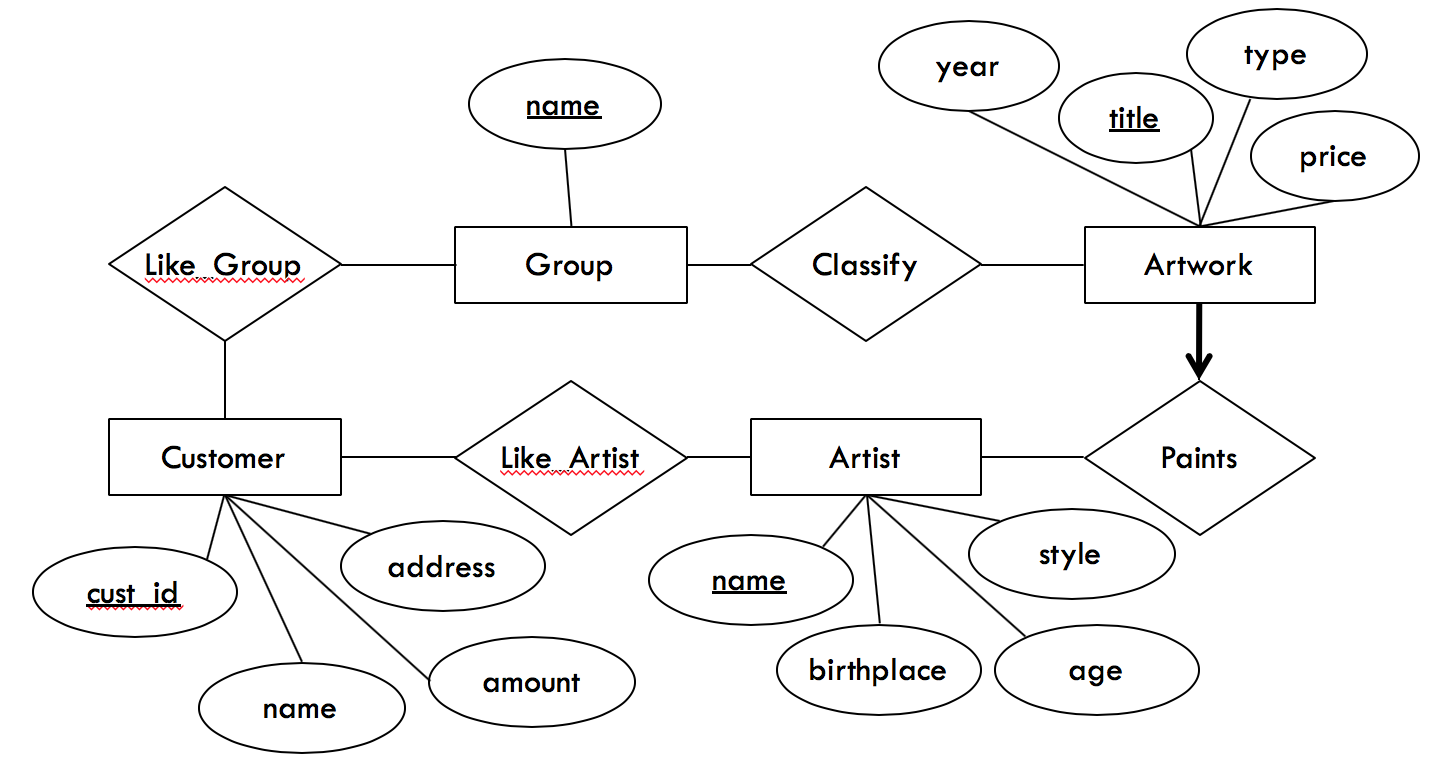
**Lesson: Building SQL**

**Description:** Although you always wanted to be an artist, you ended up being an expert on databases because you love to cook data and you somehow confused database with data baste. Your old love is still there, however, so you set up a database company, ArtBase, that builds a product for art galleries. The core of this product is a database with a schema that captures all the information that galleries need to maintain.

Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art. For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, lithograph, sculpture, photograph), and its price must be stored. Pieces of artwork are also classified into groups of various kinds, for example, portraits, still lives, works by Picasso, or works of the 19th century; a given piece may belong to more than one group.

Each group is identified by a name (like those just given) that describes the group. Finally, galleries keep information about customers. For each customer, galleries keep that person’s unique name, address, total amount of dollars spent in the gallery (very important!), and the artists and groups of art that the customer tends to like.

The Database Designers have sketched the following ER diagram for this company.



As a database developer, you are responsible to implement the SQL database for this ERD answering the following questions

1. Identify the relationships between entities.
2. Identify and data types for each attribute.

For the following steps, we will use the MySQL Command Line Client tool, which is already installed in the lab machine. To install to your own machine, please follow the steps in the document titled *MySQL-Setup-Instructions.PDF* on Blackboard.

1. Write SQL statements to create the corresponding relations and capture as many of the constraints as possible. If you cannot capture some constraints, explain why.
2. First, begin with the entities. Translating these to SQL is straightforward. Write SQL statements to create tables for each entity.

1. Next, translate the relationships. Write SQL statements to create tables for each relationship.

*Hints:* Identify the primary keys and foreign keys for these relationship tables.

1. Now, change the “Customer” table by adding a new column “phone”
2. Insert the following employees to the table “Customer”
   * + 38214, ‘Leterski’, $55,000, ‘27 Baker Street’, 518-9987
     + 54907, ‘Altvater’, $70,000, ‘1400 Washington Avenue’, 518-4455,
     + 74632, ‘Altvater’, $48,000, ‘71 Fuller Road’, 438-2211
3. Change the name of the customer 38214 to ‘Letersky’
4. Write an SQL query to show the average amount of the customers
5. Write an SQL query to show the maximum salary of the customers
6. Write an SQL query to show the customers whose phone number starts with ‘518’

Hint: you may use *like.*

1. BONUS: Insert the following information to the table “Artwork” **in batch**
   * + 2012, ‘Life is beautiful’, ‘abstract’, $114,000
     + 2013, ‘Revenge’, ‘modern’, $84,000
     + 2009, ‘Double diamond’, ‘Contemporary’, $154,080
     + 2001, ‘Rhythm’, ‘surrealism’, $101,200
     + 2018, ‘Freeway’, ‘modern’, $214,000
     + 2003, ‘Independence’, ‘abstract’, $178,000
2. Copy all your SQL statements in a text file and submit it to Blackboard.
3. Write a code to delete/drop your tables and schema.