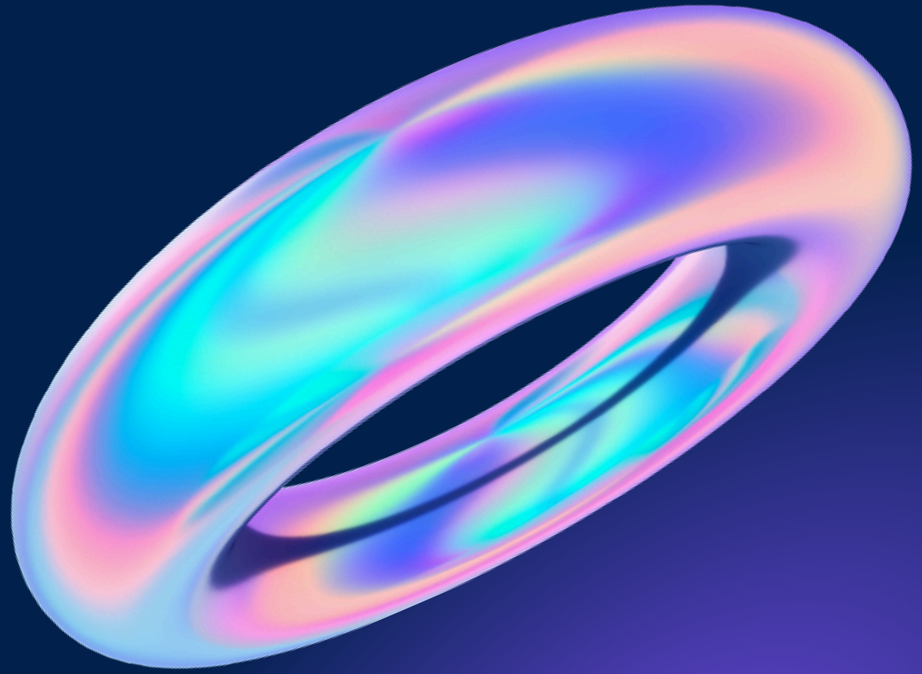
Note Sharing App

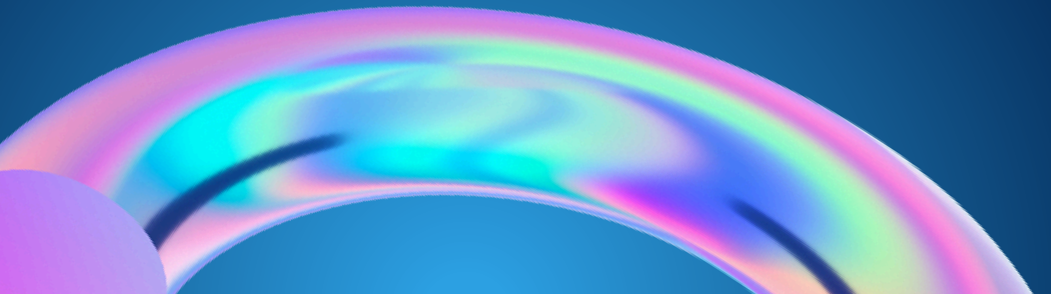
DBMS



Introduction

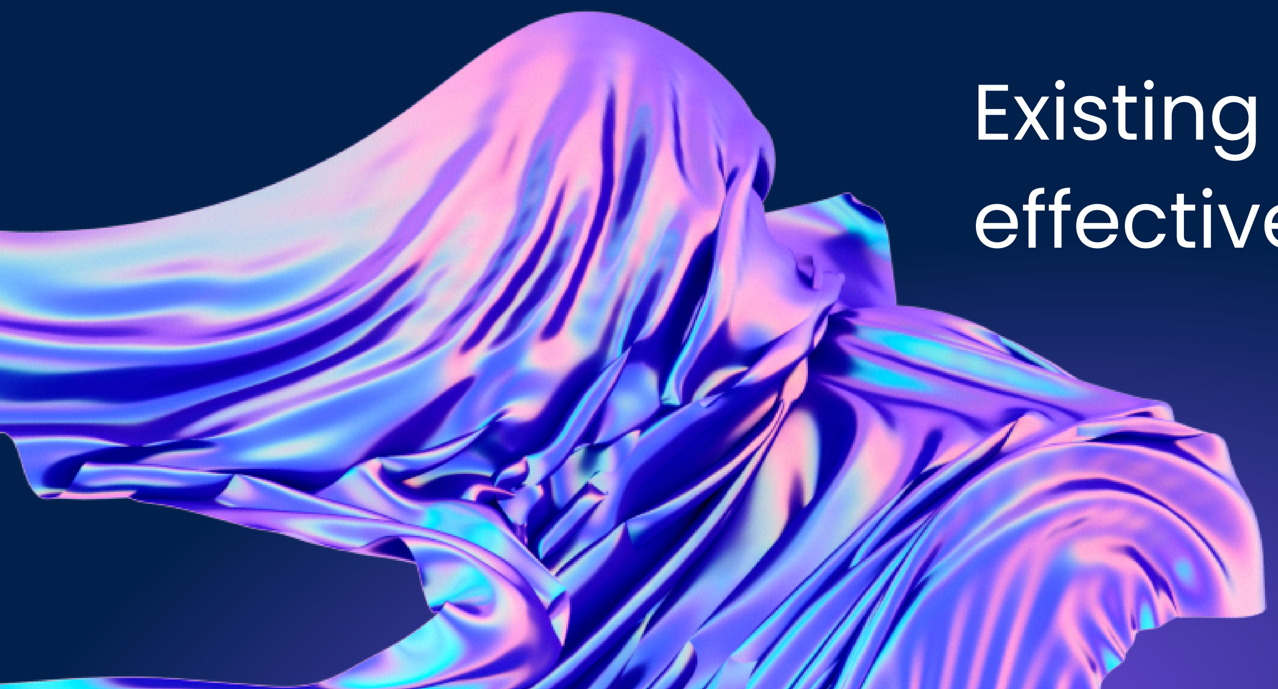
The Notes Sharing System facilitates the students to access the Notes available on the basis of Subject and Topic, the teachers share notes on the basis of the students they are teaching. The aim of this project is to design and develop a database maintaining the notes of different subjects, topics, and faculty.It is the computerized system of sharing subject notes on a single app. It can be mainly used by universities.

The option of online notes have made the process of sharing notes very much easier than ever before. This project contains an entity relationship model diagram based on Notes sharing App and introduction to relational model . There is also design of the database of the note sharing app based on relation model. Example of some SQL queries to retrieve data from the database.



Background

In a digital world, effective note-taking and collaboration are key. Traditional methods lack organization and collaboration features. Our project addresses this with a tailored Notes Sharing App.



Problem:

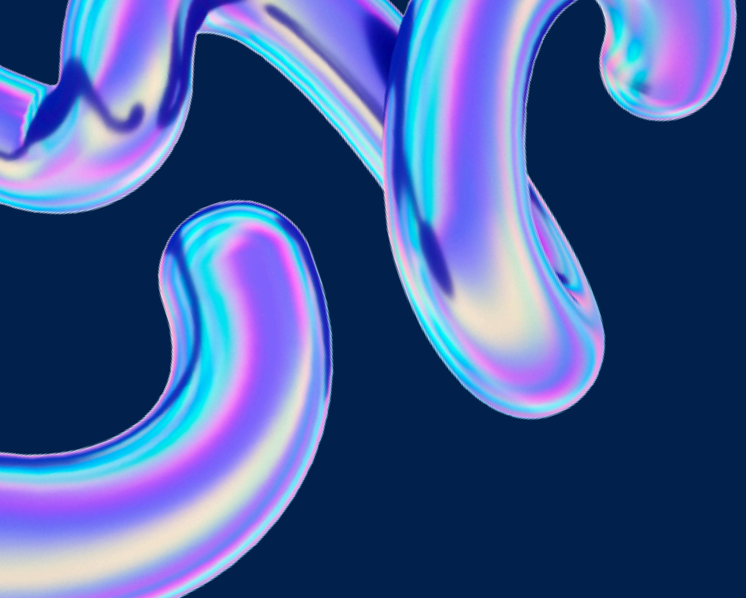
Existing note-taking methods lack efficiency and flexibility, hindering effective organization, sharing, and collaboration among users.

Objectives

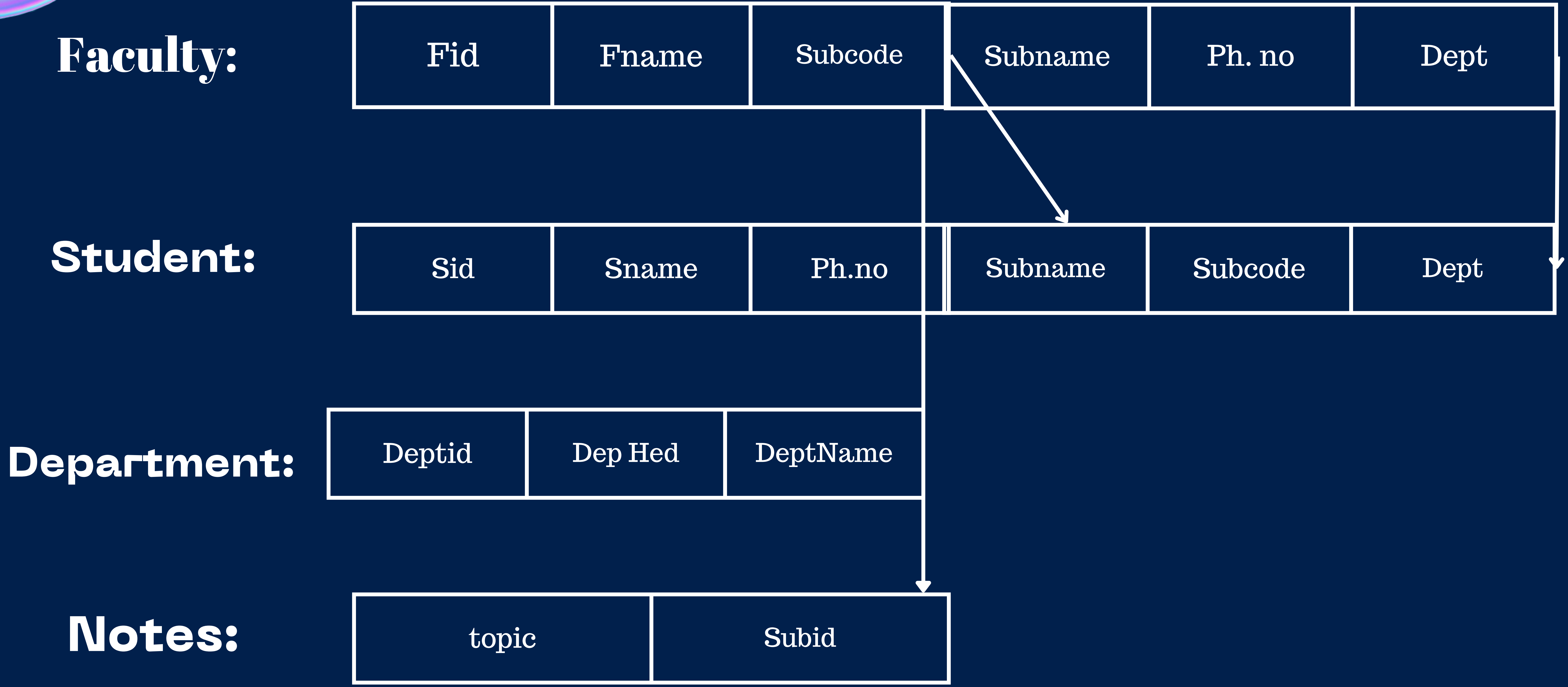
- Streamline note creation and management.
- Facilitate real-time collaboration among users.
- Ensure cross-platform accessibility and synchronization.
- Prioritize security and privacy.
- Enhance user engagement and experience.

Entities and Attributes:

Faculty	Fid (Faculty id) Fname (Faculty Name) subcode (subject code) Ph.no (Phone number) subname (subject name) Department
Student	sid (student id) sname (student name) Ph.no(Phone number) subcode (subject code) subname (subject name) Department Yr (Year of study)
Department	depid (department id) depname (department name) dephead (department head)
Subject	subid (subject id) subname (subject name)
Notes	subid (subject id) Topic



Schema :



Faculty:

Fid	Fname	Subcode	Subname	Ph. no	Dept

Student:

Sid	Sname	Ph.no	Subname	Subcode	Dept

Department:

Deptid	Dep Hed	DeptName

Notes:

topic	Subid

Normalization and Final List of Relations:

1NF: removing multivalued attributes
Phone numbers can be multivalued as people can have more than 1 phone number.
In this case we can use multiple columns to represent more than 1 phone number.

2NF: Removal of partial dependencies

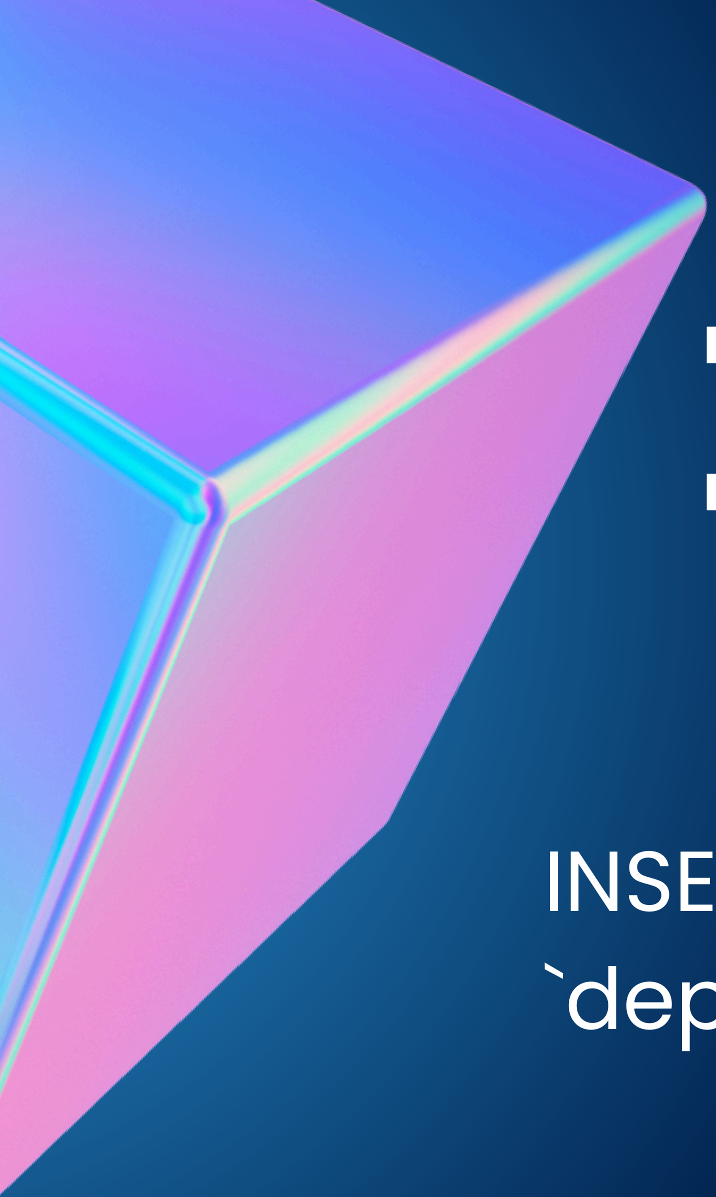
Non-prime attribute Sub Name is dependent on subcode a proper subset of the candidate key, which is a partial dependency. So, we remove this by splitting them into multiple tables..

3NF: Removal of Transitive dependencies

Fid-> Fname and Fname -> Ph.no1 and Fname -> Ph.no2 are true. So Ph.no1 and Ph.no2 are transitively dependent on Sid. It violates the third normal form. Similar in Student table So, we put them in a different table of phone numbers.

Create queries:

```
use DBMS;
Create table faculty(fid INT,fname varchar(100),subcode
varchar(100),subname varchar(100),department varchar(100));
Create table student(sid int,sname varchar(100),subcode
varchar(100),subname varchar(100),department varchar(100),yr INT);
Create table dep(depid varchar(100),depname varchar(100),dephead
varchar(100));
```



Insert queries:

```
INSERT INTO `dbms`.`faculty` (`fid`, `fname`, `subcode`, `subname`,
`department`) VALUES ('2022', 'Swati', 'MAN143', 'economics', 'SEAMS');
```

```
INSERT INTO `dbms`.`faculty` (`fid`, `fname`, `subcode`, `subname`,
`department`) VALUES ('2033', 'Kriti', 'ART123', 'History', 'SLASS');
```

Thank You.

Team:
Nivedha Sriram
Krish Nariya
Piyusha Mukherjee