Univent Database Schema – 2NF Justification

SQL Code for Database and Tables

```
CREATE DATABASE Univent;
USE Univent;

-- USER Table
CREATE TABLE User (
    user_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    age INT,
    role VARCHAR(50),
    email VARCHAR(100) UNIQUE
);

-- COLLEGE Table
CREATE TABLE College (
    college_id INT PRIMARY KEY,
    name VARCHAR(100),
    location VARCHAR(100)
);

-- SUPER_ADMIN Table
CREATE TABLE Super_Admin (
    admin_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    password VARCHAR(100),
    designation VARCHAR(100),
    college_id INT,
    FOREIGN KEY (college_id) REFERENCES College(college_id)
);

-- CLUB_OR_SOCIETY Table
CREATE TABLE Club (
    club_id INT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    Category VARCHAR(100),
    secretary_name VARCHAR(100),
    secretary_id INT,
    college_id INT,
    FOREIGN KEY (college_id) REFERENCES College(college_id),
    FOREIGN KEY (secretary_id)REFERENCES USER(user_id)
);

-- EVENT Table
CREATE TABLE Event (
    event_id INT PRIMARY KEY,
    name VARCHAR(100),
    type_of_event VARCHAR(100),
    date DATE,
    location VARCHAR(100),
    status VARCHAR(50),
```

```
    organised_BY INT,
    max_num_of_participants INT,
    FOREIGN KEY (organised_BY) REFERENCES Club(club_id)
);

-- Event Scheduler for Past Events
SET GLOBAL event_scheduler = ON;
DELIMITER $$

CREATE EVENT move_old_events_to_past
ON SCHEDULE EVERY 1 DAY
DO
BEGIN
  INSERT INTO Past_Event (event_id, name, type_of_event, date, location,
status)
  SELECT event_id, name, type_of_event, date, location, status
  FROM Event
  WHERE date < CURDATE() - INTERVAL 2 DAY;

  DELETE FROM Event
  WHERE date < CURDATE() - INTERVAL 2 DAY;
END$$

DELIMITER ;

-- COMPETITION Table
CREATE TABLE Competition (
    comp_id INT auto_increment PRIMARY KEY,
    name VARCHAR(100),
    type_of_comp VARCHAR(100),
    date DATE,
    venue VARCHAR(100),
    event_id INT,
    FOREIGN KEY (event_id) REFERENCES Event(event_id)
);

-- TRANSACTION Table
CREATE TABLE Transaction (
    trans_id INT AUTO_INCREMENT PRIMARY KEY,
    amount DECIMAL(10, 2),
    description TEXT,
    trans_type VARCHAR(50),
    transferred_to INT,
    FOREIGN KEY (transferred_to) REFERENCES Club(club_id)
);

-- REGISTERS Table
CREATE TABLE Registers (
    reg_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    event_id INT,
    UNIQUE (user_id, event_id),
    FOREIGN KEY (user_id) REFERENCES User(user_id),
    FOREIGN KEY (event_id) REFERENCES Event(event_id)
);

-- REQUESTS_APPROVAL Table
CREATE TABLE Requests_Approval (
    request_id INT PRIMARY KEY,
    club_id INT,
    source VARCHAR(100),
```

```
    status VARCHAR(50),
    approved_by INT,
    rejected_by INT,
    FOREIGN KEY (club_id) REFERENCES Club(club_id),
    FOREIGN KEY (approved_by) REFERENCES Super_Admin(admin_id),
    FOREIGN KEY (rejected_by) REFERENCES Super_Admin(admin_id)
);

-- FEEDBACK Table
CREATE TABLE Feedback (
    feedback_id INT PRIMARY KEY,
    event_id INT,
    user_id INT,
    time TIMESTAMP,
    rating INT CHECK (rating BETWEEN 1 AND 5),
    comment TEXT,
    FOREIGN KEY (event_id) REFERENCES Event(event_id),
    FOREIGN KEY (user_id) REFERENCES User(user_id)
);
```

Why the Univent Schema is in 2NF

What is Second Normal Form (2NF)?

A table is in **2NF** if:

1. It is already in **First Normal Form (1NF)** (i.e., atomic values and unique rows).
2. It has **no partial dependency**—which means that all non-prime (non-key) attributes are fully functionally dependent on the **entire primary key**, not just part of it.

Univent Schema Analysis

Here's why each table is in **2NF**:

Table Name	Primary Key	2NF Justification
User	`user_id`	All columns like name, age, role, email depend fully on user_id.
College	`college_id`	`name` and `location` are atomic and depend on the full key.
Super_Admin	`admin_id`	Attributes depend on `admin_id`; `college_id` is a foreign key, not part of PK.
Club	`club_id`	Attributes like `name`, `email`, etc., depend entirely on `club_id`.
Event	`event_id`	All fields are fully dependent on `event_id`.

Competition	`comp_id`	All fields depend on `comp_id`; `event_id` is a FK.
Transaction	`trans_id`	Each attribute is fully dependent on `trans_id`.
Registers	`reg_id` (and unique `user_id, event_id`)	Fully functionally dependent on the primary key.
Requests_Approval	`request_id`	Each attribute depends fully on `request_id`.
Feedback	`feedback_id`	All attributes are fully functionally dependent.

Key Observations

- No table has **composite primary keys** where only part of the key is used to determine a non-key attribute.
- Each table contains **atomic, non-redundant data** with all fields depending **only on their respective primary keys**.
- The **foreign keys** (like `college_id`, `event_id`, etc.) are used only to **connect related tables**, and don't violate 2NF principles.

Conclusion

The **Univent Database Schema** is fully normalized to **2NF**:

- **No partial dependencies**
- **No data redundancy**
- **Improved data integrity**

This ensures better **efficiency**, **scalability**, and **clean relational structure** for managing event and college-related data.
