Reduced Data redundancy and inconsistency.

Redundancy is duplication/repetition of same data multiple times unnecessarily. In traditional file system, same data may be repeated multiple times in different files. For Example: let us consider student information system maintained using excel file system in a college. One excel file in administration section of the college may contain personnel information of student (Rollno, name, class, address etc), other may contain library systems (Rollno, name, class, address, book borrowed, borrow date etc) in library section and account section may have separate file that contains information for accounting/fee (Rollno, name, class, address, tuition fee, total fee to be paid, paid fee etc). Here, some information (like Rollno, name, address) are repeated in each excel file. They are repeated as each section of the college (administration, exam, and account) needs such information. There are mainly two disadvantages due to data redundancy in file system. Firstly, it consumes more disk space. Secondly, the data may be in inconsistent state. Due to repetition of same information in multiple files, the update operation if required is to be performed in all the files otherwise one file shows one information and other shows other information for same student. For examples, if address of a student is to be changed, it should be changed in all files otherwise administration section shows one address and exam section shows another address for same student. Thus, consistency is a state where every relation in database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. DBMS follows rules of normalization, which splits a relation/table when any of its attributes is having redundancy in values.

Easy in accessing data.

The data accessing and management is difficult in traditional file system. The conventional file-processing environments do not allow needed data to be retrieved in a convenient and efficient manner. However, the DBMS has features of query or SQL (database command) through which we can retrieve data easily and efficiently. The query makes database operations (retrieval, insertion, update, deletion etc) easy.

Isolation of data and application

Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult using the file system. However, a database system is entirely different from its data. Database is said to active entity while data is said to be passive one on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process. The data and application are isolated such that an application program using a database can easily be changed to use different database.

Maintains Data Integrity

Data integrity refers to the accuracy and consistency of data stored in a database. We can define constraints (conditions) in the database to enforce business rules on data. Business rules specify conditions and relationships that must always be true, or must always be false. Because each company/organization defines its own policies about things like salaries, employee numbers, inventory tracking, and so on, we can specify a different set of rules for each database table. When an integrity constraint applies to a table, all data in the table must conform to the corresponding rule. When you issue a SQL statement (command) that modifies data in the table, database ensures that the new data satisfies the integrity constraint. For example, suppose the college maintains result of students. Suppose also that the college requires that the marks of students in a subject may never fall below zero. We can enforce these constraints (policy or business rule) in the database system. Then, database doesn't allow to enter negative number in the database.

Removes concurrent-access anomalies (Concurrency Control)

Concurrency control is used to address conflicts with the simultaneous accessing or altering of data that can occur with a multi-user system. Concurrency control, when applied to a DBMS, is meant to coordinate simultaneous transactions while preserving data integrity. For the sake of overall performance of the system and faster response, many systems allow multiple users to update the data simultaneously. In such an environment, interaction of concurrent updates is possible and may result in inconsistent data. Consider a bank account with balance of Rs. 10,000. If some amount (say Rs. 500 and Rs.100, respectively) is withdrawn from the same account number from different locations or systems (say one from Biratnagar branch and other from kathmandu branch, one from ATM and one from cheque) at almost exactly the same time, the result of the concurrent executions may leave the account balance in an incorrect (or inconsistent) state. Suppose that the programs executing on behalf of each withdrawal read the old balance, reduce that value by the amount being withdrawn, and write the result back. If the two programs run concurrently, they may both read the value Rs. 10,000, and write back Rs. 9500 and Rs. 9900, respectively. Depending on which one writes the value last, the account balance may contain Rs. 9500 or Rs. 9900, rather than the correct value of Rs. 9400. To guard against this possibility, the system must maintain concurrency control mechanism. Most of DBMS software supports concurrency controls. It is difficult to control the concurrency in traditional file system.

Atomicity problems

Atomicity refers to the ability of the database to guarantee that either all of the tasks of a transaction are performed or none of them. Database modifications must follow an all or nothing rule. Each transaction is said to be atomic if when one part of the transaction fails, the entire transaction fails. A computer system, like any other device, is subject to failure. In many applications, it is crucial that, if a failure occurs, the data is to be restored to the consistent state that existed prior to the failure. Consider two bank accounts A and B with balances Rs. 5000 and Rs 1000 respectively. Let us consider a transaction to transfer Rs. 500 from the account A to the account B. If a system failure occurs during the execution of the program, it is possible that the Rs 500 was removed from the balance of account A but was not credited to the balance of account B, resulting in an inconsistent database state. Clearly, it is essential to database consistency that either both the credit and debit occur, or that neither occur. That is, the funds transfer must be atomic—it must happen in its entirety or not at all. Most of databases support atomicity while it is difficult to main it in traditional file system.

Multiple Views

DBMS offers multiples views for different users. A user who works in sales department will see only sales related information. He will have a different view of database than a person working in production department. This enables user to have a concentrate view of database according to their requirements.

Enhanced Security

DBMS allows to create different users in the system and assigns corrects privileges to them. DBMS enables multiple users to have different view with different features. For example, a user in sales department cannot see data of purchase department. We can customize/manage the data in sales department how much he can see. The features like multiple views offers security at some. DBMS offers methods to impose constraints while entering data into database and retrieving data from the database. Thus, DBMS provides more security in the system than traditional file system.