

CS 5800 Hw 2

Schema for Artists and Their Paintings

ALPHA: Mounika Arukala

BETA: Justin Hershberger

Overview:

The general idea of the database is to keep track of Artists, their Paintings, Galleries that the paintings are shown in, the medium of the paintings as well as their art style. Some of the relationships are: artists make paintings therefore a painting has an artist but an artist can have several paintings. Paintings have a medium and a style of art. Mediums are things like: charcoal, water color, oil, etc. Art Styles are the style most present in a painting such as abstraction, photorealism, or pointillism. Galleries show several paintings but can show paintings of one specific medium or style. Each of these tables is a separate entity but they all connect in one way or another. We chose the entity types that we did because they met a lot of requirements and contain some potentially unnecessary entity types.

Entity Types:

Artist, Paintings, Medium, Gallery that the piece is shown in, Art Style, Master, Apprentice, Company, Freelance

Entity Relationships:

The artists table will have information about the artist (id,name, age, gender, etc)

The Paintings table will have info about the painting. (painting_id, name of painting, artist name, year made, medium)

The Medium table will be a table of specific Mediums (charcoal, oil, etc.).

The Gallery table will be a table of art galleries.

The Art Style is a table for specific art styles such as (abstract, photorealism, etc.)

This is the description of each entity type:

Artist Table - Should only contain info about the artist

Artist_id | Artist_Name | Age | Gender | Country of Origin | Painting_id | company id | medium known for | Salary | Contact info

Description:

The Artist table will have the attributes: name, age, gender, country of origin, salary, contact info, and date of birth. Painting_id is a foreign key to an artist. The reason it is a key is because each painting has an artist but not all artists have a painting but if an artist does have a painting, we want to know which ones. Company id is also a foreign key because artist can have an employer or they could work for themselves but if they work for a company, there needs to be a way to tie the artist to the company. Medium id is also a foreign key because some artist tend to have a specialty with a certain medium. Constraints are that their age needs to be a valid age and name has to be a string.

Paintings Table - should only contain info about each painting

Painting_id | Artist_Id | Painting_name | Date Created | Medium | Art Style | Gallery id | Price at Gallery

Description:

The Painting table will have the attributes: name, date created, price. The primary key is the painting_id, it will uniquely identify each painting. The foreign keys are: artist_id because each painting will have an artist that painted it so the key can be used to access information about the artist, gallery_id because paintings can be shown in galleries and we may want to know which gallery the painting is shown in, medium_id because each painting has a medium and we want to know which one so this will give access to the medium's info in the medium table, art_style_id because each painting has an art style and we may want to access the info about the art style. The constraints are that price and date created have to be valid. Another constraint is that the date created has to correspond with the art style and when it was created.

Medium Table - Should only contain information about the specific Medium(oil, canvas, charcoal, etc)

Medium_id | Medium Name | Medium Description

Description:

The Medium table is a table for specific Mediums of art (charcoal, oil, etc) it will have the attributes: name and description. The primary key is the medium_id to uniquely identify a specific medium. There are no foreign keys. The constraints are that the medium name would need to be valid and that the date of creation on any painting tied to the medium would need to be after the medium was founded.

Art Style - Should only contain information about the specific art style

Art Style id | Name of style | Era of Art Style (years in which the style was popular) | medium_id
| gallery_id | artist_id

Description:

The Art Style table is for specific Art Styles such as (abstract, pointillism, photorealism, etc.). The attributes are: name, and the era the style was founded. The primary key is the Art_Style_id to uniquely identify each Art Style. The foreign keys are: medium_id because certain styles rely on certain mediums and we might want to know more about that medium or vice versa, gallery_id because galleries can sometimes exclusively show one style, if this is the case, we might want to know which galleries show which art styles. The constraints are that the date of creation on the paintings need to be accurate to the era the art style was founded/was popular. Also, if an artist does only do specific styles, the data needs to accurately show which styles.

Gallery - Should only contain information about the Gallery

Gallery_id | Gallery name | Gallery location(country/city/state)

Description:

The gallery table is for information about galleries that show paintings. It's attributes are: name and location. It's primary key is gallery_id to uniquely identify each gallery. There are no foreign keys. The constraints are that the gallery location must exist and the name must be accurate. If a gallery shows one artist, medium, or art style then this needs to have accurate data in the table. The paintings that are in the gallery need to be accurate as well since if we wanted to go see the mona lisa but the data was wrong, we would be frustrated.

Company

Company_id | Company name | artist_id | painting_id

Description:

The company table is for companies that have artists as employees and divide out paintings that the company has been hired to do. The attributes are the company name. The primary key is the company_id so that each company can be uniquely identified. The foreign keys are: artist_id because each company can have several artists working for them and we may want to know more info about these artists, painting_id because each company can be responsible for several paintings by multiple artists so this way there is a way to see which paintings the company did. The constraints are that the artists listed and paintings listed need to actually work for or be done by artists employed by the company.

Apprentice

Apprentice_id | Apprentice Name | Artist_id | master id

Description:

The apprentice table is a table for understudy artists, they are apprenticing under another artist in hopes of being a master artist one day. The attributes are the apprentice's name. The primary key is the apprentice_id to uniquely define each apprentice. The foreign keys are: the artist_id because in order to be an apprentice, you also need to be an artist, master_id because there needs to be a master artist the apprentice studies under. The constraints are that artist_id can't be null in this table as well as master id.

Master

Master id | Name | Artist_id | Apprentice_id

Description:

The master table is for artists who are considered masters of their craft. They are still an artist but possess a higher skill level than most. They also have apprentices that learn from them. The attributes are the name. The primary key is the master id which uniquely identifies each master artist. The foreign keys are artist_id because each master is an artist and is represented in the artist table, and apprentice_id because each master has an apprentice artist studying under them. The constraints are that each master needs to be an artist and each apprentice must exist.

Freelance

Freelance_id | Freelance Name | Contact info | Artist_id | Painting_id

Description:

The Freelance table is a table for the situation when an artist takes on a job in their freetime. This is an agreement between the artist and one other person. This is a one to one relationship. The attributes are the person hiring the artist's name and contact info. The primary key is the freelance_id to uniquely identify each freelance employer. The foreign keys are: artist_id because there needs to be an artist that is hired in his free time, and painting_id for the painting that the artist has been hired to do. The constraints are that the contact info must be correct, the artist must exist, and the artist can't be too busy with actual work to take on a side job.

Note: To view all the paintings in a specific gallery, we would need to use joins on the foreign keys.

In order to have data integrity, we don't want unnecessary data in tables. A gallery table should only have data about a gallery in order to maintain data integrity and a self-describing model.

Relationship Types:

- 1) A Gallery is related to an Artist by displaying a painting

Description:

Paintings are shown in a gallery, paintings have an artist and because of this when a painting is shown in a gallery, the gallery is then related to the artist of the painting. The attributes of this are: gallery name, artist name, and painting name because the gallery shows the painting with the name on display and because of this, the gallery is then related to the artist. The constraints are that the gallery has to display the painting in order to be related to the artist. This is a many-many relationship with partial participation on both sides because many galleries show many paintings which in turn have many artists. It is partial because not every painting by an artist is in one gallery, due to the artist either not wanting to display a work or wanting to divide up the pieces they've done between several galleries. Each gallery shows part of the artist's paintings but not each one.

- 2) An Artist is related to a medium by painting with that medium

Description:

Artists create paintings using a specific medium. They are then related to a medium by painting a work with that medium. The attributes are: artist name, medium name, as well as painting name because these are what are tied to each other via the artist painting with a medium. The constraints are that in order for the relationship to exist, the artist will have to make one painting with the medium. This is a many to many relationship generally but can be one to one or one to many. Many to many would be the most common because most artists expand their tools and use several mediums and mediums can have several artists who use that medium. The one to one is a rare case but occurs when one artist uses a medium that no other artist uses, this has total participation on both sides but almost certainly won't occur. The one to many relationship is when many artists specialize and only use a specific medium.

- 3) An Artist is related to an Art Style by painting in that art style

Description:

This relationship is the same as the medium. An artist is related to an art style and vice versa when an artist creates a work using a specific style. The attributes of this are the artist name, age, art style name, and art style era because the names are how you can see the relationship and the era/age can correlate, meaning if an artist grew up in a time where a certain style was popular then they are more likely to use that style. The constraints are that the style needs to be created before an artist can be related to it as well as the artist needs to paint using

the style. This is a many to many relationship with some rare cases where it is one to one or one to many. The reason it is many to many is because many artists paint using different art styles and many art styles have many artists they are related to. The rare cases are when a sole artist paints only using a style that no one else has ever used or when many artists use only one specific style and nothing else.

4) A company is related to a painting by an giving an employee artist the job to do it

Description:

Paintings have artists and their is an obvious direct relationship between them but if an artist works for a company and the company was hired to complete a painting and that task was assigned to an artist, then the painting is related to the company by the artist painting it for them. The attributes are: painting_name, artist_name, and company_name. The constraints are that the company must have been hired to complete the painting in order to be related to the painting. Just because the artist works for them, if they paint something in their free time then the company is not related to that painting. This is a one to many relationship as one company can be hired to complete several paintings but no other company can be related to those paintings. This has partial participation from the company and total participation by the painting because each attribute of the painting is related to one aspect of the company, the company name.

5) An apprentice is an artist

Description:

Apprentices are understudies to master artists but they are still an artist and possess all of the attributes of the artist table: name, contact info, salary, age, gender, etc. The constraints are that the apprentice needs to have an id in both tables.

6) A master is an artist

Description:

Masters are artists and possess all of the artist attributes in the artist table.

7) An artist is related to a company by employment

Description:

This relationship is a common one. A company is related to an artist by being their employer and an artist is related to a company by being their employee. The attributes are: artist name, salary, painting_name(s), and company name because the company pays the artist for completing paintings. The company is hired by other companies to complete works of art. The constraints are that the artist can only get paid if they complete paintings and that they are in an agreement for employment. This is a one to many relationship as one company has many artists that work for them. This is a

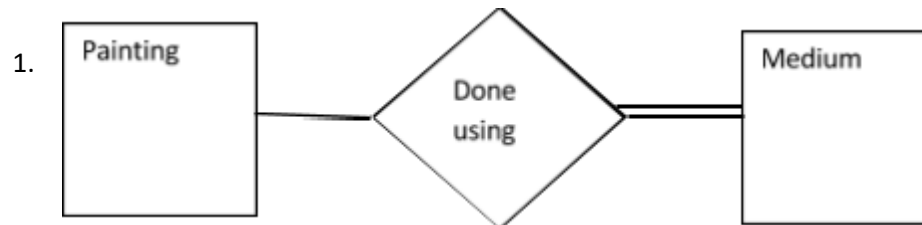
partial participation on both sides because artists can do paintings for the company but it may not be every painting they've ever done and the company gives each artist jobs to complete but it isn't every job they are hired to do.

8) An artist is related to a freelance by taking on a side job in their free time

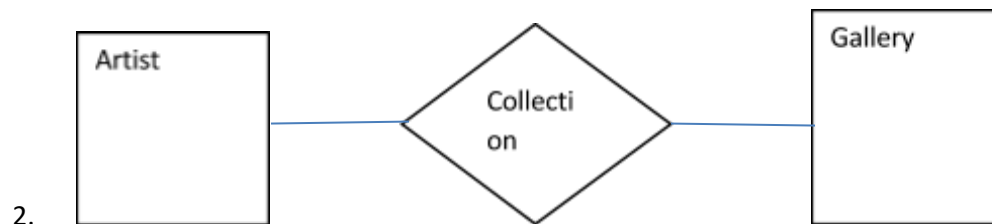
Description:

This relationship is when an artist is sought out during their free time to complete a side project for someone. The attributes of this are the person hiring the artist for the project's name, the artist's name, and the agreed upon price. The constraints are that the artist must be able to complete the project without interfering with their company's work. This is a one to one relationship with partial participation because the artist has other works that they are working on or have completed. Not every painting the artist has done is related to the freelance employer.

Alpha Section

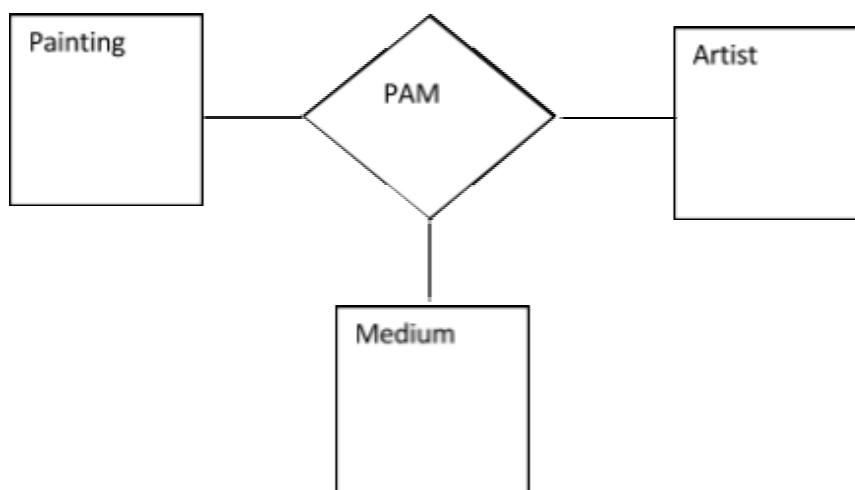


Many paintings can be done using a medium and so this satisfies **many to one Relationship**. Also, an artist cannot paint without a medium and so Medium has **total participation** and Painting has **partial participation**.

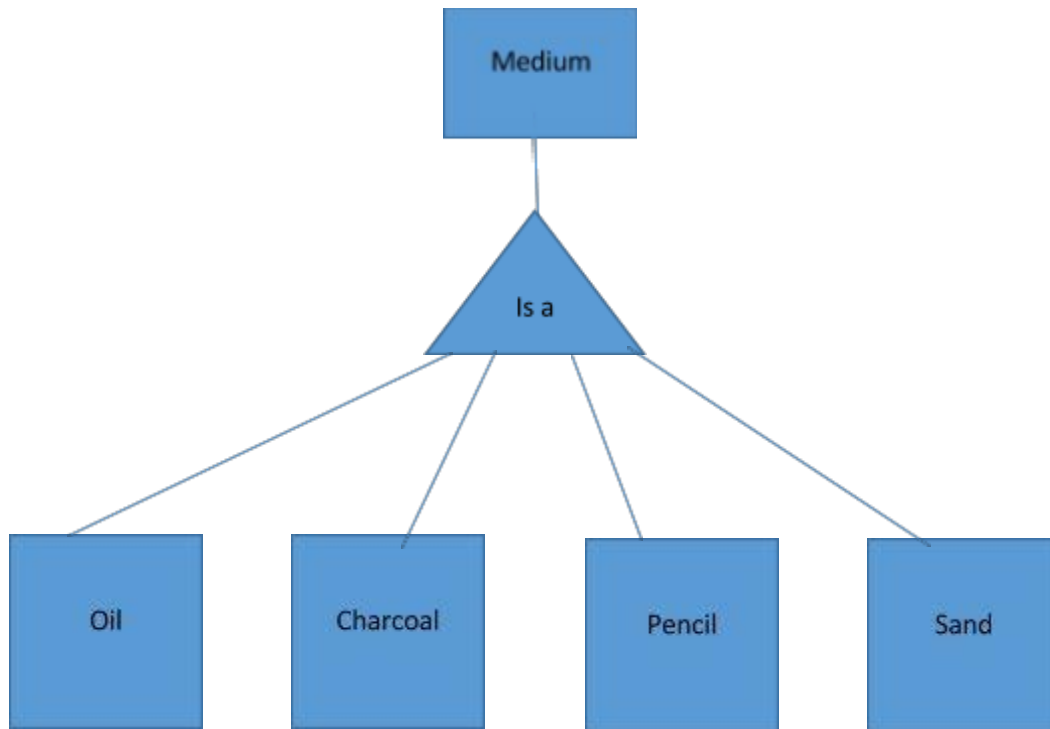


Many artists have their collections displayed in many galleries and thus the Artist and Gallery entities are satisfying **Many to Many Relationship**.

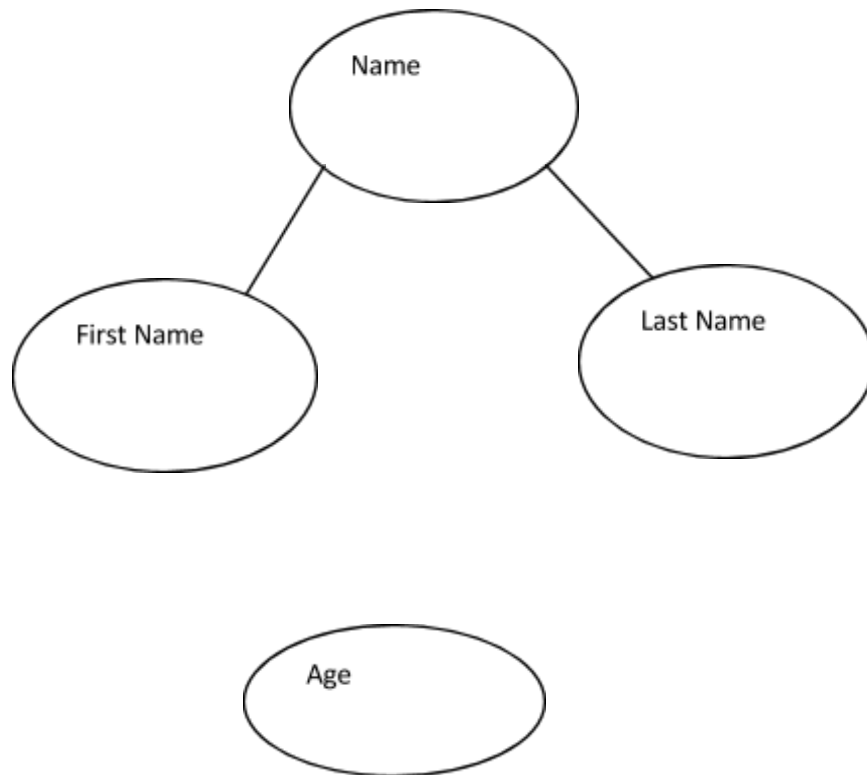
3. Ternary Relationship exists among Painting, Artist and Medium entities. Single artist can do a painting using a medium. An artist can use a medium to do multiple paintings. One painting can be done using a medium by multiple artists.



4. Here the Medium chosen to paint can exclusively either be oil or charcoal or pencil or sand but not a combination among cannot happen. To be clear, once it is oil it can't be the other three anymore. Hence this satisfies the Disjoint Inheritance hierarchy.



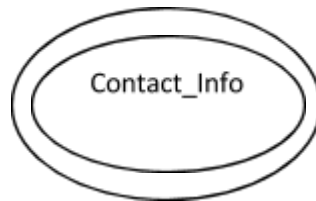
5. Name in the Artist Entity is a **COMPOSITE ATTRIBUTE** since it can be sub divided as **FIRST NAME** and **LAST NAME**.



6.

Age attribute of Artist Entity is the constraint. Excellence comes with experience and so are just concerned about artists with **Age >15**. This is the domain constraint we laid in our schema.

7.



Artist entity has an attribute called Contact_Info which can take many values for different artists and so it is a **Multi valued attribute**.

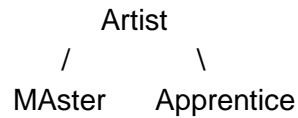
BETA section:

- 1) One to one with total participation on one side:
 - a) Artist to Apprentice where every apprentice must have an artist via being chosen as an apprentice. Total participation by the apprentice
- 2) One to many with partial participation on both sides:
 - a) Companies can have many artists but artists can work for one company
 - i) Both sides participate partially. An artist will do their job for the company but may also do freelance work. Companies will give an artist a job at a time but will not give out all the jobs to an artist. These are both partial participation.
- 3) Relationship with attributes:
 - i) An Artist painted a painting with the attribute Date Created
This attribute describes when the painting was completed by a specific artist. The painting is related to the artist with this attribute.
 - ii) A painting has been in a Gallery since a date
This relationship is when a gallery decides to display a specific painting. The relationship has the attribute of the date it first went on display.
 - iii) An artist is paid money by a company

This relationship is the employment relationship. An artist's salary is an attribute that describes the relationship they have with the company.

4) overlapping inheritance hierarchy

Both master and apprentice are an artist. Both entities possess each attribute present in the artist entity type. This is overlapping inheritance because an object could be a master or an apprentice but still be an artist.



5) A weak entity type

Apprentice is a weak entity type because without a master artist, the apprentice doesn't exist

Freelance is a weak entity type because this situation might never happen

6) A reflexive relationship

Every artist has the attributes: married to, and friends with. Artists can find love with other artists as well as find friendships with other artists. This is a reflexive relationship.

7) Check for integrity.

An artist name must be a varchar

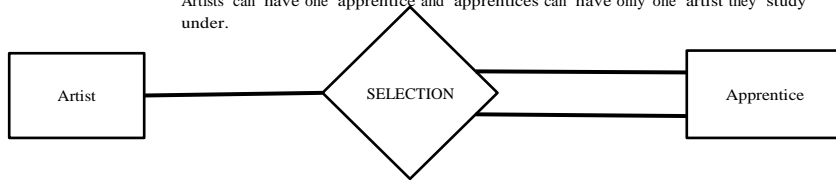
A painting must have an artist

Paintings made before an art style can't be that art style

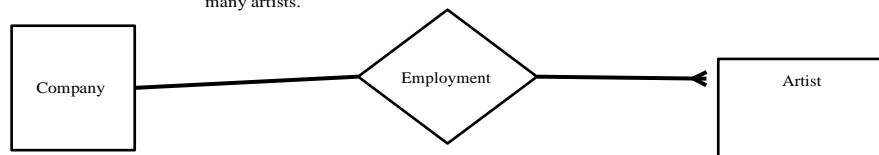
An apprentice needs to have a master

An artist can't paint a painting on a date before they are born

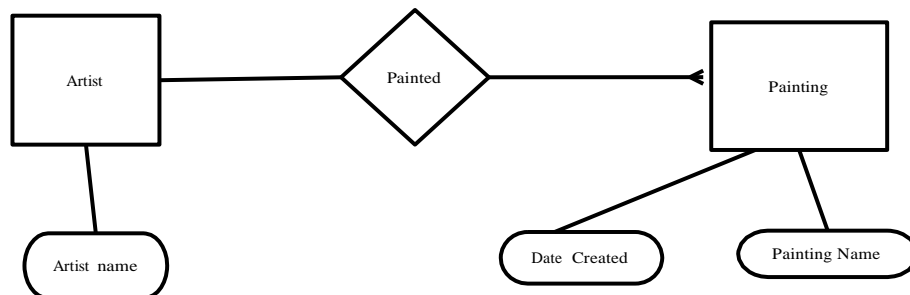
One to one with total participation by the Apprentice. An apprentice must have an artist they study under otherwise the apprentice doesn't exist. This requires total participation by the apprentice in this relationship. Not every artist has an apprentice so they participate partially in this relationship. Artists can have one apprentice and apprentices can have only one artist they study under.



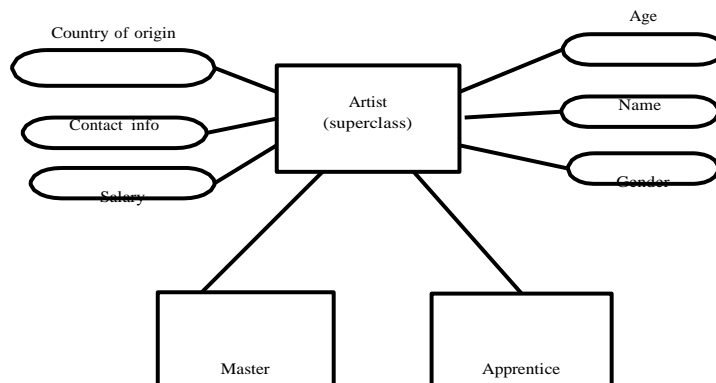
One to many with partial participation on both sides. Both company and artist participate partially with each other as the company has the artist's name and only the painting id's that the artist painted while working for the company. It isn't every single painting the artist has ever done. An artist is tied to a company by a salary and jobs/paintings they are paid for. Artist can work for one company and company's can have many artists.



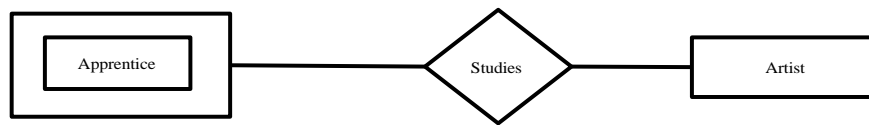
An Artist is related to a painting, this is a one to many relationship as an artist can paint several paintings but each painting can have only one artist. This relationship has attributes of date created, artist name, and painting name.



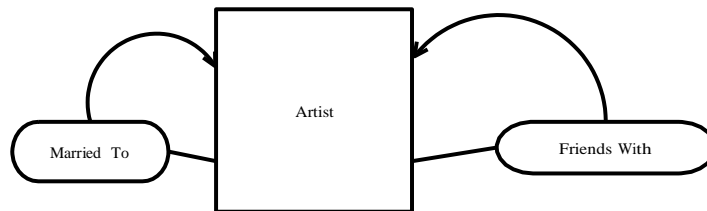
The overlapping inheritance hierarchy. This is when a superclass has multiple subclasses that share the same attributes. Both Master and Apprentice are subclasses of Artist. This is overlapping inheritance.



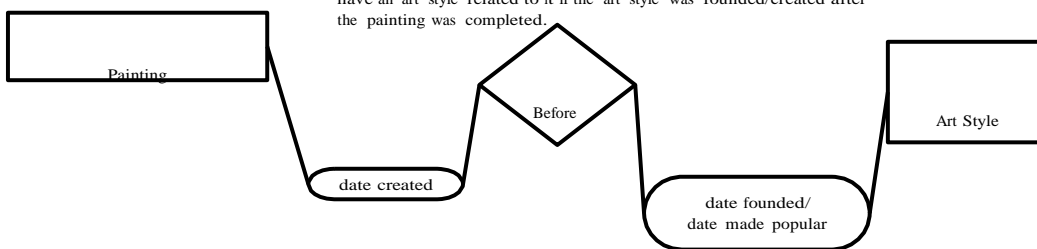
A weak entity type is an entity type that relies on its foreign key to create a primary key. This is the apprentice entity, in order for an apprentice to have a primary key of apprentice id, they must have an artist to study under.



A reflexive relationship is when an entity has attributes that are related to itself. The artist table has a couple reflexive relationships with the columns married to and friends with. Artists can find love with other artists as well as become friends with other artists.



An integrity check is when data depends on accurate information. There are several in this table but one example is that a painting can't have an art style related to it if the art style was founded/created after the painting was completed.



Physical Schema

