**Functions and Features of database approach**

**Controlling Redundancy**

Traditional software development utilizing file processing, every user maintains their own files for handling data-processing applications. For example, consider the UNIVERSITY database. Two groups of users might be the course registration office and the accounting office. In the traditional approach, each office independently keeps files on students. The accounting office keeps data on registration and related billing information, whereas the registration office keeps track of student courses and grades.

Other groups may further duplicate some or all of the same data in their own files. This redundancy in storing the same data multiple times leads to several problems.

In the database approach, we should have a database design that stores each logical data item—such as a student’s name or birth date—in only one place in the database.

**Restricting Unauthorized Access**

When multiple users share a large database, it is likely that most users will not be authorized to access all information in the database. For example, financial data such as salaries and bonuses are often considered confidential, and only authorized persons are allowed to access such data.

A DBMS should provide a security and authorization subsystem, which the Database Administrator uses to create accounts and to specify account restrictions.

**Providing Storage Structures and Search Techniques for Efficient Query Processing**

Database systems must provide capabilities for efficiently executing queries and updates. The query processing and optimization module of the DBMS is responsible for choosing an efficient query execution plan for each query based on the existing storage structures. Example IRCTC Website.

**Providing Backup and Recovery**

A DBMS must provide facilities for recovering from hardware or software failures. The backup and recovery subsystem of the DBMS is responsible for recovery.

For example, if the computer system fails in the middle of a complex update transaction, the recovery subsystem is responsible for making sure that the database is restored to the state it was in before the transaction started executing. Example Bank Transactions at ATM.

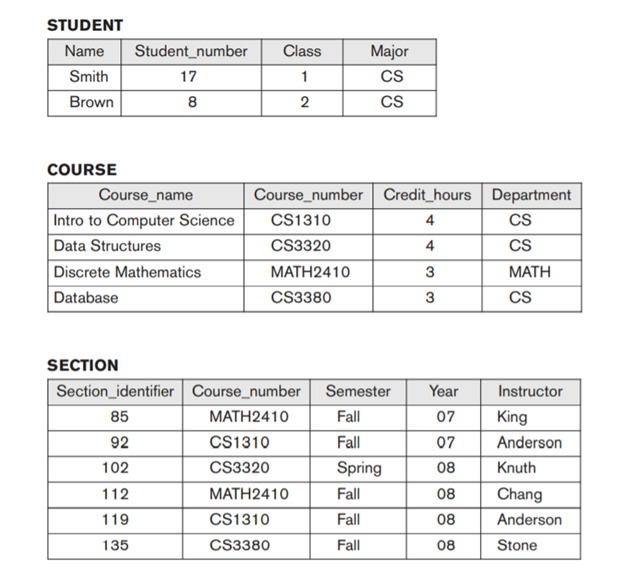
**Providing Multiple User Interfaces**

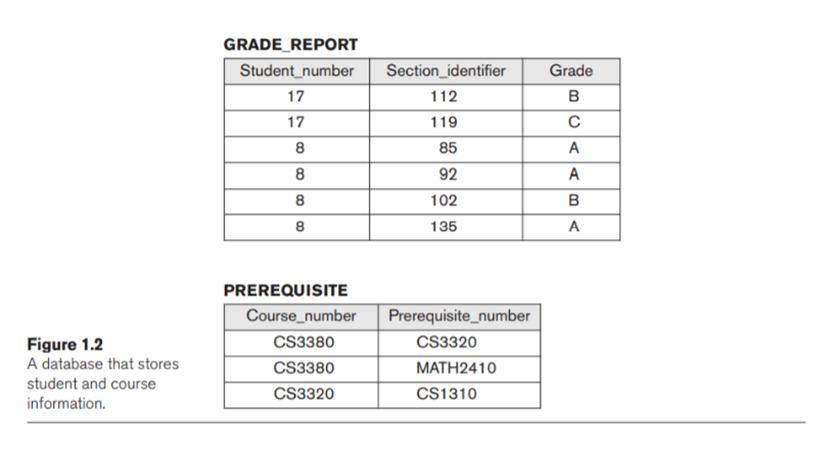
Because many types of users with varying levels of technical knowledge use a database, a DBMS should provide a variety of user interfaces. These include apps for mobile users, query languages for casual users, programming language interfaces for application programmers. Example Yono SBI, SBI card, Eazypay is mobile banking app for ICICI

**Representing Complex Relationships among Data**

A DBMS must have the capability to represent a variety of complex relationships among the data, to define new relationships as they arise, and to retrieve and update related data easily and efficiently.

Consider the example shown in Figure 1.2. The record for ‘Brown’ in the STUDENT file is related to four records in the GRADE\_REPORT file. Similarly, each section record is related to one course record and to a number of GRADE\_REPORT records—one for each student who completed that section.





**Enforcing Integrity Constraints**

Constraint specifies uniqueness on data item values, such as every course record must have a unique value for Course number in the table Course. This is known as a key or uniqueness constraint.