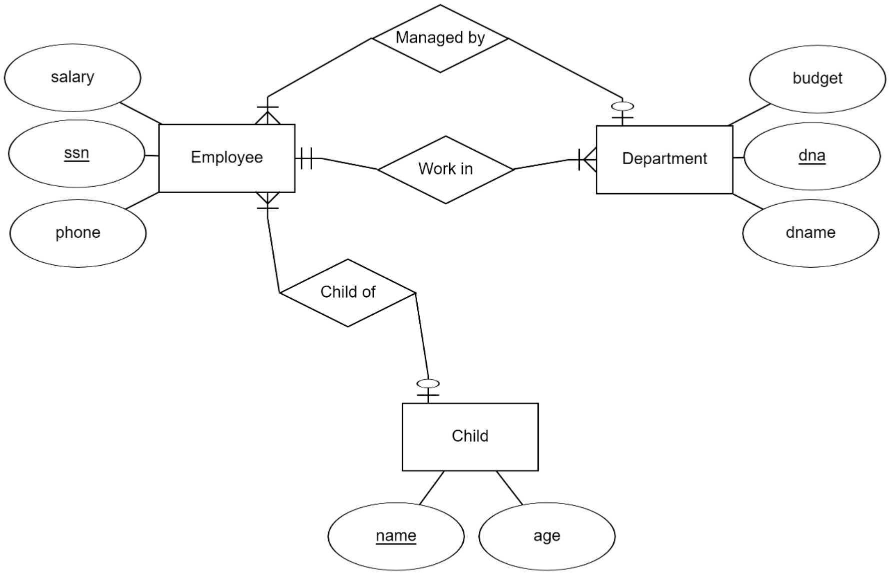
Alejandro Sherman
Lab Section 22

CS166 HW 1 ER Design and translation to Relational model

Also included in my submission with this document are my ER diagrams as image files, and my sql instructions placed into an .sql file.

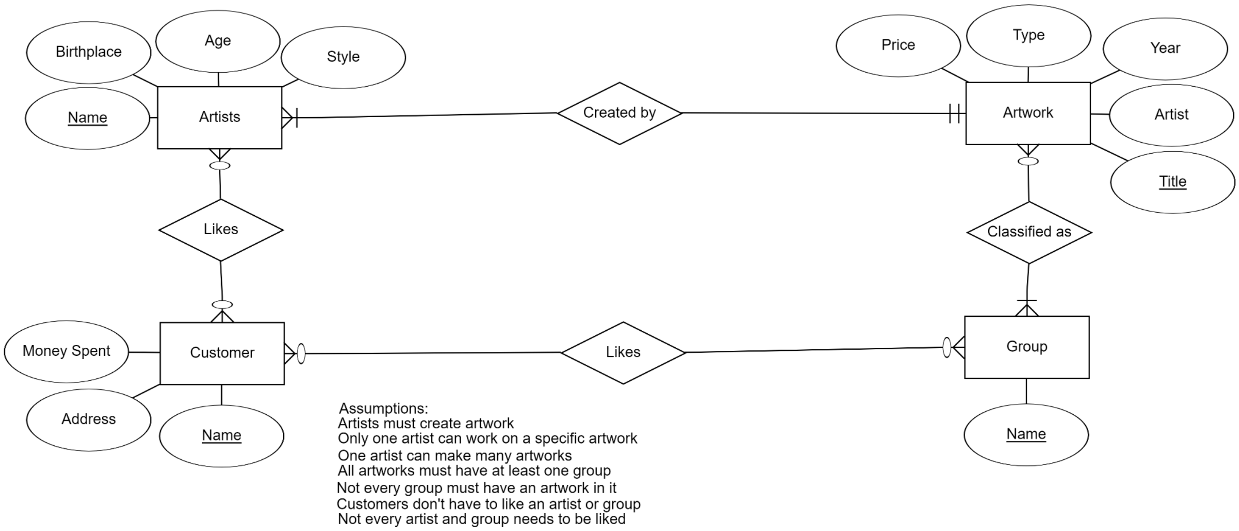
Exercise 2.4: Employee ER Diagram.



Assumtions:
An employee can only work in one department
A department can be managed by one employee
Multiple departments can be managed by the same employee
An employee can have multiple children

Please note that I utilized ERDPlus for my ER diagram. As such I utilized the format for ERDPlus in selecting the attributes for each entity connected by a relationship. For example, in this example, for the Managed by Relation, the Employee entity is set to "Optional" and "One" while the Department entity is set to "Mandatory" and "Many". ERDPlus updates the opposite arrow

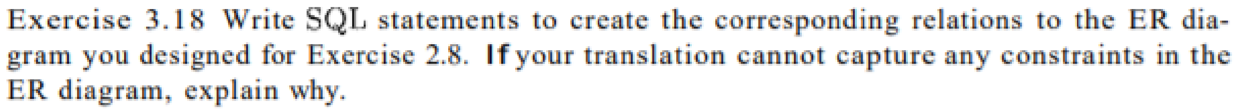
Exercise 2.8: Artist ER Diagram



Assumptions:
Artists must create artwork
Only one artist can work on a specific artwork
One artist can make many artworks
All artworks must have at least one group
Not every group must have an artwork in it
Customers don't have to like an artist or group
Not every artist and group needs to be liked

Please note that I utilized ERDPlus for my ER diagram. As such I utilized the format for ERDPlus in selecting the attributes for each entity connected by a relationship. For example, in this example, for the Created by Relation, the Artists entity is set to "Mandatory" and "One" while the Artwork entity is set to "Mandatory" and "Many". ERDPlus updates the opposite arrow

Exercise 3.18

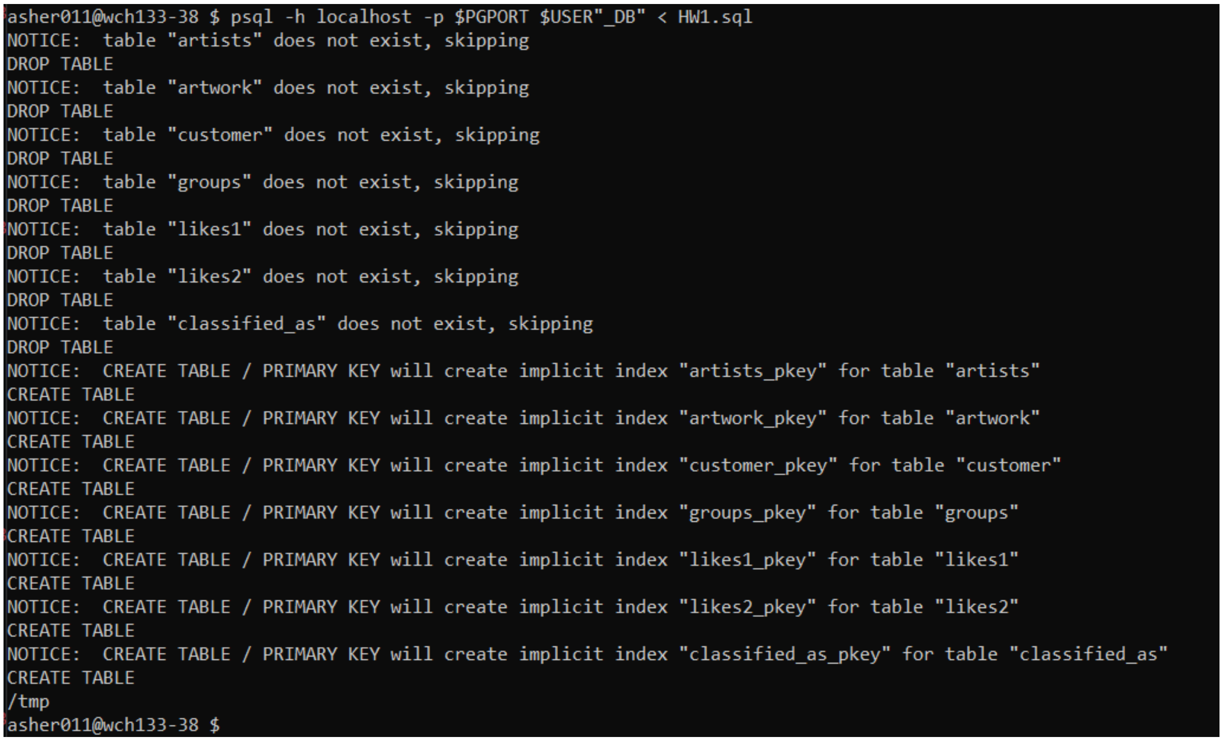


Exercise 3.18 Write SQL statements to create the corresponding relations to the ER diagram you designed for Exercise 2.8. If your translation cannot capture any constraints in the ER diagram, explain why.

SQL Statements to create the corresponding relations to my 2.8 ER Diagram above:
All constraints from the ER diagram can be translated.
Note that each name was differentiated with a subsequent letter corresponding to the entity. For instance, the name belonging to Artists is name_a. Additionally, each "Likes" relation is labeled as "1" and "2", with Likes1 referring to the relationship between Artists and Customer, and Likes2 referring to the relation between Customer and Group. Additionally, "Group" has been changed to "Groups" to avoid syntax errors.

```
DROP TABLE if EXISTS Artists CASCADE;
DROP TABLE if EXISTS Artwork CASCADE;
DROP TABLE if EXISTS Customer CASCADE;
DROP TABLE if EXISTS Groups CASCADE;
DROP TABLE if EXISTS Likes1 CASCADE;
DROP TABLE if EXISTS Likes2 CASCADE;
DROP TABLE if EXISTS Classified_as CASCADE;

CREATE TABLE Artists (
        name_a CHAR(30) NOT NULL,
        birthplace CHAR(30),
        age INTEGER,
        style CHAR(30),
        PRIMARY KEY(name_a));

CREATE TABLE Artwork (
        title CHAR(30) NOT NULL,
        price INTEGER,
        type CHAR(30),
        year DATE,
        artist CHAR(30),
        created_by CHAR(30),
        PRIMARY KEY(title),
        FOREIGN KEY(created_by) REFERENCES Artists(name_a));

CREATE TABLE Customer (
        name_c CHAR(30) NOT NULL,
        address CHAR(30),
        money_spent INTEGER,
        PRIMARY KEY(name_c));

CREATE TABLE Groups (
        name_g CHAR(30) NOT NULL,
        PRIMARY KEY(name_g));

CREATE TABLE Likes1 (
        name_a CHAR(30) NOT NULL,
        name_c CHAR(30) NOT NULL,
        PRIMARY KEY(name_a, name_c),
        FOREIGN KEY(name_a) REFERENCES Artists(name_a),
        FOREIGN KEY(name_c) REFERENCES Customer(name_c));

CREATE TABLE Likes2 (
        name_c CHAR(30) NOT NULL,
        name_g CHAR(30) NOT NULL,
        PRIMARY KEY(name_c, name_g),
        FOREIGN KEY(name_c) REFERENCES Customer(name_c),
        FOREIGN KEY(name_g) REFERENCES Groups(name_g));

 CREATE TABLE Classified_as (
        title CHAR(30) NOT NULL,
        name_g CHAR(30) NOT NULL,
        PRIMARY KEY(title, name_g),
        FOREIGN KEY(title) REFERENCES Artwork(title),
        FOREIGN KEY(name_g) REFERENCES Groups(name_g));
```

Sample running of these sql instructions on a lab machine. These instructions were placed into an sql file called "HW.sql"



```
asher011@wch133-38 $ psql -h localhost -p $PGPORT $USER"_DB" < HW1.sql
NOTICE:  table "artists" does not exist, skipping
DROP TABLE
NOTICE:  table "artwork" does not exist, skipping
DROP TABLE
NOTICE:  table "customer" does not exist, skipping
DROP TABLE
NOTICE:  table "groups" does not exist, skipping
DROP TABLE
NOTICE:  table "likes1" does not exist, skipping
DROP TABLE
NOTICE:  table "likes2" does not exist, skipping
DROP TABLE
NOTICE:  table "classified_as" does not exist, skipping
DROP TABLE
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "artists_pkey" for table "artists"
CREATE TABLE
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "artwork_pkey" for table "artwork"
CREATE TABLE
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "customer_pkey" for table "customer"
CREATE TABLE
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "groups_pkey" for table "groups"
CREATE TABLE
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "likes1_pkey" for table "likes1"
CREATE TABLE
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "likes2_pkey" for table "likes2"
CREATE TABLE
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "classified_as_pkey" for table "classified_as"
CREATE TABLE
/tmp
asher011@wch133-38 $
```