Art Patron Application Backend Database

Project 2: Group 3

[1]

[1] Image courtesy of Andy Mabbett at http://commons.wikimedia.org/wiki/File:New_Art_Gallery_Walsall_-_QRpedia_-_Portrait_of_Kitty.JPG

SECTION 1: Overview

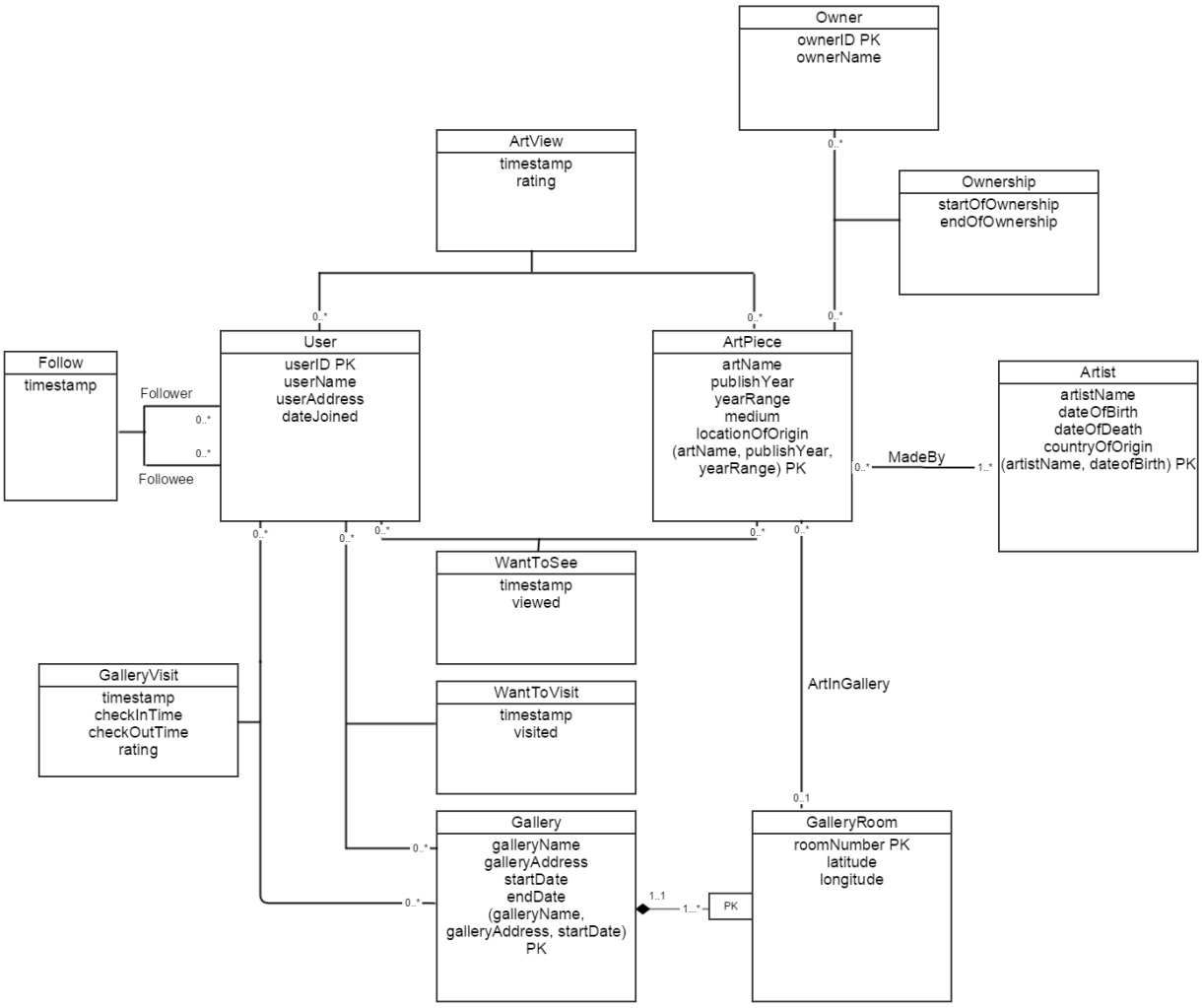
We have designed a backend database intended to be used with a mobile application. Our vision is for an app that enhances the experiences of both art lovers looking to engage in their passion as well as the galleries that serve them. An art patron using the app supported by this database will be able to develop a profile of the art they view and their opinions of the pieces, as well as connect to other art patrons over a social network. This app will provide users access to a information on a multitude of art pieces within many different galleries. A gallery using the app supported by this database will be able to help guide visitors through specialized tours of the gallery in accordance with their tastes, collect opinions in order to tailor the gallery to what visitors appreciate, and coordinate with art owners to arrange for loans of specific pieces.

Our functional specification for the utilities provided by such an application is as follows:

- Must store the location of art pieces within their galleries
- Should store relevant descriptive information about art pieces
- Must allow user to track and rate art pieces they view
- Must allow for users to follow others and track their visits and ratings
- Should allow user to store list of "To View" art pieces
- Should allow galleries to coordinate with art owners to acquire pieces for exhibition
- Must allow galleries to track aggregate ratings of pieces in their exhibits
- Should allow users to plan their routes through galleries by combining their ratings history and gallery art location data

Our database stores a variety of information in order to allow for such functionality. The database stores persistent information about art pieces and artists, such as the lifetime and nationality of artists, the specific medium of art pieces, and each art pieces approximate year of creation. The names and locations of various art galleries are also stored in the database, along with layout data about the rooms within these galleries. The database can track which art pieces are displayed within each room of each gallery, which enables customized tours to be created for patrons. The owners of art pieces that are held in the database can also have their information and ownerships recorded, facilitating the process of art lending between owners and galleries. When users sign up for the hypothetical app, their information is also stored within the database. Various tables in the database allow each user to build up their own unique profile of their taste and viewing habits. Users can track specific galleries that they desire to visit and art pieces that they desire to see. Users can also record their gallery visits and art viewings, and in the case of art pieces provide a rating for each piece in question. Finally, users can choose to follow one another so they receive updates on the preferences of their friends. The database stores information on who is following whom. For a general overview of the structure of our backend database, please refer to the UML diagram in Section 2. For specific details about the technical aspects of the database, please refer to Section 5, which contains comprehensive SQLite code detailing our implementation.

SECTION 2: Comprehensive UML Diagram of Database



SECTION 3: Functional Dependencies

Our relational schema is in Boyce-Codd Normal Form (BCNF), which means that all redundancy based on functional dependency has been removed. Therefore, our database design does not include any functional dependencies that do not stem from primary key constraints. For all functional dependencies in our relational database, the left side of the functional dependency is a super key for the schema.

SECTION 4: Assertions

The following two assertions were specified in order to enforce the two instances of one-to-many cardinality constraints in our database design. These assertions ensure that all art pieces are associated with at least one author in the database and that every gallery listed in the database has at

least one corresponding room. Unfortunately, no SQL implementation supports the ability to create assertions. Our implementation code approximates the assertion functionality through check constraints and deferrable foreign key constraints.

Check constraints in the ArtPiece and Gallery table ensure that any insert record corresponds to at least one record in their corresponding associated tables, MadeBy and GalleryRoom. Foreign key constraints in MadeBy and GalleryRoom assure that all records in those tables are also associated with a record in ArtPiece and Gallery, respectively, which could create a deadlock when trying to insert new records into ArtPiece and Gallery. The foreign key constraints are made deferrable to prevent this deadlock, by allowing new records to be inserted in pairs, first into MadeBy or GalleryRoom and then into ArtPiece or Gallery, without violating either constraints.

Our approximation unfortunately cannot handle the case where deletions are made on MadeBy or GalleryRoom such that some record in ArtPiece or Gallery is left without an associated record. In addition, SQLite cannot handle check constraints that use subqueries, which is necessary for this approximation. However, other systems can handle such queries, making them a viable choice. As such, our check constraints are left in our code but commented out to allow error-free processing by SQLite. The ideal assertions can be seen below.

```
/* If a galley exists, it must contain at least one room. Enforces one to many constraint. */
CREATE ASSERTION RoomExists (
Check( NOT EXISTS(
      SELECT galleryName, galleryAddress, startDate
      FROM Gallery
      MINUS
      SELECT UNIQUE galleryName, galleryAddress, startDate
      FROM GalleryRoom;
   )
   );

/* Every art piece has to be listed in MadeBy at least once. Enforces one to many constraint. */
CREATE ASSERTION HasCreator (
Check ( NOT EXISTS(
      SELECT artName, publishYear, yearRange
      FROM ArtPiece
      MINUS
      SELECT UNIQUE artName, publishYear, yearRange
      FROM MadeBy;
    )
   );
```

SECTION 5: Comprehensive Tables, Inserts, Deletes, Updates, Triggers

```
/* Delete the tables if they already exist */
DROP TABLE IF EXISTS User;
DROP TABLE IF EXISTS Follow;
```

```
DROP TABLE IF EXISTS ArtView;
DROP TABLE IF EXISTS ArtPiece;
DROP TABLE IF EXISTS WantToSee;
DROP TABLE IF EXISTS WantToVisit;
DROP TABLE IF EXISTS Gallery;
DROP TABLE IF EXISTS GalleryVisit;
DROP TABLE IF EXISTS GalleryRoom;
DROP TABLE IF EXISTS ArtInGallery;
DROP TABLE IF EXISTS Artist;
DROP TABLE IF EXISTS MadeBy;
DROP TABLE IF EXISTS Owner;
DROP TABLE IF EXISTS Ownership;
DROP VIEW IF EXISTS ArtPieceInfo;
DROP VIEW IF EXISTS AncientPaintings;
DROP VIEW IF EXISTS UsersGalleryRatings;
DROP TRIGGER IF EXISTS T1;
DROP TRIGGER IF EXISTS T2;
DROP INDEX IF EXISTS ArtistKeyIndex;
DROP INDEX IF EXISTS OwnerKeyIndex;
DROP INDEX IF EXISTS ArtPieceKeyIndex;
DROP INDEX IF EXISTS UserKeyIndex;
DROP INDEX IF EXISTS GalleryKeyIndex;
DROP INDEX IF EXISTS GalleryRoomKeyIndex;
DROP INDEX IF EXISTS OwnershipIndex;
DROP INDEX IF EXISTS ArtInGalleryIndex;

/* Holds information about a user*/
CREATE TABLE User (
        userID BIGINT,
        userName VARCHAR(50),
        userAddress  VARCHAR(100),
        dateJoined DATETIME NOT NULL,
        PRIMARY KEY(userID)
);

/* Shows all the users that a particular user follows.
userID1 is the follower, userID2 is the followee*/
CREATE TABLE Follow (
        userID1 BIGINT  NOT NULL,
        userID2 BIGINT  NOT NULL,
        timestamp DATETIME NOT NULL,
        PRIMARY KEY(userID1, userID2)
        FOREIGN KEY(userID1) REFERENCES User(userID) ON DELETE CASCADE,
        FOREIGN KEY(userID2) REFERENCES User(userID) ON DELETE CASCADE
```

```
);

/* This table represents the most recent time a user has recorded viewing an art piece.*/
CREATE TABLE ArtView (
	userID BIGINT NOT NULL,
	artName VARCHAR(100) NOT NULL,
	publishYear INT NOT NULL,
	yearRange INT NOT NULL,
	timestamp DATETIME NOT NULL,
	rating SMALLINT CHECK (rating >= 0 AND rating <= 5),
	PRIMARY KEY (userID, artName, publishYear, yearRange)
	FOREIGN KEY (artName, publishYear, yearRange)
		REFERENCES ArtPiece(artName, publishYear, yearRange)
		ON DELETE RESTRICT,
	FOREIGN KEY (userID) REFERENCES User(userID) ON DELETE CASCADE
);

/* Holds information describing all pieces of art in the database*/
CREATE TABLE ArtPiece (
	artName VARCHAR(100) NOT NULL,
	publishYear INT NOT NULL,
	yearRange INT NOT NULL,
	medium VARCHAR(50),
	locationofOrigin VARCHAR(100),
	PRIMARY KEY (artName, publishYear, yearRange)
	/* CHECK((artName, publishYear, yearRange)
		IN (SELECT artName, publishYear, yearRange FROM MadeBy)) */
);

/* List of art pieces a user wants to see.
Boolean value viewed updated if the user records viewing the piece*/
CREATE TABLE WantToSee (
	artName VARCHAR(100) NOT NULL,
	publishYear INT NOT NULL,
	yearRange INTNOT NULL,
	userID BIGINT NOT NULL,
	timestamp DATETIME NOT NULL,
	viewed BOOLEAN,
	PRIMARY KEY(userID, artName, publishYear, yearRange),
	FOREIGN KEY(userID) REFERENCES User(userID) ON DELETE CASCADE,
	FOREIGN KEY(artName, publishYear, yearRange)
		REFERENCES ArtPiece(artName, publishYear, yearRange)
		ON DELETE RESTRICT
);
```

```
/* List of galleries a user wants to visit.
Boolean value visited updated if the user records visiting the gallery*/
CREATE TABLE WantToVisit (
        userID BIGINT NOT NULL,
        galleryName VARCHAR(50) NOT NULL,
        galleryAddress VARCHAR(100) NOT NULL,
        startDate DATE NOT NULL,
        timestamp DATETIME NOT NULL,
        visited BOOLEAN,
        PRIMARY KEY(userID, galleryName, galleryAddress, startDate),
        FOREIGN KEY(userID) REFERENCES User(userID) ON DELETE CASCADE,
        FOREIGN KEY(galleryName, galleryAddress, startDate)
                REFERENCES Gallery(galleryName, galleryAddress, startDate)
                ON DELETE CASCADE
);

/* The gallery table stores the individual entities where art can be visited.
The table's primary keys are galleryName and location. */
CREATE TABLE Gallery (
        galleryName VARCHAR(50) NOT NULL,
        galleryAddress VARCHAR(100) NOT NULL,
        startDate DATETIME NOT NULL,
        endDate DATETIME,
        galleryOwner VARCHAR(50),
        PRIMARY KEY (galleryName, galleryAddress, startDate)
        /* CHECK((galleryName, galleryAddress, startDate)
                IN (SELECT galleryName, galleryAddress, startDate FROM GalleryRoom)) */
);

/* Represents the most recent time a user has visited a gallery.*/
CREATE TABLE GalleryVisit (
        galleryName VARCHAR(50) NOT NULL,
        galleryAddress VARCHAR(100) NOT NULL,
        startDate DATE NOT NULL,
        checkInTime DATETIME NOT NULL,
        checkOutTime DATETIME,
        rating SMALLINT CHECK (rating >= 0 AND rating <= 5),
        userID BIGINT NOT NULL,
        PRIMARY KEY(galleryName, galleryAddress, startDate, userID)
        FOREIGN KEY(galleryName, galleryAddress, startDate)
                REFERENCES Gallery(galleryName, galleryAddress, startDate)
                ON DELETE CASCADE,
        FOREIGN KEY(userID) REFERENCES User(userID) ON DELETE CASCADE
```

```
);

/* This table represents each room inside a gallery*/
CREATE TABLE GalleryRoom (
        galleryName VARCHAR(50) NOT NULL,
        galleryAddress VARCHAR(100) NOT NULL,
        startDate DATE NOT NULL,
        roomNumber SMALLINT NOT NULL,
        latitude  FLOAT,
        longitude FLOAT,
        PRIMARY KEY(galleryName, galleryAddress, startDate, roomNumber),
        FOREIGN KEY(galleryName, galleryAddress, startDate)
                REFERENCES Gallery(galleryName, galleryAddress, startDate)
                ON DELETE CASCADE
                DEFERRABLE INITIALLY DEFERRED
);

/* Holds information about each piece of art in a gallery and which room it is in.*/
CREATE TABLE ArtInGallery (
        galleryName VARCHAR(50) NOT NULL,
        roomNumber VARCHAR(50) NOT NULL,
        startDate DATE NOT NULL,
        galleryAddress VARCHAR(100) NOT NULL,
        artName VARCHAR(100) NOT NULL,
        publishYear INT NOT NULL,
        yearRange INT NOT NULL,
        PRIMARY KEY(artName, publishYear, yearRange)
        FOREIGN KEY(galleryName, galleryAddress, startDate, roomNumber)
                REFERENCES GalleryRoom(galleryName, galleryAddress, startDate, roomNumber)
                ON DELETE CASCADE,
        FOREIGN KEY(artName, publishYear, yearRange)
                REFERENCES ArtPiece(artName, publishYear, yearRange)
                ON DELETE CASCADE
);

/* Holds information about artists in the database.*/
CREATE TABLE Artist (
        artistName VARCHAR(50) NOT NULL,
        dateofBirth DATE NOT NULL,
        dateofDeath DATE,
        countryofOrigin VARCHAR(100),
        PRIMARY KEY(artistName, dateOfBirth)
);
```

```
/* The table represents the artists associated with each art piece.
There will be one entry per artist per piece.*/
CREATE TABLE MadeBy (
	artistName VARCHAR(50) NOT NULL,
	dateofBirth DATE NOT NULL,
	artName VARCHAR(100) NOT NULL,
	publishYear INT NOT NULL,
	yearRange INT NOT NULL,
	PRIMARY KEY (artistName, dateOfBirth, artName, publishYear, yearRange)
	FOREIGN KEY(artistName, dateOfBirth)
		REFERENCES Artist(artistName, dateOfBirth)
		ON DELETE CASCADE
		DEFERRABLE INITIALLY DEFERRED,
	FOREIGN KEY(artName, publishYear, yearRange)
		REFERENCES ArtPiece(artName, publishYear, yearRange)
		ON DELETE CASCADE
		DEFERRABLE INITIALLY DEFERRED
);

/* Holds information about owners of art pieces.*/
CREATE TABLE Owner (
	ownerID BIGINT,
	ownerName VARCHAR(50),
	PRIMARY KEY (ownerID)
);

/* This table represents the owners, both current and previous, of all art pieces that have
owners.*/
CREATE TABLE Ownership (
	ownerID BIGINT NOT NULL,
	artName VARCHAR(100) NOT NULL,
	publishYear INT NOT NULL,
	yearRange INTNOT NULL,
	startofOwnership DATE NOT NULL,
	endofOwnership DATE,
	PRIMARY KEY (artName, publishYear, yearRange)
	FOREIGN KEY(ownerID) REFERENCES Owner(ownerID)
		ON DELETE CASCADE
		DEFERRABLE INITIALLY DEFERRED,
	FOREIGN KEY(artName, publishYear, yearRange)
		REFERENCES ArtPiece(artName, publishYear, yearRange)
		ON DELETE CASCADE
		DEFERRABLE INITIALLY DEFERRED
);
```

```
/* View containing all the information about an art piece.
Some users of the database might only be interested in finding a specific information about
art pieces and not in the social aspect of the database. In this view, owner name could
only be available to a subset of authorized database users.*/
CREATE VIEW ArtPieceInfo AS
SELECT AP.artName, AP.publishYear, AP. yearRange, AP.medium, AP.locationofOrigin,
        MB.artistName, O.ownerName, AIG.galleryName
FROM ArtPiece AP, MadeBy MB, Owner O, Ownership OS, ArtInGallery AIG
WHERE AP.artName = AIG.artName AND AP.publishYear = AIG.publishYear
        AND AP.yearRange = AIG.yearRange AND AP.artName = MB.artName
        AND AP.publishYear = MB.publishYear AND AP.yearRange = MB.yearRange
        AND AP.artName = OS.artName AND AP.publishYear = OS.publishYear
        AND AP.yearRange = OS.yearRange AND OS.ownerID = O.ownerID;

/*View containing information about users and the ratings they gave to a specific
gallery after visiting it. Gallery representatives might be interested in the visitors'
opinion about their galleries to improve service, for example. In this view, the
private information about users (name, address) will only be available to the
official representative of a gallery. */
CREATE VIEW UsersGalleryRatings AS
SELECT U.userID, U.userName, U.userAddress, U.dateJoined,
                GV.checkInTime, GV.checkOutTime, GV.rating
FROM User U, GalleryVisit GV
WHERE GV.galleryName = 'Musee de lOrangerie'
        AND GV.galleryAddress = 'Jardin Tuileries, 75001, Paris, France'
        AND GV.startDate = '2006-05-01 00:00:00'
        AND U.userID = GV.userID;

/*View containing information about all paintings that were published before 19th century.
There could be users that are interested only in art pieces in one particular medium that
were created in art periods preceding 19th century.*/
CREATE VIEW AncientPaintings AS
SELECT artName, publishYear, yearRange, medium, locationOfOrigin
FROM ArtPiece
WHERE publishYear < 1800 AND medium = 'paint';

/* Artist information will frequently be looked up when patrons are visiting galleries or trying to
find similar art to view, but new artists rarely join the database. */
CREATE INDEX ArtistKeyIndex
ON Artist (artistName, dateOfBirth);

/* Most art is owned by collectors, so it is very rare to add new previously absent owners or
```

```
delete someone who no longer owns any art. Owner queries are not too frequent, but they occur
much more often than these updates. */
CREATE INDEX OwnerKeyIndex
ON Owner (ownerID);

/* Art piece information is one of the most central aspects of the database and will be queried very often during gallery visits. Art information is very static, and new notable pieces are created
fairly infrequently. */
CREATE INDEX ArtPieceKeyIndex
ON ArtPiece (artName, publishYear, yearRange);

/* Users will often be added and removed, but they will be queried so frequently for social networking purposes that it still makes sense to index their IDs. */
CREATE INDEX UserKeyIndex
ON User (userID);

/* Galleries will be queried often as patrons search for art they wish to view. Also, gallery information is usually very static. */
CREATE INDEX GalleryKeyIndex
ON Gallery (galleryName, galleryAddress, startDate);

/* Rooms within galleries are totally static, and will be queried very frequently as patrons plan their routes through galleries during a visit. */
CREATE INDEX GalleryRoomKeyIndex
ON GalleryRoom (galleryName, galleryAddress, startDate, roomNumber);

/* Most queries regarding owners will revolve around which art pieces they own, and ownership
is fairly static, so it makes sense to index their IDs within the Ownership table too. */
CREATE INDEX OwnershipIndex
ON Ownership (ownerID);

/* Users visiting galleries will often want to see what art is held in each room. Since room information is very stable, we also index that information in the ArtInGallery table. */
CREATE INDEX ArtInGalleryIndex
ON ArtInGallery(galleryName, galleryAddress, startDate, roomNumber);

/*Update WantToVisit table after a user visits some gallery*/
CREATE TRIGGER HasVisited
AFTER INSERT ON GalleryVisit
FOR EACH ROW
BEGIN
```

```
        UPDATE WantToVisit
        SET visited = 1, timestamp = New.checkOutTime
        WHERE userID = New.userID AND galleryName = New.galleryName
        AND galleryAddress = New.galleryAddress
        AND startDate = New.startDate;
END;

/*Update WantToSee table after a user viewed some art piece*/
CREATE TRIGGER HasSeen
AFTER INSERT ON ArtView
FOR EACH ROW
BEGIN
        UPDATE WantToSee
        SET timestamp = New.timestamp, viewed = 1
        WHERE userID = New.userID AND artName = New.artName
        AND publishYear = New.publishYear AND yearRange = New.yearRange;
END;


/* A trigger that deletes an art piece that no longer has an artist associated with it. */
CREATE TRIGGER deleteArt
AFTER DELETE ON Artist
FOR EACH ROW
BEGIN
        DELETE FROM ArtPiece
        WHERE artName IN (SELECT artName
                                FROM MadeBy
                                WHERE artistName = old.artistName
                                        AND dateOfBirth = old.dateOfBirth)
        AND publishYear IN (SELECT publishYear
                                FROM MadeBy
                                WHERE artistName = old.artistName
                                        AND dateOfBirth = old.dateOfBirth)
        AND yearRange IN (SELECT yearRange
                                FROM MadeBy
                                WHERE artistName = old.artistName
                                        AND dateOfBirth = old.dateOfBirth);
END;


/*SAMPLE DATA*/

INSERT INTO User (userID, userName, userAddress, dateJoined)
VALUES
```

(0, 'Jim Crow', '7336 lol Lane', '1999-10-03'),
(1, 'Eli Whit', '94 Smokye way', '1998-11-13'),
(2, 'Bob Chris', '3825 Washington', '1998-07-04'),
(3, 'Elias deh', 'First Street', '1998-12-03'),
(4, 'Ricky Bobby', 'A tunnel', '1997-02-03'),
(5, 'Aliance', '237 Vanderbilt', '2000-01-01'),
(6, 'Horde', 'Boulevard de le tensel', '1998-10-23'),
(7, 'Jim Raynor', 'In a museum', '1998-01-08'),
(8, 'Eliant Tom', '7364 drive Lane', '1999-05-03'),
(9, 'Ashley Robinson', '382 West Side', '1999-02-03');

INSERT INTO Follow (userID1, userID2, timestamp)
VALUES
(0,1,'2006-05-01 10:00:00'),
(1,2,'2007-05-07 00:00:00'),
(2,1,'2005-05-05 00:00:00'),
(3,4,'2006-05-04 00:00:00'),
(5,8,'2006-05-03 00:00:00'),
(5,6,'2006-04-01 00:00:00'),
(4,9,'2002-05-01 00:00:00'),
(5,7,'2001-05-01 00:00:00');

INSERT INTO ArtView (userID, artName, publishYear, yearRange, timestamp, rating)
VALUES
(0,'Starry Night', 1889, 0, '2006-05-04 00:00:00', 5),
(5,'Starry Night', 1889, 0, '2006-05-04 00:00:00', 5),
(2,'Pieta', 1498, 3, '2006-05-04 00:00:00', 4),
(2,'Starry Night', 1889, 0, '2006-05-04 00:00:00', 4),
(4,'Il Giudizio Universale', 1541, 0, '2006-05-04 00:00:00', 5),
(7,'The Last Judgement', 1505, 0, '2006-05-04 00:00:00', 4),
(9,'Venus de Milo', 100, 30, '2006-05-04 00:00:00', 5),
(8,'Il Giudizio Universale', 1541, 0, '2006-05-04 00:00:00', 5),
(3,'Venus de Milo', 100, 30, '2006-05-04 00:00:00', 4),
(6,'The Last Judgement', 1505, 0, '2006-05-04 00:00:00', 5);

INSERT INTO Artist (artistName, dateofBirth, dateofDeath, countryofOrigin)
VALUES
('Leonardo Da Vinci', '1452-01-01', '1519-01-01', 'Italy'),
('Vincent Van Gogh', '1853-01-01', '1890-01-01', 'Holland'),
('Michelangelo', '1475-03-06', '1654-04-23', 'Italy'),
('Claude Money', '1840-11-14', '1926-08-05', 'Italy'),
('Pablo Picasso', '1881-01-01', '1973-05-01', 'Spain'),
('Raphael', '1483-01-01', '1520-04-06', 'Italy'),
('Auguste Renoir', '1841-02-25', '1919-06-24', 'France'),

('Jan Vermeer', '1632-01-01', '1675-01-01', 'Holland'),
('Paul Cezanne', '1839-10-11', '1906-05-15', 'France'),
('Rembrandt', '1606-01-01', '1669-01-01', 'Italy');

INSERT INTO MadeBy (artistName, dateofBirth, artName, publishYear, yearRange)
VALUES
('Leonardo Da Vinci', '1452-01-01', 'Il Giudizio Universale', 1541, 0),
('Vincent Van Gogh', '1853-01-01', 'Persistence of Memory', 1931, 0),
('Michelangelo', '1475-03-06', 'The Last Judgement', 1505, 0),
('Claude Money', '1840-11-14', 'Nimphee', 1926, 0),
('Pablo Picasso', '1881-01-01', 'The Last Supper', 1497, 0),
('Raphael', '1483-01-01', 'Starry Night', 1889, 0),
('Auguste Renoir', '1841-02-25', 'Sposalizio della Vergine', 1504, 0),
('Jan Vermeer', '1632-01-01', 'The Thinker', 1541, 15),
('Paul Cezanne', '1839-10-11', 'Pieta', 1498, 3),
('Rembrandt', '1606-01-01', 'Venus de Milo', 100, 30);

INSERT INTO Owner (ownerID, ownerName)
VALUES
(0, 'Bob Dole'),
(1, 'Caleb'),
(2, 'Jon Snow'),
(3, 'Elena'),
(4, 'Eddie Murphy'),
(5, 'Connor'),
(6, 'MaKenzie'),
(7, 'Elphius Doge'),
(8, 'Harry Potter'),
(9, 'Doctor Who');

INSERT INTO ArtPiece (artName, publishYear, yearRange, medium, locationofOrigin)
VALUES
('Il Giudizio Universale', 1541, 0, 'paint', 'Rome'),
('Persistence of Memory', 1931, 0, 'paint', 'New York'),
('The Last Judgement', 1505, 0, 'paint', 'NULL'),
('Nimphee', 1926, 0, 'paint', 'Paris'),
('The Last Supper', 1497, 0, 'paint', 'Milano'),
('Starry Night', 1889, 0, 'paint', 'Holland'),
('Sposalizio della Vergine', 1504, 0, 'paint', 'Milano'),
('The Thinker', 1541, 15, 'marble', 'Rome'),
('Pieta', 1498, 3, 'marble', 'Firenze'),
('Venus de Milo', 100, 30, 'Marble', 'Greece');

```
INSERT INTO Ownership (ownerID, artName, publishYear, yearRange, startofOwnership,
endofOwnership)
VALUES
(0, 'Persistence of Memory', 1931, 0, 2012-01-01, NULL),
(1, 'Il Giudizio Universale', 1541, 0, 2004-03-11, NULL),
(2, 'The Last Judgement', 1505, 0, 1995-01-01, 1999-01-01),
(3, 'Nimphee', 1926, 0, 2007-03-01, NULL),
(4, 'The Last Supper', 1497, 0, 2014-10-14, NULL),
(5, 'Starry Night', 1889, 0, 2012-01-01, NULL),
(6, 'Sposalizio della Vergine', 1504, 0, 2012-01-01, NULL),
(6, 'The Thinker', 1541, 15, 2012-01-01, NULL),
(7, 'Pieta', 1498, 3, 2012-01-01, NULL),
(8, 'Venus de Milo', 100, 30, 2012-01-01, NULL);

INSERT INTO GalleryVisit (galleryName, galleryAddress, startDate, checkInTime, checkOutTime,
rating, userID)
VALUES
('Sistine Chapel','00120 Vatican City', '1984-04-08 00:00:00', '2006-05-01 04:00:00', NULL, 2, 1),
('Sistine Chapel','00120 Vatican City', '1984-04-08 00:00:00', '2006-05-01 05:00:00', NULL, 1, 2),
('Musee de lOrangerie', 'Jardin Tuileries, 75001, Paris, France', '2006-05-01 07:00:00', '2006-05-
01 00:00:00', NULL, 2, 3),
('Musee de lOrangerie', 'Jardin Tuileries, 75001, Paris, France', '2006-05-01 00:00:00', '2006-05-
01 06:00:00', NULL, 2, 4),
('Musee de lOrangerie', 'Jardin Tuileries, 75001, Paris, France', '2006-05-01 00:00:00', '2006-05-
01 07:00:00', NULL, 2, 5),
('Musee Rodin', 'Hotel Biron 79, rue De Varenne, Paris, France', '1919-01-01 10:00:00', '2006-
05-01 08:00:00', NULL, 3, 1),
('Musee Rodin', 'Hotel Biron 79, rue De Varenne, Paris, France', '1919-01-01 10:00:00', '2006-
05-01 09:00:00', NULL, 3, 3),
('Musee Rodin', 'Hotel Biron 79, rue De Varenne, Paris, France', '1919-01-01 10:00:00', '2006-
05-01 01:00:00', NULL, 5, 5),
('Musee Rodin', 'Hotel Biron 79, rue De Varenne, Paris, France', '1919-01-01 10:00:00', '2006-
05-01 02:00:00', NULL, 4, 6),
('Louvre Museum', '75001 Paris, France', '1793-08-10 11:00:00', '2006-05-01 03:00:00', NULL, 4,
1);

INSERT INTO GalleryRoom (galleryName, galleryAddress, startDate, roomNumber, latitude,
longitude)
VALUES
('Sistine Chapel','00120 Vatican City', '1984-04-08 00:00:00', 2, 30, 179),
('Sistine Chapel','00120 Vatican City', '1984-04-08 00:00:00', 1, 31, 179),
('Museum of Modern Art', '11 West 53rd Street New York, NY', '1929-11-07 5:00:00', 25, -5, -95),
('Museum of Modern Art', '11 West 53rd Street New York, NY', '1929-11-07 5:00:00', 10, -5, -96),
```

('Academy of Fine Arts Gallery', 'Schillerplatz 3, 1010 Wien, Austria', '1887-04-01 00:00:00', 1, 56, 3),
('Musee d lOrangerie', 'Jardin Tuileries, 75001, Paris, France', '2006-05-01 00:00:00', 5, 22.5, -96.1),
('Pinacoteca Di Brera', 'Via Brera 28, Milan, Italy', '1776-01-01 7:00:00', 5, 55, 13),
('Musee Rodin', 'Hotel Biron 79, rue De Varenne, Paris, France', '1919-01-01 10:00:00', 1, -89.4, -52),
('Louvre Museum', '75001 Paris, France', '1793-08-10 11:00:00', 33, 20.7, -9.6),
('Louvre Museum', '75001 Paris, France', '1793-08-10 11:00:00', 35, 20.7, -9.6);

INSERT INTO ArtInGallery (galleryName, roomNumber, startDate, galleryAddress, artName, publishYear, yearRange)
VALUES
('Sistine Chapel', 2, '1984-04-08 00:00:00', '00120 Vatican City', 'Il Giudizio Universale', 1541, 0),
('Museum of Modern Art', 25,'1929-11-07 5:00:00', '11 West 53rd Street New York, NY', 'Persistence of Memory', 1931, 0),
('Academy of Fine Arts Gallery', 1,'1887-04-01 00:00:00', 'Schillerplatz 3, 1010 Wien, Austria', 'The Last Judgement', 1505, 0),
('Musee de lOrangerie', 5,'2006-05-01 00:00:00', 'Jardin Tuileries, 75001, Paris, France', 'Nimphee', 1926, 0),
('Museum of Modern Art', 10, '1929-11-07 5:00:00', '11 West 53rd Street New York, NY', 'Starry Night', 1889, 0),
('Pinacoteca Di Brera', 5, '1776-01-01 7:00:00','Via Brera 28, Milan, Italy', 'Sposalizio della Vergine', 1504, 0),
('Musee Rodin', 1,'1919-01-01 10:00:00', 'Hotel Biron 79, rue De Varenne, Paris, France', 'The Thinker', 1541, 15),
('Louvre Museum',35, '1793-08-10 11:00:00', '75001 Paris, France','Venus de Milo', 100, 30);

INSERT INTO Gallery (galleryName, galleryAddress, startDate, endDate, galleryOwner)
VALUES
('Sistine Chapel','00120 Vatican City', '1984-04-08 00:00:00', NULL, 'Baccio Pontelli'),
('Museum of Modern Art', '11 West 53rd Street New York, NY', '1929-11-07 5:00:00', NULL, 'Glenn D. Lowry'),
('Academy of Fine Arts Gallery', 'Schillerplatz 3, 1010 Wien, Austria', '1887-04-01 00:00:00', NULL, 'Eva Blimlinger'),
('Musee de lOrangerie', 'Jardin Tuileries, 75001, Paris, France', '2006-05-01 00:00:00', NULL, 'Marie-Paule Vial'),
('Pinacoteca Di Brera', 'Via Brera 28, Milan, Italy', '1776-01-01 7:00:00', NULL, 'Sandrina Bandera'),
('Musee Rodin', 'Hotel Biron 79, rue De Varenne, Paris, France', '1919-01-01 10:00:00', NULL, 'Catherine Chevillot'),
('Louvre Museum', '75001 Paris, France', '1793-08-10 11:00:00', NULL, 'Jean-Luc Martinez');

```
INSERT INTO WantToSee (artName, publishYear, yearRange, userID, timestamp, viewed)
VALUES
('Il Giudizio Universale', 1541, 0, 0, '2014-05-10 7:00:20', 0),
('Il Giudizio Universale', 1541, 3, 3, '2014-05-10 7:00:20', 0),
('Nimphee', 1926, 0, 6, '2010-05-07 7:00:00', 0),
('The Thinker', 1541, 15, 7, '2015-01-25 2:00:15', 0),
('Starry Night', 1889, 0, 9, '2014-07-01 7:05:10', 0),
('Venus de Milo', 100, 30, 5, '2012-03-05 4:00:00', 0),
('Persistence of Memory', 1931, 0, 2, '2008-10-10 10:00:25', 0),
('Starry Night', 1889, 0, 7, '2014-07-01 7:05:10', 0),
('Venus de Milo', 100, 30, 2, '2015-02-05 10:00:15', 0),
('Sposalizio della Vergine', 1504, 0, 7, '2014-05-07 8:00:45', 0);

INSERT INTO WantToVisit (userID, galleryName, galleryAddress, startDate, timestamp, visited)
VALUES
(0, 'Louvre Museum', '75001 Paris, France', '1793-08-10 11:00:00', '2014-10-10 10:00:00', 0),
(2, 'Musee Rodin', 'Hotel Biron 79, rue De Varenne, Paris, France', '1919-01-01 10:00:00', '2014-
01-01 4:00:25', 0),
(9, 'Museum of Modern Art', '11 West 53rd Street New York, NY', '1929-11-07 5:00:00', '2014-
07-01 7:05:10', 0),
(1, 'Musee de lOrangerie', 'Jardin Tuileries, 75001, Paris, France', '2006-05-01 00:00:00', '2014-
01-03 4:15:57', 0),
(2, 'Academy of Fine Arts Gallery', 'Schillerplatz 3, 1010 Wien, Austria', '1887-04-01 00:00:00',
'2014-04-01 10:00:00', 0),
(2, 'Musee de lOrangerie', 'Jardin Tuileries, 75001, Paris, France', '2006-05-01 00:00:00', '2014-
02-03 11:00:15', 0),
(4,  'Museum of Modern Art', '11 West 53rd Street New York, NY', '1929-11-07 5:00:00', '2014-
05-02 2:00:15', 0),
(5, 'Louvre Museum', '75001 Paris, France', '1793-08-10 11:00:00', '2012-03-05 4:00:00', 0),
(7, 'Louvre Museum', '75001 Paris, France', '1793-08-10 11:00:00', '2014-01-01 11:00:15', 0),
(8, 'Louvre Museum', '75001 Paris, France', '1793-08-10 11:00:00', '2014-08-11 4:25:00', 0);

/* QUERIES */

/*Given an artpiece (name, publish year, year range),
return all of its artist and artpiece information*/
SELECT *
FROM ArtPiece, Artist, MadeBy
WHERE ArtPiece.artName = 'Starry Night'
        AND ArtPiece.publishYear = 1889
        AND ArtPiece.yearRange = 0
        AND ArtPiece.artName = MadeBy.artName
        AND ArtPiece.publishYear = MadeBy.publishYear
        AND Artpiece.yearRange = MadeBy.yearRange
```

```
        AND Artist.artistName = MadeBy.artistName
        AND Artist.dateOfBirth = MadeBy.dateOfBirth;

/*Find all of a specific User's followers*/
SELECT U.userID, U.userName, U.userAddress
FROM User U, Follow F
WHERE F.userID1 = 0
        AND F.userID2 = U.userID;

/*Find all of the pieces that a specific user has already viewed,
sorted by rating highest to lowest */
SELECT artName, publishYear, yearRange, rating
FROM ArtView AV
WHERE userID = 7
ORDER BY rating DESC;

/* Locations of art pieces a specific ID wants to see */
SELECT AIG.galleryName, AIG.galleryAddress, AIG.roomNumber
FROM ArtInGallery AIG, WantToSee WS
WHERE WS.userID = 5
        AND WS.ArtName = AIG.artName
        AND WS.publishYear = AIG.publishYear
        AND WS.yearRange = AIG.yearRange;

/*Given identifying information about a specific artpiece return the names of all owners
and the dates they owned the piece, ordered from most to least recent owner. */
SELECT Owner.ownerName, Ownership.startofOwnership, Ownership.endofOwnership
FROM Owner, Ownership
WHERE Ownership.artName = 'Starry Night'
        AND Ownership.publishYear = 1889
        AND Ownership.yearRange = 0
        AND Ownership.ownerID = Owner.ownerID
ORDER BY Ownership.startofOwnership DESC;

/*Given a specific room within the gallery and the date,
return the art piece information about all art pieces that have been located in the room */
SELECT artName, publishYear, yearRange
FROM ArtInGallery AIG
WHERE galleryName = 'Sistine Chapel'
        AND roomNumber = 2
        AND startDate <= '1984-04-08 00:00:00';

/* Return average rating and number of ratings for an artpiece.
Given an art piece whose PK is defined by variables X, Y, and Z
```

```
The IS NOT NULL is required because the check constraint on rating doesn't prevent nulls. */
SELECT AVG(AV.rating), COUNT(*)
FROM ArtView AV
WHERE AV.artName = 'Starry Night'
        AND AV.publishYear = 1889
        AND AV.yearRange = 0
        AND AV.rating IS NOT NULL;

/* Given an artpiece, show all other pieces by that artist.
The PK of the art piece in question is represented by variables X,Y,Z
Not DISTINCT because we want a piece to appear once in each artist's list.
THIS IS ONE OF OUR CORRELATED QUERIES. */
SELECT AP.artName, AP.publishYear, AP.yearRange
FROM MadeBy MB, ArtPiece AP,
        (SELECT Artist.artistName, Artist.dateOfBirth
         FROM Artist, MadeBy
         WHERE MadeBy.artName = 'Starry Night'
                AND MadeBy.publishYear = 1889
                AND MadeBy.yearRange = 0
                AND MadeBy.artistName = Artist.artistName
                AND MadeBy.dateOfBirth = Artist.dateOfBirth) A
WHERE A.artistName = MB.artistName
        AND A.dateOfBirth = MB.dateOfBirth
        AND AP.artName = MB.artName
        AND AP.publishYear = MB.publishYear
        AND AP.yearRange = MB.yearRange
        AND AP.artName <> 'Starry Night'
        AND AP.publishYear <> 1889
        AND AP.yearRange <> 0;

/* The most wanted to see and highly rated art pieces located in some city
THIS IS ONE OF OUR AGGREGATED QUERIES. */
SELECT AIG.artName, AIG.publishYear, AIG.yearRange, AIG.galleryName,
        AIG.galleryAddress, APR.avgRating, MostWS.WSNumber
FROM (SELECT artName, publishYear, yearRange,  count(*) AS WSNumber
    FROM WantToSee
    GROUP BY artName, publishYear, yearRange) AS MostWS,
   (SELECT artName, publishYear, yearRange, AVG(rating) AS avgRating
    FROM ArtView
    GROUP BY artName, publishYear, yearRange) AS APR,
    ArtInGallery AIG
WHERE APR.avgRating >= 4
        AND APR.artName = AIG.artName
        AND APR.publishYear = AIG.publishYear
```

```
        AND APR.yearRange = AIG.yearRange
        AND MostWS.artName = AIG.artName
        AND MostWS.publishYear = AIG.publishYear
        AND MostWS.yearRange = AIG.yearRange
        AND AIG.galleryAddress like '%Paris%'
ORDER BY APR.avgRating DESC, MostWS.WSNumber DESC;

/* Information about galleries visited by people a specific User follows */
SELECT DISTINCT galleryName, galleryAddress
FROM Follow F, GalleryVisit GV
WHERE F.userID1 = 0
        AND GV.userID = F.userID2;

/*Find all of a specific User's followers that want to see a specific piece*/
SELECT U.userID, U.userName, U.userAddress
FROM User U, Follow F, WantToSee
WHERE F.userID1 = 1
        AND WantToSee.artName = 'Persistence of Memory'
        AND WantToSee.publishYear = 1931
        AND WantToSee.yearRange = 0
        AND F.userID2 = U.userID
        AND U.userID = WantToSee.userID;

/* The most highly rated pieces within a certain gallery having
more than 100 number of views (sorted by rating).
Given galleryName gname, galleryAddress gaddress, startDate sd.
THIS IS ONE OF OUR AGGREGATED QUERIES.
THIS IS ONE OF OUR CORRELATED QUERIES. */
SELECT AIG.artName, AIG.publishYear, AIG.yearRange, APR.avgRating
FROM (SELECT artName, publishYear, yearRange, COUNT(*) AS views, AVG(rating) AS avgRating
          FROM ArtView
          GROUP BY artName, publishYear, yearRange) AS APR,
        ArtInGallery AS AIG
WHERE APR.views > 100
        AND APR.artName = AIG.artName
        AND APR.publishYear = AIG.publishyear
        AND APR.yearRange = AIG.yearRange
        AND 'Museum of Modern Art' = AIG.galleryName
        AND '11 West 53rd Street New York, NY' = AIG.galleryAddress
        AND '1929-11-07 5:00:00' = AIG.startDate
ORDER BY APR.avgRating DESC;

/* If user1 and user2 both rated the same artpiece a 5 rating,
query other artpieces that were given a 5 rating by user2 as suggestions for user1 */
```

```
SELECT DISTINCT V3.artName, V3.publishYear,V3.yearRange
FROM ArtView V1, ArtView V2, ArtView V3
WHERE V1.userID= 0
        AND V2.userID= 5
        AND V1.rating = 5
        AND V1.rating = V2.rating
        AND V1.artName=V2.artName
        AND V1.publishYear=V2.publishYear
        AND V1.yearRange=V2.yearRange
        AND V2.userID = V3.userID
        AND V3.rating = 5;

/* Find all of the art pieces that someone a user follows rated
a 4 or higher and the user has not yet viewed.
Given a userID uid.
THIS IS ONE OF OUR CORRELATED QUERIES. */
SELECT DISTINCT ratedPieces.rating, ratedPieces.artName,
        ratedPieces.publishYear, ratedPieces.yearRange
FROM (SELECT User.userID, rating,  artName, publishYear, yearRange
    FROM User, ArtView
    WHERE User.userID = ArtView.userID AND
    rating > 3) AS ratedPieces,
    ArtView AS  AV
WHERE ratedPieces.userID IN (SELECT userID2
                                FROM Follow
                                WHERE userID1 = 5)
        AND NOT EXISTS (SELECT *
                        FROM ArtView AS AV2
                        WHERE userID = 5
                                AND AV.artName = AV2.artName
                                AND AV.publishYear = AV2.publishYear
                                AND AV.yearRange = AV2.yearRange);

/* Given a geolocation, show list of all pieces the user wants to see within a certain distance.
Given userID U, latitude LA and longitude LO and distance D.
THIS IS ONE OF OUR CORRELATED QUERIES. */
SELECT Art.artName, Art.publishYear, Art.yearRange
FROM (SELECT artName, publishYear, yearRange
          FROM WantToSee
          WHERE WantToSee.userID = 5) Art,
        (SELECT galleryName, galleryAddress, startDate, roomNumber
         FROM GalleryRoom
         WHERE ABS((GalleryRoom.latitude - 5) + GalleryRoom.longitude - 90) <= 100) Local,
        ArtInGallery AIG
```

```
WHERE Art.artName = AIG.artName
        AND Art.publishYear = AIG.publishYear
        AND Art.yearRange = AIG.yearRange
        AND Local.galleryName = AIG.galleryName
        AND Local.galleryAddress = AIG.galleryAddress
        AND Local.startDate = AIG.startDate
        AND Local.roomNumber = AIG.roomNumber;

/* Update a user's rating of an art piece on a revision
Given user key U, new rating R, current time T, and art piece key X,Y,Z */
UPDATE ArtView
SET rating = 5, timestamp = '2015-12-03'
WHERE userID = 2
        AND artName = 'Starry Night'
        AND publishYear = 1889
        AND yearRange = 0;

/* Check off a piece on a user's Want To See list after they view it
Given user key U, current time T, and art key X,Y,Z */
UPDATE WantToSee
SET viewed = 1, timestamp = '1999-10-03'
WHERE userID = 4
        AND artName = 'Nimphee'
        AND publishYear = 1926
        AND yearRange = 0;

/* Record a user checking out of a gallery visit, along with their rating of the gallery
Given user U, current time T, rating R, and gallery keys A,B,C
The rating may be null if the user wishes */
UPDATE GalleryVisit
SET checkOutTime = '1984-04-08 10:00:00', rating = 4
WHERE userID = 7
        AND galleryName = 'Sistine Chapel'
        AND galleryAddress = '00120 Vatican City'
        AND startDate = '1984-04-08 00:00:00';

/* List the owners who have the most highly rated art, from highest to lowest average rating.
THIS IS ONE OF OUR AGGREGATED QUERIES.*/
SELECT O.ownerID, O.ownerName
FROM Owner O, Ownership OS, ArtView AV
WHERE OS.artName = AV.artName
        AND OS.publishYear = AV.publishYear
        AND OS.yearRange = AV.yearRange
        AND OS.ownerID = O.ownerID
```

```
GROUP BY OS.ownerID, O.ownerName
HAVING AVG(AV.rating) >= 4
ORDER BY AVG(AV.rating) DESC;

/*Select gallery in some location that has art pieces with average rating =5
THIS IS ONE OF OUR AGGREGATED QUERIES*/
SELECT DISTINCT AIG.galleryName, AIG.galleryAddress, AIG.startDate
FROM ArtInGallery AIG,
        (SELECT AV.artName, AV.publishYear, AV.yearRange, AVG(AV.rating) AS avgR
         FROM ArtView AV
         GROUP BY AV.artName, AV.publishYear, AV.yearRange) AS AVR
WHERE AIG.galleryAddress LIKE '%Paris%'
        AND  AIG.artName = AVR.artName
        AND AIG.publishYear = AVR.publishYear
        AND AIG.yearRange = AVR.yearRange
        AND AVR.avgR > 4;

/*Given an owner find galleries where his pieces are displayed.*/
SELECT AIG.galleryName, AIG.galleryAddress, AIG.startDate
FROM Ownership O, ArtInGallery AIG
WHERE O.ownerID = 6 AND O.artName = AIG.artName
        AND O.publishYear = AIG.publishYear AND O.yearRange = AIG.yearRange ;

/* Mark that a user wants to see all works by a particular artist */
INSERT INTO WantToSee (artName, publishYear, yearRange, userID, timestamp)
SELECT MB.artName, MB.publishYear, MB.yearRange, 4, datetime('now')
FROM MadeBy MB
WHERE MB.artistName = 'Pablo Picasso'
        AND MB.dateOfBirth = '1881-01-01';

/* Mark that a user wants to visit all galleries in some city */
INSERT INTO WantToVisit (userID, galleryName, galleryAddress, startDate, timestamp)
SELECT 3, G.galleryName, G.galleryAddress, G.startDate, datetime('now')
FROM Gallery G
WHERE G.galleryAddress LIKE '%Paris%';


/* Delete the tables if they already exist */
DROP TABLE IF EXISTS User;
DROP TABLE IF EXISTS Follow;
DROP TABLE IF EXISTS ArtView;
DROP TABLE IF EXISTS ArtPiece;
DROP TABLE IF EXISTS WantToSee;
DROP TABLE IF EXISTS WantToVisit;
```

```
DROP TABLE IF EXISTS Gallery;
DROP TABLE IF EXISTS GalleryVisit;
DROP TABLE IF EXISTS GalleryRoom;
DROP TABLE IF EXISTS ArtInGallery;
DROP TABLE IF EXISTS Artist;
DROP TABLE IF EXISTS MadeBy;
DROP TABLE IF EXISTS Owner;
DROP TABLE IF EXISTS Ownership;
DROP VIEW IF EXISTS ArtPieceInfo;
DROP VIEW IF EXISTS AncientPaintings;
DROP VIEW IF EXISTS UsersGalleryRatings;
DROP TRIGGER IF EXISTS T1;
DROP TRIGGER IF EXISTS T2;
DROP INDEX IF EXISTS ArtistKeyIndex;
DROP INDEX IF EXISTS OwnerKeyIndex;
DROP INDEX IF EXISTS ArtPieceKeyIndex;
DROP INDEX IF EXISTS UserKeyIndex;
DROP INDEX IF EXISTS GalleryKeyIndex;
DROP INDEX IF EXISTS GalleryRoomKeyIndex;
DROP INDEX IF EXISTS OwnershipIndex;
DROP INDEX IF EXISTS ArtInGalleryIndex;
```