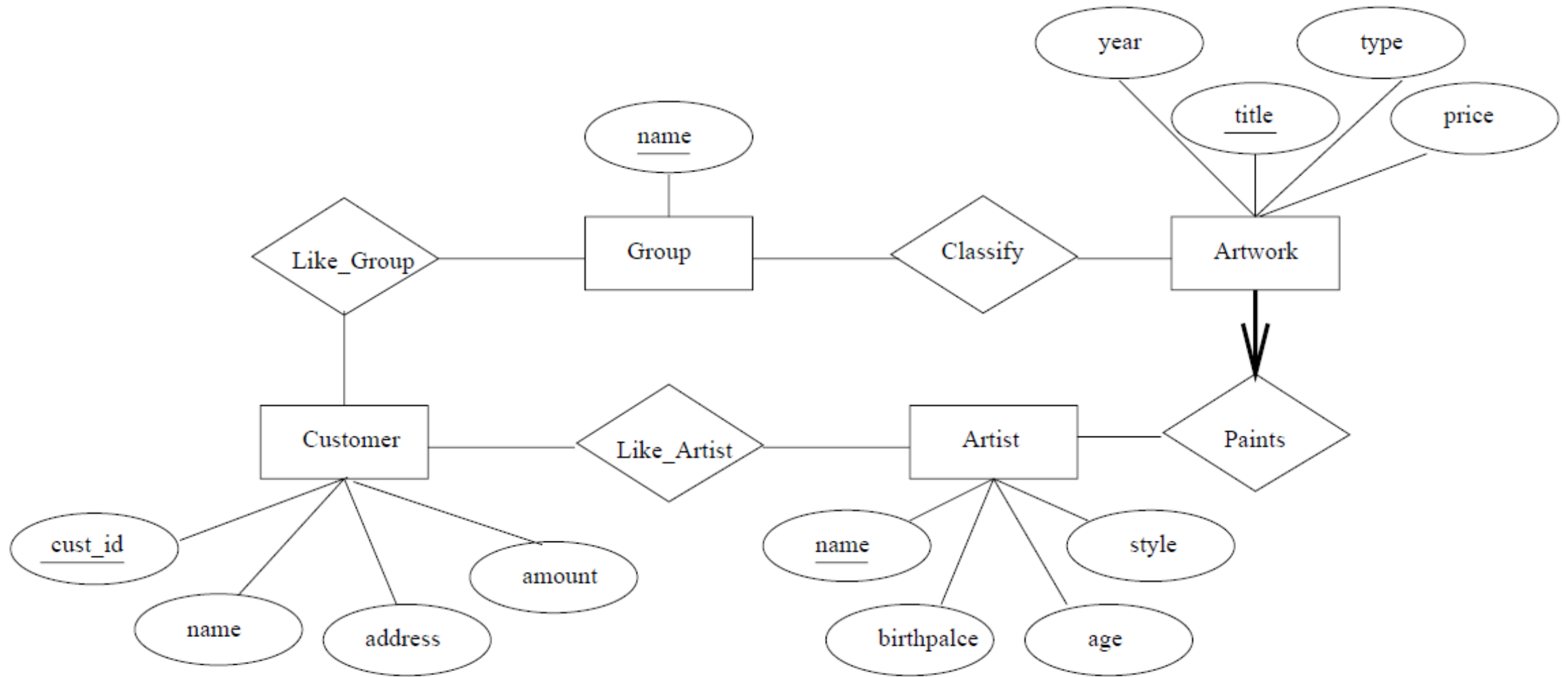


Exercise 2.8

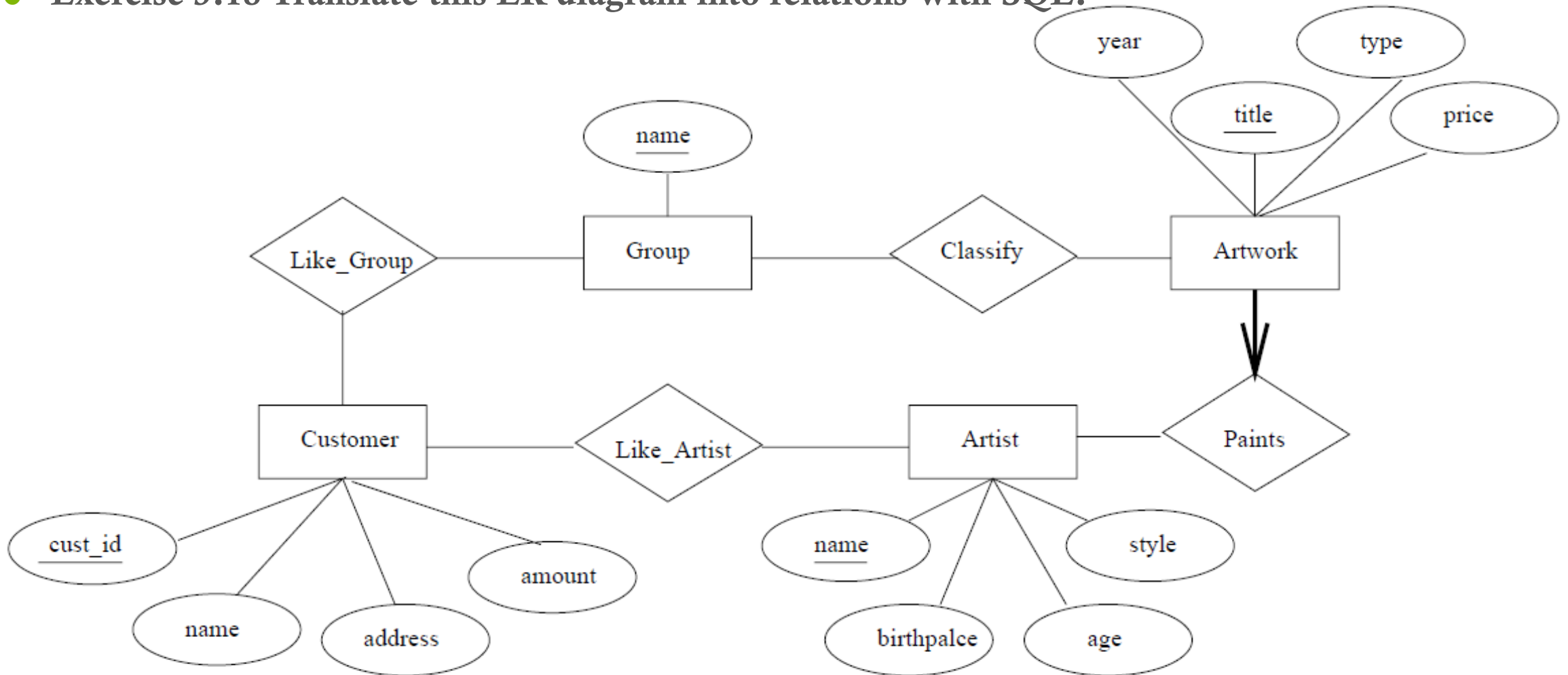
- ◆ **Exercise 2.8** Although you always wanted to be an artist, you ended up being an expert on databases because you love to cook data and you somehow confused *database* with *data baste*. Your old love is still there, however, so you set up a database company, ArtBase, that builds a product for art 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 who painted it, 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 lifes, 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 id, 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.

Exercise 2.8



Exercise 3.18

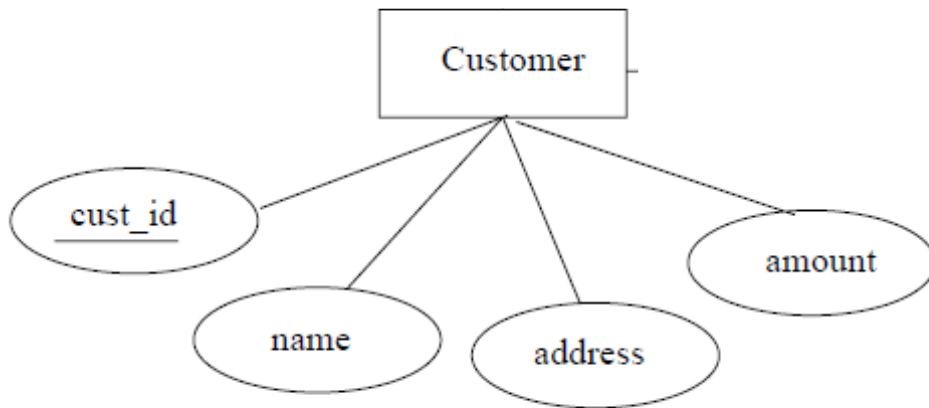
- Exercise 3.18 Translate this ER diagram into relations with SQL.



Exercise 3.18

- Exercise 3.18 Translate this ER diagram into relations with SQL.

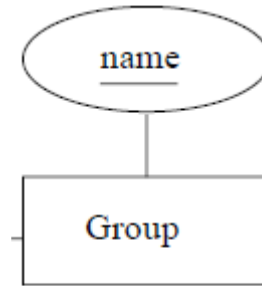
```
CREATE TABLE Customers
(  
    cust_id CHAR(20),  
    name     CHAR(20),  
    address  CHAR(50),  
    amount   REAL,  
    PRIMARY KEY (cust_id)  
)
```



Exercise 3.18

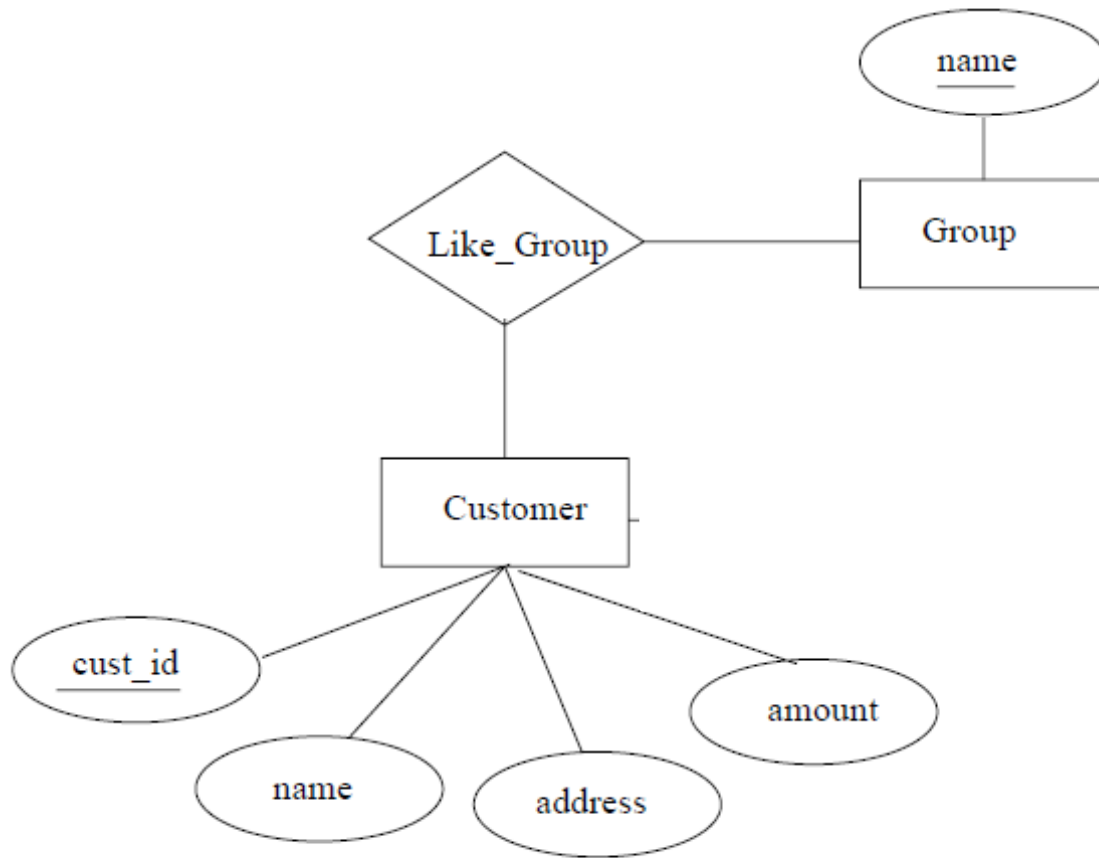
- Exercise 3.18 Translate this ER diagram into relations with SQL.

```
CREATE TABLE Groups
(  
  name      CHAR(20),  
  PRIMARY KEY (name)  
)
```



Exercise 3.18

- Exercise 3.18 Translate this ER diagram into relations with SQL.

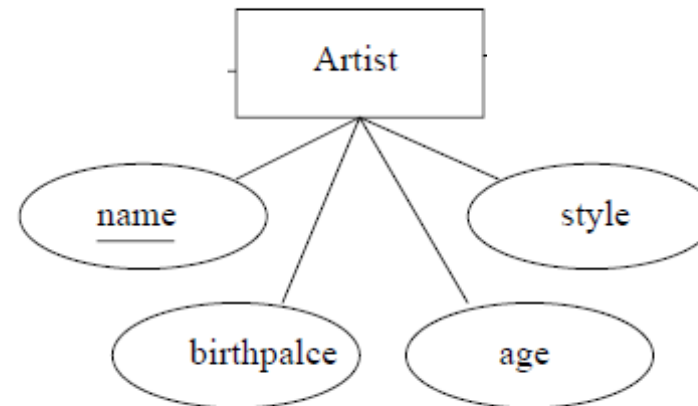


```
CREATE TABLE Like_Group
(
    name      CHAR(20),
    cust_id   CHAR(20),
    PRIMARY KEY (name, cust_id),
    FOREIGN KEY (name) REFERENCES Groups,
    FOREIGN KEY (cust_id) REFERENCES Customers
)
```

Exercise 3.18

- Exercise 3.18 Translate this ER diagram into relations with SQL.

```
CREATE TABLE Artists
(  
  name          CHAR(20),  
  birthplace    CHAR(30),  
  age           INTEGER,  
  style         CHAR(30),  
  PRIMARY KEY (name)  
)
```

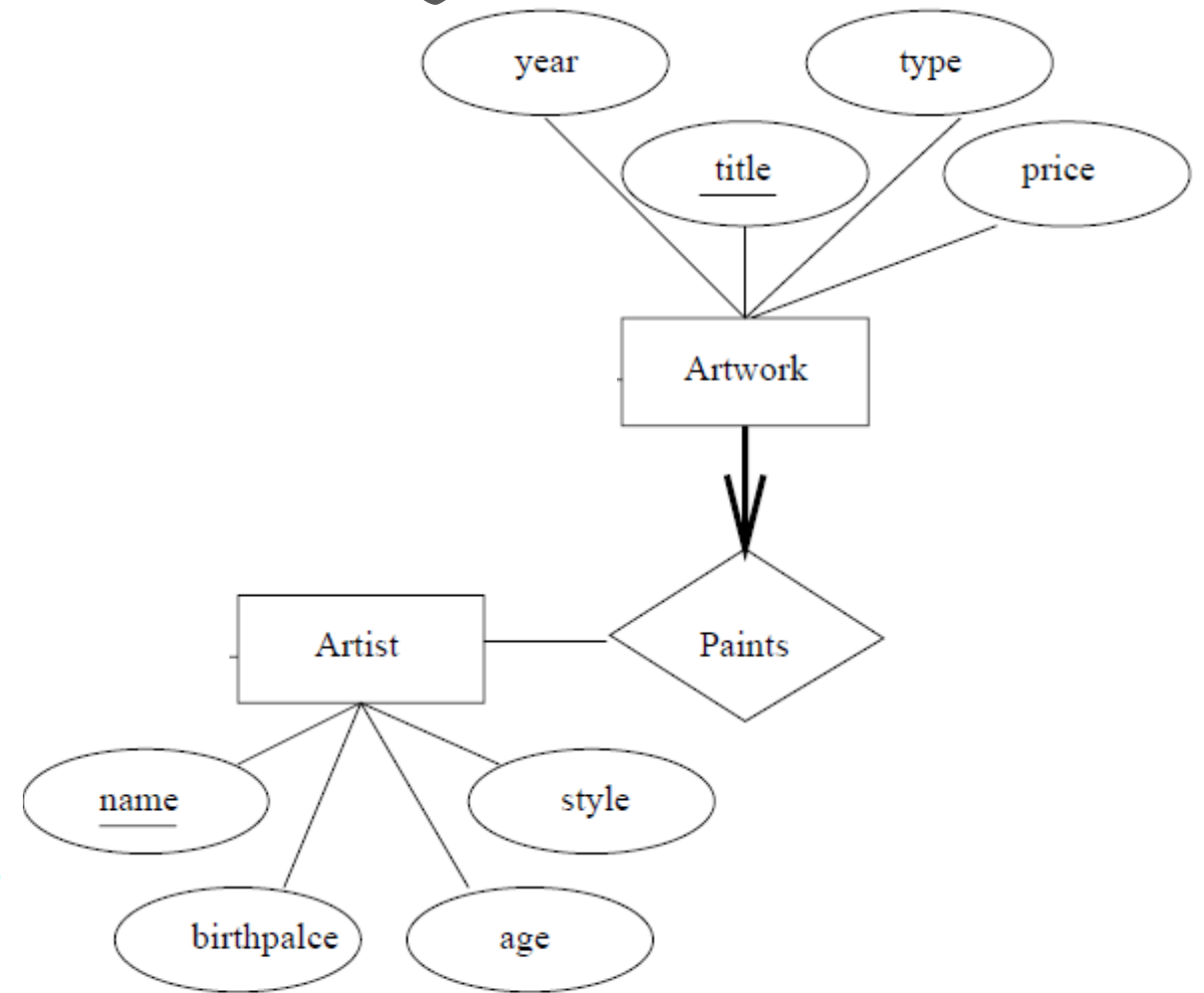


Exercise 3.18

- Exercise 3.18 Translate this ER diagram into relations with SQL.

```
CREATE TABLE Artworks
(  
  title    CHAR(20),  
  year     INTEGER,  
  type     CHAR(30),  
  price    REAL,  
  PRIMARY KEY (title)  
)
```

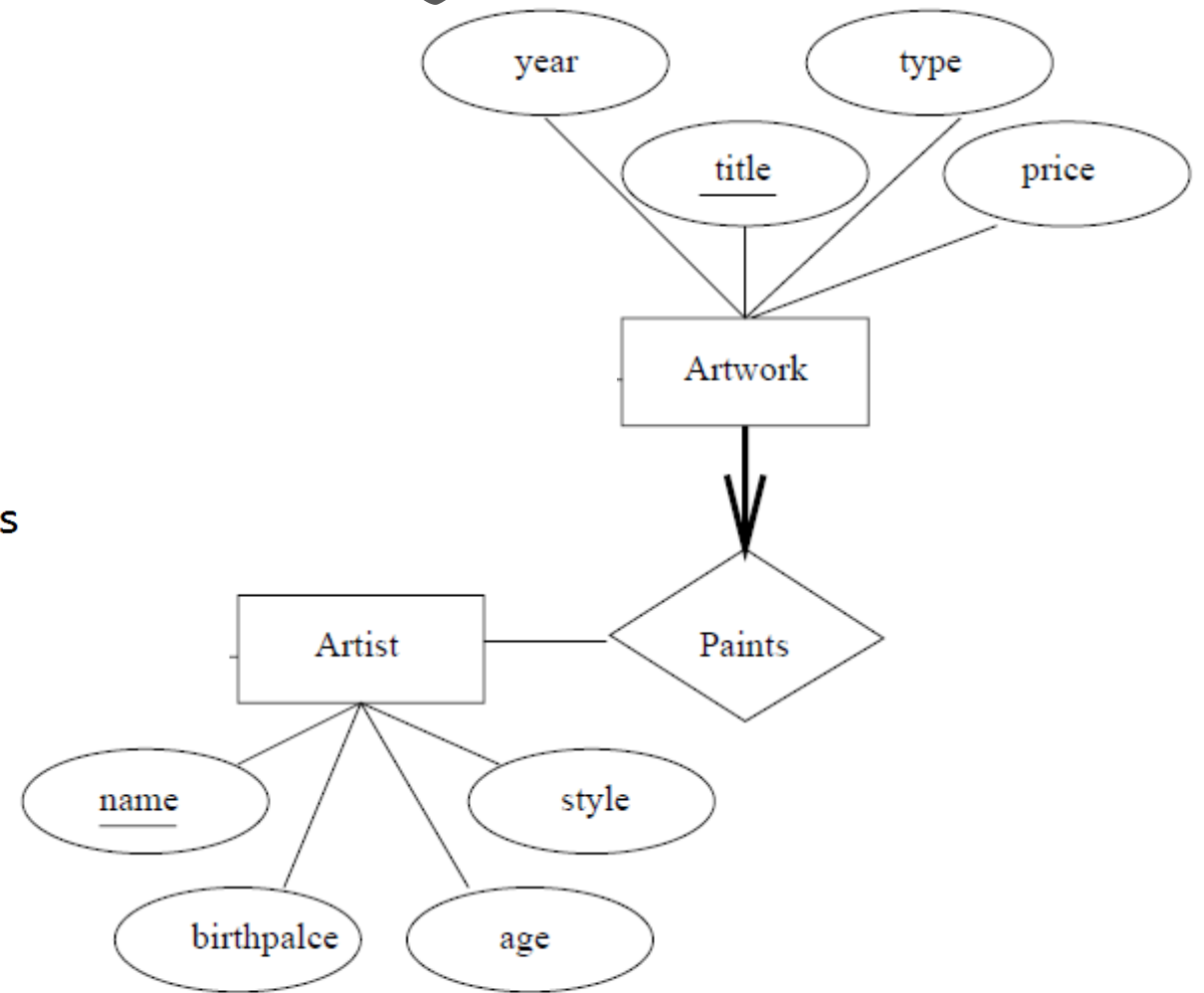
```
CREATE TABLE Paints
(  
  title    CHAR(20),  
  name     CHAR(20),  
  PRIMARY KEY (title),  
  FOREIGN KEY (name) REFERENCES Artists,  
  FOREIGN KEY (title) REFERENCES Artworks  
)
```



Exercise 3.18

- Exercise 3.18 Translate this ER diagram into relations with SQL.

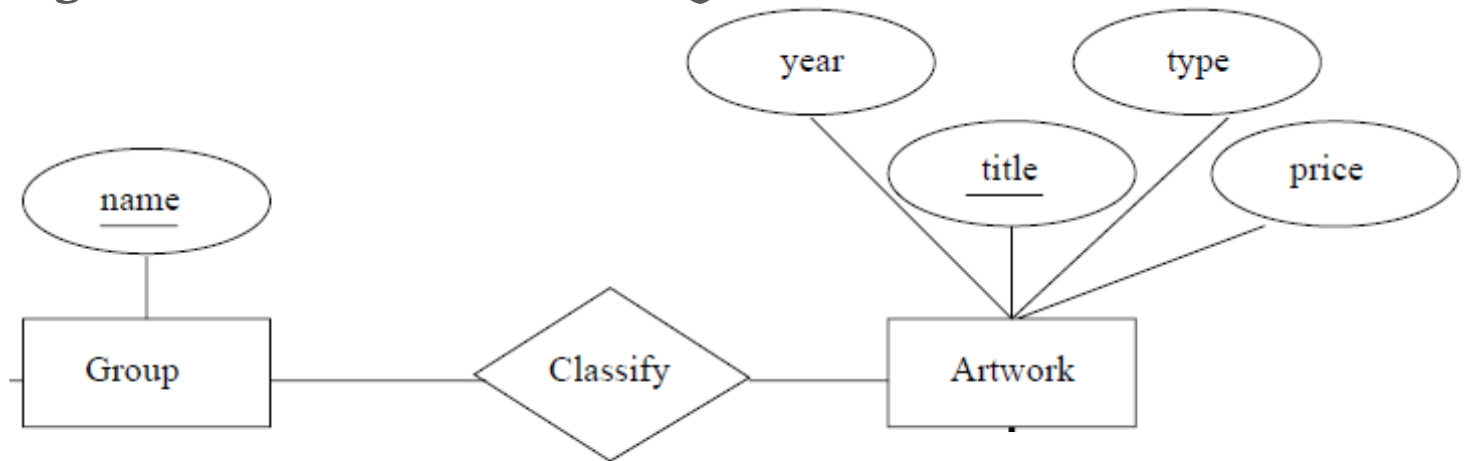
```
CREATE TABLE Artwork_Paints
(  
  title    CHAR(20),  
  name     CHAR(20) NOT NULL,  
  type     CHAR(30),  
  price    REAL,  
  year     INTEGER,  
  PRIMARY KEY (title),  
  FOREIGN KEY (name) REFERENCES Artists  
)
```



Exercise 3.18

💧 Exercise 3.18 Translate this ER diagram into relations with SQL.

```
CREATE TABLE Classify
(  
  name      CHAR(20),  
  title     CHAR(20)  
  PRIMARY KEY (name,title),  
  FOREIGN KEY (name) REFERENCES Groups,  
  FOREIGN KEY (title) REFERENCES Artworks  
)
```



```
CREATE TABLE Like_Artist
(  
  cust_id CHAR(20),  
  name    CHAR(20)  
  PRIMARY KEY (cust_id,name),  
  FOREIGN KEY (cust_id) REFERENCES Customers,  
  FOREIGN KEY (name) REFERENCES Artists  
)
```

