* 1. Banking: For account information, transfer of funds, banking transactions.

Universities:  For student information, online assignment submissions, course registrations, and grades.

 Airlines: For reservation of tickets, and schedule information.

 Online news sites:  For updating new, maintenance of archives.

1.7)  Both systems contain a collection of data and a set of programs which access that data. A database management system coordinates both the physical and the logical access to the data, whereas a file processing system coordinates only the physical access.

 A database management system reduces the amount of data duplication by ensuring that a physical piece of data is available to all programs authorized to have access to it, whereas data written by one program in a file-processing system may not be readable by another program.

A database management system is designed to allow flexible access to data (i.e., queries), whereas a file-processing system is designed to allow pre-determined access to data (i.e., compiled programs).

 A database management system is designed to coordinate multiple users accessing the same data at the same time. A file-processing system is usually designed to allow one or more programs to access different data files at the same time. In a file-processing system, a file can be accessed by two programs concurrently only if both programs have read-only access to the file.

* 1. Physical data independence is the ability to modify the physical scheme without

making it necessary to rewrite application programs. Such modifications include changing from unblocked to blocked record storage, or from sequential to random access files. Such a modification might be adding a field to a record; an application program’s view hides this change from the program.

* 1. a. interaction with the file manager.

No DBMS can do without this. If there is no file manager interaction then nothing stored in the files can be retrieved.

b. integrity enforcement.

Consistency constraints may not be satisfied, for example an instructor may belong to a non-existent department, two students may have the same ID, account balances could go below the minimum allowed, and so on.

c. security enforcement.

Unauthorized users may access the database, or users authorized to access part of the database may be able to access parts of the database for which they lack authority. For example, a low-level user could get access to national defense secret codes, or employees could find out what their supervisors earn (which is presumably a secret).

d. backup and recovery.

Data could be lost permanently, rather than at least being available in a consistent state that existed prior to a failure.

e. concurrency control.

Consistency constraints may be violated despite proper integrity enforcement in each transaction. For example, incorrect bank balances might be reflected due to simultaneous withdrawals and deposits on the same account, and so on.

1.11) The component of the database that prevents both students from getting the last seat is: transaction isolation

Data that enters the database are expected to maintain accuracy and also be consistent with the database structure.

So, when both students request for the last seat, the possibilities are:

* Student A getstheseat
* Student B getstheseat

The following is not a possibility

* Bothstudentsgettheseat
* None of the students gets the seat

The above highlights means that, only one of the students would get the seat.

This is possible because of the concept called transaction isolation.

The transaction isolation ensures that the data requested by a user is complete and such data maintains competency.

So, when a student gets the last seat, the next student would not get the same seat (or any other seat), because a transaction has already been completed.

1.15) **a**) A users table containing users, with attributes such as account name, real name, age, gender, location, and other profile information

**b.** A content table containing user provided content, such as text and

images, associated with the user who uploaded the content.

**c**. A friends table recording for each user which other users are connected to that user. The kind of connection may also be recorded

in this table.

**d**. A permissionstable, recording which category of friends are allowed

to view which content uploaded by a user. For example, a user may

share some photos with family but not with all friends.