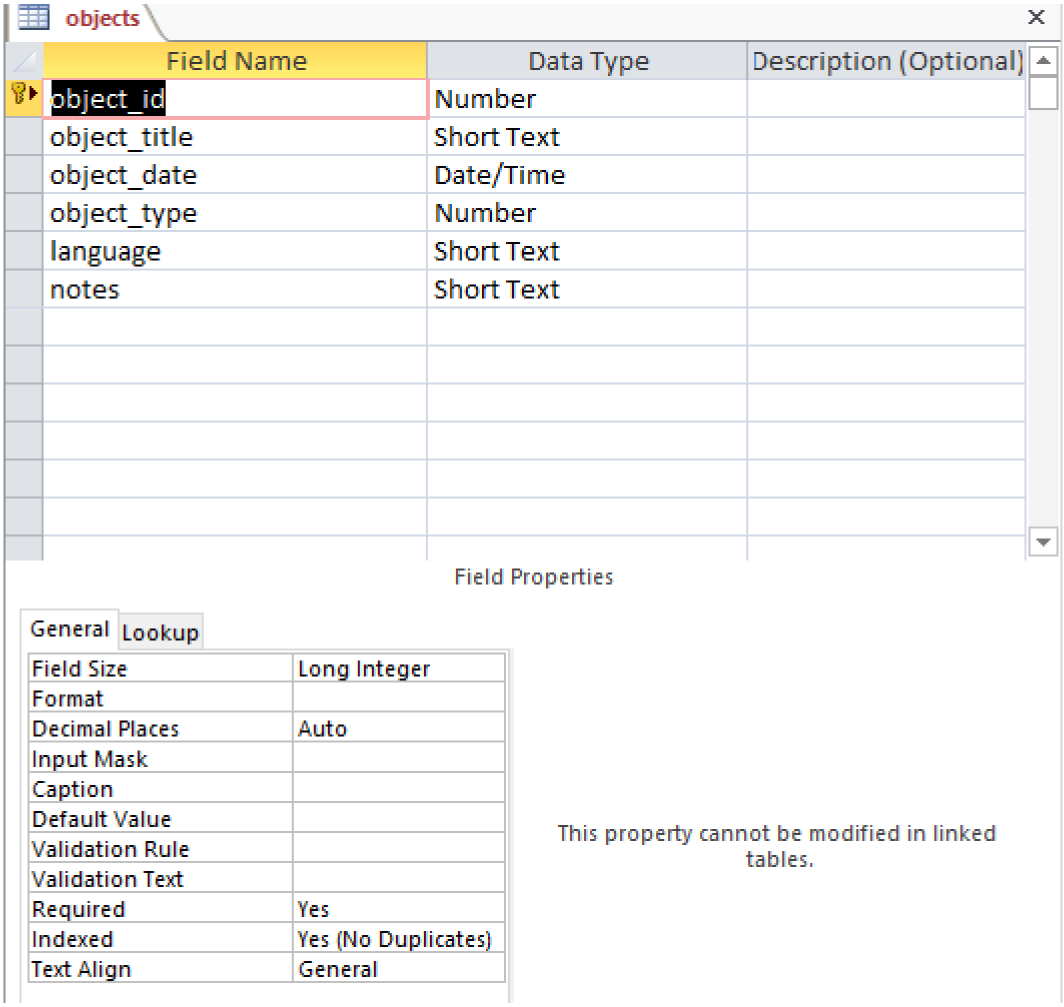
Field Name	Data Type	Description (Optional)
object_types_id	Number	
object_types_name	Short Text	

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

objects

This table stores metadata for a given object.



Field Name	Data Type	Description (Optional)
object_id	Number	
object_title	Short Text	
object_date	Date/Time	
object_type	Number	
language	Short Text	
notes	Short Text	

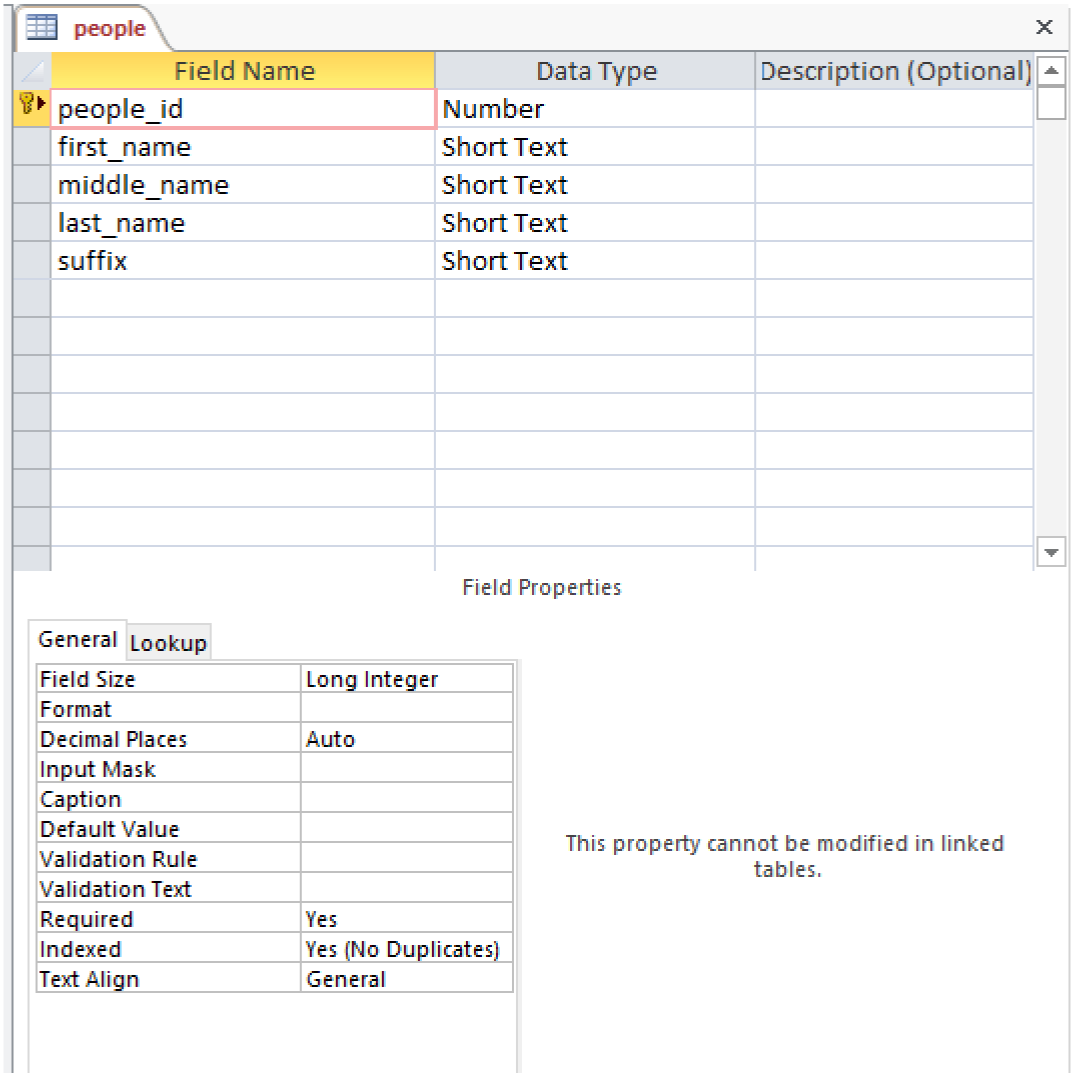
Field Properties

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

People

This table stores all the metadata about people in the database.



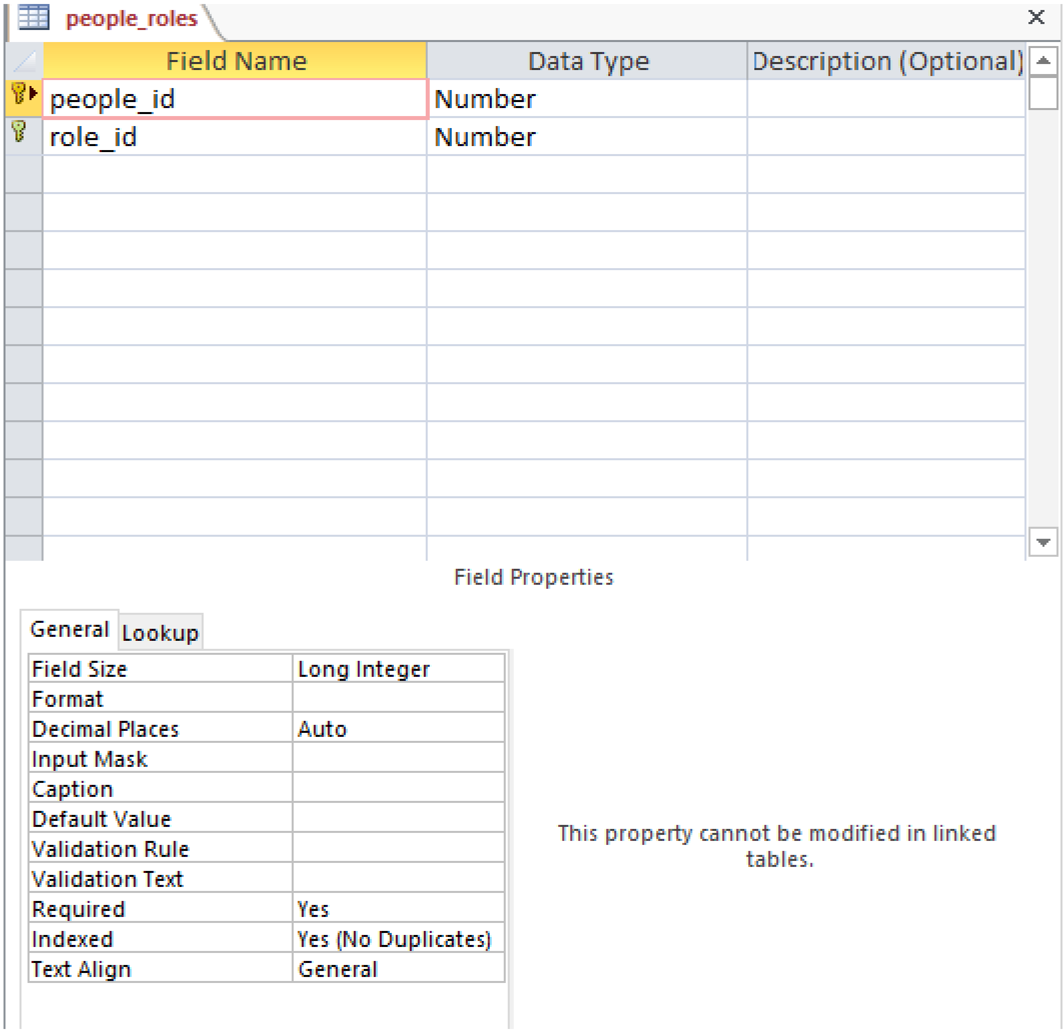
Field Name	Data Type	Description (Optional)
people_id	Number	
first_name	Short Text	
middle_name	Short Text	
last_name	Short Text	
suffix	Short Text	

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

people_roles

This table would link people with roles.



Field Name	Data Type	Description (Optional)
people_id	Number	
role_id	Number	

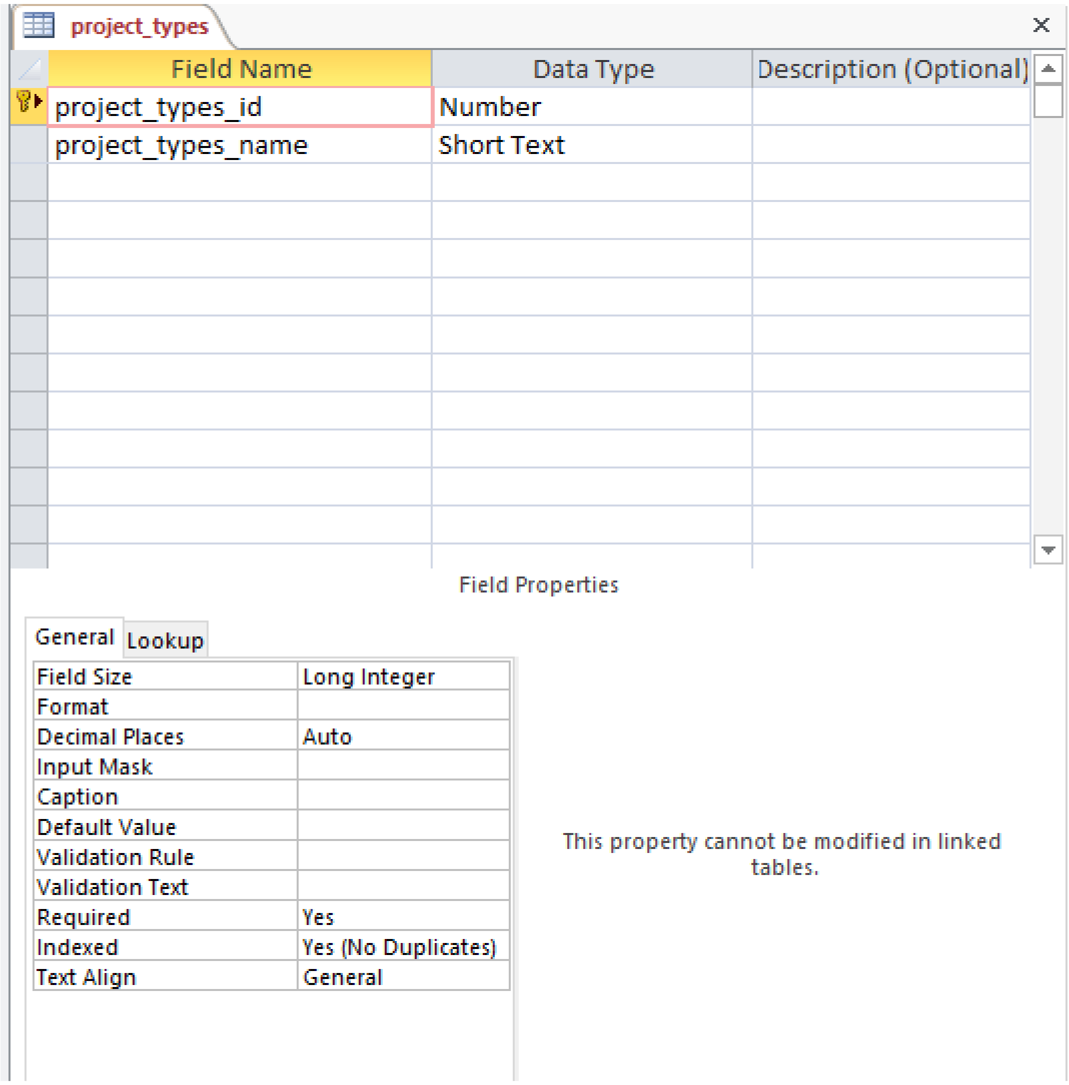
Field Properties

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

project_types

This table would link projects with project types.



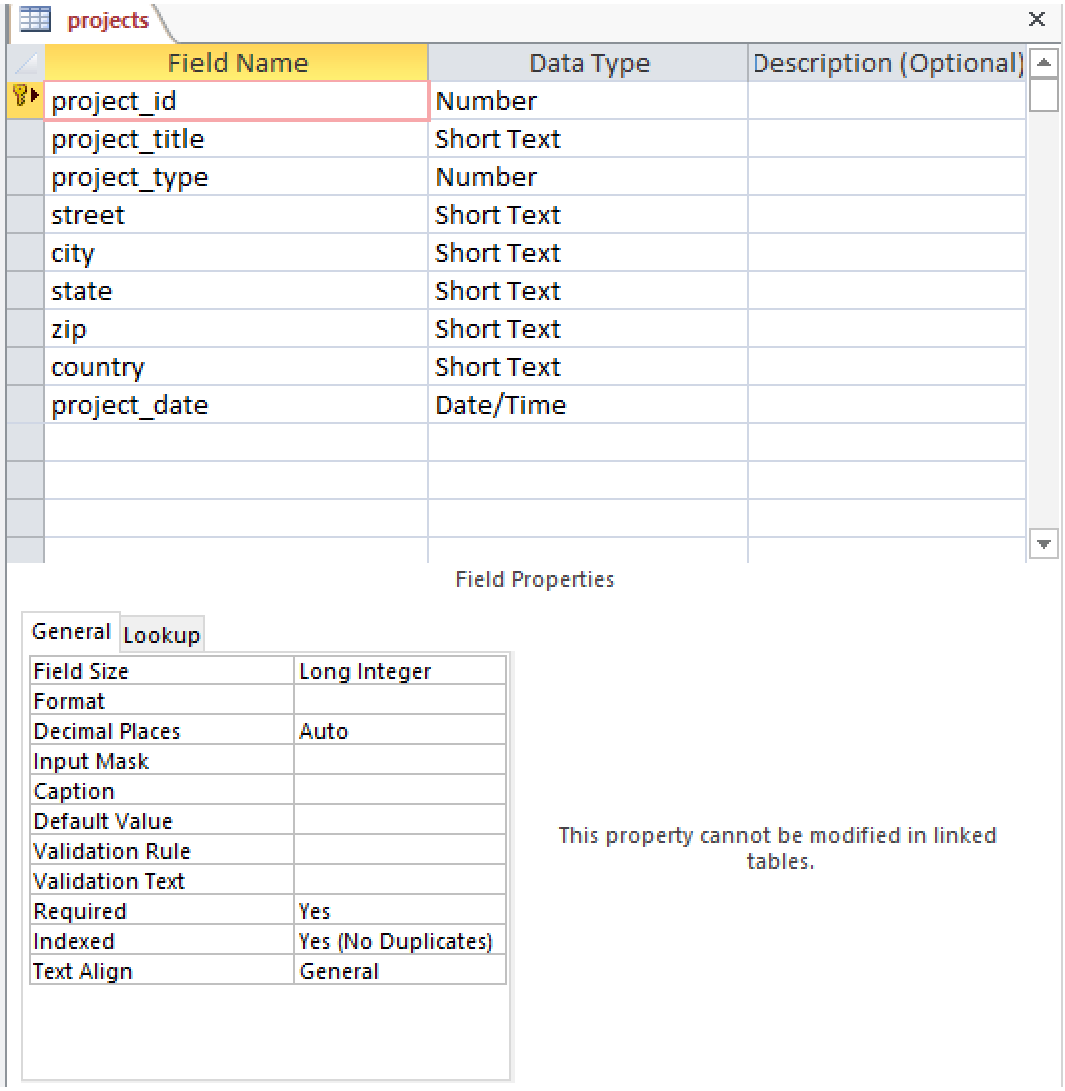
Field Name	Data Type	Description (Optional)
project_types_id	Number	
project_types_name	Short Text	

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

Projects

This table would store any project information.



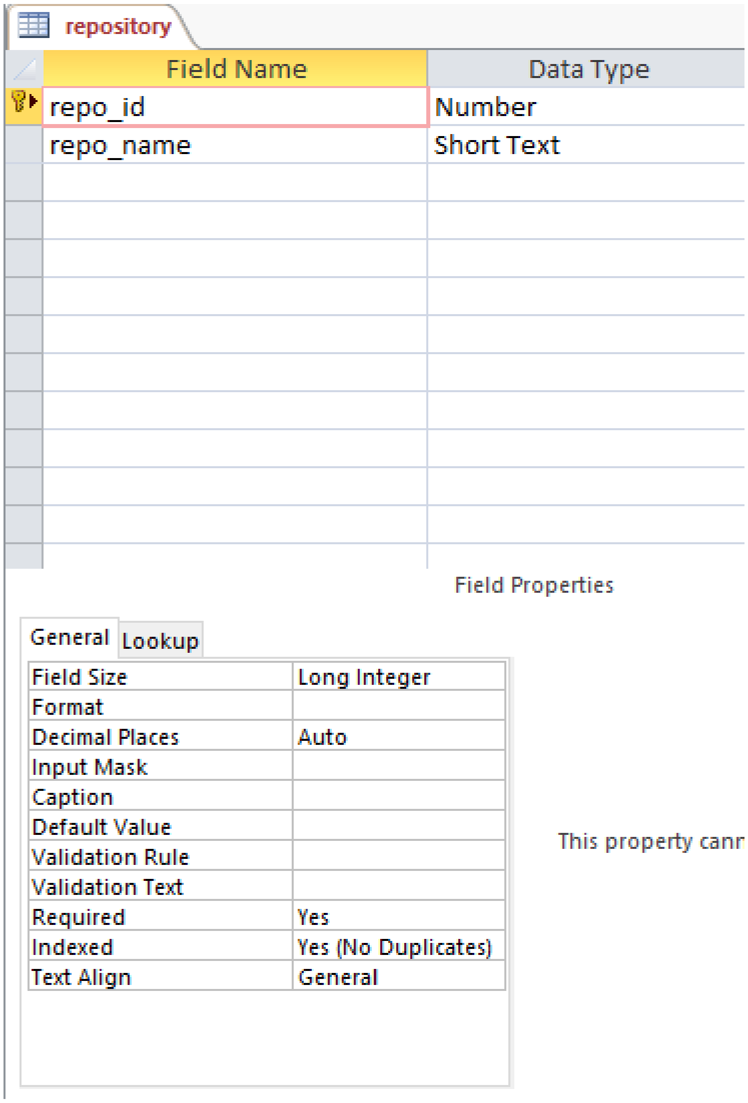
Field Name	Data Type	Description (Optional)
project_id	Number	
project_title	Short Text	
project_type	Number	
street	Short Text	
city	Short Text	
state	Short Text	
zip	Short Text	
country	Short Text	
project_date	Date/Time	

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

repository

This table would store the repositories that the objects belong in.



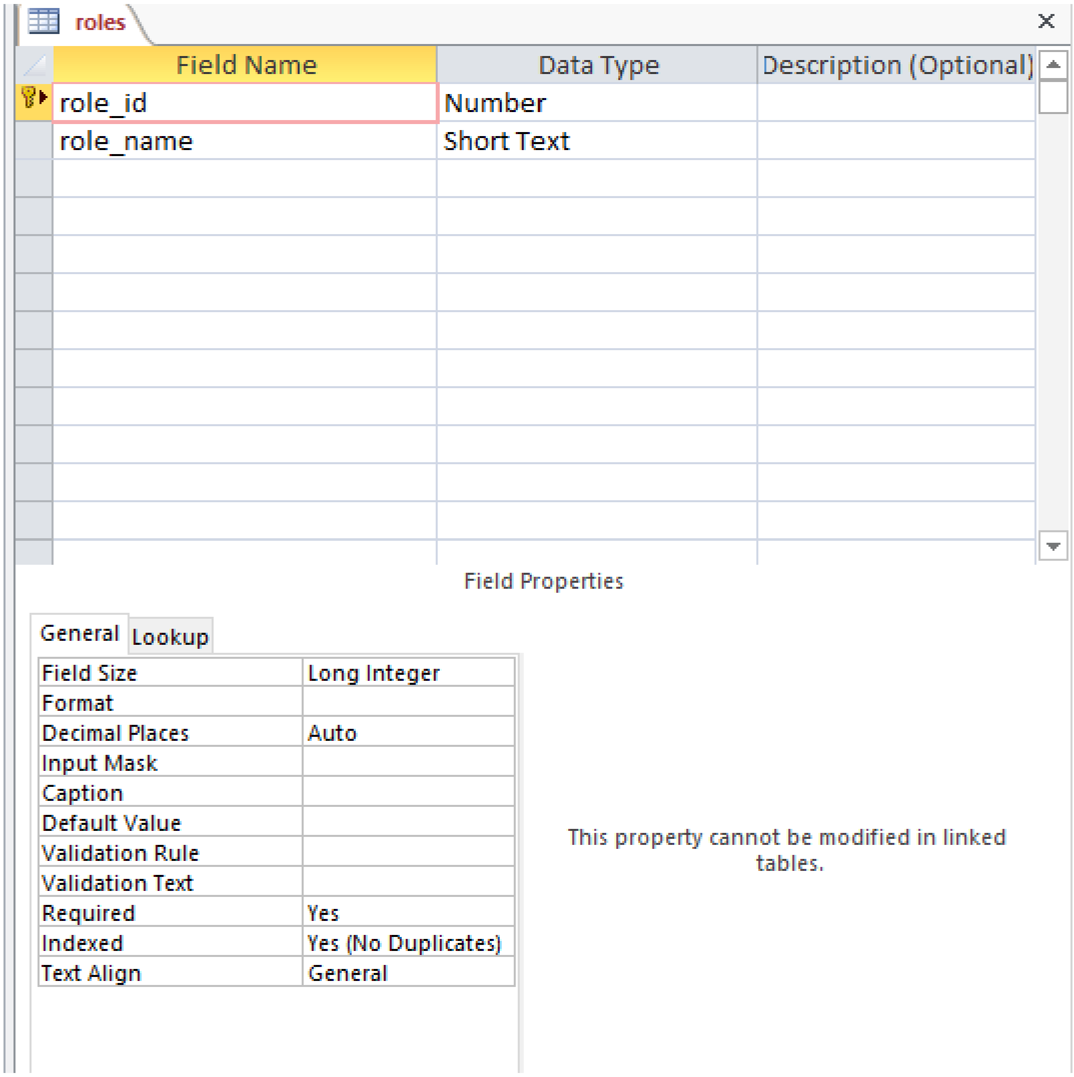
Field Name	Data Type
repo_id	Number
repo_name	Short Text

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property canr

roles

This table links peoples roles id with a role name.



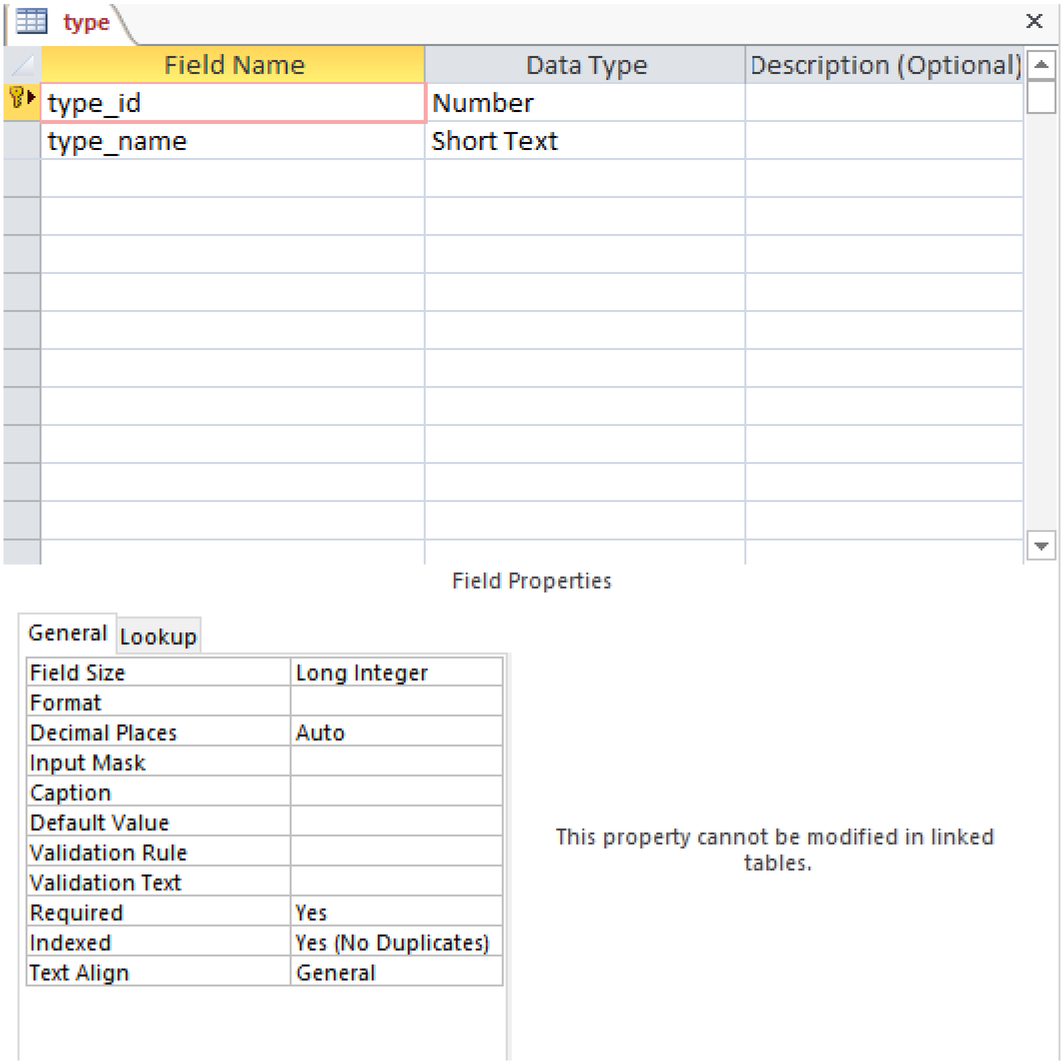
Field Name	Data Type	Description (Optional)
role_id	Number	
role_name	Short Text	

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.

type

This table links the image types with the type name.



Field Name	Data Type	Description (Optional)
type_id	Number	
type_name	Short Text	

Field Properties

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

This property cannot be modified in linked tables.