

# DBMS PROJECT

Vivlr: The Ultimate Campus Companion

Y.Snehith-22CSB0C29

E.Dhanush-22CSB0B18

## ***Problem Statement:***

NIT Warangal lacks a centralized platform for facilitating spontaneous social interactions among students and accessing information about club events.

Existing solutions are fragmented and do not cater to the specific needs of the campus community. To address this, the project aims to design and implement a database for "Vivlr," a social connection platform tailored for NIT Warangal.

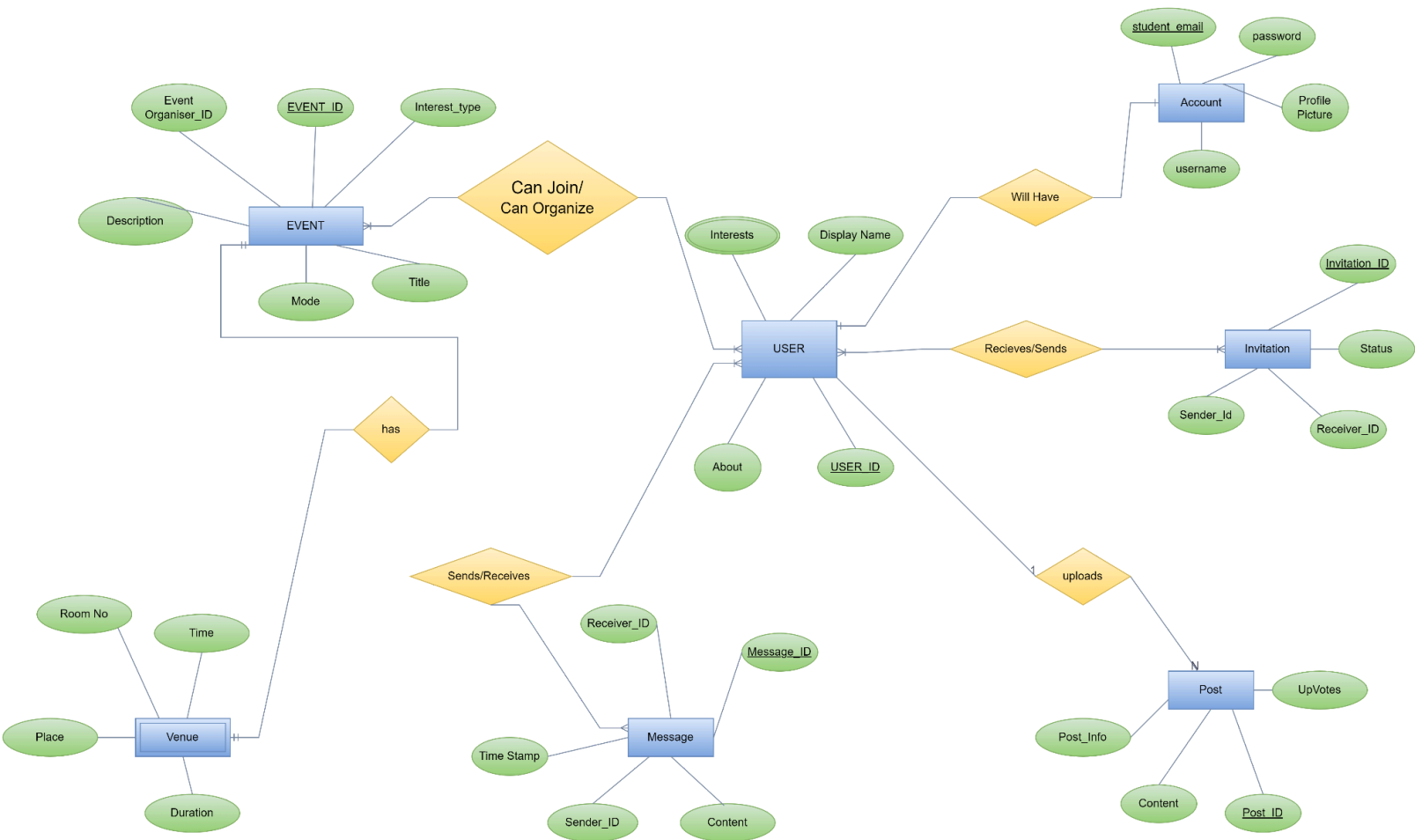
## ***Objective:***

The objective of the project is to create a robust database infrastructure for Vivlr that seamlessly integrates features for spontaneous social interactions, club event management, and campus-wide discussions. This database will serve as the backbone of Vivlr, providing students with a unified platform for staying connected, discovering events, and engaging in meaningful conversation

## Contents

- ER Diagram
- Working of the database (ER assumptions)
- Normalization of the tables
- Tables
- Relational Schema
- SQL Operations  
(creation of tables and insertion of values)

# Entity-Relationship Model



# WORKING OF THE DATABASE

- People log in to the system using their own username and password.
- Once logged in, they can see their own details like their username, email, full name, and profile picture.
- People can check out events happening around them and decide which ones they want to attend or organize.
- They can talk to each other by sending messages, either one-on-one or as part of an event.
- People can share their thoughts or things they find interesting by creating posts. Others can like and comment on these posts.
- They can see what others are saying about events they're interested in.
- People can invite others to join events they're planning.
- They can also see invitations from others and decide if they want to join.
- Details about events like the title, location, and time are available for people to see.
- People can interact with events by joining or leaving them, depending on if they want to go or not.
- The system keeps track of all these interactions to give people updates and notifications about events they're involved in.
- Some events might be private, meaning only certain people can join, while others are public and open to everyone.
- The system makes sure that everyone's information is safe and that only authorized actions can be taken.

# NORMALIZATION

## 1)USER

USER\_ID -> {DISPLAY NAME,ABOUT}

ON PRIMARY CLOSURE OF USER\_ID WE GET :- (USER\_ID)<sup>+</sup>->R

**PRIMARY KEY -> USER\_ID**

**NORMALISATION:**

PRIME ATTRIBUTES :- USER\_ID

NON PRIME ATTRIBUTES :- USERNAME,ABOUT

THERE IS **NO PARTIAL DEPENDENCY** THEREFORE THE TABLE IS IN 2NF.

THERE IS **NO TRANSITIVE DEPENDENCY** THEREFORE THE TABLE IS IN 3NF.

ALL DEPENDENCIES ARE FROM **CANDIDATE KEY** THEREFORE THE TABLE IS **BCNF**.

## 2)ACCOUNT

USERNAME -> {STUDENT\_EMAIL,PASSWORD,PROFILE PICTURE}

ON PRIMARY CLOSURE OF USERNAME WE GET :- (USERNAME)<sup>+</sup>->R

**PRIMARY KEY -> USERNAME**

**NORMALIZATION:**

PRIMARY ATTRIBUTES:- USERNAME

NON PRIMARY ATTRIBUTES:- PROFILE PICTURE,PASSWORD,

STUDENT\_EMAIL

THERE IS **NO PARTIAL DEPENDENCY** THEREFORE THE TABLE IS IN 2NF.

THERE IS **NO TRANSITIVE DEPENDENCY** THEREFORE THE TABLE IS IN 3NF.  
ALL DEPENDENCIES ARE FROM **CANDIDATE KEY** THEREFORE THE TABLE IS BCNF .

### **3)INTERESTS**

THIS IS A WEAK ENTITY SET, THEREFORE IT FORMS PRIMARY KEY WITH THE HELP OF A FOREIGN KEY.

USER\_ID -> {INTEREST\_TYPES}

**PRIMARY KEY & FOREIGN KEY -> USER\_ID**

### **4)INVITATION**

INVITATION\_ID -> {SENDER\_ID,RECEIVER\_ID,STATUS}  
ON PRIMARY CLOSURE OF INVITATION\_ID WE GET :-  
(INVITATION\_ID)+->R

**PRIMARY KEY -> INVITATION\_ID**

**NORMALIZATION:**

PRIMARY ATTRIBUTES:- INVITATION\_ID

NON PRIMARY ATTRIBUTES:- SENDER\_ID,RECEIVER\_ID,STATUS

THERE IS **NO PARTIAL DEPENDENCY** THEREFORE THE TABLE IS IN 2NF.  
THERE IS **NO TRANSITIVE DEPENDENCY** THEREFORE THE TABLE IS IN 3NF.  
ALL DEPENDENCIES ARE FROM **CANDIDATE KEY** THEREFORE THE TABLE IS BCNF .

## **5)POST**

THIS WILL BE IN FULL PARTICIPATION WITH THE USER ENTITY

POST\_ID -> {POST\_INFO,CONTENT,UPVOTES}

ON PRIMARY CLOSURE OF POST\_ID WE GET :- (POST\_ID) $\rightarrow$ R

**PRIMARY KEY -> POST\_ID**

**NORMALIZATION:**

PRIMARY ATTRIBUTES :- POST\_ID

NON PRIMARY ATTRIBUTES :-POST\_INFO , CONTENT,UPVOTES

THERE IS **NO PARTIAL DEPENDENCY** THEREFORE THE TABLE IS IN 2NF.

THERE IS **NO TRANSITIVE DEPENDENCY** THEREFORE THE TABLE IS IN 3NF.

ALL DEPENDENCIES ARE FROM **CANDIDATE KEY** THEREFORE THE TABLE IS BCNF .

## **6)MESSAGE**

MESSAGE\_ID -> {SENDER\_ID,RECEIVER\_ID,TIME\_STAMP,CONTENT}

ON PRIMARY CLOSURE OF MESSAGE\_ID WE GET :- (MESSAGE\_ID) $\rightarrow$ R

**PRIMARY KEY ->MESSAGE\_ID**

**NORMALIZATION:**

PRIMARY ATTRIBUTES :- MESSAGE\_ID

NON PRIMARY ATTRIBUTES :- SENDER\_ID,RECEIVER\_ID,TIME\_STAMP,  
CONTENT.



THERE IS NO PARTIAL DEPENDENCY THEREFORE THE TABLE IS IN 2NF.  
THERE IS NO TRANSITIVE DEPENDENCY THEREFORE THE TABLE IS IN 3NF.  
ALL DEPENDENCIES ARE FROM CANDIDATE KEY THEREFORE THE TABLE IS BCNF .

## 7) EVENT

```
EVENT_ID -> { EVENT_ORGANIZER , TITLE , DESCRIPTION ,
INTEREST_TYPE
                , MODE }
```

ON PRIMARY CLOSURE OF EVENT\_ID WE GET :- (EVENT\_ID)<sub>+</sub> → R

**PRIMARY KEY ->MESSAGE\_ID**

## NORMALIZATION:

PRIMARY ATTRIBUTES :-EVENT ID

NON PRIMARY ATTRIBUTES :- EVENT\_ORGANIZER , TITLE , DESCRIPTION ,  
INTEREST\_TYPE , MODE

## 8) VENUE

THIS IS A WEAK ENTITY SET, THEREFORE IT FORMS PRIMARY KEY WITH THE HELP OF A FOREIGN KEY.

# TABLES

## • USER

ATTRIBUTE	DATATYPE	CONSTRAINTS-
USER_ID	INTEGER	PRIMARY_KEY NOT NULL
Display Name	VARCHAR(20)	NOT NULL
About	VARCHAR(50)	NOT NULL

## •ACCOUNT

ATTRIBUTE	DATATYPE	CONSTRAINTS
Student_email	VARCHAR(20)	PRIMARY KEY NOT_NULL
Password	VARCHAR(20)	NOT_NULL
Profile_Picture	VARCHAR(20)	NULL
UserName	VARCHAR(20)	NOT_NULL
USER_ID	INTEGER	FOREIGN KEY NOT_NULL

## •INVITATION

ATTRIBUTE	DATATYPE	CONSTRAINTS
Invitation_ID	INTEGER	PRIMARY_KEY NOT_NULL
Sender_ID	INTEGER	NOT_NULL
Receiver_ID	INTEGER	NOT_NULL
Status	INTEGER	NOT_NULL

## •POST

ATTRIBUTE	DATATYPE	CONSTRAINTS
User_ID	INTEGER	FOREIGN KEY NOT_NULL
Post_Info	VARCHAR(20)	NOT_NULL
Post_ID	INTEGER	PRIMARY KEY NOT_NULL
Content	VARCHAR(50)	NOT_NULL
Upvotes	INTEGER	NULL

## •MESSAGE

ATTRIBUTE	DATATYPE	CONSTRAINTS
Message_ID	VARCHAR(20)	PRIMARY_KEY NOT_NULL
Sender_ID	VARCHAR(20)	NOT_NULL
Receiver_ID	VARCHAR(20)	NOT_NULL
TimeStamp	TIME	NOT_NULL
Content	VARCHAR(20)	NOT_NULL

## • EVENT

ATTRIBUTE	DATATYPE	CONSTRAINTS
Event_ID	VARCHAR(20)	PRIMARY_KEY NOT_NULL
TITLE	VARCHAR(20)	NOT_NULL
EVENT ORG_ID	INTEGER	NOT_NULL
DESCRIPTION	CHAR(50)	NOT_NULL
MODE	CHAR(20)	NOT_NULL
INTEREST_TYPE	VARCHAR(20)	NOT_NULL

### • VENUE

ATTRIBUTE	DATATYPE	CONSTRAINTS
Event_ID	INTEGER	PRIMARY_KEY FOREIGN_KEY NOT_NULL
Duration	TIME	NOT_NULL
Place	VARCHAR(20)	NOT_NULL
Time	TIME	NOT_NULL
Room No	INTEGER	NULL

### • Sends\_Receives\_Message

ATTRIBUTE	DATATYPE	CONSTRAINTS
Message_ID	INTEGER	PRIMARY_KEY FOREIGN_KEY NOT_NULL
USER_ID	INTEGER	PRIMARY_KEY FOREIGN_KEY NOT_NULL

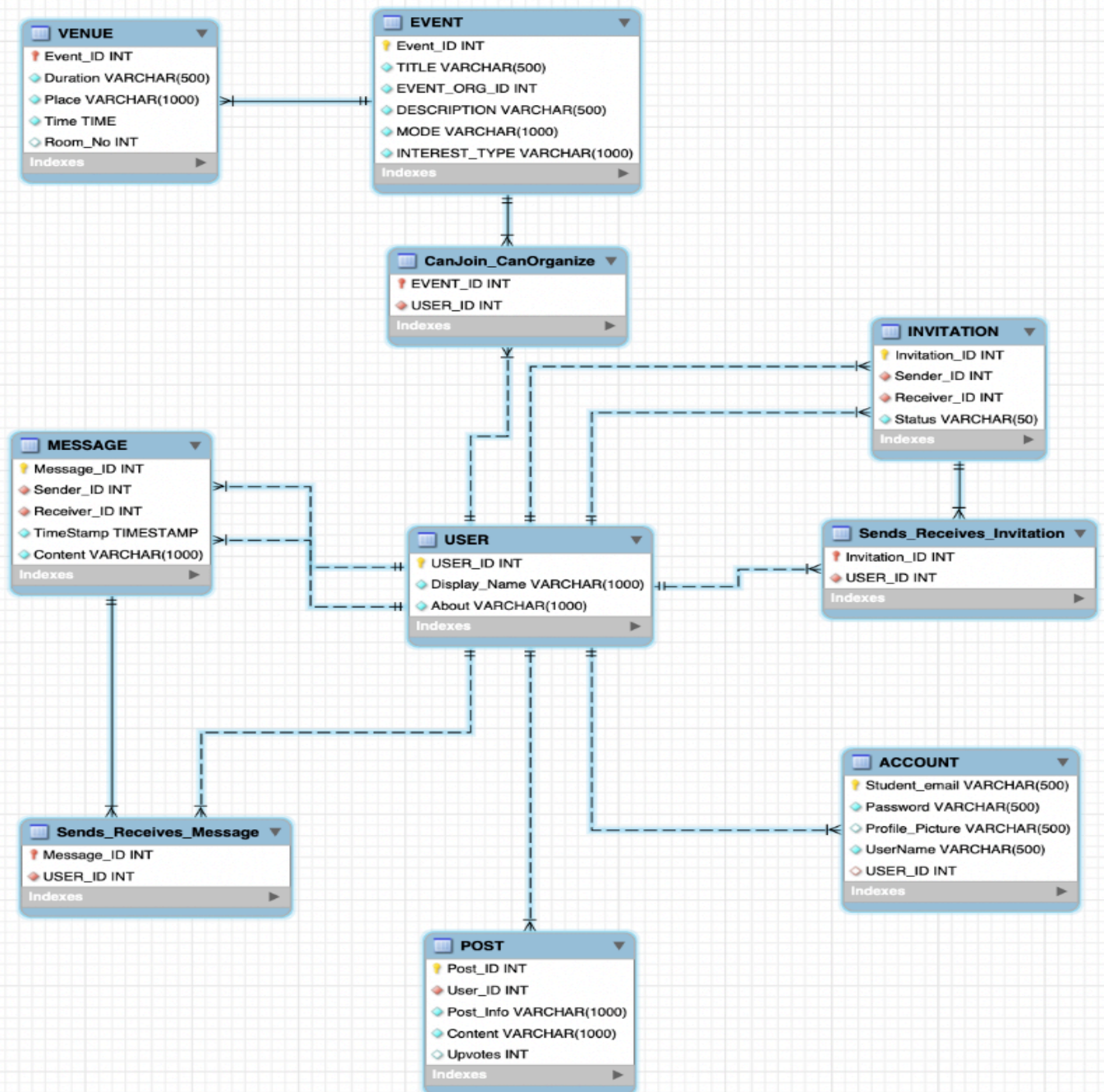
### •CanJoin\_CanOrganize

ATTRIBUTE	DATATYPE	CONSTRAINTS
EVENT_ID	INTEGER	PRIMARY_KEY FOREIGN_KEY NOT_NULL
USER_ID	INTEGER	PRIMARY_KEY FOREIGN_KEY NOT_NULL

- **Sends\_Receives\_Invitation**

ATTRIBUTE	DATATYPE	CONSTRAINTS
Invitation_ID	INTEGER	PRIMARY_KEY FOREIGN_KEY NOT_NULL
USER_ID	INTEGER	PRIMARY_KEY FOREIGN_KEY NOT_NULL

## RELATIONSHIP SCHEMA



# **Table Creation**

```
CREATE DATABASE VIVLR;  
USE VIVLR;
```

## **USER**

```
CREATE TABLE USER  
( USER_ID INTEGER PRIMARY KEY NOT NULL,  
  Display_Name VARCHAR(1000) NOT NULL,  
  About VARCHAR(1000) NOT NULL);
```

## **ACCOUNT**

```
CREATE TABLE ACCOUNT (  
  Student_email VARCHAR(500) PRIMARY KEY NOT NULL,  
  Password VARCHAR(500) NOT NULL,  
  Profile_Picture VARCHAR(500),  
  UserName VARCHAR(500) NOT NULL,  
  USER_ID INTEGER,  
  FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID));
```

## **INVITATION**

```
CREATE TABLE INVITATION (  
  Invitation_ID INTEGER PRIMARY KEY NOT NULL,  
  Sender_ID INTEGER NOT NULL,  
  Receiver_ID INTEGER NOT NULL,  
  Status VARCHAR(50) NOT NULL,  
  FOREIGN KEY (Sender_ID) REFERENCES USER(USER_ID),  
  FOREIGN KEY (Receiver_ID) REFERENCES USER(USER_ID));
```



## **POST**

```
CREATE TABLE POST (  
    Post_ID INTEGER PRIMARY KEY NOT NULL,  
    User_ID INTEGER NOT NULL,  
    Post_Info VARCHAR(1000) NOT NULL,  
    Content VARCHAR(1000) NOT NULL,  
    Upvotes INTEGER,  
    FOREIGN KEY (User_ID) REFERENCES USER(USER_ID));
```

## **MESSAGE**

```
CREATE TABLE MESSAGE (  
    Message_ID INTEGER PRIMARY KEY NOT NULL,  
    Sender_ID INTEGER NOT NULL,  
    Receiver_ID INTEGER NOT NULL,  
    TimeStamp TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    Content VARCHAR(1000) NOT NULL,  
    FOREIGN KEY (Sender_ID) REFERENCES USER(USER_ID),  
    FOREIGN KEY (Receiver_ID) REFERENCES USER(USER_ID));
```

## **EVENT**

```
CREATE TABLE EVENT (  
    Event_ID INTEGER PRIMARY KEY NOT NULL,  
    TITLE VARCHAR(500) NOT NULL,  
    EVENT_ORG_ID INTEGER NOT NULL,  
    DESCRIPTION VARCHAR(500) NOT NULL,  
    MODE VARCHAR(1000) NOT NULL,  
    INTEREST_TYPE VARCHAR(1000) NOT NULL);
```

## **VENUE**

```
CREATE TABLE VENUE (  
    Event_ID INTEGER PRIMARY KEY NOT NULL,  
    Duration VARCHAR(50) NOT NULL,  
    Place VARCHAR(1000) NOT NULL,  
    Time TIME NOT NULL,  
    Room_No INTEGER,  
    FOREIGN KEY (Event_ID) REFERENCES EVENT(Event_ID));
```

## **Sends Receives Message**

```
CREATE TABLE Sends_Receives_Message (  
    Message_ID INTEGER PRIMARY KEY NOT NULL,  
    USER_ID INTEGER NOT NULL,  
    FOREIGN KEY (Message_ID) REFERENCES MESSAGE(Message_ID),  
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID));
```

## **CanJoin\_CanOrganize**

```
CREATE TABLE CanJoin_CanOrganize (  
    EVENT_ID INTEGER PRIMARY KEY NOT NULL,  
    USER_ID INTEGER NOT NULL,  
    FOREIGN KEY (EVENT_ID) REFERENCES EVENT(Event_ID),  
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID));
```

## **Sends Receives Invitation**

```
CREATE TABLE Sends_Receives_Invitation (  
    Invitation_ID INTEGER PRIMARY KEY NOT NULL,  
    USER_ID INTEGER NOT NULL,  
    FOREIGN KEY (Invitation_ID) REFERENCES INVITATION(Invitation_ID),  
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID));
```

# **INSERTING THE DATA**

## **USER**

```
INSERT INTO USER (USER_ID, Display_Name, About) VALUES
(1, 'Ananya Sharma', 'Studying Computer Science at NIT Warangal'),
(2, 'Rahul Patel', 'Aspiring Electrical Engineer from NIT Warangal'),
(3, 'Priya Singh', 'Final Year Civil Engineering Student at NIT Warangal'),
(4, 'Ravi Kumar', 'Enthusiastic Mechanical Engineering Student at NIT Warangal'),
(5, 'Sneha Verma', 'Passionate about Chemical Engineering at NIT Warangal'),
(6, 'Amit Gupta', 'Mechanical Engineering Graduate from NIT Warangal'),
(7, 'Neha Sharma', 'Computer Science Junior at NIT Warangal');
```

## **ACCOUNT**

```
INSERT INTO ACCOUNT (Student_email, Password, Profile_Picture, UserName,
USER_ID) VALUES
('ananya.sharma@nitw.ac.in', 'password123', NULL, 'ananya123', 1),
('rahul.patel@nitw.ac.in', 'patel456', NULL, 'rahul456', 2),
('priya.singh@nitw.ac.in', 'singh789', NULL, 'priya789', 3),
('ravi.kumar@nitw.ac.in', 'kumar123', NULL, 'ravi123', 4),
('sneha.verma@nitw.ac.in', 'verma456', NULL, 'sneha456', 5),
('amit.gupta@nitw.ac.in', 'gupta789', NULL, 'amit789', 6),
('neha.sharma@nitw.ac.in', 'neha123', NULL, 'neha123', 7);
```

## **INVITATION**

```
INSERT INTO INVITATION (Invitation_ID, Sender_ID, Receiver_ID, Status) VALUES
(101, 1, 2, 'Sent'),
(102, 2, 3, 'Accepted'),
(103, 3, 4, 'Pending'),
(104, 4, 5, 'Sent'),
(105, 5, 6, 'Accepted'),
(106, 6, 7, 'Pending'),
(107, 7, 1, 'Sent');
```

## **POST**

```
INSERT INTO POST (Post_ID, User_ID, Post_Info, Content, Upvotes) VALUES
(201, 1, 'Tech Club Meeting', 'Discussing upcoming projects', 15),
(202, 2, 'Sports Club Event', 'Football tournament finals', 20),
(203, 3, 'Cultural Fest Announcement', 'Seeking volunteers', 10),
(204, 4, 'Robotics Workshop', 'Hands-on experience with robots', 8),
(205, 5, 'Chemical Engineering Seminar', 'Guest lecture on new advancements', 12),
(206, 6, 'Alumni Meetup', 'Networking and career guidance', 25),
(207, 7, 'Programming Contest', 'Registration now open', 18);
```

## **MESSAGE**

```
INSERT INTO MESSAGE (Message_ID, Sender_ID, Receiver_ID, TimeStamp, Content)
VALUES
(301, 1, 2, '2024-04-01 10:30:00', 'Hi Rahul, How are you?'),
(302, 2, 3, '2024-04-01 11:45:00', 'Hey Priya, Would you like to join our club?'),
(303, 3, 4, '2024-04-01 12:15:00', 'Hi Ravi, Do you have the notes for today?'),
(304, 4, 5, '2024-04-01 13:00:00', 'Hello Sneha, Let"s discuss the project tomorrow.'),
(305, 5, 6, '2024-04-01 14:20:00', 'Hey Amit, Are you attending the alumni meetup?'),
(306, 6, 7, '2024-04-01 15:30:00', 'Hi Neha, How"s your project coming along?'),
(307, 7, 1, '2024-04-01 16:45:00', 'Hello Ananya, Let"s catch up soon.');
```

## **EVENT**

```
INSERT INTO EVENT (Event_ID, TITLE, EVENT_ORG_ID, DESCRIPTION, MODE,
INTEREST_TYPE) VALUES
(401, 'Tech Fest', 1, 'Annual technical festival', 'Online', 'Technology'),
(402, 'Cultural Night', 2, 'Celebrating diversity through performances', 'Offline', 'Cultural'),
(403, 'Sports Meet', 3, 'Inter-college sports competition', 'Offline', 'Sports'),
(404, 'Robotics Expo', 4, 'Showcasing innovative robotics projects', 'Offline', 'Technology'),
(405, 'Alumni Talk', 5, 'Interactive session with NIT Warangal alumni', 'Online', 'Career'),
(406, 'Chemical Symposium', 6, 'Exploring recent trends in chemical engineering', 'Online',
'Education'),
(407, 'Debate Competition', 7, 'Engaging discussions on contemporary issues', 'Offline',
'Education');
```

## **VENUE**

```
INSERT INTO VENUE (Event_ID, Duration, Place, Time, Room_No) VALUES
(401, '3 hours', 'NIT Warangal', '2024-05-15 10:00:00', 102),
(402, '4 hours', 'NIT Warangal Auditorium', '2024-06-20 18:00:00', 201),
(403, 'Full day', 'NIT Warangal Sports Complex', '2024-07-10 08:00:00', 301),
(404, '2 days', 'NIT Warangal Robotics Lab', '2024-08-05 09:00:00', 401),
(405, '2 hours', 'Online', '2024-09-15 14:00:00', NULL),
(406, '3 hours', 'Online', '2024-10-20 16:00:00', NULL),
(407, '5 hours', 'NIT Warangal Seminar Hall', '2024-11-10 11:00:00', 502);
```

## **Sends\_ Receives\_Message**

```
INSERT INTO Sends_Receives_Message (Message_ID, USER_ID) VALUES
(301, 1),
(302, 2),
(303, 3),
(304, 4),
(305, 5),
(306, 6),
(307, 7);
```

### **CanJoin\_CanOrganize**

```
INSERT INTO CanJoin_CanOrganize (EVENT_ID, USER_ID) VALUES  
(401, 1),  
(402, 2),  
(403, 3),  
(404, 4),  
(405, 5),  
(406, 6),  
(407, 7);
```

### **Sends\_Receive Invitation**

```
INSERT INTO Sends_Receive Invitation (Invitation_ID, USER_ID) VALUES  
(101, 1),  
(102, 2),  
(103, 3),  
(104, 4),  
(105, 5),  
(106, 6),  
(107, 7);
```