Sequential File Design

If a particular record must be found in a file and the number of records in the file is very small, then it may not be difficult to search the file from beginning to end to find the desired record. For files containing large numbers of records, however, this method is inefficient. A special ordering technique is needed so that records can be retrieved more easily. For this reason, records may be arranged according to a **key value** (often referred to as a **primary key**). **The key is one data field chosen to identify the record.** Since a key is used to locate a particular record, it must be **unique**; that is, no two records in a file can have the same key value. In the example below, the social insurance number field is used as the key.

SIN (key)	**Last Name**	**First Name**
628-306-591	Smith	Terry
632-233-579	Wilson	Barbara
633-306-591	White	Doug
640-354-966	Smith	Heather
641-692-853	McMillan	Terry

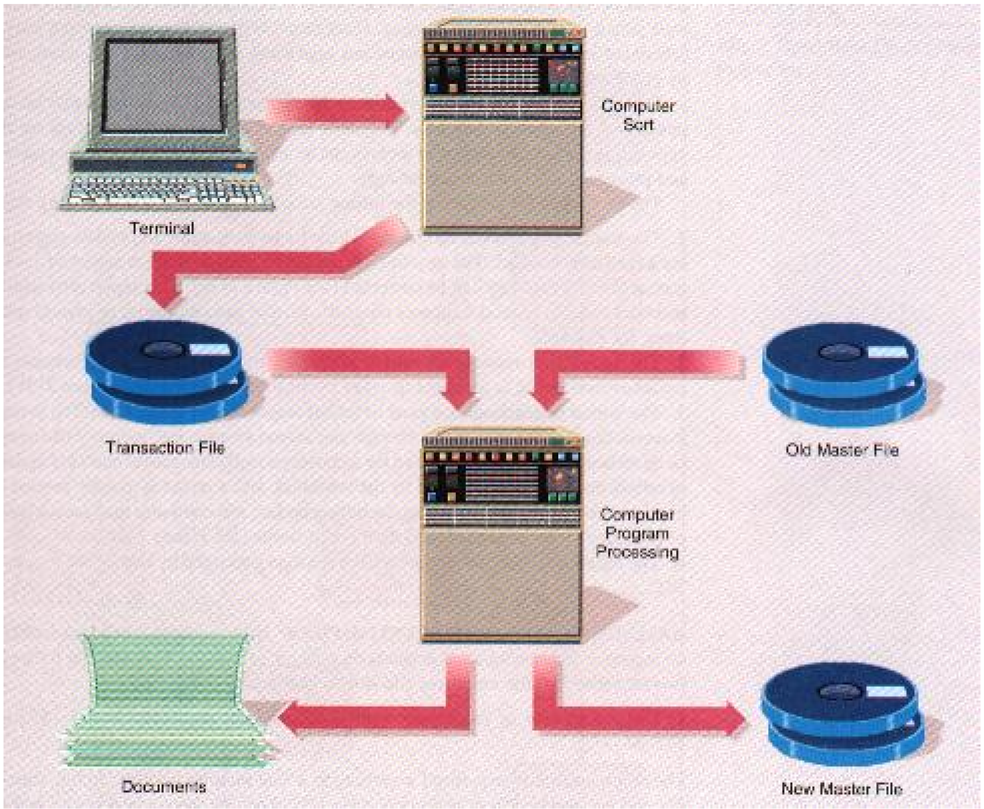
Employee File

Social insurance numbers are excellent keys for employee records because no two people in the Canada have the same number. An employee record is located by searching for the appropriate value in the social insurance field. The key value in an inventory file could be the item number. When records are ordered according to their key values, a **sequential file** is formed.

Updating a sequential file involves two sets of files: the master file and transaction file. The **master file** is the file containing all existing records. The **transaction file** is the file containing the changes to be made to the master. During the updating process a third file, the **new master file**, is created.

Before updating begins, the **old master file** must be in sequential order according to the **key** field. The **transaction file** should then be sorted on the **same key** as the **master file**. The computer compares the key of the first master file record with the key from the first record in the transaction file. If the keys do not match, the computer writes the record to the **new master file** as is, and then it reads the next record on the **master file**. When a match between the master and transaction records occurs, the computer updates the **new master file** record. Sometimes if a transaction record has no matching master record, the computer may generate an error message. Some transactions may add a new record, some may modify a record, while others may delete an existing record.

With sequential processing there is no direct way to locate the matching master record for a transaction. Each time a certain master record is to be updated, the entire master file must be read and a new master file created. Since this makes updating one record in a file very time-consuming, transactions are collected over a period of time and processed in one run (see below). Therefore, **batch file access must be used with sequential file design**.



Sequential Processing

EXAMPLE OF SEQUENTIAL PROCESSING

The preparation of customer bills is well suited to sequential processing. Customers' bills are usually prepared only at scheduled intervals. Standard procedures apply and large numbers of records must be processed.

MAKING INQUIRIES TO SEQUENTIAL FILES

How inquiries into a sequential file on magnetic tape are handled depends on the type of inquiry. Consider the following two inquiries into the employee file shown below.

1. List the records of employees with social security numbers 234-02-1532 and 372-84-7436.

   The employee file is sequenced according to social security number. In this case the file will be searched from beginning to end by checking the social security number key. As soon as the required social security numbers are located and the required records listed, the search is stopped. Of course, if the numbers are the last two records on the file, then the entire file must be searched.

2. List all employees from the area with zip code 43403.

   For this inquiry the entire file will again be searched. In this case, the zip code field of each record must be checked to see if it matches 43403. This illustrates one problem with referring to a non-key field on sequential files. If an inquiry is based on a field other than the key, a great deal of time may be wasted in the search process. To alleviate this problem a second employee file, ordered by zip code, could be created; however, this approach requires multiple files with duplicate data.

ASSESSMENT OF SEQUENTIAL FILE DESIGN

In determining if sequential file design is appropriate for a particular application, two factors must be considered--activity and volatility.

Activity refers to the number of records that must be updated during a batch processing run. For example, if 400 of a total of 500 records must be updated during a run, the activity rate would be 80 percent. This would be considered a high rate of activity.

Volatility refers to how often a particular record may be updated during some period of time. If a record gets updated once every two weeks, the volatility might be considered low.

Sequential processing and file design are suitable for applications with high activity and low volatility. Examples of such applications (requiring large numbers of records to be updated at specific times) include payroll processing, updating the addresses of magazine subscribers, and preparing student grades.

Advantages of sequential file design include the following:

- It can be cost effective when at least half the records in a master file are updated during one processing run.
- The design of sequential files is simple.
- Magnetic tape, a low cost medium, can be used to maximum advantage.
- Old master and transaction files provide automatic backup for the system.

Certain disadvantages characterize this method of processing, however:

- The entire master file must be processed and a new master file created even when only a few master records have to be updated.
- Transactions must be sorted in a particular sequence; this takes time and can be expensive.
- The master file is only as up to date as the last processing run. In many instances, using information from a master file that has not been recently updated results in the use of old, and sometimes incorrect, information.
- The sequential nature of the file organization makes it difficult to provide information for unanticipated inquiries such as the status of a particular record, because all information is retrieved by reading the file from beginning to end.

Last Updated Jan.6/99