Week 2: Logical Schema & SQL Implementation

1. Relational Schema

TABLE: Faculty

Column	**Type**	**Key**
Faculty_ID	INTEGER	PK
Faculty_Name	VARCHAR(100)	
Email	VARCHAR(100)	UNIQUE
Phone_Number	VARCHAR(20)	

TABLE: Students

Column	**Type**	**Key**
NIM	INTEGER	PK
Full_Name	VARCHAR(100)	
Email	VARCHAR(100)	UNIQUE
Major	VARCHAR(50)	
DPA_Faculty_ID	INTEGER	FK

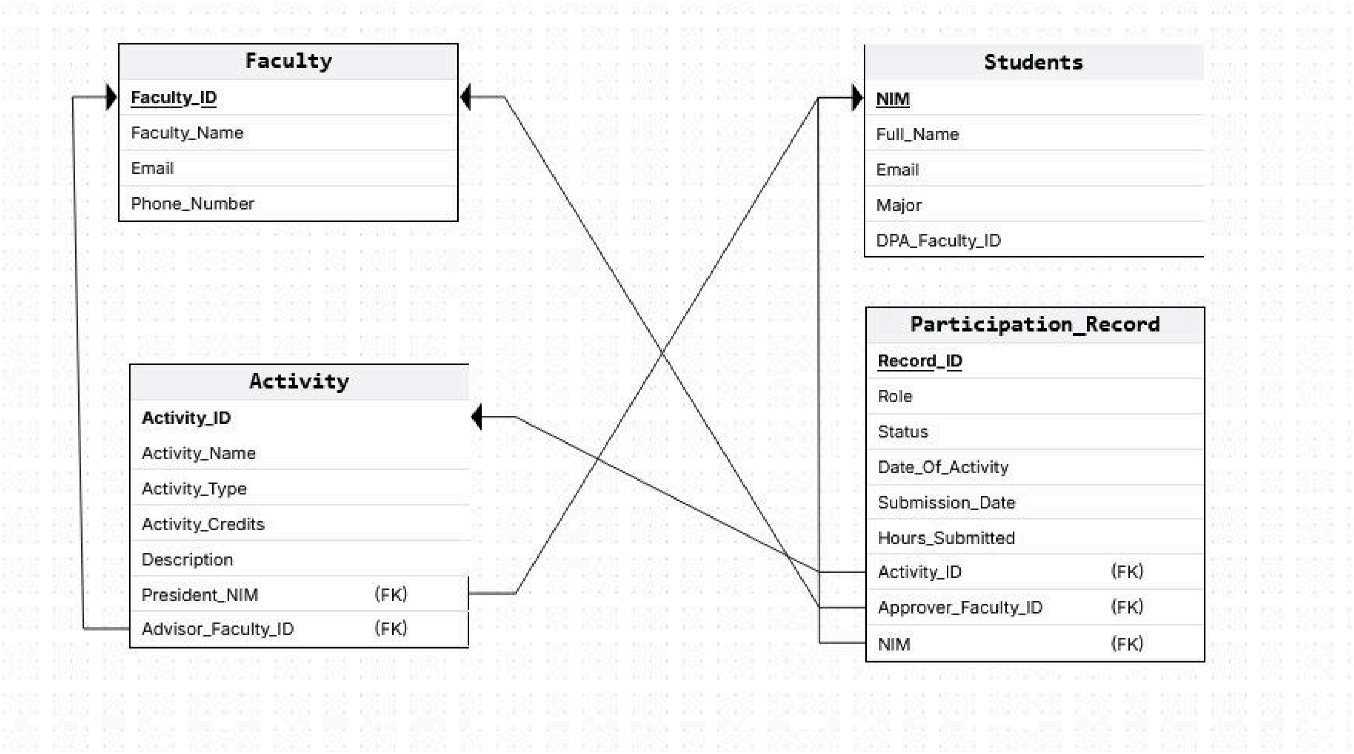
TABLE: Activity

Column	**Type**	**Key**
Activity_ID	INTEGER	PK
Activity_Name	VARCHAR(100)	
Activity_Type	VARCHAR(50)	
Activity_Credits	INTEGER	
Description	TEXT	
Advisor_Faculty_ID	INTEGER	FK
President_NIM	INTEGER	FK

TABLE: Participation_Record

Column	Type	Key
Record_ID	INTEGER	PK
NIM	INTEGER	FK
Activity_ID	INTEGER	FK
Role	VARCHAR(100)	
Status	VARCHAR(20)	
Date_Of_Activity	DATE	
Submission_Date	DATE	
Hours_Submitted	INTEGER	
Approver_Faculty_ID	INTEGER	FK

Relational Schema:



2. Third Normal Form (3NF) Justification

Our database design meets the 3NF requirement by eliminating transitive dependencies and calculated fields.

1NF (First Normal Form):

All tables are in 1NF as every column contains atomic (indivisible) values. There are no repeating groups or arrays. For example, Student.Full_Name is a single value, and we do not store multiple activities in a single row.

2NF (Second Normal Form):

All tables are in 2NF as all non-key attributes (like Student.Full_Name) are fully dependent on the entire primary key (Student.NIM). There are no partial dependencies.

3NF (Third Normal Form):

All tables are in 3NF because we have eliminated transitive dependencies. A transitive dependency is when a non-key attribute depends on another non-key attribute. We fixed several 3NF violations from our initial ERD:

Violation 1: DPA (Student's Advisor)

- Problem: Storing the advisor's name in the Students table. The DPA's details (like email) would depend on the DPA name, not on the Student.NIM.

- 3NF fix: We removed the DPA text field and replaced it with DPA_Faculty_ID. This foreign key references the Faculty table. All of the advisor's details (Faculty_Name, Email) are now correctly stored in one place (Faculty table) and do not depen on a non-key attribute in the Students table.

Violation 2: President and Staff in Activity

- Problem: Storing the President's name or Staff name directly in the Activity table

- 3NF fix: We replaced these text fields with foreign keys: President_NIM (which points to Faculty.Faculty_ID) and Advisor_Faculty_ID (which points to Faculty.Faculty_ID). This ensures that the president's details are stored in the Students table and the advisor's details are stored in the Faculty table, eliminating the transitive dependency.

Violation 3: Calculated fields

- Problem: Our ERD had fields like SKS, GPA, No. of Courses, and Amount of Members. These are all calculated fields, which violate 3NF principles as they depend on other data that may change.

- 3NF fix: We removed all calculated fields from the schema. The application will be responsible for calculating these values when needed. This ensures data integrity and avoids update anomalies.

3. SQL Implementation

```
CREATE TABLE Faculty (
    Faculty_ID INTEGER PRIMARY KEY AUTOINCREMENT,
    Faculty_Name VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    Phone_Number VARCHAR(20)
);

CREATE TABLE Students (
    NIM INTEGER PRIMARY KEY,
    Full_Name VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    Major VARCHAR(50),
    DPA_Faculty_ID INTEGER,
    FOREIGN KEY (DPA_Faculty_ID) REFERENCES Faculty(Faculty_ID)
);


CREATE TABLE Activity (
    Activity_ID INTEGER PRIMARY KEY AUTOINCREMENT,
    Activity_Name VARCHAR(100) NOT NULL,
    Activity_Type VARCHAR(50),
    Activity_Credits INTEGER DEFAULT 0,
    Description TEXT,
    Advisor_Faculty_ID INTEGER,
    President_NIM INTEGER,
    FOREIGN KEY (Advisor_Faculty_ID) REFERENCES Faculty(Faculty_ID),
    FOREIGN KEY (President_NIM) REFERENCES Students(NIM)
);
```

```
CREATE TABLE Participation_Record (
   Record_ID INTEGER PRIMARY KEY AUTOINCREMENT,
   NIM INTEGER NOT NULL,
   Activity_ID INTEGER NOT NULL,

   Role VARCHAR(100),
   Status VARCHAR(20) NOT NULL DEFAULT 'Pending',
   Date_Of_Activity DATE,
   Submission_Date DATE NOT NULL,
   Hours_Submitted INTEGER,

   Approver_Faculty_ID INTEGER,

   FOREIGN KEY (NIM) REFERENCES Students(NIM) ON DELETE CASCADE,
   FOREIGN KEY (Activity_ID) REFERENCES Activity(Activity_ID) ON DELETE
CASCADE,
   FOREIGN KEY (Approver_Faculty_ID) REFERENCES Faculty(Faculty_ID)
);
```

Explanation:
This is the main SQL script. This SQL schema is in the 3NF and fixes issues from the initial ERD.

Fixes:

1. REMOVED: All calculated fields ('SKS', 'GPA', 'No. Of_Courses', 'Amount Of Members').
2. REMOVED: All transitive dependencies ('DPA', 'President', 'Staff') and replaced them with Foreign Keys.
3. REMOVED: The redundant 'Participation' M:N relationship, as this is handled by 'Participation_Record'.

4. REMOVED: The ambiguous 'Part_Of' table. The DPA relationship is now correctly handled by a Foreign Key in the Students table.

Table 1: Faculty

- Stores information about faculty members who can be advisors or approvers

Table 2: Students

- Stores information about students
- ‘DPA_Faculty_ID’ is the foreign key to the student’s advisor (DPA), fixing the 3NF violation

Table 3: Activity

- Stores information about the extracurricular activities
- ‘Advisor_Faculty_ID’ and ‘President_NIM’ are foreign keys, fixing the 3NF violations from the initial ERD

Table 4: Participation_Record

- This table links Students and Activities (M:N) and stores the details of the participation for approval
- This is the only table that links Students and Activities

Data Integrity Constraints

We use constraints to enforce data rules at the database level.

- NOT NULL: Used for critical fields like Full_Name and Email to ensure they are never left blank
- DEFAULT: Used on Participation_Record.Status to set its value to ‘Pending’ automatically. This simplifies the application logic for the “approval” workflow.
- UNIQUE: Used for Email in both Students and Faculty tables to prevent duplicate accounts.