```sql
DROP TABLE IF EXISTS EVENT;
DROP TABLE IF EXISTS OFFICIAL;
DROP TABLE IF EXISTS SPORT;

CREATE TABLE SPORT
(
      SPORTID SERIAL PRIMARY KEY,
      SPORTNAME VARCHAR(100) NOT NULL UNIQUE
);

INSERT INTO SPORT (SPORTNAME) VALUES ('Archery');          -- 1
INSERT INTO SPORT (SPORTNAME) VALUES ('Athletics');        -- 2
INSERT INTO SPORT (SPORTNAME) VALUES ('Badminton');        -- 3
INSERT INTO SPORT (SPORTNAME) VALUES ('Basketball'); -- 4
INSERT INTO SPORT (SPORTNAME) VALUES ('Boxing');           -- 5
INSERT INTO SPORT (SPORTNAME) VALUES ('Diving');           -- 6
INSERT INTO SPORT (SPORTNAME) VALUES ('Fencing');          -- 7
INSERT INTO SPORT (SPORTNAME) VALUES ('Golf');             -- 8
INSERT INTO SPORT (SPORTNAME) VALUES ('Handball');         -- 9
INSERT INTO SPORT (SPORTNAME) VALUES ('Hockey');           -- 10
INSERT INTO SPORT (SPORTNAME) VALUES ('Ice Hockey'); -- 11
INSERT INTO SPORT (SPORTNAME) VALUES ('Judo');             -- 12
INSERT INTO SPORT (SPORTNAME) VALUES ('Karate');           -- 13
INSERT INTO SPORT (SPORTNAME) VALUES ('Luge');             -- 14
INSERT INTO SPORT (SPORTNAME) VALUES ('Rowing');           -- 15
INSERT INTO SPORT (SPORTNAME) VALUES ('Rugby');            -- 16
INSERT INTO SPORT (SPORTNAME) VALUES ('Sailing');          -- 17
INSERT INTO SPORT (SPORTNAME) VALUES ('Shooting');         -- 18
INSERT INTO SPORT (SPORTNAME) VALUES ('Snowboard');        -- 19
INSERT INTO SPORT (SPORTNAME) VALUES ('Weightlifting');    -- 20

CREATE TABLE OFFICIAL
(
      OFFICIALID SERIAL PRIMARY KEY,
      USERNAME VARCHAR(20) NOT NULL UNIQUE,
      FIRSTNAME VARCHAR(50) NOT NULL,
      LASTNAME VARCHAR(50) NOT NULL,
      PASSWORD VARCHAR(20) NOT NULL
);

INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
('-','Not','Assigned','000');            -- 1
INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
('JohnW','John','Waith','999');                 -- 2
INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
('ChrisP','Christopher','Putin','888');  -- 3
INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
('GuoZ','Guo','Zhang','777');            -- 4
INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
('JulieA','Julie','Ahlering','666');           -- 5
INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
('MaksimS','Maksim','Sulejmani','555');  -- 6
INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
('KrisN','Kristina','Ness','444');       -- 7
INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
('ZvonkoO','Zvonko','Ocic','333');       -- 8
INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
('SusanF','Susan','Fischer','222');      -- 9
INSERT INTO OFFICIAL (USERNAME,FIRSTNAME,LASTNAME,PASSWORD) VALUES
```

```
('KevinB','Kevin','Boyd','111');               -- 10

CREATE TABLE EVENT
(
      EVENTID SERIAL PRIMARY KEY,
      EVENTNAME VARCHAR(50) NOT NULL,
      SPORTID INTEGER REFERENCES SPORT,
      REFEREE INTEGER REFERENCES OFFICIAL,
      JUDGE INTEGER REFERENCES OFFICIAL,
      MEDALGIVER INTEGER REFERENCES OFFICIAL

);

INSERT INTO EVENT (EVENTNAME,SPORTID,REFEREE,JUDGE,MEDALGIVER) VALUES ('Men''s
Singles Semifinal',3,2,3,4);        -- 1
INSERT INTO EVENT (EVENTNAME,SPORTID,REFEREE,JUDGE,MEDALGIVER) VALUES ('Women''s
Long Jump Final',2,1,5,6);          -- 2
INSERT INTO EVENT (EVENTNAME,SPORTID,REFEREE,JUDGE,MEDALGIVER) VALUES ('Men''s Team
Semifinal',1,3,4,5);          -- 3
INSERT INTO EVENT (EVENTNAME,SPORTID,REFEREE,JUDGE,MEDALGIVER) VALUES ('Men''s
Tournament Semifinal',4,1,2,6);     -- 4
INSERT INTO EVENT (EVENTNAME,SPORTID,REFEREE,JUDGE,MEDALGIVER) VALUES ('Women''s
Lightweight Final',5,4,6,1);  -- 5

-- FUNCTION: public.searchassociatedevents(integer)

-- DROP FUNCTION public.searchassociatedevents(integer);

CREATE OR REPLACE FUNCTION public.searchassociatedevents(
      official_id integer)
    RETURNS TABLE(peventid integer, peventname character varying, psport character
varying, preferee character varying, pjudge character varying, pmedalgiver
character varying)
    LANGUAGE 'plpgsql'

AS $BODY$
BEGIN
RETURN QUERY
      SELECT e.eventid, e.eventname, s.sportname, r.username, j.username,
m.username
      FROM event e
      LEFT JOIN sport s ON e.sportid = s.sportid -- search sportid
      LEFT JOIN official r ON e.referee = r.officialid -- search referee id
      LEFT JOIN official j ON e.judge = j.officialid -- search judge id
      LEFT JOIN official m ON e.medalgiver = m.officialid -- search medalgiver id
      WHERE e.referee = official_id
      OR e.judge = official_id
      OR e.medalgiver = official_id
      ORDER BY s.sportname;
END;
$BODY$;

-- FUNCTION: public.searchspecifiedevents(character varying)

-- DROP FUNCTION public.searchspecifiedevents(character varying);

CREATE OR REPLACE FUNCTION public.searchspecifiedevents(
      searchstring character varying)
    RETURNS TABLE(peventid integer, peventname character varying, psport character
varying, preferee character varying, pjudge character varying, pmedalgiver
character varying)
    LANGUAGE 'plpgsql'

AS $BODY$
BEGIN
RETURN QUERY
      SELECT e.eventid, e.eventname, s.sportname, r.username, j.username,
m.username
      FROM event e
      LEFT JOIN sport s ON e.sportid = s.sportid -- search sportid
      LEFT JOIN official r ON e.referee = r.officialid -- search referee id
      LEFT JOIN official j ON e.judge = j.officialid -- search judge id
      LEFT JOIN official m ON e.medalgiver = m.officialid -- search medalgiver id
      WHERE CONCAT(e.eventname, s.sportname, r.username, j.username, m.username)
ILIKE CONCAT('%', searchString, '%') --CONCAT:Concatenate two or more strings to
form a single string; ILIKE:Case insensitive
      ORDER BY s.sportname;
END;
$BODY$;


-- FUNCTION: public.searchids(character varying, character varying, character
varying, character varying)

-- DROP FUNCTION public.searchids(character varying, character varying, character
varying, character varying);

CREATE OR REPLACE FUNCTION public.searchids(
      sport_name character varying,
      referee_name character varying,
      judge_name character varying,
      medalgiver_name character varying)
    RETURNS TABLE(sport_id integer, referee_id integer, judge_id integer,
medalgiver_id integer)
    LANGUAGE 'plpgsql'

AS $BODY$
BEGIN
RETURN QUERY
      SELECT (SELECT sportid FROM sport WHERE sportname ILIKE sport_name) AS
sprot_id, -- search sport id
               (SELECT officialid FROM official WHERE username ILIKE referee_name)
AS referee_id, -- search referee id
               (SELECT officialid FROM official WHERE username ILIKE judge_name) AS
judge_id, -- search judge id
               (SELECT officialid FROM official WHERE username ILIKE
medalgiver_name) AS medalgiver_id; -- search medalgiver id
END;
$BODY$;


COMMIT;
```