

Art Archive

http://web.engr.oregonstate.edu/~volairn/art_database/artArchiveHome.html



Table of Contents:

Outline	1
Database Outline in Words	1-2
ER Diagram of Database	3
Database Schema	4
Table Creation Queries	5-6
General Use Queries	7-10

Outline

This database describes a universe of artwork created by one of the collaborators over the last ten years. This universe contains entities representing the actual artwork, mediums, series, and exhibitions as well as their relationships. The artwork is made as single pieces or as part of a series. The goal is to have these series displayed in exhibitions. Series which are exceptional may be displayed in several exhibitions. The artwork is often composed of one or more mediums, except in the case of conceptual works which do not have mediums. Mediums can be used for many works of art. There is often a trend in mediums throughout a series.

There are over a hundred works of art in this universe, so cataloging and displaying in generic photo viewing applications is a bit cumbersome and also often obscures relevant information. This database provides a more personalized solution which matches the artist's working style and tracks the entire process of art creation and display. There is a wealth of interesting source material to organize, and future additions of new artwork make this set of data a good candidate for a database.

Database Outline in Words- *Paragraph Format*

Artwork can be composed of several mediums, and mediums can be used in many artworks. Artwork doesn't have to have a medium, for example conceptual pieces. Medium also doesn't have to be used on an artwork, such as in the case of experimental mediums which might be associated with future artwork if the results are successful. Some artworks are made in a similar style with an overarching theme and are grouped into a series. An artwork may only be in one series, or could be a standalone work not associated with any series. A series must be associated with a work of art. If there are no available works of art to associate to a series, it cannot be created. If a series is successful, it will hopefully be shown in an exhibition. One series can be featured in several different exhibitions, but doesn't have to be featured in an exhibition. Exhibitions must feature exactly one series.

Database Outline in Words- *Outline Format*

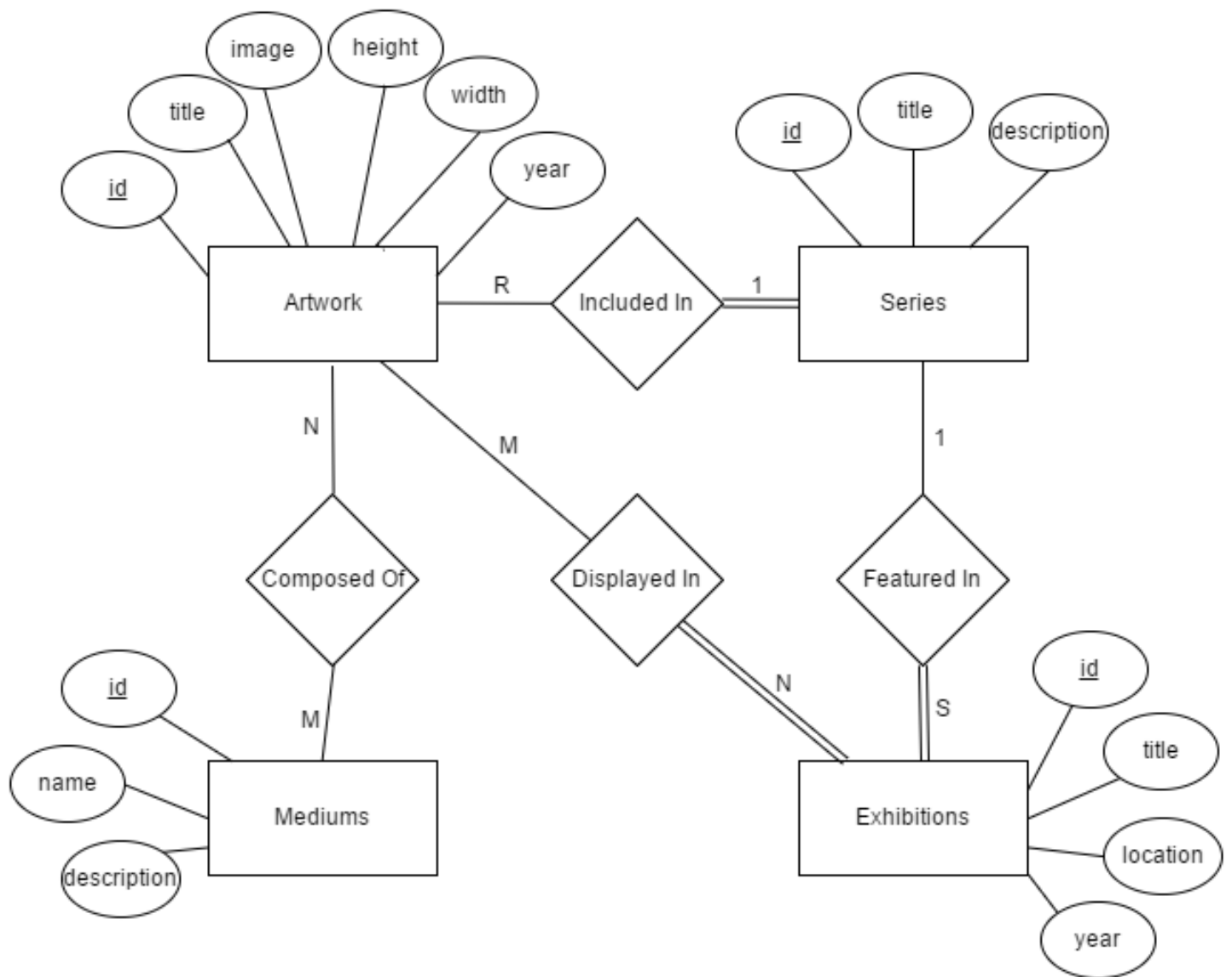
The entities we will have in our database are:

1. **Artwork** – Each work of art has an id, title, image, sid, height, width, and year
 - id is the primary key which cannot be null and must be unique
 - title cannot be null and must be unique
 - image is a url which links to a photo of the artwork
 - if there is no image, an artwork can be added without one
 - if the image is found to be null, a generic photo will be displayed instead
 - if an incorrect url is entered, the photo will simply display as an empty photo box.
 - sid is a foreign key which relates to a series id
2. **Medium** – Each medium has an id, name and may have a description
 - id is the primary key which cannot be null and must be unique
 - name cannot be null and must be unique
 - description is for nonconventional materials such as nail polish scales
3. **Series** – Each series has an id, title and description
 - id is the primary key which cannot be null and must be unique
 - title cannot be null and must be unique
4. **Exhibition** – Each exhibition has an, id, title, location, and year
 - id is the primary key which cannot be null and must be unique
 - sid which is a foreign key which cannot be null and must be unique it relates to a series id
 - title cannot be null and must be unique
 - location could be a specific place name, city, or even a country

The relationships we will have:

1. **Artwork – Medium** is a many-to-many relationship
 - Works of art can be composed of zero to many mediums
 - Mediums can be used on zero to many works of art
 - ❖ **Composed_of** is the table representing this relationship
 - an aid relates to artwork and an mid relates to a series
2. **Artwork – Series** is one-to-many relationship
 - A single series includes one to many works of art
 - A work of art does not have to be included in a series
3. **Series – Exhibition** is a one-to-many relationship
 - An exhibition must feature one series
 - One series may be feature in zero to many exhibitions
4. **Artwork – Exhibition** is a many-to-many relationship
 - A work of art may be featured in zero to many exhibitions
 - An exhibition must feature one to many works of art
 - ❖ **Series** is the table that enforces this relationship
 - A series must include one or more works of art
 - An exhibition must feature exactly one series

ER Diagram of Database



Database Schema

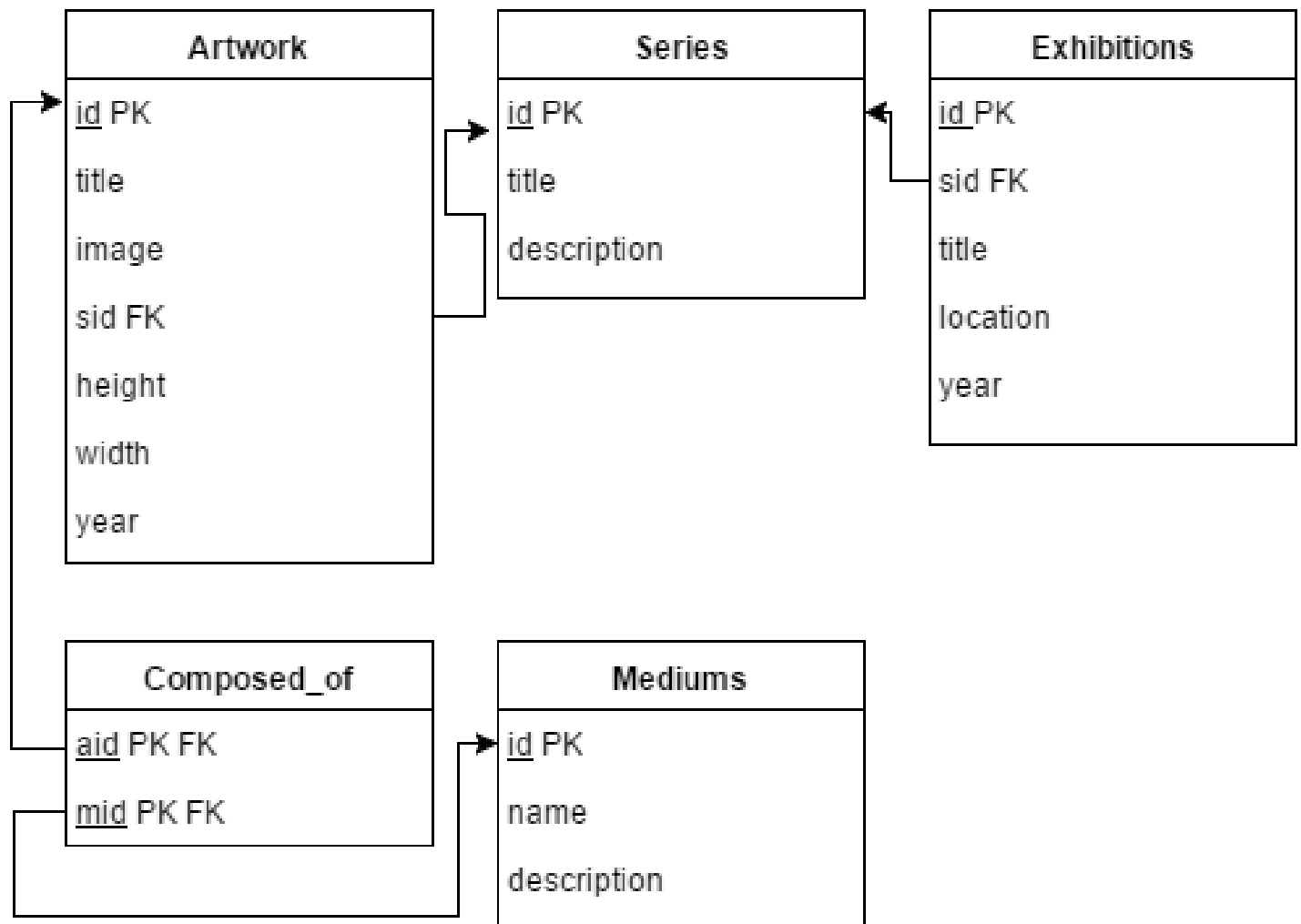


Table Creation Queries

--Table Creations

```
CREATE TABLE artwork (  
    id int(11) AUTO_INCREMENT,  
    title varchar(255) NOT NULL UNIQUE,  
    image varchar(255),  
    sid int(11),  
    width int(11),  
    height int(11),  
    year year(4),  
    PRIMARY KEY (id),  
    FOREIGN KEY (sid) REFERENCES series(id)  
)  
ENGINE=InnoDB;
```

```
CREATE TABLE mediums (  
    id int(11) AUTO_INCREMENT,  
    name varchar(255) NOT NULL UNIQUE,  
    description text,  
    PRIMARY KEY (id)  
)  
ENGINE=InnoDB;
```

```
CREATE TABLE series (  
    id int(11) AUTO_INCREMENT,  
    title varchar(255) NOT NULL UNIQUE,  
    description text,  
    PRIMARY KEY (id)  
)  
ENGINE=InnoDB;
```

```

CREATE TABLE exhibitions (
    id int(11) AUTO_INCREMENT,
    title varchar(255) NOT NULL UNIQUE,
    location text,
    sid int(11),
    year year(4),
    PRIMARY KEY (id),
    FOREIGN KEY (sid) REFERENCES series(id)
) ENGINE=InnoDB;

```

```

CREATE TABLE composed_of (
    aid int(11),
    mid int(11),
    PRIMARY KEY (aid, mid),
    FOREIGN KEY (aid) REFERENCES artwork(id),
    FOREIGN KEY (mid) REFERENCES mediums(id)
) ENGINE=InnoDB;

```

--Initial Insertions

```

INSERT INTO mediums (id, name, description) VALUES (id, [name], [description]);
INSERT INTO series (id, title, description) VALUES (id, [title], [description]);
INSERT INTO artwork (id, title, image, sid, width, height, year)
    VALUES (id, [title], [image url],
    (SELECT id FROM series WHERE title = [series title]),
    [width], [height], [year]);
INSERT INTO exhibitions (id, title, location, sid, year)
    VALUES (id, [title], [location], (SELECT id FROM series WHERE title = [artwork title]), [year]);
INSERT INTO composed_of (aid, mid)
    VALUES ((SELECT id FROM artwork WHERE title = [artwork title]),
    (SELECT id FROM mediums WHERE name = [medium name]));

```

General Use Queries

User Inserts

-- Mediums

```
INSERT INTO mediums (id, name, description) VALUES (id, [name], [description]);
```

-- Series Creation: artwork is associated afterwards

```
INSERT INTO series (id, title, description) VALUES (id, [title], [description]);
```

-- Series

Note: artwork insertions must still be done, so a series is made and an artwork is added

--\$SeriesTitle is returned from the insertion to properly update the selected artwork

```
INSERT INTO series(title, description) VALUES ([series title], [series description])
```

```
UPDATE artwork SET sid = (SELECT id FROM series WHERE title = [$SeriesTitle])
```

```
WHERE id = [artwork id]
```

-- Artwork

```
INSERT INTO artwork (id, title, image, sid, width, height, year)
```

```
VALUES (id, [title], [image url],
```

```
(SELECT id FROM series WHERE title = [series title]), [width], [height], [year]);
```

-- Exhibitions

```
INSERT INTO exhibitions (id, title, location, sid, year)
```

```
VALUES (id, [title], [location],
```

```
(SELECT id FROM series WHERE title = [artwork title]), [year]);
```

-- Composed_of

```
INSERT INTO composed_of (aid, mid)
```

```
VALUES ((SELECT id FROM artwork WHERE title = [artwork title]),
```

```
(SELECT id FROM mediums WHERE name = [medium name]));
```

User Selections

--Artwork queries

--Select all artwork

```
SELECT a.title, a.image, s.title, a.width, a.height, a.year FROM artwork a
      LEFT JOIN series s ON a.sid = s.id
      ORDER BY s.title, a.year DESC
```

--Select artwork not in a series

```
SELECT a.title, a.image, s.title, a.width, a.height, a.year FROM artwork a
      LEFT JOIN series s ON a.sid = s.id
      WHERE a.sid IS NULL
      ORDER BY a.title, a.year DESC
```

--Select artwork by title

```
SELECT a.title, a.image, s.title, a.width, a.height, a.year FROM artwork a
      LEFT JOIN series s ON a.sid = s.id
      WHERE a.title = [artwork title]
```

--Select artwork by size

```
SELECT a.title, a.image, s.title, a.width, a.height, a.year FROM artwork a
      LEFT JOIN series s ON a.sid = s.id
      WHERE a.width <= [artwork width] AND a.height <= [artwork height]
      ORDER BY s.title, a.year DESC
```

--Select artwork by year

```
SELECT a.title, a.image, s.title, a.width, a.height, a.year FROM artwork a
      LEFT JOIN series s ON a.sid = s.id
      WHERE a.year = [artwork year]
      ORDER BY s.title, a.year DESC
```


--Select earliest artwork

```
SELECT a.title, a.image, s.title, a.width, a.height, a.year FROM artwork a
    LEFT JOIN series s ON a.sid = s.id
    WHERE a.year = (SELECT MIN(year) FROM artwork) ORDER BY s.title, a.year DESC
```

--Select latest artwork

```
SELECT a.title, a.image, s.title, a.width, a.height, a.year FROM artwork a
    LEFT JOIN series s ON a.sid = s.id
    WHERE a.year = (SELECT MAX(year) FROM artwork) ORDER BY s.title, a.year DESC
```

--Select artwork without images

--Note: this selects artwork entered without images, artwork with incorrect images links will not be included

```
SELECT a.title, a.image, s.title, a.width, a.height, a.year FROM artwork a
    LEFT JOIN series s ON a.sid = s.id
    WHERE a.image IS NULL ORDER BY s.title, a.year DESC
```

--Mediums Queries

--Select all mediums

```
SELECT name, description FROM mediums
```

--Select Composed_of relationships between artwork and mediums

```
SELECT a.title, a.image, m.name FROM artwork a
    INNER JOIN composed_of c ON a.id = c.aid
    INNER JOIN mediums m ON m.id = c.mid
    ORDER BY a.title
```

--Select artwork by medium

```
SELECT a.title, a.image, m.name FROM mediums m
    INNER JOIN composed_of c ON c.mid = m.id
    INNER JOIN artwork a ON a.id = c.aid
    WHERE m.id = [medium id] ORDER BY a.title ASC
```

--Series Queries

--Select all series

```
SELECT s.title, s.description, COUNT(a.id) FROM series s
      LEFT JOIN artwork a ON a.sid = s.id
      GROUP BY s.title ORDER BY s.title
```

--Select artwork by series

```
SELECT a.title, a.image, s.title, a.width, a.height, a.year FROM artwork a
      INNER JOIN series s ON a.sid = s.id
      WHERE s.id = [series id]
      ORDER BY s.title ASC, a.year DESC
```

--Exhibition Queries

--Select all exhibitions

```
SELECT e.title, e.location, s.title, e.year FROM exhibitions e
      INNER JOIN series s ON e.sid = s.id
      ORDER BY e.year DESC
```

--Select artwork by exhibition

```
SELECT a.title, a.image, s.title, e.title, e.location, e.year FROM artwork a
      INNER JOIN series s ON a.sid = s.id
      INNER JOIN exhibitions e ON e.sid = s.id
      WHERE e.id = [exhibition id]
      ORDER BY a.title ASC
```

--Select exhibition by year

```
SELECT e.title, e.location, s.title, e.year FROM exhibitions e
      INNER JOIN series s ON e.sid = s.id
      WHERE e.year = [exhibition year]
      ORDER BY e.title ASC
```