TRAINING COURSE

ON

SOFTWARE FOR GROUNDWATER DATA MANAGEMENT

UNDER

WORLD BANK FUNDED HYDROLOGY PROJECT

LECTURE NOTES
ON

DATA MANAGEMENT WITH dBASE

BY

D CHALISGAONKAR

ORGANISED BY

NATIONAL INSTITUTE OF HYDROLOGY
ROORKEE - 247 667
INDIA

DATA MANAGEMENT WITH DBASE

1.0 INTRODUCTION

Analysis of large volume of hydrological data collected over a considerable period of time, is required for formulation of any water resources project besides determining its scope. This data is required to be collected and preserved even after a project is completed.

The relevant data collected are recorded in bulky registers even now. This practice of data storage has many limitations and disadvantages. It is very difficult to keep an up to date inventory of data. Many times data are lost due to damage to registers or files. If the data are to be manually copied whenever required there is every possibility of errors in copying besides taking considerable time. If there is any change in the format of data, the data have to be rewritten in the desired format.

The advent of computers has facilitated the storage of voluminous data and their retrieval in a easy fashion so that the same can be processed and presented in the manner required. In computer storage systems the related data items are collected into records and the space reserved for each data item is called a field. Here the similar records are grouped together into files and several files are normally help in one storage device.

To handle these problems a Data Base Management computer software can be used very efficiently. dBase is one of the Data Base Management software package available on PCs. dBase is capable of adding new data records and modifying/retrieving the desired data from the existing database. Using dBase, time series hydrometeorological parameters like ground water elevation, daily rainfall, hourly rainfall, humidity, evaporation, minimum temperature, mean temperature, maximum temperature, stage, discharge, sediment load, stage during flood, reservoir level, water quality parameters etc, can be easily handled. It also provides simple statistical information.

2.0 FEATURES OF DBASE

The dBase is a second generation relational database program using easily recognized English like commands. The main features are:

- It has its own programming language with a large number of simple and powerful commands with which it is easy to organize and reproduce information.

- Program and data are independent of each other. The user can change the structure of the data without disturbing the data already created.

- Each record can contain up to 128 separate fields and 4000 characters.

- It can use up to 10 files simultaneously to obtain the information.

- The user can easily add, edit, delete and retrieve the data with minimum programming.

- Custom screen formats can be created for data entry so that the data entry operator has to fill up the blanks only.

3.0 HOW TO USE DBASE

Set the directory containing dBase software as default directory. Type dbase at the command prompt and press <Enter> key. Some information about version is displayed followed by a dot prompt at the lower left corner of the screen. The commands of dBase can be entered at this prompt. The commands are terminated by pressing the <Enter> key.

3.1 Creating the Database

The CREATE command is used to form a new database. To create a database type CREATE and press <Enter> key. For example,

CREATE

The following message will appear on the screen :

Enter the name of new file :

After entering the file name, following will be displayed on the screen.

```
drive name : File name Bytes
Remaining Fields defined 0

Field name  Type  Width  Dec
|________| |___| |_____| |___|
```

dBASE allows a maximum of 128 fields.

3.2 Rules Governing Field Name :

- It can be maximum 10 characters long.
- It may be formed by alphabets, numerals or underscore character (_) but the first can be only the alphabet and the last character can not be the underscore character (_).
- The alphabets, if entered in lower case letters, will automatically be converted to uppercase letters.
- First character cannot be blank.

3.3 Rules Governing Field Type :

dBASE allows five types of fields. They are as follows:

Field Type	Denoted by	Contents possible
ALPHANUMERIC	C	Any alphabet, number or special characters
NUMERIC	N	Any number
LOGICAL	L	T or F (denoting True or False)
MEMO	M	Long passages of text
DATE	D	Date in the form MM/DD/YY

3.4 Rules Governing Field Width :

- Field width can be maximum of 254 characters
- If the number to be entered is real then width includes the number of digits before the decimal and the number of digits after decimal.
- If field-type is 'C', the maximum field width permissible is 254, if field-type is 'N', the same is 19.

3.5 Rules Governing Dec :

DEC denotes the number of digits after decimal. It can be zero, denoting no significant digit after decimal. The range of Dec width is 0-15 and should be at least 2 less than the field width.

4.0 TO CREATE THE DATA STRUCTURE

To fill in information on CREATE command, type the field name and press the <Enter> key. The cursor will move to the 'Type' column. Press C, N, L, M or D, as the case may be, followed by <Enter> key. The cursor will move to 'Width' column. Enter the width of the field. If field type is N (numeric) the cursor will move to 'Dec' column, otherwise to the next field. To come out of this process, press <Enter> key at field name. Following will be the message displayed:

HIT RETURN to confirm any other key to resume

Now press the <Enter> Key. The file structure will be saved by the name of the file entered at the time of creation and following message will be displayed:

Input data records now (Y/N)

In this case either one would either like to enter the data just after one has created the structure, or some other time. In the first condition, one would enter Y; in second option, one would enter N.

5.0 SELECTING AND DESELECTING A DATABASE

USE command selects and deselects a database. This command brings the structure of the desired file in computer's memory, or the database will be ready to accept the data. The syntax is :

```
USE <file_name>
```

USE command without parameter closes the file in use.

6.0 ADDING DATA IN SELECTED DATABASE

After creating data base, user can add the data in the data in the database. This can be done with the APPEND command. The syntax is :

APPEND

This command displays a form to be filled on the screen. Another form will be displayed after the first one has been filled. <Ctrl>+<W> can be pressed together to save the data and to come back to dot prompt.

7.0 DEFINING FUNCTION KEYS

Sometimes, the user may have to type same words a number of times. e.g. if we are entering data in a database having field CITY then we might have to type city name as many times as number of records. It becomes very cumbersome and time consuming when we have to write same thing many times. This job could be made very simple with the help of this command. To do this we have to
reassign keyboard function keys (F2-F10).

Suppose we have to write word "LUCKNOW" in every or many records of a database then we can assign value "LUCKNOW" to any function key except F1. e.g.

```
.SET FUNCTION 5 TO "LUCKNOW"
```

Now F5 key is assigned value "LUCKNOW". Now the word "LUCKNOW" can be entered with a single key stroke (just pressing F5 key) whenever required.

8.0 TO MOVE POINTER TO A PARTICULAR RECORD

GO N or N	Move Pointer to n th record
Go TOP	Move Pointer to the top of file in use.
GO BOTTOM	Move pointer to the end of file in use.
SKIP	Move pointer to next record.

9.0 MODIFYING A DATA FILE

There are two commands to edit the contents of the database file.

9.1 EDIT

This command edits one record at a time. For example

Edit	Edits current record
Edit 5	Edits record number 5

9.2 BROWSE

This command allows the user to scroll through the database both horizontally and vertically to edit or add records. In BROWSE, one screenful of records are shown at a time. The screen can be shifted to the left if some of the fields are hidden from view beyond the right edge. The syntax of this command is as follows:

BROWSE

10.0 DELETING AND UNDELETING RECORDS

10.1 DELETE

The DELETE command can be used to mark records for deletion, however, to delete the records permanently PACK command is used, e.g.

DELETE NEXT n (a block of next n records)
DELETE REST remaining file.

e.g.

DELETE RECORD 1
will mark record number 1 for deletion.

If the user gives DISPLAY or LIST command, he can observe a * (Asterisk) sign in front of the first record which has been marked for deletion. To check the deleted records, the command is :

LIST FOR DELETED()

10.2 PACK

This command deletes the records permanently which have been marked for deletion. The syntax is :

PACK

10.3 RECALL

This command recalls the records marked for deletion. It removes the asterisk sign prefixed before the marked records, e.g.

RECALL RECORD 1	Recalls record number 1
RECALL ALL	Recalls all records marked for deletion.

11.0 SEARCHING AND LOCATION RECORDS

11.1 FOR & WHILE Parameters

These parameters can be used for selecting the records which satisfy the desired conditions. The WHILE condition displays records as long as successive records fulfill the same conditions, whereas the FOR condition displays all records that satisfy the condition. The < (less than), > (greater than), < = (less than or equal to), > = (greater than or equal to), < > (not equal to) and # (not equal to) signs can be used for checking these conditions. These conditions can be used for following type of fields:

- Numeric Fields
- Character Fields
- Logical Fields
- Date Fields

11.2 DISPLAY and LIST

Both these commands can be used to view the contents of the database file. The basic difference between DISPLAY and LIST is that only one record is shown if DISPLAY command is given whereas all the records of the database are shown if LIST command is issued. The syntax of DISPLAY and LIST command is as follows:

```
DISPLAY [OFF][Scope][field list][Condition For or WHILE] [TO PRINT]
```

Where

[]	Optional part
OFF	Suppress the record numbers
Field list	The fields which are required to be displayed.
Scope	Numbers of records to be included in display or list. They may be :

- ALL

- RECORD n (a specific record)
- NEXT n (a block of next n records)
- REST (remaining file)

Following are the examples of normal DISPLAY command and DISPLAY command with FOR condition :

DISPLAY	Displays the current record
DISPLAY STRUCTURE	Displays the structure of the database
DISPLAY ALL	Displays all records. The display will halt temporarily after displaying one screen of records.
DISPLAY NEXT 5	Display next 5 records
DISPLAY ALL FOR SALARY > 1000	Displays records in which salary > 1000.
DISPLAY ALL FOR SALARY # 2500	Displays records in which salary is not equal to 2500.
DISPLAY ALL FOR NAME = "GUPTA"	Display all records in which NAME is GUPTA
DISPLAY ALL FOR NAME < "R"	Displays all records with NAME beginning with any letter below R in alphabets i.e. A to Q.
DISPLAY ALL FOR NAME < > "GUPTA"	Displays all records which do not contain "GUPTA" in NAME.
DISPLAY ALL FOR NAME="GUPTA".AND. SALARY>2500	Displays all records with NAME = GUPTA and SALARY > 2500
DISPLAY ALL FOR SUBSTR(NAME,1,1)="N"	Displays all records having N as first character of the NAME field.
DISPLAY TO PRINT	Displays and prints all records

The syntax of LIST is as follows:

LIST [Scope] [OFF] [FOR or WHILE Condition] [TO PRINT]

LIST DESIGN,SALARY	Displays the fields DESIGN and SALARY of all the records.

LIST FOR DESIGN ="OFFICER"	Displays all records for which DESIGN = "OFFICER"

11.3 LOCATE

This command is used for locating the position of a record based upon a desired characteristics. It only displays the record number. e.g.

```
LOCATE FOR NAME = "RAM"
```

will display the record number at which the name RAM is present. If no record with NAME = "RAM " is found then dBASE will display the message. "End of Locate Scope".

12.0 MANAGING NUMBERS IN A DATABASE

12.1 COUNT

This command counts the number of records in the active database that meet the specified condition. e.g.

```
COUNT FOR CODE = "C001"
```

will count the total number of records with CODE = "C001"

```
COUNT NEXT 5 FOR CODE = "B001"
```

will count the total number of records with CODE = "B001" in next 5 records.

12.2 AVERAGE

This command computes the arithmetic mean of numeric expressions. e.g.

```
AVERAGE QTY,AMOUNT
```

will give the average of QTY field and average of AMOUNT field.

```
AVERAGE QTY,AMOUNT FOR DESCRIP = "PEN"
```

will give the average of the field QTY and average of the field AMOUNT for the record satisfying the condition DESCRIP ="PEN"

12.3 SUM

This command totals expression involving numeric fields in the active database file. e.g.

```
SUM QTY, AMOUNT
SUM QTY, AMOUNT FOR CODE = "C001"
SUM QTY, AMOUNT FOR DESCRIP = "PEN"
```

12.4 TOTAL

This command sums the specified numeric fields of the active database file, and creates a new database file containing the sums of totalled fields as well as all other fields. The source file must be INDEXed or SORTED. e.g.

```
TOTAL ON CODE TO NEW FIELDS QTY, AMOUNT
```

An another file NEW.DBF will be created having totals of QTY and AMOUNT codewise.

13.0 DATE FUNCTIONS

Suppose a database ABC.DBF has the following records

CODE	DESCRIPTION	QTY	AMOUNT	DATE
1012	RUBBER	200	2.00	02/08/93

- **CDOW** Returns the name of day of date in characters. e.g. the command LIST CDOW(DATE) will give Monday.
- **DOW** Returns the day of the date in number.
- **CMONTH** Returns the name of the month of the date in character e.g. LIST CMONTH(DATE) will give February.
- **MONTH** Returns the month of the date in number.
- **DAY** Returns the day of the date in number.
- **TIME()** Returns the time within the system in HH:MM:SS format
- **DATE()** Returns date within system in MM/DD/YY
- **CTOD** Character to Date conversion

In manipulating the date type fields in database the comparison between two dates is possible only on values of DATE type. It is not sufficient to write the string in MM/DD/YY format to do either of these type of operations. for example

- **DTOC** Date to character conversion. This will convert a DATE type value into

Character type. e.g. LIST FOR DTOC(DATE) = "02/10/93". Here DATE is of type DATE. Now dBASE will list the records in which field DATE will match the string "02/10/93".

14.0 STRING HANDLING FUNCTIONS

- **UPPER** Converts lowercase letters into uppercase. ?UPPER ('ABC') will give ABC
- **LOWER** Converts uppercase letters into lowercase. ?LOWER ('ABC') will give abc
- **SUBSTR** Selects SOME character from a string. The syntax is:
  SUBSTR(STRING, N, M) Where, STRING specifies string to be used in command, N is the character position where chopping will start and M tells the number of characters to be chopped, e.g.
  ?SUBSTR ("NAVEEN",2,2) will give "AV"

15.0 SORTING & INDEXING

15.1 SORT

This command sorts the database file in use and create another database file with sorted records. The syntax is :

```
SORT ON <key field> TO <field name>
```

e.g.

```
SORT ON DESIG TO TEMP
```

will sort the database file in use on key fields DESIG and create another file with name TEMP.DBF. The sorted records can be seen as follows:

```
.USE TEMP.LIST
```

15.2 INDEX

This command sorts the database file in use and creates an index file. The syntax is as follows :

```
INDEX ON <key field> To <file name>
```

e.g.

```
INDEX ON NAME TO TEMP
```

will index the database file in use on key field NAME. The indexed file can be seen by using the TEMP file the syntax for which will be :

USE<Original file name> INDEX <Indexed file name>

The difference between SORT and INDEX command is, in SORT record number will also change after sorting, but in INDEX record number will remaining same after indexing.