-- Database: Pingpong

-- DROP DATABASE IF EXISTS "Pingpong";

CREATE DATABASE "Pingpong"

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'English\_United States.1252'

LC\_CTYPE = 'English\_United States.1252'

LOCALE\_PROVIDER = 'libc'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;

-- USER TABLE

-- Create the address type

CREATE TYPE address\_type AS (

street VARCHAR(30),

city VARCHAR(10),

state VARCHAR(10),

zip INT

);

-- Create the appuser table

CREATE TABLE appuser (

ID SERIAL PRIMARY KEY,

first\_name VARCHAR(18),

last\_name VARCHAR(18),

date\_of\_birth DATE,

mobile\_no INT[],

bio VARCHAR(150),

password VARCHAR(64),

image BYTEA,

skill\_title VARCHAR(10),

skill\_level VARCHAR(20) CHECK (skill\_level IN ('beginner', 'intermediate', 'advanced', 'pro')),

address address\_typE,

OWNED\_BADGES INT REFERENCES BADGES(ID);

);

-- Create the Age function

CREATE OR REPLACE FUNCTION calculate\_age(date\_of\_birth DATE) RETURNS INT AS $$

DECLARE

age INT;

BEGIN

SELECT EXTRACT(YEAR FROM AGE(CURRENT\_DATE, date\_of\_birth)) INTO age;

RETURN age;

END;

$$ LANGUAGE plpgsql;

-- CREATE PROJECT TABLE

-- Create the composite type with user\_id as a foreign key

CREATE TYPE contribution\_info AS (

user\_id INT,

contribution\_percentage INT

);

-- Create the Project table

CREATE TABLE Project (

ID SERIAL PRIMARY KEY,

name VARCHAR(30),

date\_of\_creation DATE,

visibility VARCHAR(10) CHECK (visibility IN ('public', 'enlisted', 'private')),

vote INT,

user\_id INT[], -- Define user\_id as an INT array with foreign key constraint

contribution\_info contribution\_info[]

);

-- CREATE FOLDER

-- Create the Folder table

CREATE TABLE Folder (

ID SERIAL PRIMARY KEY,

NAME VARCHAR(18),

PROJECT\_ID INT REFERENCES Project(ID),

FOLDER\_ID INT REFERENCES Folder(ID)

);

-- CREATE TABLE TEXT

CREATE TABLE TEXTPROJECT (

ID SERIAL PRIMARY KEY,

NAME VARCHAR(30),

BODY TEXT, -- Change the data type to TEXT to accommodate larger files

EXTENSION VARCHAR(10)

);

--CREATE TABLE FILE

CREATE TABLE FILEPROJECT (

ID SERIAL PRIMARY KEY,

NAME VARCHAR(30),

BLOB BYTEA, -- BLOB column to store binary data of any type

EXTENSION VARCHAR(10)

);

--CREATE TABLE CHATUSER

CREATE TABLE CHATUSER (

ID SERIAL PRIMARY KEY,

MESSAGE TEXT,

DATETIME TIMESTAMP,

SENDER INT REFERENCES AppUser(ID),

RECEIVER INT REFERENCES AppUser(ID)

);

CREATE TABLE BLOGS (

ID SERIAL PRIMARY KEY,

MESSAGE TEXT,

IMAGE BYTEA -- Change to BYTEA for storing binary data

USER\_ID INT REFERENCES AppUser(ID)

);

CREATE TABLE REPLY (

ID SERIAL PRIMARY KEY,

MESSAGE TEXT,

USER\_ID INT REFERENCES AppUser(ID),

REPLY\_OF\_BLOG INT REFERENCES BLOGS(ID),

REPLY\_OF\_REPLY INT REFERENCES REPLY(ID)

);

CREATE TABLE BADGES (

ID SERIAL PRIMARY KEY,

TITLE VARCHAR(18),

LEVEL INT

);

SELECT \* FROM APPUSER;

SELECT \* FROM PROJECT;

SELECT \* FROM FOLDER;

SELECT \* FROM TEXTPROJECT;

SELECT \* FROM FILEPROJECT;

SELECT \* FROM CHATUSER;

SELECT \* FROM BLOGS;

SELECT \* FROM REPLY;

SELECT \* FROM BADGES;