

# Assignment 4

## DBMS LAB

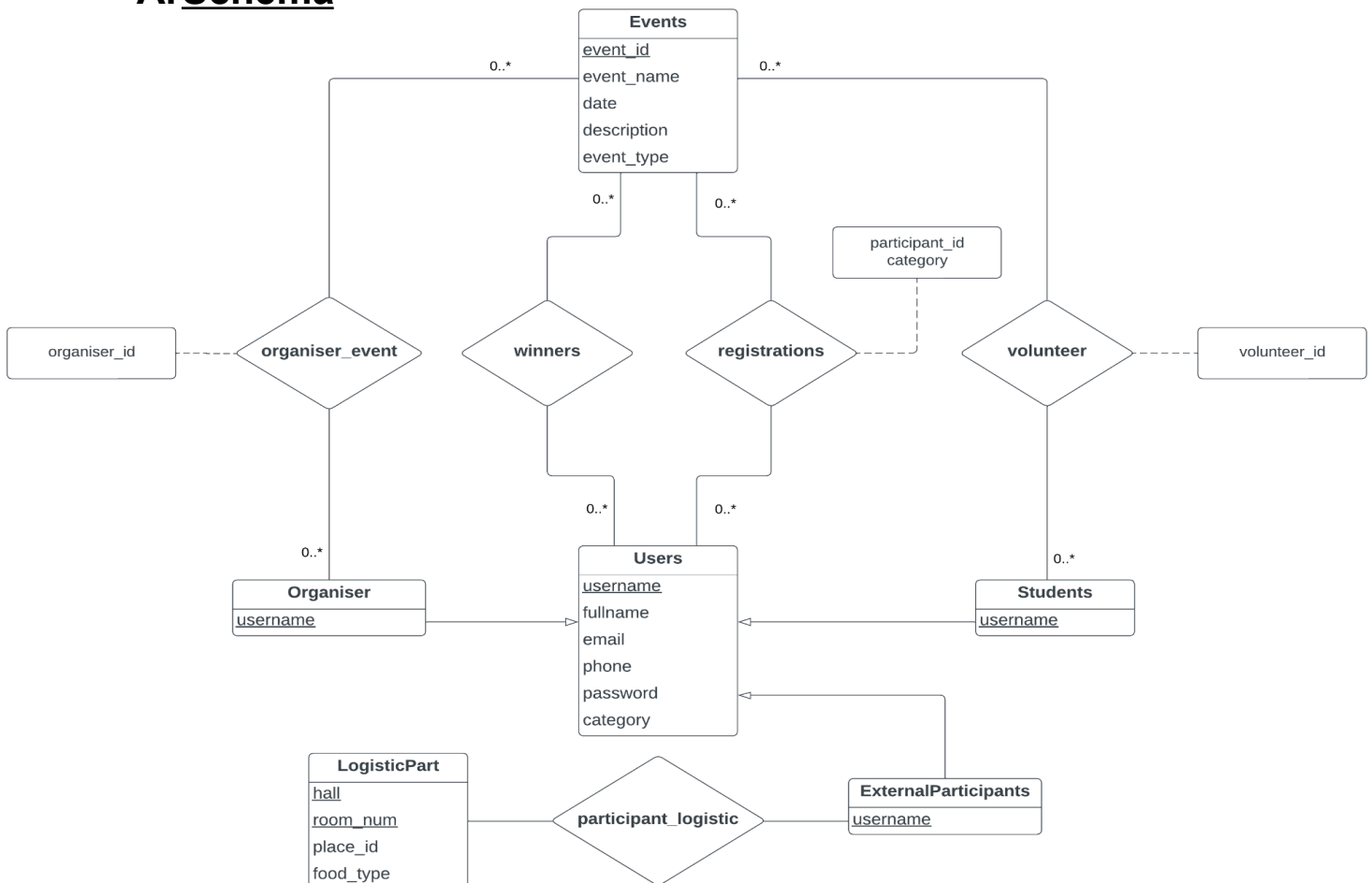
Group Name: SHMAD

Members:

1. 21CS10042 M HARSHITH CHOWDARY
2. 21CS10057 SANSKAR MITTAL
3. 21CS10041 MARLA MAYUKHA
4. 21CS10004 ANIRUDH M
5. 21CS30037 DATTA KSHEERAJ P

## Deliverables

### A. Schema



## B. Triggers

```
CREATE OR REPLACE FUNCTION update_medals_trigger()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.gold = OLD.silver THEN
        NEW.silver = NULL;
        RETURN NEW;
    END IF;
    IF NEW.gold = OLD.bronze THEN
        NEW.bronze = NULL;
        RETURN NEW;
    END IF;
    IF NEW.silver = OLD.gold THEN
        NEW.gold = NULL;
        RETURN NEW;
    END IF;
    IF NEW.silver = OLD.bronze THEN
        NEW.bronze = NULL;
        RETURN NEW;
    END IF;
    IF NEW.bronze = OLD.gold THEN
        NEW.gold = NULL;
        RETURN NEW;
    END IF;
    IF NEW.bronze = OLD.silver THEN
        NEW.silver = NULL;
        RETURN NEW;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER update_medals_trigger
BEFORE INSERT OR UPDATE ON winners
FOR EACH ROW EXECUTE FUNCTION update_medals_trigger();
```

- update\_medals\_trigger() function implements the trigger operation used when assigning winners for a particular event

- It is used to dynamically update the winner table whenever there are changes in the winner section
- For example, if a player changes from gold to silver, this trigger function will be called

## C. Queries

### A. Login

```
-- Select users based on name given in login-page
cur.execute('SELECT users.fullname from users where username = %s',
[username])
```

```
-- Select users with given username and password
cur.execute('SELECT * FROM users WHERE username = %s AND password =
%s', (username, password))
```

### B. Event

```
-- Get the registration details for given event_id and username
cur.execute("SELECT * FROM registrations WHERE event_id = %s AND username =
%s", (event_id1, username))
-- If there's none with given event_id and username
cur.execute('INSERT INTO registrations (username, event_id, category)
VALUES (%s, %s, %s) ', (username, event_id1, 'ExternalParticipant'))
-- Order the events by id
cur.execute('SELECT * FROM events order by event_id')

cur.execute('INSERT INTO events (event_id, event_name, date, description,
event_type)' 'VALUES (%s, %s, %s, %s, %s)', (event_id, event_name, date,
description, event_type))
```

### C. Registration

```
-- User Registration
cur.execute('INSERT INTO users (username, fullname, email, phone, password,
category)' 'VALUES (%s, %s, %s, %s, %s, %s)', (username, fullname, email,
```

```
phone, password, category))
```

```
cur.execute('INSERT INTO students (username, fullname, email, phone, password, category)' 'VALUES (%s, %s, %s, %s, %s, %s)', (username, fullname, email, phone, password, category))
```

```
cur.execute('INSERT INTO externalparticipants (username, fullname, email, phone, password, category)' 'VALUES (%s, %s, %s, %s, %s, %s)', (username, fullname, email, phone, password, category))
```

```
cur.execute('INSERT INTO organiser (username, fullname, email, phone, password, category)' 'VALUES (%s, %s, %s, %s, %s, %s)', (username, fullname, email, phone, password, category))
```

## D. Profile

```
-- User Profile
cur.execute('''\
    UPDATE users
    SET email = %s,
        phone = %s,
        password = %s
    WHERE username = %s
        AND password = %s
        AND category = 'Student'
    ''',
    (emailid, phone, newpass, username, curpass))
```

## E. Winner

```
-- Get winners for each events
cur.execute('''
    SELECT events.event_id, events.event_name, users.username,
    winners.gold, winners.silver, winners.bronze
    FROM events
    JOIN organiser_event ON events.event_id =
    organiser_event.event_id
    JOIN registrations ON events.event_id = registrations.event_id
    JOIN users ON registrations.username = users.username
    JOIN winners ON events.event_id = winners.event_id
    WHERE organiser_event.username = %s
```

```
ORDER BY events.event_id  
'', [username])
```

## F. Volunteer

```
-- Get volunteers for the events  
cur.execute('''  
    SELECT events.event_id, events.event_name, users.username,  
    users.fullname  
    FROM events  
    JOIN volunteer ON events.event_id = volunteer.event_id  
    JOIN users ON volunteer.username = users.username  
''')
```

## G. Organise

```
-- Get organiser(/s) for the events  
cur.execute('SELECT * FROM organiser_event WHERE username = %s and  
event_id = %s', (username, eventid))  
  
cur.execute('INSERT INTO organiser_event (username, event_id) VALUES  
(%s, %s)', (username, eventid))  
-- List all events an organiser is organising  
cur.execute('SELECT * FROM organiser_event WHERE username = %s',  
(username))
```

## H. Logistics

```
-- Set logistic information  
cur.execute('SELECT logisticpart.hall, logisticpart.roomnum,  
logisticpart.foodtype FROM logisticpart WHERE logisticpart.username =  
%s', [username])  
-- Update logisticinfo  
cur.execute('''  
    UPDATE logisticpart  
    SET foodtype = NULL, username = NULL  
    WHERE username = %s;  
    ''', [username])
```

```

-- Get Non-Null users and their logistic information
cur.execute('''
    SELECT users.username, users.fullname, logisticpart.hall,
    logisticpart.roomnum, logisticpart.foodtype
    FROM users
    LEFT OUTER JOIN logisticpart ON users.username =
    logisticpart.username
    WHERE logisticpart.username IS NOT NULL
''')

cur.execute('''SELECT logisticpart.hall, logisticpart.roomnum
    FROM logisticpart
    WHERE logisticpart.username IS NULL
    ORDER BY logisticpart.hall, logisticpart.roomnum;
''')

```

## I. Winner

```

-- Add none winners when an event is added
cur.execute('INSERT INTO winners (event_id, gold, silver, bronze)'
'VALUES (%s, %s, %s, %s)', (event_id, None, None, None))

-- Get event names and winner names from event_id
cur.execute('''
    SELECT events.event_name, winners.gold, winners.silver,
    winners.bronze
    FROM events
    JOIN winners ON events.event_id = winners.event_id
    WHERE events.event_id = %s
''', [event_id2])

-- Update winners
cur.execute('UPDATE winners SET gold = %s WHERE event_id = %s;',
[goldw, eventid])

cur.execute('UPDATE winners SET silver = %s WHERE event_id = %s;',
[silverw, eventid])

```

```

cur.execute('UPDATE winners SET bronze = %s WHERE event_id = %s;')

-- View winners
cur.execute('''
    SELECT events.event_id, events.event_name, users.username,
    winners.gold, winners.silver, winners.bronze
    FROM events
    JOIN organiser_event ON events.event_id =
    organiser_event.event_id
    JOIN registrations ON events.event_id =
    registrations.event_id
    JOIN users ON registrations.username = users.username
    JOIN winners ON events.event_id = winners.event_id
    WHERE organiser_event.username = %s
    ORDER BY events.event_id
''', [username])
cur.execute('''
    SELECT events.event_name, winners.gold, winners.silver,
    winners.bronze
    FROM events
    JOIN winners ON events.event_id = winners.event_id
    WHERE events.event_id = %s
''', [event_id])

```

## J. Deletes

```

-- Delete a user => student/externalparticipant/organiser
cur.execute('SELECT * FROM students WHERE username = %s', [deluser])
cur.execute('SELECT * FROM external WHERE username = %s', [deluser])
cur.execute('SELECT * FROM organiser WHERE username = %s', [deluser])
cur.execute('DELETE FROM users WHERE username = %s', [deluser])

```

## D. Forms

### A. Sign-In

```
# Request username and password
username = request.form['username']
password = request.form['password']
```

### B. Registration

```
# User information for registration
username = request.form['username']
fullname = request.form['fullname']
email = request.form['email']
phone = request.form['phone']
password = request.form['password']
confirm_password = request.form['confirm_password']
category = request.form['category']
```

### C. Add Event

```
# Event information
event_id = request.form['eventid']
event_name = request.form['eventname']
date = request.form['date']
description = request.form['description']
event_type = request.form['eventtype']
```

### D. Profile Change

```
# Information during profile change/update
emailid = request.form['emailid']
phone = request.form['phone']
newpass = request.form['newpass']
curpass = request.form['curpass']
```

### E. Winner

```
eventid = request.args.get('eventid')
goldw = request.args.get('goldw')
```



```
silverw = request.args.get('silverw')  
bronzew = request.args.get('bronzew')
```

## F. Change Logistics

```
hall = request.args.get('hall')  
roomnum = request.args.get('roomnum')  
foodtype = request.args.get('foodtype')
```