```
CREATE SCHEMA IF NOT EXISTS swoosh;
--DROP TABLES in specific order to avoid foreign key issues
--Relationships
DROP TABLE IF EXISTS swoosh.ThumbsUp;
DROP TABLE IF EXISTS swoosh.Teaches;
DROP TABLE IF EXISTS swoosh.EnrolledIn;
DROP TABLE IF EXISTS swoosh.WatchedSegment;
DROP TABLE IF EXISTS swoosh.Watched;
DROP TABLE IF EXISTS swoosh.Attended;
--Entities
DROP TABLE IF EXISTS swoosh.PostTopics;
DROP TABLE IF EXISTS swoosh.Post;
DROP TABLE IF EXISTS swoosh.Recording;
DROP TABLE IF EXISTS swoosh.Meeting;
DROP TABLE IF EXISTS swoosh.Recurrence;
DROP TYPE IF EXISTS swoosh.week_days;
DROP TABLE IF EXISTS swoosh.Course;
DROP TABLE IF EXISTS swoosh.InstructorEducation;
DROP TABLE IF EXISTS swoosh.Instructor;
DROP TABLE IF EXISTS swoosh.Student;
DROP TABLE IF EXISTS swoosh.Users;


--------------
-- Entities
--------------
CREATE TABLE swoosh.Users (
      user_id text,
      email text NOT NULL,
      first_name text NOT NULL,
      last_name text NOT NULL,
      PRIMARY KEY(user_id)
);

CREATE TABLE swoosh.Student (
      user_id text,
      occupation text,
      PRIMARY KEY(user_id),
      --isA User
      FOREIGN KEY(user_id) REFERENCES swoosh.Users(user_id) ON DELETE CASCADE
);
```

```
CREATE TABLE swoosh.Instructor (
      user_id text,
      title text,
      PRIMARY KEY(user_id),
      --isA User
      FOREIGN KEY(user_id) REFERENCES swoosh.Users(user_id) ON DELETE CASCADE
);

--Multivalued attribute "Education" of Instructor
CREATE TABLE swoosh.InstructorEducation (
      instructor_id text,
      education_id text,
      degree text,
      major text,
      school text,
      graduation_year integer,
      PRIMARY KEY(instructor_id, education_id),
      FOREIGN KEY(instructor_id) REFERENCES swoosh.Instructor(user_id) ON DELETE CASCADE
);

CREATE TABLE swoosh.Course (
      course_id text,
      course_name text NOT NULL,
      description text,
      PRIMARY KEY(course_id)
);

CREATE TYPE swoosh.week_days AS ENUM('Mon', 'Tue','Wed', 'Thu','Fri','Sat','Sun');

CREATE TABLE swoosh.Recurrence (
recurr_id text,
repeat_on week_days NOT NULL,
end_date DATE NOT NULL,
PRIMARY KEY(recurr_id)
);

CREATE TABLE swoosh.Meeting (
      meeting_id text,
      meeting_name text NOT NULL,
      passcode text,
      start_at timestamp NOT NULL,
      duration integer NOT NULL,
      mute_participants boolean NOT NULL,
```

```
      course_id text NOT NULL, -- Course--<Associated>--Meeting, total participation
      instructor_id text NOT NULL, --Meeting--<HostedBy>--Instructor, total participation
      recurr_id text, --Meeting--<RecursOn>--Recurrence
      PRIMARY KEY(meeting_id),
      FOREIGN KEY(course_id) REFERENCES swoosh.Course(course_id) ON DELETE CASCADE,
      FOREIGN KEY(instructor_id) REFERENCES swoosh.Instructor(user_id) ON DELETE CASCADE,
      FOREIGN KEY(recurr_id) REFERENCES swoosh.Recurrence(recurr_id) ON DELETE CASCADE
);

CREATE TABLE swoosh.Recording (
      recording_id text,
      start_time time,
      end_time time,
      meeting_id text NOT NULL, --Recording--<Recorded>--Meeting, total participation
      PRIMARY KEY(recording_id),
      FOREIGN KEY(meeting_id) REFERENCES swoosh.Meeting(meeting_id) ON DELETE CASCADE
);

CREATE TABLE swoosh.Post (
      post_id text,
      post_type text NOT NULL,
      body text,
      created_at timestamp NOT NULL,
      user_id text NOT NULL, -- Post--<PostedBy>--User, total participation
      meeting_id text NOT NULL, -- Post--<PostAbout>--Meeting, total participation
      replied_to_post_id text, -- Post--<RepliedTo>--Post
      PRIMARY KEY(post_id),
      FOREIGN KEY(user_id) REFERENCES swoosh.Users(user_id) ON DELETE CASCADE,
      FOREIGN KEY(meeting_id) REFERENCES swoosh.Meeting(meeting_id) ON DELETE CASCADE,
      FOREIGN KEY(replied_to_post_id) REFERENCES swoosh.Post(post_id) ON DELETE CASCADE
);

--Multi-valued attribute "topics" for Post
CREATE TABLE swoosh.PostTopics (
      post_id text,
      topic text,
      PRIMARY KEY(post_id, topic),
      FOREIGN KEY(post_id) REFERENCES swoosh.Post(post_id) ON DELETE CASCADE
);
```

```
----------------------
-- Relationships
----------------------

-- Student <-- Attended --> Meeting
CREATE TABLE swoosh.Attended(
      user_id text,
      meeting_id text,
      PRIMARY KEY(user_id, meeting_id),
      FOREIGN KEY(meeting_id) REFERENCES swoosh.Meeting(meeting_id) ON DELETE CASCADE,
      FOREIGN KEY(user_id) REFERENCES swoosh.Student(user_id) ON DELETE CASCADE
);


-- Student <-- Watched --> Recording
CREATE TABLE swoosh.Watched(
      recording_id text,
      user_id text,
      PRIMARY KEY(recording_id, user_id),
      FOREIGN KEY(user_id) REFERENCES swoosh.Student(user_id) ON DELETE CASCADE,
      FOREIGN KEY(recording_id) REFERENCES swoosh.Recording(recording_id) ON DELETE CASCADE
);

-- Watched Segment Multi-valued attribute
CREATE TABLE swoosh.WatchedSegment(
      recording_id text,
      user_id text,
      segment_id text,
      watched_from time NOT NULL,
      watched_to time NOT NULL,
      PRIMARY KEY(recording_id, user_id,segment_id),
      FOREIGN KEY(user_id,recording_id) REFERENCES swoosh.Watched(user_id,recording_id) ON DELETE
CASCADE
);

-- Student <-- EnrolledIn --> Course
CREATE TABLE swoosh.EnrolledIn(
      user_id text,
      course_id text,
      enroll_date date NOT NULL,
      PRIMARY KEY(user_id, course_id),
      FOREIGN KEY(user_id) REFERENCES swoosh.Student(user_id) ON DELETE CASCADE,
      FOREIGN KEY(course_id) REFERENCES swoosh.Course(course_id) ON DELETE CASCADE
```

```
);

-- Instructor <-- Teaches --> Course
CREATE TABLE swoosh.Teaches(
      user_id text,
      course_id text,
      PRIMARY KEY(user_id, course_id),
      FOREIGN KEY(user_id) REFERENCES swoosh.Instructor(user_id) ON DELETE CASCADE,
      FOREIGN KEY(course_id) REFERENCES swoosh.Course(course_id) ON DELETE CASCADE
);

-- User <-- ThumbsUp --> Post
CREATE TABLE swoosh.ThumbsUp(
      user_id text,
      post_id text,
      PRIMARY KEY(user_id, post_id),
      FOREIGN KEY(user_id) REFERENCES swoosh.Users(user_id) ON DELETE CASCADE,
      FOREIGN KEY(post_id) REFERENCES swoosh.Post(post_id) ON DELETE CASCADE
);
```