

# R S I T कम्प्यूटर कॉलेज करकेली

शिव मंदिर के पास, रेलवे स्टेशन रोड़ करकेली  
जिला- उमरिया (म.प्र.) 484661



## Subject – Fox Pro

Session 2018-19

- **PGDCA**
- **DCA**
- **DCPA**
- **ADCP**
- **MCIT**

एवं अन्य 8 वीं पास सर्टिफिकेट कोर्सेस

Prepared by: -

Prakash Dwivedi (BE) 8982505087

Abhilash Pathak (MCA) 8517906324, 7974259812

Website: [www.rudrasoftech.in](http://www.rudrasoftech.in)

## UNIT - I

➤ **Explain DBMS and RDBMS in detail.**

➤ **DBMS: -**

DBMS का full form Data Base Management System है। Data Base दो शब्दों से मिलकर बना है। (Data + Base) जिसका अर्थ है, Data का प्रबन्धन अथवा रखरखाव। Data कुछ भी हो सकता है, जैसे- Number, name of person, name of city इत्यादि। Data को Process करने पर हमें information मिलती है जो किसी भी Organization के लिए important होती है। Data को computer में records के रूप में Store कर सकते हैं। अतः Data base व्यवस्थित आँकड़ों का collection है।

इस Data Base को manage करने के लिए DBMS का use किया जाता है। DBMS का use Data-Base को manage करने अर्थात् stored records से कुछ हटाना, Records के structure में correction, उन्हें किसी order में arrange करना, report तैयार करना आदि किया जाता है। Data Base के निम्नलिखित Part होते हैं -

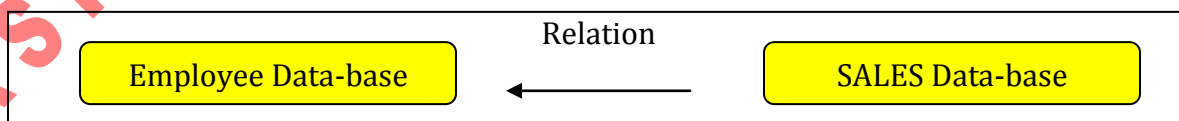
1. **Field: -** Data-Base Table की एक column, जिसमें एक ही type के data present हो Field कहलाता है।
2. **Record: -** Data-Base Table की एक Row की Information जो अनेक Field से मिलकर बनी हो Record कहलाता है।
3. **Table Name: -** Data-Base Table की Field का नाम Table Name कहलाता है।
4. **Field Name: -** Field के शीर्षक (heading) को Field Name कहते हैं।

	Field				
Field Name	Roll No.	Name	Address	Date of Birth	Table Name
	1	Prakash	Vikatganj Umaria	11- June-1987	}
	2	Abhilash	Single Tola Umaria	01-April-1990	
	3	Rupesh	Forest Naka Umaria	18-June-1991	
	4	Akash	Near Bus Stand Umaria	02-July-2005	
					Records

➤ **RDBMS: -** RDBMS का full form Relational Data Base Management System है। किसी Data Base System में निम्नलिखित कमियाँ हो सकती हैं -

- Repetition of Information
- निश्चित Information को present करने में अक्षमता।

उपर्युक्त कमी को दूर करने के लिए एक और Database system develop की गई जिसे RDBMS कहते हैं। इसके द्वारा Related Data को Join करके आवश्यकतानुसार Record को Display किया जा सकता है। RDBMS में दो Database files में Relation एक common field के द्वारा स्थापित करते हैं।



EMP_Code	NAME	CITY
101	Shadat Khan	Shahdol
102	Anuj	Umaria
103	Abhilash	Katni
104	Prakash	Umaria

Date	EMP_Code	Sales
10/23/15	101	10,000.00
05/05/15	103	8000.00
20/07/15	104	5000.00
10/06/14	102	8000.00

Relation  
↔

105	D.P. Singh	Umara
-----	------------	-------

02/02/14	105	15,000.00
----------	-----	-----------

**Primary Key****Foreign Key****❖ FoxPro: -**

FoxPro एक DBMS है। यह FoxBASE+ का संशोधित (edited) version है। FoxPro को RDBMS (Relational Data Base Management System) भी कहते हैं। सन् 1979-80 में सर्वप्रथम DBMS Software As then-Tate ने develop किया था, जिसका नाम D-Base-II था। उसके बाद Company ने Second Version सन् 1984 में किया जिसका नाम D-Base-III था। उसके बाद FoxPro को develop किया गया।

**➤ FoxPro में हम निम्नलिखित कार्य कर सकते हैं –**

- Database files को आवश्यकतानुसार Design करना।
- Database files में Data insert करना।
- Insert data को Organized करना।
- Insert data को जोड़ना, उसमें बदलाव करना तथा delete करना।
- Inset data का आपस में Relation स्थापित करना।
- आवश्यकतानुसार Data को Display या print करना।

**➤ Version of FoxPro –**

- Fox Base
- Fox Base +
- Fox Pro
- FoxPro 2.0
- FoxPro2.5
- FoxPro 2.6
- Microsoft Visual Fox Pro 6.0

**➤ FoxPro में आवश्यक Software and Hardware को समझाइए?**

FoxPro को चलाने के लिए Required Software and Hardware निम्नलिखित हैं–

1. 80286 अथवा High Processor
2. Hard Disk
3. Mouse
4. VGA Monitor
5. At least 4 MB RAM
6. Window 95 अथवा उसके बाद के Operating System.

**Question: - FoxPro में Data Base file Create करने के steps को समझाइए।**

**Ans:** Fox Pro में New Data Base File तैयार करने के लिए create command का Use किया जाता है। इस command का Use दो प्रकार से किया जाता है–

1. File Menu के New option पर click करके।
2. Command Window में Create Command Type करके।

**(1) File Menu द्वारा :-** यदि हम Menu द्वारा New File बनाना चाहते हैं तो File Menu में New Option को click करते हैं। जिससे एक Dialog Box Open होता है। इस Dialog Box में Radio Button Table/DBF पर click करके New Button पर Click करते हैं। जिससे Screen पर Table Structure Dialog Box Open होता है।

**(2) Create Command द्वारा :-** Create Command का Syntax निम्न प्रकार होता है–

**Create <Data Base File Name>**

FoxPro में कोई भी Command, उसके पहले 4 character Type करके Perform की जा सकती है। Command Window में Syntax के Base पर Create के साथ file का नाम देकर Enter Button Press करते हैं। जिससे Table Structure Dialog Box Open होता है जिससे Data Base File से Related information दी जाती है।

**Example:** - Field Name, Tata Type, Width, Decimal Value.

➤ **Field Name** देने के **Rules** निम्नलिखित हैं –

1. यह Maximum 10 character का हो सकता है।
2. यह हमेशा Alphabet से ही start होता है।
3. इसमें 0 to 9 digit का use किया जा सकता है, परन्तु Special Character में Under Score ( \_ ) का ही प्रयोग किया जाता है।

**Example** – Name

➤ **FoxPro** में निम्नलिखित **7 Data Types** का **Use** किया जाता है –

1. Character
2. Numeric
3. Float
4. Date
5. Logical
6. Memo
7. General

Width, Field की लम्बाई के लिए दिया जाता है। Decimal Value दशमलव वाले अंको के लिए प्रयोग किए जाते हैं। (जैसे - 102.5) Table Structure Window में Field की information देने के बाद OK Button पर click करते हैं या Shortcut Key (Ctrl+w) को press करते हैं। जिससे यह Record Enter करने को पूछता है। Yes पर click करने पर Data Enter करने के लिए End Window display होती है जिसमें Field Name एक के बाद एक Horizontally Display हो रहे होते हैं। Record Type करने पर दूसरी खाली Record Display हो जाता है। Last में Ctrl+w या Menu द्वारा Save करके File को Close किया जा सकता है, एवं Command Window में List Command Type करके बनाई गयी Data Base Table को Display कर सकते हैं।

**Que: - Explain the following Command with example.**

- |                      |                 |             |
|----------------------|-----------------|-------------|
| 1. Create            | 7. Copy File    | 13. Replace |
| 2. List              | 8. Locate/Go To | 14. Delete  |
| 3. Append            | 9. Edit         | 15. Recall  |
| 4. Close             | 10. Change      | 16. Pack    |
| 5. Quit              | 11. Browse      | 17. Zap     |
| 6. Display Structure | 12. Rename      | 18. Set     |

**(1) Create Command:** - इस Command का use New Data Base File, Report, Menu, Label आदि बनाने के लिए किया जाता है।

**Syntax:** - Create <Data Base File Name>

**Example:** - Create Employee

**(2) List Command:** - List Command का Use Data Base File के समस्त Record को Screen पर Display करने के लिए किया जाता है। ऐसा करने के लिए हमें File Open करनी होगी जिसका Data हमें देखना है।

**Syntax:** - List

**Example:** - Use Employee

List

उपरोक्त Command के बाद Data Base File Employee के सारे Records Screen पर Display हो जायेगा। List Command के द्वारा Selected Field का Data देखने के लिए निम्न Command का Use होगा –

LIST NAME. BASIC

**(3) Append Command:** - इस Command का Use बनाए गए Data Base File में New Records को Add करने के लिए किया जाता है। इस Command को Run करने पर record edit window open होती है जिससे horizontally New Record Add किए जाते हैं।

**Syntax:** - Use Employee  
Append

**(4) Close Command:** - इस Command का प्रयोग Data Base File को Close करने के लिए किया जाता है।

**Syntax:** - Close All  
**Example:** - Close All

**(5) Quit Command:** - इस Command का प्रयोग Fox Pro की window को Close करने के लिए किया जाता है।

**Syntax:** - Quit

**(6) Display Structure Command:** - इस Command का Use Current Record को Display करने के लिए किया जाता है। यह List Command से अलग होती है।

**Syntax:** - Display < Condition >  
**Example:** - Locate Record 4  
Display Next 3

**(7) Copy File Command:** - Copy File Command के द्वारा किसी File की Copy बनाई जा सकती है।

**Syntax:** - Copy File < Source File > To < Destination File >  
**Example:** - Copy File Sales .DBF To Sal - MAS .DBF  
1024 bytes copied

FoxPro File को Copy करने के बाद Number of bytes Copied को भी Display करता है।

**(8) Locate/Go to Command:** - इस Command का Use Data Base File को किसी Record को Point करने के लिए किया जाता है।

**Syntax:** - Locate For < Condition >  
Use Employee

**Example:** - Locate Record 4  
Display

इस Command के द्वारा वे सभी Record Lock हो जायेंगे जो दी गई शर्त को पूरा कर सकते हैं। उन Records को Display command के द्वारा देखा जा सकता है।

**Example:** - USE EMPLOYEE  
LOCATE FOR NAME = "PRAKASH DWIVEDI"  
DISPLAY

**(9) Edit/ Change Command:** - इस Command का Use Data Base Table के पुराने Record में बदलाव करने के लिए किया जाता है। Edit and Change Command के द्वारा Horizontally Record को Edit कर सकते हैं, एवं Browse Command के द्वारा Vertically Edit कर सकते हैं।

**Syntax:** - USE EMPLOYEE  
**EXAMPLE:** - EDIT/CHANGE

उपरोक्त Command के प्रयोग से File के सभी Records Screen पर Correction के लिए Present हो जायेंगे।

USE EMPLOYEE  
EDIT FIELD EMP\_CODE, NAME

इस Command के द्वारा केवल 2 ही field Correction के लिए Present होंगे।

USE EMPLOYEE  
EDIT FOR DEPT\_CODE = "D20"

उपरोक्त Command द्वारा Department Code D20 वाले Records Correction के लिए present होंगे।

**(10) Browse Command:** - यह Command एक Window में खुली हुई Data Base Files के सभी Records एक साथ देखने तथा उसमें Correction करने की facility provide करती है। यह Window, Browse Window कहलाती है। Browse Command देने पर निम्न Window Display होती है –

```
USE EMPLOYEE
BROWSE FIELDS EMP_CODE BASIC
```

BROWSE COMMAND के प्रयोग द्वारा किसी Special Field के Records में Correction के लिए FREEZE Option का Use किया जाता है।

```
BROWSE FREEZE BASIC
```

Browse Command के द्वारा Field का Size भी Change किया जा सकता है, तथा उसमें New Records को add या remove किया जा सकता है।

**(11) Rename Command:** - इस Command के द्वारा किसी File को दूसरा नाम दिया जा सकता है। अर्थात् उसका नाम बदला जा सकता है।

**Syntax:** - RENAME < old file Name > To < New file Name >

**(12) Replace Command:** - इस Command का Use Data Base Table के किसी निश्चित Record को जो Table में एक से ज्यादा बार Use किया जाता है, एक साथ बदलने के लिए किया जाता है।

**Syntax:** - Replace <field Name> With < Old Record > For < Field Name > = < New Record >

**Example:** - Replace Name with "Abhilash" For Name = "Prakash"

```
USE EMPLOYEE
REPLACE BASIC WITH BASIC +200
```

**(13) Delete Command:** - Delete Command का Use Data Base File के Records को अस्थायी रूप से Delete करने के लिए किया जाता है। यह Command वर्तमान Record को Delete के चिन्ह (\*) से Marked कर देती है।

**Syntax:** - DELETE ALL

इसके द्वारा Data Base File के सारे Records Correction के लिए Marked हो जाते हैं।

**(14) Recall Command:** - इस Command का Use अस्थायी रूप से हटाए गए Record को वापस लाने के लिए या Unmark करने के लिए किया जाता है।

**Syntax:** - Recall < Record No. >

**Example:** - Delete Record 4

Recall Record 4

- (1) USE EMPLOYEE  
RECALL ALL
- (2) RECALL NEXT 3
- (3) RECALL FOR AGE > = 20
- (4) RECALL REST

**(15) Pack Command:** - PACK Command के द्वारा खुली हुई Data Base File में Delete के लिए चिन्हित (Mark for Deletion) Record ही स्थायी रूप से Delete किये जा सकते हैं।

**Syntax:** - DELETE ALL  
PACK

**(16) Zap Command:** - ZAP Command के द्वारा खुली हुई Data Base File के सभी Records को बिना Mark किये स्थायी रूप से Delete किया जा सकता है। इसके प्रयोग के बाद File में केवल ZAP Field Structure ही शेष रह जाता है। ZAP Command में File Delete नहीं होते।

**(17) Set Command:** - इस Command का Use Data Base के Status को ON करने के लिए करते हैं। इस Command का प्रयोग निम्नलिखित रूप से कर सकते हैं –

**Syntax:** Set Start ON

**Que: - What is Memo Field? Explain in detail with suitable Example.**

**Ans.** FoxPro में Memo Field का Use Long Text को Store करने के लिए किया जाता है। Character Field 254 Character से ज्यादा Size का नहीं हो सकता। अतः इससे ज्यादा Character Store करने के लिए Memo Field का Use किया जाता है। इसके लिए Memo Field में Data Base Create करना होगा।

**Memo Field बनाने के लिए –**

- EMP File को Active करे।
- Use EMP Command Type करे।
- MODI STRU Type करे।
- Table Structure Window दिखाई देगा।
- Name Column में REMARKS Type करें।
- Type Column में जाने के बाद M दबायें।
- Memo को Select करे।
- OK पर Click करें।

Memo Field, Data Base File में 10 Bytes का स्थान लेता है। जिसमें आप असीमित Text Store कर सकते हैं। Memo Field का Data other Data Base File में Store होता है, जिससे Main Data Base File का Size कम हो जाता है। Memo Field का Extension. FPT होता है।

**Example: -** PRODUCT NAME से Table बनाई, जिसका structure निम्न प्रकार का है –

Field Name	Type	Width
Pno	Character	6
Pname	Character	20
Desc	Memo	10

Memo Field में Data Insert करने के लिए Courser को Edit Window में Memo Field पर लाकर Ctrl+Home या Ctrl+PgDn Key Press करे, या Memo Field पर Double Click करें। इसको करने पर एक New Window Open हो जाएगी, जिसमें आप Text inset कर सकते हैं। Save करने के लिए Ctrl+W दबायें।

Memo Field को Display करने के लिए List Command के साथ Memo Field का नाम देकर Memo Field के Contents को Screen पर Display किया जा सकता है –

LIST OFF Pno, Desc.

**Que: - FoxPro में विभिन्न Data Types को समझाइए।**

**Ans.:** - FoxPro में 7 प्रकार के Data Type होते हैं, जिनका Use विभिन्न उद्देश्य के लिए किया जाता है। जो निम्न लिखित हैं –

1. **Character:** - इस प्रकार की Data type में सभी Character (A to Z), digit (0 to 9) and Special Character (@, !, %) का प्रयोग किया जाता है। इस Field का आकार By Default 10 होता है, तथा इसमें maximum 254 Character Store किये जा सकते हैं। इस Field में कोई Mathematical Operation का Use नहीं किया जाता है।
2. **Numeric:** - इस प्रकार के Data Type का प्रयोग digits को Store करने के लिए किया जाता है। इस Field का Size By Default 8 Character होता है एवं Maximum Size 19 Character हो सकता है। इसमें Decimal Numbers भी Use किये जा सकते हैं।
3. **Float:** - यह Data Type Numeric Data Type की तरह ही होता है। इस Data Type पर Scientific Calculation की जाती है। इसमें +, - चिन्ह एवं दशमलव का Use कर Accurate Result पाए जा सकते हैं। इसका Maximum Size 20 Character होता है। Numeric and Float में अंतर सिर्फ इतना होता है कि Numeric Data Type में Mathematical Calculation की जाती है एवं Float में Scientific Calculation की जाती है।
4. **Date:** - यह Data Type, Date को Store करने के लिए Use किया जाता है। जैसे – Date of Birth, Bill Date आदि। इसका Size By Default 8 Character होता है एवं Syntax mm/dd/yyyy होता है।
5. **Logical:** - यह Data Type सिर्फ T(True) या F(False) के रूप में Store करती है। इस Field का Size सिर्फ 1 Character होता है। Yes/No में Y/N का प्रयोग किया जाता है।

6. **Memo:** - यह Data Type Character Data Type की तरह ही होता है। इस Data Type में Character Data Type की अपेक्षा अधिक Character Store किये जा सकते हैं। इसका Size By Default 10 Character होता है। एवं Maximum Size Computer में उपस्थित Total Free Space होता है।
7. **General:** - यह विशेष प्रकार की Data Type होती है। जिससे Picture, Sound, Word Document, Spreadsheet Store किये जा सकते हैं। यह Field सिर्फ Window पर आधारित FoxPro में उपयोग की जा सकती है।

❖ **Difference between LIST and DISPLAY ALL:** - DISPLAY Command मुख्यतः निम्नलिखित formats में Command window में प्रयुक्त किया जाता है –

**Syntax 1:**

DISPLAY

यह Command खुली हुई Data Base File के वर्तमान Record को Display करता है।

**Syntax 2:**

DISPLAY ALL

यह Command खुली हुई Data Base File के सभी Records को एक - एक करके Screen पर Display करता है।

**Syntax 3:**

DISPLAY ALL< fieldname1 >< fieldname2 >

यह command खुली हुई Data Base File के सभी Records को सिर्फ दिये गये Field का Data Display करती है।

**Example: -**

USE < filename >

DISPLAY ALL< fieldname1 >, < fieldname2 >, < fieldname3 >

**Syntax 4:**

DISPLAY Record< Number >

यह command खुली हुई Data Base File के केवल दिये गये Number वाले Record को Display करता है।

**Example: -**

USE < filename >

DISPLAY Record 7

**LIST Command:** - LIST Command के द्वारा भी खुली हुई Data Base File के records Screen पर Display किये जा सकते हैं।

**Syntax: - LIST**

यह खुली हुई Data Base File के सभी Records को Screen पर लगातार प्रदर्शित करता है –

(1) LIST OFF

(2) LIST TO PRINTER

(3) LIST FIELDS< fieldname1 >, < fieldname2 > ...

(4) LIST FOR< condition >

(5) LIST FOR< condition1 > AND < condition2 >

(6) LIST FOR< condition1 > TO PRINTER

(7) LIST FOR< condition1 > AND < condition2 > TO PRINTER

**Que: - Explain various features of FoxPro.**

**Ans: -** FoxPro को RDBMS (Relational Database Management System) भी कहते हैं। RDBMS एक ऐसी System है जिसमें कि कई Related Data Base को Check किया जा सकता है। FoxPro में निम्न कार्य सुविधापूर्वक किये जा सकते हैं –

- (1) FoxPro में Data Base File/Table सरलतापूर्वक तैयार की जा सकती है। उस Table का नाम बदला जा सकता है। Table के Structure में परिवर्तन किया जा सकता है तथा Table को स्थायी रूप से Delete किया जा सकता है।



- (2) Data Base Table में Records जोड़े जा सकते हैं, Store Records में से कुछ Delete किया जा सकता है, उन्हें किसी भी order में व्यवस्थित करके Screen पर display किया जा सकता है।
- (3) Data Base में Store Data के बारे में Query तैयार की जा सकती है तथा Query द्वारा चुने हुये Records एवं Field को Screen पर दिखाया जा सकता है।
- (4) Reports तैयार की जा सकती है।
- (5) Arithmetic (अंकगणित) एवं Scientific Calculation की जा सकती है।

FoxPro का Use करने के लिए इसे Computer में Install करना अनिवार्य है। FoxPro के आरम्भ होने पर सबसे पहले Catalog Manager Present होता है। Catalog Manager को बन्द करने पर FoxPro Software एक application Window और एक Command Window Present करता है। FoxPro में कार्य करने के लिए दो प्रकार के निर्देश दिये जा सकते हैं—

- (1) Command Window में Type करके अथवा  
 (2) Application Window में विभिन्न Menus के Options को चुनकर।

**Question:** - Write the difference between the following:

- (1) Display and Display All  
 (2) Browse and Edit  
 (3) Recall and Pack  
 (4) Replace and Change

**Answer:** - (1) Display and Display All

Display	Display All
<p>(i) यह Command खुली हुई Data Base के Current Records को Display करता है।</p> <p>(ii) इसका Syntax है – DISPLAY</p> <p>(iii) यह Command खुली हुई Data base के Present Record से केवल दिये गये Number वाले Record को Display करती है।</p> <p><b>Ex:</b> USE&lt;file name&gt;            DISPLAY NEXT 4            Display</p>	<p>(i) यह Command खुली हुई Data Base के सभी Record को एक-एक करके screen पर Display करता है।</p> <p>(ii) इसका Syntax है – DISPLAY ALL</p> <p>(iii) यह Command खुली हुई Data base के सभी Record के सिर्फ दिए गए Fields के Data को Display करती है।</p> <p><b>Ex:</b> USE&lt;file name&gt;            DISPLAY ALL&lt;field name 1&gt;, &lt;field name2&gt;,            &lt;field name 3&gt;</p>

(2) Browse and Edit

Browse	Edit
<p>(i) यह Command एक Window में खुली हुई Data Base File के सभी Records एक साथ देखने तथा उसमें संशोधन</p>	<p>(i) यह Command खुली हुई Data Base File के सभी Record में संशोधन करते हुए screen पर Display करता है।</p>

<p>करने की सुविधा प्रदान करती है।</p> <p><b>(ii)</b> इसका Syntax है – Browse Width 10</p> <p><b>(iii)</b> इस window को Browse Window कहते हैं।</p> <p><b>Ex:</b> USE EMPLOYEE BROWSE FIELDS EMP_CODE BASIC</p>	<p><b>(ii)</b> इस window को Browse Window कहते हैं।</p> <p><b>(iii)</b> इसका Syntax है – USE EMPLOYEE EDIT FIELDS EMP_CODE NAME</p>
--	---

### (3) Recall and Pack

Recall	Pack
<p><b>(i)</b> यह Command Delete किए हुए Records को वापस लाने के लिए उपयोगी होता है।</p> <p><b>(ii)</b> Delete करने के बाद Records को Recall किया जा सकता है।</p> <p><b>(iii)</b> इसका Syntax है – RECALL</p>	<p><b>(i)</b> यह Command Delete Command द्वारा Marked Records को स्थायी रूप से मिटा देती है।</p> <p><b>(ii)</b> Pack करने के बाद Records को Recall नहीं किया जा सकता है।</p> <p><b>(iii)</b> इसका Syntax है – PACK</p>

### (4) Replace and Change

Replace	Change
<p><b>(i)</b> Replace Command की सहायता से किसी खुली हुई Data Base File के एक या अनेक Fields में Data प्रतिस्थापित किया जा सकता है।</p> <p><b>(ii)</b> इस Command से प्रतिस्थापना File में अनेक अथवा सभी Records के लिए की जा सकती है।</p> <p><b>(iii)</b> इसका Syntax है – REPLACE ALL &lt;field name&gt; WIDTH</p>	<p><b>(i)</b> Change Command के द्वारा वर्तमान Records के वांछित Fields में बदलाव लाया जा सकता है।</p> <p><b>(ii)</b> यह Command Single Field में बदलाव करने का सबसे fast method है।</p> <p><b>(iii)</b> इसका Syntax है – CHANGE FIELD &lt;field name 1&gt;,&lt;field name 2&gt; For&lt;condition&gt;</p>

## UNIT - II

➤ **SORTING:** - Data Base File के Records को किसी Field के आधार पर Ascending या Descending Order में व्यवस्थित करने का Action Sorting कहलाता है। किसी Data Base File के Records को Sort करने का कार्य दो प्रकार से किया जा सकता है –

1. Menu की सहायता से अथवा
2. Command window में Command Type करके।

Sorting करने पर एक अन्य Data Base File Create होती है जिसका Extension .DBF होता है। इस Command द्वारा Records को Alphabetical, Numeric या Alphanumeric Order में Sort कर सकते हैं।

### Syntax:

SORT ON< fieldname> To< new fieldname> A/D/C

यहाँ /A का अर्थ है Ascending Order व /D का अर्थ है Descending Order तथा /C का अर्थ है Ignore Case. अगर कोई Option Chose नहीं किया है तो FoxPro By Default Ascending Order में Sorting करता है।

### Example:

```
USE EMPLOYEE
SORT TO EMP ON EMP_CODE
BROWSE
```

**I. Sorting With the Help of MENU:** - File को Sort करने से पहले उसे प्रयोग में लाना आवश्यक है, जैसे आपको employee file को sort करना है, तो पहले employee.DBF file को Open करेंगे इसके बाद Data Base menu के sort option को select करना होगा। जिससे एक dialog box open होता है। इस window में 6 part होते हैं।

1. **All Fields:** - इस Part में Data Base File के समस्त Field Display होते हैं। जिस भी Field पर Sorting करनी है, उस Field को Mouse द्वारा select कर Mouse Button पर Click करने पर इस Field का नाम Sort Order में चला जाता है, जो यह दर्शाता है कि इस Field पर Sorting की गई है।
2. **Field Options:** - इस part में एक check Box तथा दो Radio Button Ascending और Descending होते हैं।
3. **Sort Order:** - All Field से Move की गई Field Sort Order List में दिखाई देने लगती है। Remove Button का प्रयोग करके इस Part से Field का नाम पुनः All Field में Normal Field की तरह लाया जा सकता है।
4. **From Table:** - इस Field में प्रयोग में लाई जा रही Table का नाम Display होता है। इस Part का Use एक से अधिक Data Base को Sort करने के लिए किया जाता है।
5. **Input:** - इस Part में तीन Check Box होते हैं जो क्रमशः SCOPE, FOR and WHILE होते हैं। इनका प्रयोग किसी Condition के आधार पर Field को Short करने के लिए किया जाता है।
6. **Output:** - इस Part में जिस नाम से Sort File को Save करना है, कर दिया जाता है। अन्त में OK Button पर Click करके Sorting की जाती है।

**II. Sorting With the Help Of COMMAND: -**

Sort Command का Use करके भी Sorting की जा सकती है।

**Syntax:**

`SORT TO <New File> ON <Field/A/D/C>, <Field....>`

A = Ascending, D = Descending, C = Ignore Case

**अनेक Field पर Sorting:** - अक्सर एक से ज्यादा Field पर Sort करने की आवश्यकता पड़ती है। For Example, किसी Organization में 4000 से 5000 तक Employee है तथा ये Employee अलग-अलग Department में कार्य करते हैं, जो Disk पर अधिक जगह घेरती है। प्रत्येक Department के नाम के अनुसार Employees को Order में व्यवस्थित करने के लिए आपको एक से ज्यादा Field पर Sorting करनी पड़ेगी। इस Case में Deptt\_Code, Name दोनों पर Sorting होगी।

```
USE EMPLOYEE
SORT TODSSHORT ON DEPT_CODE, NAME
BROWSE
```

➤ **Advantages of Sorting:** -

1. Merit list तैयार करने तथा क्रमानुसार Records व्यवस्थित करने में उपयोगी रहते हैं।
2. SORT Command से एक New Data Base File तैयार हो जाती है।

➤ **Disadvantages of Sorting:** -

1. SORT Command के द्वारा किसी बड़ी Data Base File को Sort करने में अधिक समय लगता है।
2. Output File .DBF Main File से अलग बनती है जो Disk पर अधिक जगह घेरती है।
3. यदि Original File में कोई परिवर्तन होता है तो उसे दोबारा से Sort करना पड़ता है। अतः Data Base को Original करने के लिए SORT Command का प्रयोग नहीं करना चाहिए। इसके स्थान पर INDEX Command का प्रयोग करना चाहिये।
4. Sorting एक Slow Process है।

➤ **INDEXING:** - Index द्वारा Data Base File को किसी Order में व्यवस्थित किया जाता है। Data Base File की Indexing करने पर एक New Index File बनती है तथा Index में Data Base File के Record क्रमबद्ध रूप में store होते हैं। Original File के Records में कोई परिवर्तन नहीं होता है।

Index, Book के Index के तरह ही होता है जिससे यह पता चलता है कि कौन सा Topic कौन से Page पर है। Indexing करने पर Record संख्या में परिवर्तन केवल Index File में ही होता है। Index File का Extension .CDX अथवा .IDX होता है। .IDX Index File में एक Field पर Index होती है जबकि .CDX एक से अधिक Fields पर Index की जा सकती है। ये Index File स्वतः .DBF File के Use होने पर Operate हो जाती है। Data Base File में Data को जोड़ने या उनमें किसी प्रकार के परिवर्तन होने की दिशा में REINDEX Command का Use करके File को दोबारा से Index किया जा सकता है।

INDEX Command का Syntax निम्न है – USE <table name>

INDEX ON<expresser> TAG <tag name>

**Example:** - USE EMPLOYEE

INDEX ON EMP\_CODE TAG E1

Record No.	Name		Record No.	Name
1.	Prakash	→	1.	Prakash
2.	Abhilash	→	3.	Arti
3.	Arti	→	6.	Hemant
4.	Aks	→	2.	Abhilash
5.	Vikash	→	4.	Aks

6.	Hemant
----	--------

5.	Vikash
----	--------

➤ Type of Indexing: -

Data Base File में Indexing दो प्रकार से किया जा सकता है –

1. Single Index द्वारा।
2. Compound Index द्वारा।

1. Single Index File: - Single Index File का Extension .IDX होता है। Index को Update रखने के लिए Index File को Open करना होता है। यह भी दो प्रकार का होता है –

- (i) Compact
- (ii) Non – Compact

Compact Index File, Disk में कम जगह घेरती है तथा प्रभावशाली होती है। इसको Access करने में कम समय लगता है।

Compact Index File बनाने का **Syntax** है –

```
USE< table name >
INDEX ON < expr >TO<. IDX File >
COMPACT
```

Non – Compact Index File बनाने का **Syntax** है –

```
USE< table name >
INDEX ON < expr >TO<. IDX File >
```

2. Compound Index File: - Compound Index File का Extension .CDX होता है। Compound Index File एक से ज्यादा Index एक ही File में रख सकते हैं। प्रत्येक Index Compound Index File में Index Tag कहलाती है। इस प्रकार की File Compact Format में होती है तथा कम Space लेती है और अधिक सफलतापूर्वक कार्य को पूर्ण करती है। इसका **Syntax** निम्न है –

```
USE< table name >
INDEX ON < expr >TAG< tagname >
OF<. CDX File >
```

Compound Index File भी दो प्रकार की होती है –

- (i) **Structural**
- (ii) **Independent**

(i) Structural Compound Index file: - File का नाम Data Base का ही होता है, जबकि Extension.CDX होता है। Structural Compound Index File स्वतः ही Data Base File के Use होने पर Perform हो जाती है।

Structural Compound Index File का **Syntax** निम्न होता है –

```
USE< table name >
INDEX ON < expr >TAG< tagname >
```

(ii) Independent Compound Index file: - File का नाम Data Base File के नाम के अतिरिक्त दूसरा भी हो सकता है, परन्तु इसका भी Extension.CDX होता है। Independent Compound Index File, Data Base File के Use होने पर स्वतः Perform नहीं होती है। इसको Open करना पड़ता है। अर्थात् यदि हम Indexing के बाद और Record Add करने हैं तो हमें Independent Compound Index File को REINDEX Command द्वारा Update करना होगा। इसका **Syntax** निम्न होता है –

```
USE< table name >
INDEX ON < expr >TAG< tagname >
```

OF<. CDX File >

**Example: -**

USE ADDRESS  
INDEX ON NAME TAG NN OF IND

यदि आप Independent Compound Index File को Online Update करना चाहते हैं तो आपको इसको Data Base File के साथ ही Open कर रखना होगा।

USE ADDRESS INDEX IND  
OR  
USE ADDRESS  
SET INDEX TO IND

➤ **Difference Between Sorting and Indexing: -**

S. No.	<b>SORTING</b>	S. No.	<b>INDEXING</b>
1.	Sorting ज्यादा समय लेती है।	1.	Indexing कम समय लेती है।
2.	File का Extension .DBF होता है और ज्यादा जगह लेती है।	2.	File का Extension .IBX या .CDX होता है और कम जगह लेती है।
3.	यह एक .DBF File होती है, अतः Open करनी पड़ती है।	3.	INDEX File Data Base Open करने पर स्वतः Open हो जाती है।
4.	इसकी File को बार - बार Sort करना पड़ता है।	4.	इसकी File को बार-बार Index नहीं करना होता है।
5.	इसमें Data Base का structure Copy कर सकते हैं।	5.	इसमें Data Base का structure Copy नहीं कर सकते हैं।
6.	इसमें Fields को कमबद्ध कर सकते हैं।	6.	इसमें Fields को कमबद्ध नहीं कर सकते हैं।
7.	Record Number में कोई बदलाव नहीं होता है।	7.	Record Number भी Change होते हैं।

➤ **Explain the Following:** -

- FIND Command
- SEEK Command
- Printing a Report
- Display Directory
- Single Key Sorting
- Multiple Key Sorting
- Rushmore Technology

(1) **FIND Command:** - Find Command का Use सिर्फ Character String के लिए किया जा सकता है। Character String को Quotes के बीच में नहीं लिखा जाता है।

इसका **Syntax** है:

FIND <character string>

**Example:**

USE EMPLOYEE INDEX CNAME  
FIND Prakash  
DISPLAY

उपरोक्त Command Prakash Name को EMPLOYEE Data Base में Search करेगी तथा DISPLAY Command के द्वारा उसका Result Screen पर प्रदर्शित होता है।

(2) **SEEK Command:** - SEEK Command का प्रयोग Character and Numeric दोनों Sting के लिए किया जाता है।

USE EMPLOYEE INDEX CNAME  
STORE "Prakash Dwivedi" TO mm

SEEK mm

(3) **Printing a Report:** - Report present करने या Print करने के लिए निम्नलिखित command का Use करते है -  
REPORT FROM <report file name> TO PRINT

**Example: -**

```
USE EMPLOYEE
REPORT FROM EMPLOYEE TO PRINT
```

(4) **Single Key Sorting:** - Single Key Sorting में Data Base File को किसी एक Field पर Sort किया जाता है।

**Example: -**

```
USE EMPLOYEE
SORT TO EMP ON EMP_CODE
```

उपरोक्त Command Data Base को EMP\_CODE field पर Sort करेगी और एक अन्य Data Base File EMP में Copy कर देगी। इस Command का Result देखने के लिए BROWSE, DISPLAY, LIST में किसी एक Command का प्रयोग करना होगा।

(5) **Multiple Key Sorting:** - दो Data Base को Multiple Field पर Sort करना ही Multiple Key Sorting कहलाता है।

**Example: -**

```
USE EMPLOYEE
SORT TO DSORT ON DEMP_CODE, NAME
BROWSE
```

उपरोक्त Command EMPLOYEE Data base को DEPT\_CODE तथा NAME दोनों field पर Sort करेगी तथा Sort हुये Data को अन्य Data Base File DSORT में Store कर देगी।

(6) **Rushmore Technology:** - Rushmore Technology एक Data Access Technology है जिससे Record के समूह पर अधिक तीव्रता से Searching process perform होती है। Rushmore Technology की Help से जटिल Data Bases की प्रक्रिया में हजारों गुना तीव्र गति प्रक्रिया उत्पन्न कर ली जाती है। यह Technology FOR Clause को Utilize करके खुली Index के माध्यम से प्रक्रिया गति में वृद्धि कर देती है। Rushmore Technology से Information की Searching Fast and Easy हो जाती है। निम्नलिखित Command जो FOR Clause के साथ प्रयुक्त होते है, Rushmore Technology को Fast कर देती है –

**BROWSE, CHANGE, EDIT DELETE, LIST, LOCATE, SORT, INDEX, REPORT, LABSUM, REPLACE, RECALL, COPY** आदि।

(7) **Display Directory:** - FoxPro में जब हम किसी Command को लिखते हैं तो इसके लिये FoxPro में Command Window में सभी Commands लिखे जाते हैं। इसी प्रकार जब हम किसी Data Base या Program File को Save करते हैं तो वह File by default FoxPro की Directory में Save होती है। यदि हम FoxPro में उपलब्ध सभी Directory की List देखना चाहते हैं, इसके लिए Data Base Prompt पर dir command का प्रयोग करते हैं। dir command के द्वारा हम Directory को Display करते हैं। FoxPro में Directory को Display करने के लिए dir command को निम्नलिखित रूप में लिखते हैं –

Dir ↓

इसके बाद Current directory के अंतर्गत सभी Sub directory and files display होती है। इन्हे देखने के लिए हम F4 Function Key का Use कर सकते हैं।

❖ FoxPro में Report तैयार करने के Steps समझाइए।

FoxPro में किये गये कार्य का last important result REPORT होता है। Data Base के Record को व्यवस्थित एवं निर्धारित रूप से प्रस्तुत करना Report कहलाता है। Report को हम Screen पर विभिन्न Feature के साथ अच्छे Format में देख सकते हैं। Report File का Extension .FRS होता है। किसी Data Base File में Report तैयार करने के लिए File Menu के New Option पर Click करते हैं। Display Dialog Box में से Report Radio Button select कर New Button पर Click करते हैं, जिससे Report Written Window Display होती है। इस Menu में Reporting से Related विभिन्न Option Present होते हैं। FoxPro में Reporting करने के दो Method हैं –

- 1) Quick Report
- 2) Customs Report

1) **Quick Report:** - Quick Report, Report Menu के Quick Report Option को select करके तैयार की जाती है। Generally Quick Report में Data Base File के सभी Records होते हैं, परन्तु इच्छानुसार Selected Field के अनुसार भी reporting की जा सकती है। किसी भी Data Base File पर Report तैयार करने से पहले उसको प्रयोग में लाना होता है। Report Menu के Quick Report Option को चुनते ही Dialog Box Display होता है। इस Dialog Box में OK Button पर Click करते ही Data Base File के सभी Record Report Writer Window में Display होने लगता है। इस Window के Tool Box का प्रयोग कर Report को Customize भी किया जा सकता है। इस प्रकार इच्छानुसार Layout चुनकर Quick Report तैयार की जा सकती है। इस Dialog Box के Field Layout वाले भाग में दो तरह के Layout होते हैं –

1. Column Oriented Layout

2. Form Oriented Layout

इसके नीचे बने Check Box Titles को चुनने पर हमारी Report में Field Name Titles के रूप में Display होते हैं। Field Check Titles पर Click करके Field Picker Dialog Box द्वारा हम जिन-जिन Field को Report में Show करना चाहते हैं चुन लेते हैं तथा OK Button पर Click करने पर Report तैयार की जा सकती है।

2. **Custom Report:** - Custom Report में हम Condition के साथ Reporting कर सकते हैं। File Menu के New Option पर Click करके Radio Button को Select करने पर Report Writer Display होता है तथा Report Menu, Menu Bar से जुड़ जाता है। इस Features के द्वारा हम Footer में आवश्यक बदलाव, Body को Format and Heading को Design करना आदि कार्य सम्पन्न कर सकते हैं। Report writer Window को निम्न Command के द्वारा भी Open किया जा सकता है।

```
USE EMPLOYEE
CREATE REPORT FILE1
```

Report Writer Window में तीन Layer होती है, जिन्हें कि Report Band कहा जाता है। प्रत्येक Report में कम से कम तीन Band अवश्य होते हैं, ये तीन Band निम्नलिखित हैं –

1. Page Header Band
2. Detail Band
3. Page Footer Band

इसके अलावा Optional Title एवं Optional Group Band भी हो सकते हैं। Page में Header and Footer का Use कर सकते हैं तथा अन्य कार्य भी कर सकते हैं।

Page Header Band Report के प्रत्येक Page पर ऊपर Display होता है। Page Header में हम Report की Heading, Report की Date तथा यह किस समय तैयार की गयी है, यह जानकारी दे सकते हैं। Detail Band में report से Related Record होते हैं। Page Footer Band Report के प्रत्येक Page पर नीचे Display होती है। Page Footer में हम जानकारियां दे सकते हैं जो page Header Band में देना उपर्युक्त नहीं है। Report को आवश्यकतानुसार तैयार करने के बाद उसको Save करना जरूरी है, ताकि हम भविष्य में उसका प्रयोग कर सकें। Report को Run कराने के लिए हमें Report Menu में RUN Option को चुनना होगा अथवा Command Window में निम्न Command देना होगा –

REPORT FORM FILE 1



➤ **REPORT में GROUP अथवा SUBGROUP ADD करना:-**

Report Menu के Option Data Grouping के द्वारा आप Data की Grouping कर सकते हैं। Example : माना ऐसे Record की आवश्यकता है जिसमें सभी व्यक्तियों की City Umaria है। अतः Umaria City में रहने वाले व्यक्तियों की Grouping करना अथवा सभी व्यक्तियों जिनका Department Code 20 है, उन व्यक्तियों की Grouping करनी है।

Report Menu के Data Grouping Options को चुनने के बाद Data Grouping Dialog Box Display होता है। आप Add, Change, Delete Button की मदद से Report में Groups Add, Change, and Delete कर सकते हैं।

Add Button Press करके Group Info नामक Dialog Box में Group Push Button के सामने बने Text Box में उस Field का नाम Enter करते हैं, जिस पर Grouping करनी है। Group Push Button को Click करके Expression Builder में Valid FoxPro Expression के आधार पर भी Records को Group किया जा सकता है।

➤ **FoxPro में Mailing Address Label तैयार करने के Steps को समझाइए।**

FoxPro में Mailing Address Label बनाने की सुविधा भी होती है। यह सुविधा तब तक उपयोगी होती है जब इससे अधिक मात्रा में Print करवाने होते हैं। यह किसी भी Data Base File को पहले से Sort या Index किया गया है तो Labels उसी Order में Print होंगे जिस Order में Data Base को Sort या Index किया है।

Mailing Address Labels बनाने की प्रक्रिया बिल्कुल Report बनाने की तरह ही है। यह प्रक्रिया निम्न प्रकार है –

1. उस Data Base File को Open करना जिसमें नाम या पते आदि दिये गये हैं।
2. Label Designee Window को Open करना।
3. Data Base File की Field को select करना।
4. Margin आदि को Select करना।
5. Label बनी हुई File को Store करना।

Address Label बनाने के लिये Address Data Base File Create करते हैं, जिसमें निम्नलिखित Field हो सकते हैं –

1. Name
2. City
3. State
4. Pin Code

Command Window द्वारा Label Window Open करने के निम्नलिखित Command दी जाती है –

**Syntax:**

Create Label <Label File Name>

**Example:**

Create Label Mailing

Label को Screen पर या Printer द्वारा Print करने के लिये Data Base File को खोलते हैं।

**Example: -**

Use Address

इसके पश्चात् Label File को Perform करने के लिए Command देते हैं –

Label from Mailing

इस Command को देने के पश्चात् सभी Address Field एक-एक करके Screen पर Display होते हैं –

**Guided by: Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)**

यदि Label का Printout लेना है तो निम्नलिखित Command देते हैं –  
Label from Mailing to Printer

## UNIT - III

### ➤ Explain the various Functions in FoxPro.

Fox Pro में विभिन्न प्रकार के Function का Use किया जाता है –

1 - Date and Time Function: - FoxPro में Date and Time से Related अनेक Functions हैं –

- |               |             |                      |
|---------------|-------------|----------------------|
| • TIME ( )    | • DATE ( )  | • SECOND ( )         |
| • DAY ( )     | • MONTH ( ) | • CMONTH ( )         |
| • YEAR ( )    | • DOW ( )   | • CDOW ( )           |
| • DTOC ( )    | • CTOD ( )  | • SET CENTURY ON/OFF |
| • SET MARK TO | • DMY ( )   |                      |

(1) TIME ( ): - Time ( ) function Operating System में Load System Time को screen पर दिखाता है। अतः FoxPro में Time Function का प्रयोग करके हम सही समय ज्ञात कर सकते हैं। Time ( ) का प्रयोग निम्न प्रकार से कर सकते हैं –

```
? TIME ( )
17: 40: 00
```

FoxPro Time को 24 Hours Format में Display करता है। FoxPro screen पर यदि Time 12 Hours Format में Display करना है तो उसे SET HOUR TO 12 तथा यदि 24Hour Format में Display करना है तो उसे SET HOUR TO 24 में Set करते हैं।

(2) DATE ( ): - Date Function Operating System की System Date को Screen पर Display करता है।

```
? DATE ( )
04/03/2012
```

उपरोक्त Function Date को American Format {mm/dd/yyyy} में Screen पर दिखाता है।

(3) SECOND ( ): - यह Function Time के Seconds की गणना को Screen पर Display करता है।

```
? SECOND ( )
80446.070
```

(4) DAY ( ): - FoxPro में यह Function Date Expression के Day के Numerical Value को screen पर दर्शाने के लिए प्रयोग किया जाता है।

```
STORE DATE ( ) TO CD
```

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

```
07/02/2005
? DAY (CD)
07
```

**(5) MONTH()** :- यह Function किसी Date Expression के Month के Numerical Value को Screen पर Display करता है।

```
STORE MONTH ( ) TO CD
07/02/2005
? MONTH (CD)
07
```

**(6) CMONTH()** :- यह Function किसी भी Date Expression के Month के नाम को Character Expression को Screen पर दिखाता है।

```
? CMONTH ( )
JULY
```

**(7) YEAR()** :- यह Function किसी Date Expression के Year के Numerical Value को Screen पर Display करता है।

```
? YEAR (CD)
2005
```

**(8) DOW()** :- इसका Full Form Day Of Week है। यह Date Expression के Day के Numerical Value को Screen पर Display करता है।

```
? DOW ( )
5
```

**(9) CDOW()** :- इसका Full Form Character Day Of Week है। यह Function किसी Date Expression के Day के Character Value को Screen पर Display करता है।

```
? DOW ( )
Thursday
```

**(10) DTOC()** :- यह एक Important Function है जो कि Date Type Data को Character Type Data में Change करता है।

```
STORE DTOC (CD) TO TODAY
06/11/87
C_ DATE = DTOC (DATE ( ) )
06/11/87
```

**(11) CTOD()** :- CTOD Function का तात्पर्य Character To Date से है। यह DTOC के विपरीत Function है। CTOD Function Character Type Data को Date Type Data में Change करता है।

```
C_ CTOD = "06/11/87"
STORE CTOD (C_ DATE) TO DATE_ VAR
```

**(12) SET CENTURY ON/OFF** :- FoxPro में इस Command के प्रयोग द्वारा यह निर्धारित किया जाता है कि Screen पर Date Display करते समय Century को दिखाना है या नहीं? By Default Century OFF होती है। यदि आप FoxPro को 4 Numerical Format में दिखाना चाहते हैं तो आप Set Century Command का प्रयोग कर सकते हैं।

```
? DATE ( )
07/02/05
SET CENTURY ON
```

? DATE ( )  
07/02/2005  
SET CENTURY OFF  
? DATE ( )  
07/02/05

**(13) SET MARK TO:** - Set Mark To Command निर्धारित करती है कि Date Format को किस प्रकार Screen पर दिखाना है। Set Mark To Command द्वारा इसे बदला जा सकता है।

? DATE ( )  
07/02/05  
SET MARK TO "-"  
? DATE ( )  
07-02-05

**(14) DMY ( ):** - DMY ( ) Function Day-Month-Year Format में Character Expression को Return करता है। इसका Syntax होता है –

DMY (d Expression/t Expression)

d Expression, Date Expression को तथा t Expression, Time Expression को Specify करता है।

**2 - Mathematical/Statistical Function:** - FoxPro में Numerical Mathematical Command की भी सुविधा होती है, जिनके द्वारा Addition, Subtraction, Multiplication, and division के साथ-साथ दिये गये Formula के अनुसार गणना की जा सकती है।

**Example:**  $5 + 2 = 7$

इसके अलावा कुछ अन्य Functions भी होते हैं, जिन्हें प्रयोग करने पर किसी भी एक प्रकार की गणना की जा सकती है। ये निम्नलिखित हैं –

- SQRT ( )
- ABS ( )
- MOD ( )
- CNT ( )
- BETWEEN ( )
- ROUND ( )
- LOG ( )
- MIN ( )
- SUM ( )
- CALCULATE ( )
- FLOOR ( )
- STD ( )
- INT ( )
- MAX ( )
- AVG ( )
- SIGN ( )
- CEILING ( )
- NPV ( )

**1) SQRT ( ):** - SQRT Function के द्वारा भी Positive Number का Square Root Find किया जा सकता है।

Ex: - ? SQRT (49)

7

**2) LOG ( ):** - Log Function के द्वारा भी Positive Number का Logarithm Find किया जा सकता है।

Ex: - ? LOG (100)

4.61

**3) INT ( ):** - किसी भी Numeric Value का Integer Part Find करने के लिए INT Function का Use करते हैं। यह दशमलव संख्या हटा देता है।

Ex: - ? INT (1263.789)

1263

**4) ABS ( ):** - इस Function के द्वारा किसी भी Negative Value को Positive Value में Change किया जा सकता है। अगर दी हुई Value Positive है तो वह Positive ही रहती है।

Ex: - ? ABS (- 125.25)

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

125.25

**(5) MIN ( ):** - इस Function के द्वारा विभिन्न संख्याओं में जो सबसे छोटी होती है, उसे ज्ञात किया जाता है।

Ex: - ? MIN (10, 5, 25, 2. 9)

2

? MIN ("BA", "CA", "ZD", "AZ")

"AZ"

**(6) MAX ( ):** - इस Function के द्वारा विभिन्न संख्याओं में जो सबसे बड़ी होती है, उसे ज्ञात किया जाता है।

Ex: - ? MIN (10, 5, 25, 2. 9)

25

? MIN ("BA", "CA", "ZD", "AZ")

"ZD"

**(7) MOD ( ):** - इस Function के द्वारा Reminder को ज्ञात किया जाता है।

Ex: - ? MOD (965, 100)

65

**(8) SUM ( ):** - इस Function के द्वारा Data Base File की सभी Numeric Field का Sum Find किया जाता है।

Ex: - USE SALARY

SUM

10 RECORDS SUMMED

SALARY

987650.00

**(9) AVG ( ):** - Data Base File में दिये गये Numeric Field की Values का Average Find किया जाता है।

Ex: - USE SALARY

AVERAGE

RECORDS AVERAGED

**(10) SIGN ( ):** - Sign Function 1, 0, -1 Numeric Value देता है यदि Numeric Expression Positive है तो इसका मान 1 आता है। यदि Numeric Expression Negative है तो इसका मान -1 आता है। यदि Numeric Expression Zero (0) आता है तो इसका मान 0 होता है।

Ex: - STORE ( ) TO NUM

? SIGN (NUM)

0

? SIGN (20.34)

1

? Sign (-3)

-1

**(11) Between ( ):** - यह Function निर्धारित करता है कि दिये गये Expression का मान अन्य दूसरे दो Expression की Value के बीच आता है या नहीं?

Ex: - ? BETWEEN (X, Y, Z)

.T.

Between Check करता है कि Data Variable X, Data Variable Y, Data Variable Z के बीच आता है या नहीं? यहाँ X (Current Date), Y (Tomorrow Date) and Z (Yesterday Date) है। अतः X, Y and Z के बीच में आता है इसलिए यह Command .T. Value दिखाती है।

**(12) CNT ( )** :- इस Function का प्रयोग Data Base File में दिये गये Numeric Field को Count करने के लिए किया जाता है।

**Ex:** - USE EMP  
COUNT ( )

**(13) CALCULATE ( )** :- इस Function का प्रयोग Table में Column की Numeric value को ज्ञात करने के लिए करते हैं।

**(14) FLOOR ( )** :- यह Function दिये गये Numeric Value argument से छोटे या बराबर सबसे Nearest Integer Value की गणना करता है।

**Syntax:** - FLOOR (Numeric Value)

**Ex:** - ? FLOOR (19.45)  
19

**(15) CEILING ( )** :- यह Function दिये गये Numeric Value या Number से बड़े या बराबर सबसे Nearest Integer Value की गणना करता है। यदि Value Negative है तो यह दशमलव से पहले वाला भाग Negative Sign के साथ ही लौटता है।

**Ex:** - CEILING (-19.45)  
-19

? CEILING (111.11) का Result 112 प्रदर्शित होगा।

**(16) ROUND ( )** :- यह Function Natural Exponent (चरघातांकी) Operator है। यह किसी संख्या के Exponent Value की गणना करता है।

**Syntax:** - ? ROUND (संख्या, दशमलव अंकों की संख्या)

**Ex:** - SET DECIMALS TO 4

? ROUND (1234.1962, 3)

**RESULT:** 1234.196

? ROUND (1234.1962, 2)

**RESULT:** 1234.20

? ROUND (1234.1962, 0)

**RESULT:** 1234

? ROUND (1234.1962, -1)

**RESULT:** 1230

**(17) STD ( )** :- इस Function का प्रयोग Standard Division को Calculate करने के लिए किया जाता है। FoxPro Programming में जब हम Mathematical Expression में Standard Division Value को Calculate करते हैं तब इस Function का प्रयोग करते हैं।

**(18) NPV ( )** :- इस Function का प्रयोग Net Present Value को Find करने के लिये किया जाता है। FoxPro में Interest को Calculate करने के लिए Net Present Value को इस Function के द्वारा ज्ञात किया जाता है।

**3- String Function:** - FoxPro में विभिन्न प्रकार के String Function Support करता है।

ASC Function के प्रयोग द्वारा Character String के First Character की ACS II Value find की जा सकती है।

**Syntax:** - ASC (<Character String>)

**Ex:** - ? ASC ("A")

65

- LEN ( )
- LEFT ( )
- RIGHT ( )
- UPPER ( )
- LOWER ( )
- CHR ( )

- LTRIM ( )
- RTRIM ( )
- ISUPPER ( )
- ISLOWER ( )
- ISDIGIT ( )
- PROPER ( )

(1) **LEN ( )**: - इस Function के प्रयोग द्वारा Character String की Length Find की जा सकती है।

**Ex:** - ? LEN ("Dwivedi Prakashan")

(2) **LEFT ( )**: - इस Function के प्रयोग द्वारा Character Expression में से Left Side से Numeric Expression में दी गई संख्या तक के Character, Screen पर Display किये जा सकते हैं।

**Syntax:** LEFT (<Character Expression>, < Numeric Expression>)

**Ex:** - ? LEFT ("Dwivedi Prakashan"8)

Dwivedi P

(3) **RIGHT ( )**: - इस Function के प्रयोग द्वारा Character Expression में से Right Side से Numeric Expression में दी गई संख्या तक के Character, Screen पर Display किये जा सकते हैं।

**Syntax:** RIGHT (<Character Expression>, < Numeric Expression>)

**Ex:** - ? RIGHT ("Dwivedi Prakashan"8)

(4) **UPPER ( )**: - इस Function के प्रयोग द्वारा किसी भी String को Capitalized किया जा सकता है।

**Syntax:** UPPER (<Character Expression>)

**Ex:** - ? UPPER ("Dwivedi")

DWIVEDI

(5) **LOWER ( )**: - इस Function के प्रयोग द्वारा किसी भी String को Lower Case या Small Letter में Change किया जा सकता है।

**Syntax:** LOWER (<Character Expression>)

**Ex:** - ? LOWER ("Dwivedi")

(6) **CHR ( )**: - इस Function के प्रयोग द्वारा ACS II Code से संबंधित Character Value Find की जा सकती है। CHR ASC ( ) का Opposite है।

**Syntax:** - CHR (<ASC II Code>)

**Ex:** - ? CHR (65)

A

(7) **LTRIM ( )**: - इस Function के प्रयोग Character Expression के Left Side में उपस्थित Space को हटाने के लिए किया जाता है।

**Syntax:** - LTRIM (<Character Expression>)

**Ex:** - Store " Umaria" To Town

? Town

Umaria

? LTRIM (Town)

Umaria

(8) **RTRIM ( )**: - इस Function के प्रयोग Character Expression के Right Side में उपस्थित Space को हटाने के लिए किया जाता है।

**Syntax:** - RTRIM (<Character Expression>)

**Ex:** - Store "Umaria " To Town

? Town

Umaria

? RTRIM (Town)

Umaria

(9) **ISUPPER ( )**: - यदि Specified Character Expression का First Character Upper Case में है तो ISUPPER (ISUPPER ?) Function का Result True (T) या False (F) आता है।

**Ex:** - STORE "Dwivedi Prakashan" TO PUB  
? ISUPPER (PUB)  
.T.  
? ISUPPER ("Dwivedi")  
.F.

(10) **ISLOWER ( )**: - यदि Specified Character Expression का First Character Lower Case में है तो ISLOWER Function का Result True (T) या False (F) आता है।

**Ex:** - ? ISLOWER (PUB)  
.F.  
? ISLOWER ("Dwivedi")  
.T.

(11) **ISDIGIT ( )**: - यदि Specified Character Expression का First Digit (0 – 9) है तो ISDIGIT Function का Result True (T) आता है।

**Ex:** - ? ISDIGIT ("123", UMARIA)  
.T.

(12) **PROPER ( )**: - इस Function के प्रयोग द्वारा Character String के प्रत्येक शब्द के First Digit को Capital Letter में प्राप्त किया जा सकता है।

**Syntax:** PROPER (<Character Expression>)

**Ex:** - ? PROPER ("Dwivedi Prakashan")

उपरोक्त Command का Output Dwivedi Prakashan आयेगा।

4 - **File Manipulation Function**: - FoxPro में विभिन्न प्रकार के File Manipulation Function को Support करता है।

- DBF ( )
- BOF ( )
- Field ( )
- EOF ( )
- File ( )

(1) **DBF ( )**: - DBF Function के द्वारा प्रयोग की जा रही Data Base File का नाम Display होता है।  
C:\fox\Address .DBF

(2) **Field ( )**: - इस Function के प्रयोग के द्वारा Use की जा रही DBF की Field का नाम ज्ञात किया जा सकता है।  
**Ex:** - ? Field (1)

उपरोक्त Command Use की जा रही Data Base File के First Field का नाम Screen पर Display करेगा।  
? FIELD (2, 3)

उपरोक्त Command Third Work Area में खुली हुई Data Base File के दूसरे Field का नाम Screen पर Display करेगा।

(3) **File ( )**: - इस Function के द्वारा यह ज्ञात करने के लिए किया जा सकता है कि Necessary File Disc में है अथवा नहीं? यदि File Disc में है तो .T. अथवा .F. Result प्राप्त होगा।

(4) **BOF ( )**: - {Beginning Of File} DBF में इस Function की Value तभी सत्य होती है जब Record Pointer DBF के प्रारम्भ में हों।

GO TO RECORD <FIRST RECORDNO>  
SKIP -1

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*



**(5) EOF ( ):** - {End Of File} यह Function BOF Function का Opposite है जो यह दर्शाता है कि Record Pointer DBF के Record के अंत में पहुँच गया है।

GO TO RECORD <LAST RECORD NO>

### ❖ Memory Variable: -

FoxPro Version के प्रयोग द्वारा हम Computer की Memory RAM में अस्थायी रूप से Data Store कर सकते हैं। Data किसी भी प्रकार का हो सकता है। हमारे Computer में 256 विभिन्न Data Items को एकत्र किया जा सकता है, जैसे – Bill No, Date, Result etc. RAM में एकत्र होने वाले Data Items के नाम को Memory Variable कहते हैं। Memory Variable के नामकरण के लिए आवश्यक Rule निम्नलिखित है –

1. Memory Variable का नाम Maximum 10 Character का हो सकता है।
2. Memory Variable का नाम किसी character (A – Z) से Start होना चाहिए।
3. (0 – 9) तक के अंक तथा Underscore ( \_ ) के अलावा हम किसी Special Character का प्रयोग नहीं कर सकते हैं।

**Example:** Space, Punctuation Mark etc.

हालांकि हम Memory Variable का नाम Underscore ( \_ ) से शुरू कर सकते हैं। ऐसे Variable System Memory Variable कहलाते हैं।

### ❖ Type of Memory Variable: -

FoxPro में चार प्रकार के Memory Variable होते हैं –

- 1) **Date Variable:** - Date Variable के अन्दर हम दिनांक को Store करते हैं। Date Variable में दिनांक को Date Format में ही Store किया जाता है। Date Format में 6 digit होते हैं, पहले दो अंक माह, दूसरे दो अंक दिनांक तथा अन्तिम दो अंक वर्ष के होते हैं। इसका Format MM/DD/YYYY में होना अति आवश्यक है।
- 2) **Numeric Variable:** - इस Variable के द्वारा हम कोई भी अंक दशमलव सहित Store कर सकते हैं। इस Variable में Store किये गये मान के साथ आंकिक गणनायें; जैसे – गुणा, भाग, आदि की जाती हैं।
- 3) **Logical Variable:** - Logical Variable में केवल एक ही Value को Store किया जा सकता है। यह Value Yes (Y) or No (N) अथवा True (T) or False (F) के रूप में Store हो जाती है।
- 4) **Character Variable:** - इस Variable में किसी भी Character की string को store किया जा सकता है, जो Number या Letter भी हो सकते हैं। Character Variable में कोई अंक भी store किया जा सकता है, परन्तु इस Store Digit द्वारा Analysis नहीं की जा सकती है।

### ❖ Creating Memory Variable: - Memory Variable Create करने के दो तरीके हैं –

#### i) STORE COMMAND: -

इस Command के द्वारा Special Variable का प्रयोग किया जाता है। तथा उसे एक Special Value दी जाती है—

**Example:** - STORE 20 TO NUM

यह Command NUM नाम से Memory में एक Numeric Memory Variable का निर्माण करेगी तथा इसकी Value 20 इसमें Store करेगी। यदि आप Character Type Memory Variable का निर्माण करना चाहते हैं तो Data Value को Single Quotation या Double Quotation या Square Bracket के अन्दर लिखते हैं।

**Example:** - Name नामक Variable में एक Value Unit निम्न प्रकार से Store हो सकती है –

STORE "PRAKASH DWIVEDI" TO NAME

STORE "PRAKASH DWIVEDI" TO NAME

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

STORE [PRAKASH DWIVEDI] TO NAME

STORE .T.TO CURRENT

STORE {06/11/1987} TO DOB

यह Command CURRENT नामक Logical Type Memory Variable का निर्माण करेगी तथा .T. मान इसमें Store हो जाता है। Logical Type Memory Variable का निर्माण करते समय Data के मान को DOT में बन्द कर देते हैं।

STORE .F. TO RAIN

यह Command RAIN नाम से memory Variable का निर्माण करेगी तथा .F. उसका मान store करेगी।

STORE DATE ( ) TO CUR\_DATE

जब हम FoxPro में Store Command का Use करते हैं तो यह बात ध्यान रखनी चाहिए कि FoxPro Variable में store Value को Screen पर दिखाता है। यदि FoxPro Data को Screen पर नहीं दिखाना चाहते हैं तो Memory Variable का निर्माण करते समय SET TALK OFF Command का पहले प्रयोग करना चाहिए।

**Example: -**

```
SET TALK OFF
STORE 12 TO HR
STORE .F.TO RAINED
```

यहाँ SET TALK Command का Use हुआ है इसलिए Fox Pro Variable में Store Data को Screen पर ही दिखायेगा। SET TALK OFF का प्रभाव SET TALK ON Command का प्रयोग करके समाप्त किया जा सकता है।

**ii) Equal (=) Command: -**

Equal (=) Symbol Store Command की तरह ही है। इस Command का प्रयोग हम निम्न प्रकार से करते हैं –

```
CUR_DATE = DATE ( )
PASS = .F.
NAME = PRAKASH
DATE = 12/30/1987
```

❖ **DISPLAY COMMAND:** -

हम Store या Equal Command के द्वारा कई Memory Variable का निर्माण करने के बाद यह जानने के लिए कि कितने Variable का निर्माण हुआ है, Display Command का Use करते हैं। इसके द्वारा हम Variable Type या Data को Screen पर Display कर सकते हैं।

**Syntax:** DISPLAY MEMORY LIKE <VARIABLE NAME>

**Example:** STORE 20 TO NUM  
DISPLAY MEMORY LIKE NUM,

**PUBLIC VARIABLE:** - इस Variable की Range उस Procedure तक नहीं होती बल्कि सम्पूर्ण Program में Global रूप में होती है।

**PRIVATE VARIABLE:** - Private Variable, Public Variable से बिल्कुल अलग होता है। इसमें अगर कोई Module की Procedure से Call करता है और हमने इसमें Variable का निर्धारण Private रूप से किया है तो scope केवल उसी Procedure तक ही सीमित रहता है।

## ❖ ARRAY: -

परस्पर Related एकसमान Data Items का एक Group जिसका एक ही नाम हो Array कहलाता है। Array एक Variable के समान Memory में स्थिति का एक नाम होता है, जहाँ एकसमान Type के Constants Store होते हैं। एक ही Variable में एक से अधिक Data Values को Memory में Store करने के लिए इस प्रकार के Data Type का प्रयोग किया जाता है।

Array में One-Dimensional and Two- Dimensional Array का Use होता है। Array का निर्माण करते समय FoxPro में DIMENSION और DECLARE Command का Use करते हैं।

```
DIMENSION SAL (10)
DECLARE SAL (10)
DECLARE SAL [10]
```

**One-Dimensional Array:** - ऐसा Array जिसमें Element की संख्या को व्यक्त करने के लिए एक Subscript का प्रयोग किया जाता है One-Dimensional Array कहलाता है।

### Syntax:

Dimension Array Name (Row, Column)

### Example:

```
DIMENSION MARKS (5)
DECLARE A (20), B (20), C (20)
DECLARE AGE [20]
```

### Example: -

```
STORE 200 TO MARKS (1)
```

यह Command MARKS (1) Element में 200 store कर देगी।

**Two-Dimensional Array:** -ऐसा Array जिसमें दो Subscript का प्रयोग किया जाता है, Two-dimensional Array कहलाता है।

```
DIMENSION MARKS [3, 5]
DECLARE MARKS (3, 5)
```

उपरोक्त Array में Value हम निम्न प्रकार से Store कर सकते हैं –

```
STORE 50 TO MARKS [1, 2]
```

## ❖ Array Memory Variable का निर्माण :-

Memory Variable के अलावा One and Two- Dimensional Array Memory Variable का निर्माण कर सकते हैं। Array का निर्माण करते समय FoxPro में DIMENSION और DECLARE Command का Use करते हैं।

```
DIMENSION SAL (10)
DELARE SAL (10)
DECLARE SAL [10]
```

उपरोक्त तीनों में से एक Command का प्रयोग करने पर FoxPro एक One-Dimensional Variable Array Sal का निर्माण करता है। जिसका Declare Size 10 है। अर्थात् इसमें 10 Element Store हो सकते हैं।

SAL (1), SAL (2), SAL (3), SAL (4) ..... SAL (10) एक Array का निर्माण करने के बाद Array के किसी Element में Data Store करने के लिए Store Command का प्रयोग किया जाता है।

### Example: -

```
STORE 1000 TO SAL (1)
```

यह Command SAL Array के First Element में 1000 Store करेगी। SAL (1) Element में हम Numeric, Character, Logical आदि Data Store कर सकते हैं।

### Example: -

```
STORE "SUM" TO SAL (2)
```

SAL (3) = T  
STORE {06/11/1987} TO SAL (4), SAL (5)

ठीक इसी तरह Variable के किसी भी Element का Data Change किया जा सकता है।

**Example: -**

STORE 200 TO SAL (1)

यह Command SAL (1) Element में Store Data 1000 को 200 में Change कर देगी। इसी प्रकार हम Pre-declare Array के Base को घटा या बढ़ा भी सकते हैं। इसको निम्न उदाहरण के द्वारा समझा जा सकता है –

DECLARE SAL 20

यह Pre-declared SAL नाम Array Memory Variable का Size 10 से बढ़ाकर 20 कर देगी। एक से अधिक Array Memory Variable के निर्धारण के लिए या तो DECLARE Command का अलग-अलग या फिर एक ही Command का उपयोग कर सकते हैं।

### ❖ Method of Create Keyboard Macros: -

FoxPro हमें बहुत सी Keys के साथ किसी Text या Command आदि को जोड़ने की सुविधा प्रदान करता है। ये Keys निम्नलिखित हैं –

1. Function Keys, From F1 To F10.
2. SHIFT Key + F1 To F9 Function Keys
3. CTRL + कोई भी Function Key या कोई एक अक्षर
4. SHIFT + CTRL + कोई भी Function Key या कोई एक अक्षर
5. ALT + कोई भी Function Key (सिर्फ F10 को छोड़कर) या कोई एक अक्षर
6. SHIFT + ALT + कोई भी Function Key (सिर्फ F10 को छोड़कर) या कोई एक अक्षर
7. SHIFT + CTRL + ALT + कोई भी Function Key (सिर्फ F10 को छोड़कर) या कोई एक अक्षर

उपरोक्त Key Combination हमें लगभग 200 Key – Board Macros बनाने की सुविधा प्रदान करते हैं। प्रत्येक Key Combination में 1024 Key Strokes Store कर सकते हैं।

**Creating Macros: -** Key – Board द्वारा Macros का निर्माण निम्न प्रकार से किया जाता है –

सबसे पहले FoxPro Menu System से System, Macros को Select करते हैं। Select करने पर Keyboard Macros Dialog Box Open होता है।

Macro Set वाले Part में चार Push Button Save, Restore, Set Default and Clean All उपस्थित हैं, जिनका उपयोग Macros Set करने के लिए किया जाता है।

Macro Dialog Box का Individual Macro/Key - Board Macro Box हमारे द्वारा बनाए गए और Default Function Keys की List दर्शाता है। यदि हमने कोई Macro सही बनाया है, तो यह केवल F2 से F9 तक तथा Shift 1 से Shift 9 तक की Function Keys दर्शाएगा। जैसे-जैसे हम Macro बनाते जायेंगे वह इस List में Add होते जायेंगे।

उदाहरण के लिए, “SPSCT, Computer Education” के लिए एक Macro बनाना तथा इसे ALT+T Key को Assign करना है। यहाँ ALT+T में + का मतलब यह है कि Alt key और T key को साथ Press करना है। कोई भी New Macro बनाने के लिए Keyboard Macro Dialog Box के Right Side के Box में New Button पर Click करना होगा। Click करने पर Macro Edit Dialog Box Screen पर Display होगा।

इस Dialog Box में आपको Macro का नाम उसके Contents देने के साथ ही साथ Macro Key भी Define करनी होगी। इसके दबाने के बाद Cursor अपने आप ही Macro नामक Text box में पहुँच जायेगा तथा यहाँ आपको Macro का नाम देना होगा। जो भी नाम आप देना चाहे Macro Name नामक Text Box में Type कर दें तथा Enter Key Press करें।

यदि आप Default Macro Name (ALT+T) चाहते हैं तो Macro Name Text box में कुछ भी Type किये बिना Enter Key Press करें।

अब Cursor Macro Contents नामक Text box में Blink करता है। इस Text Box में आप उन Commands या Text को Insert करते हैं जिसे आप Macro Key के लिये define करना चाहते हैं, जैसे – SPSCT, Computer Education, Type करके Enter Key Press करें। Last में OK Button पर Click करें।

FoxPro इस Macro को SPSCT नाम से Save कर देगा तथा आप पुनः Key – Board Macros Dialog Box के Middle Box में आ जाते हैं। आपके द्वारा बनाये गये Macro का नाम Ker – Board Macro Dialog Box के Middle में स्वतः लिखा जाता है।

यदि आप कोई दूसरा Macro Define करना चाहते हैं तो इस Method का पुनः Process करें। एक बार Macro Define होने के बाद आप उसे कभी भी Use कर सकते हैं। किन्तु FoxPro से Exit होने पर Macro स्वतः ही समाप्त हो जायेगा। यदि आप इस Condition से बचना चाहते हैं तो Macro को एक File में Save कर दें।

### ❖ Explain the following Commands: -

**? Or?? Command:** - इस Command के द्वारा हम अपने Data को अच्छी तरह से Present कर सकते हैं, क्योंकि इस Command की सहायता से हम अपने Data के अधिकार: जैसे – Fields, Memory Variable and Fix Value FoxPro Function का Result आदि Screen पर निर्धारित अथवा प्रदर्शित कर सकते हैं।

**Ex: - USE EMPLOYEE**

? NAME

Prakash Dwivedi

उपरोक्त Command द्वारा EMPLOYEE Data Base Open हो जायेगी तथा ? NAME Command FoxPro से Current Record के NAME Field के Content को Screen पर दिखाने के लिये करती है। FoxPro Prakash Dwivedi को Screen पर दिखाता है जो कि Current Record का Content है।

? Command के द्वारा हम एक ही Record को दो Fields के Content को निम्न प्रकार से Screen पर देख सकते हैं –

USE EMPLOYEE

? NAME, BASIC

Prakash Dwivedi 50,000

? Command से Field को जोड़ने के साथ हम कोई Message भी Screen पर Display कर सकते हैं।

USE EMPLOYEE

? "First employee name", NAME

First employee name Prakash Dwivedi

? Name, "Works in ", DEPT\_NAME

Prakash Dwivedi Works in Account

? Command द्वारा Various Function का Result भी Screen पर देखा जा सकता है।

**Ex: - CLEAR**

? DATE ( )

07/11/87

CLEAR Command Screen को Clear कर देगी तथा ? DATE ( ) Command Operating System (DOS) की System Date को Screen पर Display कर देगी।

**Ex: - CLEAR**

? DATE ( ) AT 20

उपरोक्त Command Computer की System Date की 20<sup>th</sup> Column पर Show करेगी।

### Use of?? Command: -

?? Command, ? Command की तरह ही Perform अथवा Information को Next Row में दिखाती है, जबकि ?? Command Information को उसी Row में दिखाती है।

?? NAME

?? BASIC

? Command and ?? Command के result को Printer पर देखने के लिए SET PRINTER ON Command का Use किया जाता है।

### ??? Command: -

FoxPro के Old Version में ?? Command Printer को Control करती थी, परन्तु Printer को Control (Ex – Bold Printing) करने वाले Code Printer पर भेजने से पहले SET PRINTER ON Command का प्रयोग करने की आवश्यकता थी। आजकल FoxPro में Printer को Control करने के लिए ?? Command का प्रयोग किया जाता है।

??? CHR (015)

इस Command में Condensed Printing को select करने के लिए ASCII Code 015 का Use किया गया है। Printer Control Code की पूरी list हमें Printer की दिशा – निर्देशिका (Manual) में उपलब्ध होती है।

## UNIT - IV

➤ **BRANCHING:** - जब एक से अधिक Options में से किसी एक Option के सत्य होने पर इस Option से Related statement को Operate करते हैं तो इसे Branching कहते हैं। Branching के कुछ Control Statement निम्नलिखित हैं -

- **IF – END IF:** - FoxPro Design लेने के लिए If – End If and If – Else – End If Command का प्रयोग किया जाता है। इसका Format निम्न रूप में होता है –

**Syntax: -**

If <Condition>

-----

<Command>

-----

-----

End if

FoxPro If के द्वारा दी गई Condition को Check करता है, यदि यह Condition सही है तो if end if के बीच स्थित सभी Command को Perform कर देता है।

**Ex: -** Input “Enter Your Age” To Age

If Age <18

? “You are not valid for voting”

End If

- **IIF ( ) Function:** - IIF Function को Immediate IF भी कहते हैं। यह IF\_ELSE\_END IF का One Line Version है। IIF Function के साथ-साथ उसी Line में Expression भी लिखते हैं। यदि Condition True होती है तो यह उससे Related Expression को Perform करती है तथा Condition False होने पर दूसरे Expression को Perform करती है।

**Ex: -**

? IIF (SALARY > 5000, “Salary is more than 5000; Salary is not more than 5000”)

IIF Function के तीन Parts होते हैं – first Part Condition (SALARY>5000) है, अगले दोनों Parts Expression हैं। इसे Compact IF Function भी कहते हैं।

- **Do Case - End Case:** - यह Statement आपको बहुत सी Condition में से किसी एक को आसानी से चुनने की सुविधा प्रदान करता है। यह Multiple Construct है। इसका प्रयोग करने से Complicated Multiple IF Else End IF के प्रयोग से बचे जाते हैं। आप Do Case End Case Command के साथ बहुत से Alternative प्रयोग करते हैं।

**Syntax: -**

```
DO CASE
Case < Condition 1>
    <List of Commands>
Case < Condition 2>
    <List of Commands>
Case < Condition 3>
    <List of Commands>
```

.....

.....

**OTERWISE**

<List of Commands>

**END CASE**

प्रत्येक DO CASE के लिए एक END CASE लगाना आवश्यक होता है। DO CASE\_END CASE को नाचे दिए गए उदाहरण से समझा जा सकता है –

```
CLEAR
?
INPUT "Enter Number between 1 and 4;" To NUM
?
DO CASE
    CASE NUM = 1
        ? "You Entered One"
    CASE NUM = 2
        ? "You Entered Two"
    CASE NUM = 3
        ? "You Entered Three"
    CASE NUM = 4
        ? "You Entered Four"
    OTHERWISE
        ? "Wrong Number"
END CASE
?
```

यदि FoxPro में कोई Condition True नहीं मिलता तो वह OTHERWISE तथा END CASE के बीच में स्थित Command Set को Perform करता है। उदाहरण के लिए, 7 Input करने पर उपरोक्त Program में कोई Condition True नहीं मिलती है। फलस्वरूप यह Screen पर Wrong Number Display करेगा।

- **If Else - End If:** - If Else End If, IF End If का एक Version है, जो आपको दो Command Set उपलब्ध कराता है। First Command Set IF Condition के True तथा Second Command Set IF Condition के False होने पर Operate होता है।

**Syntax: - IF <Condition>**

<Condition Set 1>

Else

<Command Set 2>

End IF

**Ex: -** Input "Enter Your Salary" To Salary

If Salary > 5000

? "Your Salary is more than 5000"

Else

? "Your Salary is not more than 5000"

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

End If

➤ **LOOPING:** - FoxPro Programming में Statement के एक set को किसी निश्चित संख्या तक बार-बार दोहराने की प्रक्रिया Looping कहलाती है। Looping Statement निम्नलिखित होते हैं –

- **Do While - End Do:** - यह एक Important Command है। इसकी उपयोगिता को समझने के लिए हम एक Problem चुनते हैं। यदि आपको 100 से 110 तक की संख्याओं का square Root Find करना है। इसके लिए FoxPro में SQRT ( ) Available है। यह Function किसी भी Positive Number का Square Root बताता है।

**Ex: - ? SQRT (25)**  
5

इसी प्रकार यदि हमें 100 से 110 तक के Square Root Find करने हैं तो निम्न Command द्वारा ज्ञात कर सकते हैं –

? SQRT (100)  
? SQRT (101)  
? SQRT (102)  
.....  
? SQRT (110)

यह Program हमारी Problem का Solution कर देगा पर यह Time नष्ट करने वाला Program है। यदि आपको 100 से 500 या इससे भी अधिक संख्याओं का Square root Find करना हो तो इसमें बहुत समय लगेगा। इस Problem से बचने के लिए DO While End Do Command का Use किया जाता है। DO While End Do Command के द्वारा उपरोक्त Program निम्न प्रकार बनेगा –

**Ex: -** Store 100 to Num  
Do While Num (<111)  
Num = Num + 1  
End Do

- **For - End For:** - इसका प्रयोग Statement के एक set को किसी निश्चित संख्या तक Loop में Perform करने के लिए तो होता है। यह एक Memory Variable या एक Array Element को Counter की तरह प्रयोग करता है, जो यह निश्चित करता है कि Loop के अन्दर Statement को कितनी बार Perform करना है।

**Syntax: -**

For <Memory Variable> = <exp 1> TO <exp 2> [Step <exp 3>]

-----  
End For/Next

**Ex: -** Store 100 to NUM

For Num = 100 to 110  
? SQRT (Num)  
End For

FOR – END FOR Counter को स्वतः ही एक संख्या आगे बढ़ाता है। यदि आप चाहे तो STEP Option द्वारा Counter की वृद्धि इच्छानुसार कर सकते हैं। STEP Option का प्रयोग FOR के साथ किया जाता है।

- **Scan - End Scan:** - Scan – End Scan एक दूसरे प्रकार का प्रभावशाली Structure है। यह Do – While Loop की तरह ही Act करता है। यह Table File को Sequentially Read करता है। इस Command का Syntax निम्न है –

SCAN [Scope] [For Condition] [WHILE Condition] [Commands]  
END SCAN

Scan Loop में SKIP Command Use करने की आवश्यकता नहीं है, क्योंकि FoxPro स्वतः ही जब उसे ENDSCAN मिलता है, Next Record पर Pointer को पहुँचा देता है।

**Ex: -** USE EMPLOYEE

SCAN FOR DEPT\_NAME = "HR"  
? "Department" + DEPT\_NAME

**Guided by: Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)**



## END SCAN

• **Text - End text:** - ? Command के द्वारा Screen पर Message Display किए जाते हैं, परन्तु यदि आप किसी बड़े Message को Screen पर Display करना चाहते हैं तो आपको बहुत सी ? Command का Use करना पड़ेगा। इस कार्य को सरल तरीके से करने के लिए FoxPro, TEXT\_END TEXT Structure की सुविधा उपलब्ध कराता है, जिससे आपको ? Command बार-बार लिखने की आवश्यकता नहीं पड़ती है। जब आप ? Command के स्थान पर TEXT\_END TEXT Structure का उपयोग करते हैं तो आपको Quotation (“ ”) Mark लगाने की आवश्यकता नहीं होती है। आप इच्छानुसार जितनी Line चाहें TEXT\_END TEXT के अन्दर लिख सकते हैं।

**Ex: -** USE EMPLOYEE  
SET TALK OFF  
CLEAR  
TEXT

Program to maintain the EMPLOYEE .DBF File

- (1) Edit a Specific Record
  - (2) PACK the delete Record
  - (3) ADD NEW Record
  - (4) QUIT FoxPro
- END TEXT

INPUT “Enter Ur Choice” TO CH

उपरोक्त विधि से आप किसी भी Program में TEXT\_END TEXT Structure का प्रयोग कर सकते हैं। File को Save करने के लिए Ctrl+W Press करें।

**Q.:** - Explain the various Common Error Message in FoxPro?

**Ans:** - FoxPro एक Program file को Perform करते समय यदि किसी Command को नहीं समझ पाता तो वह इसे Perform नहीं करता तथा उससे Related Error Message Screen पर दिखा देता है। Error Message को Alert Window में दिखाने के साथ-साथ FoxPro निम्नलिखित Options भी Screen पर दिखाता है –

- (1) Ignore
- (2) Cancel
- (3) Suspend

**(1) Ignore:** - Ignore Push Button Select करने पर FoxPro next Command को Perform करने की कोशिश करता है।

**(2) Cancel:** - यदि आप Cancel Push Button को Select करते हैं तो FoxPro Program File को Terminate कर देता है, तथा Cursor, Program Window में पहुँचा देता है।

**(3) Suspend:** . Suspend Option Program में क्रियान्वयन को अस्थायी रूप से रोक देता है। DBF के Record Pointer आदि को बिना Disturb किये आपको Program Window में पहुँचा देता है।

FoxPro में programming के दौरान होने वाली Error का detail निम्न प्रकार है –

**(1) Syntax Error:** - यह एक important Error है। FoxPro में प्रत्येक Command का अपना एक Syntax होता है, जिसमें आपको उसे Enter करना होता है। Syntax Error कई कारणों से हो सकती है जैसे – Wrong Spelling, Wrong Variable Name, Clause न लिखने पर, Program को सही क्रम में न लिखने पर आदि।

**(2) Data Type Mismatch:** - जब आप एक ऐसे Expression का Use Program में करते हैं जो Various Data Type से मिलकर बना है तब यह Error Message आता है। इस Problem से बचने के लिये आपको इस Type के Data को दूसरे Data Type में Function की सहायता से बदलने की आवश्यकता होती है।

**(3) Variable Not Found:** - इस Error Message का आशय यह है कि FoxPro Command के साथ आपके द्वारा दिए गए Variable को Search करने में असमर्थ होता है।

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

**(4) File in Use:** - इस Error के आने का मुख्य कारण यह है कि जब आप किसी Already Open File को Open करते हैं या उसका नाम बदलना चाहते हैं, तो इस समस्या से समाधान के लिए आपको नाम बदलने से पहले File को Close करना होगा।

**(5) Can Not Create File:** - इस Error का मुख्य सूत्रधार Operating System होता है। यदि Disk या Directory पूरी भर चुकी है तो Operating System New File Create करने में असमर्थ होता है।

**(6) End of File Encountered:** - जब Record Pointer File के Last Record से आगे हो तो यह Error message आता है।

**(7) Operator/Operand Type Mismatch:** - यह Error तब Display होता है जब किसी Arithmetic, String, Logical Operator या Function के साथ Invalid Data Type का Use करते हैं। यह Error Message File को Compile करते समय आती है।

**(8) Unrecognized Command Verb:** - यदि आप किसी Command की Wrong Spelling लिख देते हैं, तो इस प्रकार की Error आती है। उदाहरण के लिए – यदि आप DISPLAY के स्थान पर DISPLY आदि कुछ Wrong Spelling लिख देते हैं तो “Unrecognized Command Verb” Error Screen पर आती है।

**(9) IF/ELSE/END IF Mismatch:** - इस प्रक्रिया के द्वारा जब आप IF Statement के साथ इससे Related Statement END IF या ELSE का प्रयोग नहीं करते हैं, तब आपको इस समस्या का सामाना करना पड़ता है।

**(10) MEMO File is Missing or Invalid:** - जब आप किसी Data Base File को Open करते हैं जिसकी Memo File या तो Delete हो चुकी है या remove हो चुकी है तो आपको इस Error का सामाना करना पड़ता है।

**(11) Keyword in Command:** - जब आप किसी Command के साथ कोई ऐसी Clause लिखते हैं जिसे FoxPro Read नहीं कर पाता है, तो तब आपको इस Error का सामाना आपको करना पड़ता है।

**(12) Nesting Error:** - यह Error Message तब आता है जब आप DO के साथ END DO, SCAN के साथ END SCAN, FOR के साथ END FOR नहीं लगाते। अथवा Loop या Exit Clause Do While - End Do, SCAN – END SCAN या For - End For Command के बाहर लिखते हैं।

**(13) Position is off the Screen:** - जब हम @ Command के साथ Screen पर पर्याप्त Row या Column Number से बड़े Row या Column Fix करते हैं तो हमें इस प्रकार के Error का सामाना करना पड़ता है।

**(14) Record is not in Index:** - जब हम Active Data Base File के किसी Index की Field में Related Index File को बिना Active इसमें Correction करते हैं, तब हमें यह Error Message मिलता है।

**Question:** - Write the Program to compute the factorial of a given number.

**Solution:**

\* Program to compute the factorial of a given number

\* Name of the File is FACT.PRG

SET TALK OFF

CLEAR

INPUT “Enter a Number” TO NUM

F = 1

FOR J = 1 TO NUM

F = F\*J

END FOR

```
? "The Factorial of"
?? N
?? 'Is'
?? F
```

उपरोक्त File को निम्न Command द्वारा Run कराते है –  
DO CASE

**OUTPUT: -**

```
Enter a Number: 5
The Factorial of 5 is 120
```

**Question:** Write a program to sum all the Digits of a given number entered by User.

**Answer:**

- \* Program to sum all the digits of a number
- \* Name of the File is TRY.PRG

```
CLEAR
SET TALK OFF
SUM = 0
INPUT "Enter a Number" TO NUM
DO WHILE NUM > 0
A = MOD (NUM, 10)
SUM = SUM + A
NUM = NUM/10
END DO
? "The Sum of Given Number"
?? SUM
```

उपरोक्त File को DO TRY Command द्वारा Run कराने पर निम्न OUTPUT मिलता है –  
Enter a Number 1234  
The Sum of the Given Number = 10

**Question: -** FOR – END FOR का Use करते हुए 1 से 100 तक की संख्या Print करने के लिए Program लिखिए।

**Solution: -** FOR i = 1 to step 1  
? i  
End For

इस Program को लिखने के बाद इसे FoxPro Command Prompt पर Run कराते है तो यह 1 से 100 तक की संख्याओं को Screen पर Print करेगा।

**Question: -** Write the program in FoxPro to find out Sum of any 15 odd numbers.

**Solution: -**

```
Declare Number (15)
? "Enter any 15 odd numbers"
For i = 1 to 15 step 1
INPUT "Enter number" to number[i]
End For
i = 1
Do While i <= 15
If number [i] % 2! = 0
Sum = Sum + Number [i]
End If
```

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

End Do  
? "Sum of 15 odd numbers is", Sum

**Question:** - Write a program to find the Sum of the following Series.

**Answer:** -

```
Clear
Store ( ) to sum
m = n - 1
For i = 0 to 2 * m step 2
Sum = sum + i
End For
? "Sum of series is", sum
```

### ❖ Program File In FoxPro: -

FoxPro में Function & Command को एक Sequence में प्रयोग कर सकते हैं, जिससे कोई भी कार्य पूरा किया जा सकता है। इस प्रकार के Sequence and Command को Program File कहा जाता है। Program File में Store Command को याद रखने की आवश्यकता नहीं है बल्कि FoxPro से अनभिज्ञ व्यक्ति भी इस Program File को आसानी से Execute कर सकता है। Command & Execution में बहुत कम समय लगता है। जिससे कि समय की बचत होती है, क्योंकि आपको प्रत्येक Command Key Board से Type करने की आवश्यकता नहीं होती है। आप इन Command का प्रयोग Program File के अन्दर ही कर सकते हैं।

Program File का निर्माण FoxPro में दो प्रकार से किया जा सकता है –

**1) Menu द्वारा :-** File Menu में New Option द्वारा हम Program File का निर्माण कर सकते हैं। File Menu के New Option पर Click करने पर Dialog Box में Program Radio Button को Select कर New Button पर Click करते हैं। इतना करने पर FoxPro एक खाली Window Screen पर Display करता है। Program File का Extension .prg होता है। इस Window में Command Type करने के बाद File को Save करने के लिए Ctrl+W Key को Press करते हैं।

**2) Command द्वारा :-** FoxPro में Menu System के अलावा हम Modify Command की सहायता से Program File Create कर सकते हैं। इस Command द्वारा File बनाने के लिए हमें Command द्वारा Window में निम्न Command देनी पड़ती है –

#### Modify Command Demo

उपरोक्त Command द्वारा Demo.prg नाम से एक Program File Create होती है जिससे Command देने के बाद Ctrl+W Key का Use कर इस File को save कर सकते हैं।

❖ **Program File को Run कराना :-** Program File के Store Command को Do Command के द्वारा Perform कर सकते हैं। इसके लिए Do Demo Command Type करते हैं, जहाँ Demo File name है फिर Enter Key Press करते हैं। जिससे Program Perform हो जाता है। Do Command के अलावा Fox Pro Menu System के द्वारा भी Program File को Run कराया जाता है। इसके लिए Program Menu से Do Option को चुनते हैं। Do Option पर Click करने से Do Dialog Box Screen पर Display होता है, जिसमें Program File और अन्य दूसरी File की List होती है, जिसे FoxPro Perform कर सकता है।

**Question:** - Explain ACCEPT, INPUT, & WAIT Command.

**(1) ACCEPT Command:** - ACCEPT Command का प्रयोग Screen पर specified Message (Prompt) Display करने तथा Key Board द्वारा Enter किए गए Data को Character Variable में Store करने के लिए किया जाता है। जिस variable में data Store करना है उसका नाम ACCEPT command के साथ दिया जाता है।

**Ex:** - ACCEPT "Enter Your Name: TO NAME

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

उपरोक्त Command देने पर FoxPro Screen पर एक Message, "Enter Your Name" Screen पर Display करेगा और User का Data Enter करने का Wait करेगा। Name Type कर Enter Key Press करने पर FoxPro इस नाम को Name Variable में Store कर देगा।

**(2) INPUT Command:** - INPUT Command, ACCEPT Command की तरह ही होता है, अन्तर केवल इतना है कि ACCEPT Command के द्वारा केवल Character Variable ही बनाया जा सकता है, जबकि INPUT Command के द्वारा Numeric, Character, Logical & Data किसी भी Type का Variable बनाया जा सकता है।

**Ex:** - INPUT "Enter Data" TO CHOICE

उपरोक्त Command देने पर FoxPro Screen पर "Enter Data Message" Screen पर Display करके User का wait, data enter करने के लिए करेगा। जिस तरह का Data Enter होगा उसी प्रकार का CHOICE Variable बनेगा।

**(3) WAIT Command:** - WAIT Command का प्रयोग किसी Program में उसका Extension अस्थायी रूप से रोकने के लिए किया जाता है। WAIT Command का प्रयोग करने पर FoxPro Screen पर निम्न Message Display करता है।

**Ex:** - Press any Key to Continue.....

इसके पश्चात् FoxPro User द्वारा किसी Key के Press करने का wait करता है। WAIT Command का प्रयोग Specified Message देने में किया जा सकता है।

WAIT "Make a Printer Ready and Press a Key"

यदि आप WAIT Command के Response में Press की गई Key को Character Variable में Store भी कर सकते हैं।

WAIT TO CHECK

WAIT WINDOW TO CHECK

WAIT "Do use want to continue" TO ANS

### ❖ **Debugging Technology:** -

कई बार आपका Program Correct होता है, फिर भी वह Run नहीं होता। ऐसा इसलिए होता है, क्योंकि आपके Program में Logic Error है। जिससे कारण वह Run नहीं हो पाता। जैसे— If Else—End If, Do While—End Do, Scan—End Scan आदि का गलत प्रयोग, गलत Command या गलत Function का Use होना। यह Programmer की जिम्मेदारी है कि वह Command की Sequence को Check कर Program को Run करे। क्योंकि Program को ठीक करके Run करने का Shortcut नहीं होता।

इन गलतियों को ठीक करने के लिए FoxPro कई लाभदायक Command प्रदान करता है —

**(1) Type Program:** - जब आपका Program ठीक से Run ना हो तो सबसे पहले आपको अपने Program का एक Printout निकाल लेना चाहिए। अपने Program को Type या Display कराने के लिए Type Command का Use करें।

**Ex:** - TYPE EMP .PRG TO PRINTER

**(2) Screen Activity को Alternate File में Safe करना :-** आप Screen Activity को एक Text File में Store कर सकते हैं। इस Text File को Alternate File कहते हैं। एक बार इस Alternate File के बन जाने पर यह Screen पर प्रकट होने वाले सारे Message के साथ Key Board के द्वारा Enter कराये जाने वाले Text को भी Store कर लेती है। बाद में इस File का Use आप Program की गलतियों को खोजने में कर सकते हैं। FoxPro इस उद्देश्य के लिए निम्नलिखित दो Commands का Use करता है —

**(i)** SET ALTERNATE TO <FILE NAME> {ADDITIVE}

**(ii)** SET ALTERNATE ON/OFF

First Command Text File को Alternate File के रूप में Screen की Actions को Store करने के लिए Open कर देती है।

**Ex: - SET ALTERNATE TO SAMPLE**

बाद में आप TYPE Command का प्रयोग करके SAMPLE .TXT File के Contents को देख सकते हैं।

TYPE SAMPLE .TXT अथवा

TYPE SAMPLE .TXT TO PRINTER

**(3) Trace Window में संचालित होते हुए Program को देखना :-** Trace Window द्वारा Perform हो रहे Program की सारी Command को बारी-बारी से देखा जा सकता है। FoxPro Perform हो रहे Program को दिखाता है तथा Commands को Highlight भी करता है।

Trace Window को Window Menu द्वारा अथवा निम्न Command द्वारा Open किया जा सकता है –

SET ECHO ON

SET STEP ON

ACTIVE WINDOW TRACE

FoxPro आपके सामने Trace Window को Open कर देगा।

FoxPro के Trace Window को Open करने के बाद जब आप Program को Execute करेंगे तो FoxPro Trace Window में एक के बाद एक Perform होने वाले Commands को दिखता जाएगा।

**(4) Program में Break Point का Use:-** यदि आपका Program बहुत बड़ा है और आप जानते हैं कि Program कहाँ तक सही है और व उससे आगे कहाँ गलती होने की संभावना है? इस Condition में पूरे Program में Error को Single Step Mode में Search करना बहुत कठिन होगा और Time भी अधिक लगेगा।

इस Condition में Break Point का Use कर सकते हैं। माना कि Line 50 के बाद Error होने की संभावना है तो आप 50 वीं Line पर एक Break Point Set कर सकते हैं। इस प्रकार आपका Program 50 वीं line तक सामान्य रूप से संचालित होगा तथा 51 वीं line पर आकर रुक जाएगा। आप Program को फिर से Single step Mode में संचालित कर Error को खोज सकते हैं। किसी भी line पर Break point Set करने के लिए Cursor को वहाँ ले जायें तथा Space bar या Enter Key को Press करें। अथवा Mouse से Click करें।

यदि आप Break Point को हटाना चाहते हैं तो Mouse से उस पर दोबारा Click करें या फिर Cursor को उस पर ले जाकर पुनः Space bar या Enter Key दबा दें। सारे Break Point एक बार में हटाने के लिये ^B Press करें या Trace Window के Program Menu में Clear Break Point को select करें।

**(5) Program को One Step में संचालित करना :-** Program के Trace window में load हो जाने के बाद आप इसको Single step mode या फिर Normal Execution mode में संचालित कर सकते हैं। Single step mode में FoxPro एक बार में एक Command को संचालित करने के बाद रुक जाता जाएगा। Single step mode का use करने के लिए S को दबा सकते हैं। दूसरी Command को संचालित करने के लिए दोबारा S दबायें। इस प्रकार एक-एक करके Commands को संचालित किया जा सकता है।

**(6) Debug Window में Variable को Check करना :-** Debug Window आपको Program संचालित करते समय किसी भी Variable या FoxPro के Function को Monitor करने की सुविधा प्रदान करती है। Program के संचालित होते समय जब भी आप किसी Variable की Value देखना चाहते हैं, केवल उसको Debug Window में सम्मिलित कर दें। जब आप Program को Run करेंगे FoxPro स्वतः ही Debug Window में सम्मिलित सभी Variable की Value को Display करता रहेगा। Debug Window को आप दो प्रकार से Open कर सकते हैं।

(1) Window Menu के Debug Option को Select करके।

(2) Command Window में ACTIVATE WINDOW DEBUG Command Enter करके।

Debug Window दो Part में Divide होती है। Left Side आपके द्वारा Enter किए गए Variable तथा Function एवं right side उनके result show होते हैं।

Debug Window को Window Menu में से Debug Window पर click करके अथवा ESC Key Press करके, Close किया जा सकता है।

## UNIT - V

### ❖ Multiple Data Base File: -

DBMS के अंतर्गत हमने बहुत सारी Data Base File को Open किया तथा उन पर कार्य किया। ज्यादातर एक समय में एक ही Data Base File को Open करते हैं। Real Life Situation में Data Base Application के लिए एक से अधिक Data Base file की आवश्यकता होती है। इस Data Base file का Size Small तथा Program File को Efficient and Modular बनाता है। इस प्रकार Disc Space की आवश्यकता को भी कम करता है।

❖ Opening Multiple Data Files: - FoxPro में Data Base File USE Command का प्रयोग करके Open की जा सकती है। Data Base file के Open होने पर FoxPro States Bar में Open की गयी File का नाम Display करता है। यदि आप USE Command का प्रयोग करके दूसरी Data Base file Open करते हैं, तो पहले वाली Data Base file Close हो जाती है। अर्थात् एक बार में केवल एक ही Data Base file Open होती है।

किसी work Area को select करने के लिए SELECT Command का Use किया जाता है। Second Work Area को Select करने के लिए SELECT B या SELECT 2 का प्रयोग किया जाता है। दो Files को निम्न प्रकार Separate Work Area में Open करते हैं। पहले Work Area 2 में Deposit .DBF को Open करने के लिए निम्न syntax होगा –

```
SELECT 2
USE DEPOSIT
```

**Ex:** - एक छोटी Bank Branch अपने Fix Deposit के कार्य को Computerized करना चाहती है तो इसके लिए Fix Deposit की Detail के लिए Data Base file का Structure निम्नलिखित होगा –

Field Name	Width	Type
NAME	20	Character
ADDRESS	40	Character

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

CITY	20	Character
PIN	6	Character
AMOUNT	7	Numeric
DEP_DATE	8	Date
PERIOD	5	Numeric

इस Structure में Amount और Period Numeric Type Field है। DEP\_DATE एक Date Type Field है, और शेष सभी Field Character Type हैं। इस Data Base की Record Length 107 हो सकती है।

#### Formula: -

$$\text{Record length} = \text{Sum of the Width of All Field} + 1$$

+1 प्रत्येक Record के लिए Delete Status को Store करने के लिए एक Extra Byte 1 होता है। इस प्रकार Fix Deposit का कार्य प्रारम्भ हो जाता है। जल्दी ही Data Base File का Size बढ़ने लगता है। इसमें record की संख्या 10,000 तक बढ़ सकती है और File की Size 10,70,000 तक बढ़ सकता है। Analysis करने पर पता चलता है कि Bank की इस Branch में लगभग 800 customers ऐसे हैं जो Fixed Deposit में पैसा जमा करते हैं।

#### ❖ Explain the following Commands.

- (1) SET RELATION
- (2) UPDATE
- (3) COPY TO
- (4) JOIN
- (5) APPEND FROM

(1) **SET RELATION:** - यह Command दो या दो से अधिक Work Area में Open Data Base File में Relation Establish करता है।

दो या दो से अधिक DBF File को Common Field पर Set Relation Command द्वारा जोड़ा जाता है।

#### Ex: -

मान लीजिए हमें उन Costumer या Depositor के नाम एवं पते चाहिए जिन्होंने Fix Deposit कराया है। इस Query के लिए हमें Deposit .DBF से ACCOUNT NO. तथा उस ACCOUNT NO. के लिए नाम व पते दूसरी DBF Master .DBF से लेने की आवश्यकता है। First Deposit File को Present File and Second File को Child कहते हैं। आप इन दोनों File को Common Field ACCOUNT NO. पर Set Relation द्वारा Relate कर सकते हैं। इन Field में relation Set करने के लिए Child Data Base को Common Field पर Index करना होता है, तो निम्नलिखित Command द्वारा होगा -

```
USE MAST
INDEX ON CODE TAG CD
```

अब हम दोनों File में Relation Establish कर सकते हैं। दो File को link करने के लिए Separate Work Area में Open करना पड़ता है।

```
SELECT 1
USE MAST
SELECT 2
USE DEPOSIT
```

उपरोक्त Command द्वारा दोनों DBF Open हो जाती है। Last Command अपने आप ही Master .CDX File को Open कर देती है।

```
INDEX TO CD ON ACCOUNT NO
```

अब अगला कार्य Present file को चुनना है तथा SET RELATION Command का Use करना है

```
SELECT 2
SELECT DEPOSIT
```

अब Relation बनाने के लिए निम्न Command का Use होगा।

```
SET RELATION TO CODE INTO MAST
```



उपरोक्त Command का First Part Command Field को Specify करता है। तथा Second Part Work Area को Specify करता है। इस प्रकार दोनो Data Base File में SET RELATION Command द्वारा Relation Establish हो जाता है।

(2) **UPDATE:** - UPDATE Command एक Common Field के साथ दो Data Base पर कार्य करती है। File Common Field पर Index या Short होता चाहिए। ये Data Base File अलग-अलग Work Area में Open भी होनी चाहिए, तभी UPDATE Command का Use करना चाहिए। FoxPro अपने आप ही दोनों Files के सभी Records को Scan करता है तथा आवश्यक Field को Update कर देता है। इसका **Syntax** निम्न है –

```
UPDATE ON <common field name>FROM<data base>
REPLACE <field name> WITH <expression>
```

MAST .DBF व DEPOSIT .DBF को अलग-अलग Work Area में Related Tag and Index के साथ Open करते हैं।

```
SELECT 1
USE MAST ORDER CODE
SELECT 2
USE DEPOSIT ORDER CODE
SELECT 1
UPDATE ON CODE FROM DEPOSIT REPLACE _ DEP
WITH GR _ DEP + DEPOSIT →AMOUNT
4 Records Updated
```

UPDATE Command के दो भाग होते हैं –

→ First Part UPDATE ON CODE FROM DEPOSIT FoxPro को inform करती है कि Common Field Code पर आधारित DEPOSIT .DBF के Record से Current Data Base (MAST .DBF) Update किया गया है।

→ Second Part REPLACE GR \_ DEP WITH GR \_ DEP + DEPOSIT → AMOUNT FoxPro को inform करता है कि Field Update हो चुका है।

(3) **COPY TO:** - Copy To Command द्वारा आप उपयोग की गई Data Base File के चुने हुए या सारे Records को Copy करके एक New Data Base File Create कर सकते हैं।

```
USE MAST
COPY TO A: SAMPLE
```

यह COPY TO Command Disk में SAMPLE .DBF नाम से एक New File Drive A में बना देती है।

```
COPY TO MAST 2 FOR CODE < "105">
COPY TO MAST 3 FIELDS CODE, NAME, ADDRESS
```

पहली Command MAST .DBF से केवल उन Records को MAST2 में Copy करता है जिनका CODE का नाम 105 से कम है। इसी प्रकार दूसरी Command केवल CODE, NAME, ADDRESS Field वाले Records को MAST3 में Copy करती है।

COPY Command स्वतः ही Structural Compound Index File, Copied Data Base के लिये कर सकते हैं।

```
USE MAST
COPY TO MAST 4 NAET 4 WITH CDX
```

(4) **JOIN:** - JOIN Data Base के द्वारा दो Open Data Base File को Combine कर एक New Data Base Create की जा सकती है। इस New Data Base File में कुछ Special या सारे क्षेत्र को जोड़ सकते हैं।

**Ex** – MAST .DBF and DEPOSIT .DBF Data Base File को Equal Copy Value में जोड़ा जा सकता है। इन दोनो Files को जोड़ने तथा जुड़े हुए Records को अलग File में Save करने के लिये निम्न Command दी जाती है –

```
SELECT 1
USE MAST ORDER CODE
USE DEPOSIT ORDER CODE
JOIN WITH DEPOSIT TO BANK FOR CODE = DEPOSIT →CODE
```

FIELDS CODE, NAME, DEPOSIT → AMOUNT

DEPOSIT → PERIOD

उपरोक्त Command देने पर दोनों Data Base File अलग-अलग Work Area में Open हो जायेंगी तथा Common Field पर उनकी Joining होगी। दोनों Data Base File (MAST.dbf, DEPOSIT.dbf) के Join होने पर एक तीसरी Data Base file BANK नाम से बनेगी, जिसमें कि चार Fields CODE, NAME, ACCOUNT, PERIOD होंगे। इस New BANK Data Base File को Structure निम्न Command के द्वारा देखा जा सकता है –

USE BANK

DISPLAY STRUCTURE

**(5) APPEND FROM:** - APPEND FROM Command का Use एक File के Records, other File में Add करने के लिए किया जाता है। कभी-कभी आपको एक File के Records को other file में Transfer करने की आवश्यकता होती है।

**Ex:** - आपके पास MAST.DBF File से Related कुछ Records MAST.DBF File, जो कि Floppy में है, उपस्थित है। आप चाहते हैं कि MAST.DBF File (Floppy) से ये Record MAST.DBF File में Transfer हो जाये, जो कि Hard Disk पर है तो इस कार्य के लिए आपको APPEND FROM Command की आवश्यकता होती है। इस Command को इस प्रकार Use करते हैं –

USE MAST

APPEND FROM A: MAST

APPEND FROM Command, Drive A में MAST.DBF के सभी Records को Read करता है और इन Records को Hard Disk पर उपस्थित MAST.DBF File में Copy कर देता है। यदि दोनों Data Base Files का Structure Same होता है तो सभी Records जिनमें Data होता है, APPEND हो जाते हैं। यदि दोनों Data Base File के Structure Different होते हैं, तब केवल दोनों Files में Common fields का Data ही APPEND होता है।

**Q.:** - Write notes on the following:

**(i) Custom Screen Designing:** - FoxPro में Custom Screen Designing Part वह कहलाता है जो User का Data I/O (input/output) कराने के लिये उनके बीच Design किया जाता है तथा User की आवश्यकता के लिए हम जिस Screen का निर्माण करते हैं वह Custom Screen कहलाती है। इसी की Help से User अपनी आवश्यकता के अनुसार I/O Message, Variable के नाम और Screen पर उनका उचित स्थान निश्चित कर सकता है। इस Screen को पूर्ण रूप से बनाने के लिए @GET, @Say – Get read आदि Command का प्रयोग किया जाता है।

@ Command एक most important command है जो बहुत से कार्य करने में सहायक होती है। जैसे –

(1) इच्छित Data को अपने अनुसार Format में Display करना।

(2) Data को Field या Variable में Input करना।

(3) Data को Check Box, Radio Button, Push Button आदि द्वारा Input करना।

(4) Box और Lines को बनाना या मिटाना।

(5) Custom Screen Format Files बनाना।

Computer की Screen पर Total 25 Rows and 80 Column होते हैं। Row की Numbering 0 से 24 तक तथा Column की Numbering 0 से 79 तक होती है।

Screen पर Every Condition Rows and Column Coordination द्वारा Show की जाती है। इस प्रकार की First left condition 0, 0 Coordinate तथा last Bottom right condition 24, 79 होती है।

@ Command के द्वारा Data Display कराने का Format निम्न है –

@ROW, COLUMN [SAY Character, string]

**Ex:** -

@ 10, 5 SAY DATE ( )

@ 5, 10 SAY "WELCOME"

उपरोक्त पहली Command Current Date को Screen पर 10, 5 Coordinate पर Display करती है, जो वास्तव में 11, 6 Coordinate को Refer करता है।

**Ex:** -

USE EMPLOYEE

@ 12, 5 SAY EMP \_ CODE

@ 25, 5 SAY "Name" + NAME

### Inputting the Data with @ .... GET

@ ....GET Command का प्रयोग User से Data प्राप्त करने के लिये किया जाता है। ये Command READ Command के साथ Field, Memory Variable या Array Element में Data Input करती है।

CLEAR

STORE "Prakash" TO NAME

@ 5, 5 SAY "Enter Your Name" GET NAME

उपरोक्त Command के क्रियान्वित होने पर Screen की 5<sup>th</sup> Row तथा Column पर निम्नलिखित Message Display होगा

—

Enter Your Name Prakash

यदि आप GET Variable में Data Enter करना चाहते हैं तो आपको READ Command का प्रयोग करना होगा।

Ex: -

STORE "Prakash Dwivedi" TO NAME

CLEAR

@ 5, 5 SAY "Enter Your Name"

@ 5, 20 GET NAME

READ

? NAME

**Custom Screen Designing Tools in FoxPro:** - FoxPro में Custom Screen Designing हेतु निम्नलिखित Tools पाए जाते हैं —

**(1) Screen Builder** - यह FoxPro की बहुत ही Important Utility है। इस Utility के द्वारा Simple एवं Complex दोनों तरह की Screen को बहुत ही आसानी से तैयार किया जा सकता है। Screen पर उपस्थित Fields एवं Variable को Objects कहा जाता है, तथा एक ही Screen पर एक से ज्यादा Data Base से Data Display कराया जा सकता है। एक बार Screen बनाने के बाद FoxPro से Screen के Code को Generate कराया जा सकता है।

SCREEN BUILDER को Start करने से पहले Data Base File को Open किया जाता है। उदाहरण के लिए, यदि हम EMPLOYEE .dbf Data Base File के लिए Entry Screen बनाना चाहते हैं, तो सर्वप्रथम हम इस Data Base File को Open करेंगे।

USE EMPLOYEE

CREATE SCREEN

उपरोक्त Command द्वारा हम Screen को Create कर सकते हैं। इसके अलावा FoxPro Menu System से File, New और Screen को Select करके भी Screen Builder को Start किया जा सकता है। जो भी Screen आप Design करते हैं वह Screen File में Save होती है।

SCREEN BUILDER के Start होते ही Screen पर एक Screen Design Window Display होती है। Window Title पर UNTITLED .Sex Display होता है। यह Screen Design File का नाम है। यदि आप Screen Design, File Name के साथ करते हैं तो Window Title पर उससे Related File Name Display होता है।

**(2) Designing a Custom Screen Using @ Command:** - FoxPro में @ ....SAY और @ .... SAY .... GET Command की एक Series, Special Type Data entry Screen बनाने के लिए Use होती है। इसके लिए एक Format File बनाने की आवश्यकता होती है। इस File में निम्नलिखित Contents होते हैं —

1. Screen के किसी भी भाग पर कोई भी Message Display करने के लिए @ ... SAY Command की Series Use होती है।
2. @ ..... SAY ..... GET या @ .... GET की एक Series Screen पर इच्छित स्थानों पर Fields के Display करने के लिए Use होती है।

3. @ .... To Command Screen के किसी भी भाग को Box में Enclose करने के लिए प्रयोग की जाती है। @ 05, 07 To 17, 15 DOUBLE.
4. @ ..... SAY Command FoxPro का कोई Function भी अतिरिक्त Information देने के लिए कर सकती है। जैसे – DATE Function etc.

**(3) @ ..... edit Command-** इस Command का प्रयोग अधिकतर हम Long Character Field Variable या Memo Field के लिए करते हैं। इसके प्रयोगानुसार Editing region के चारों तरफ एक line दिखाई पड़ती है, जो Text Box का Look प्रदान करती है। इसका प्रयोग हम निम्न प्रकार सकते हैं –

@ 5, 6 EDIT Address size 3.20

**(4) Designing a Custom Screen with Create Screen**- FoxPro Screen ही User के लिए Program Interface को पूर्ण रूप से तैयार करने के लिए Create Screen Command को उपलब्ध कराती है।

USE Address

CREATE Screen Company add,

इस Command के प्रयोग से एक Empty Screen Window Display होती है। इसको Report की तरह ही Design कर सकते हैं तथा FoxPro द्वारा यह .SCX विस्तारक नाम के साथ Save करता है।

- **Check Box** - Check Box का सबसे Important work यह होता है कि इसके प्रयोग द्वारा हम Information को Easily Select कर सकते हैं। इसको हम Main Window में Design कर सकते हैं। इसके सफल प्रयोग के कारण हम Information को select कर Store कर सकते हैं। हम एक या एक से Information का भी Selection कर सकते हैं। इसका **Syntax** निम्न प्रकार है –

@ ROW. COLOUMN GET <VARIABLE NAME> FUNCTION “\*C<OPTION”

- **Radio Button** - यह Button Check Box से मिलता जुलता ही होता है, परन्तु इन दोनों की कार्यविधि में अन्तर होता है। Check Box के सारे Options को Select किया जा सकता है और Radio Button में केवल एक ही Button को Select कर पाते हैं। यही इन दोनों में Main Difference है। इसका **Syntax** निम्न प्रकार है –

@ROW, COLOUMN GET<VARIABLE NAME>FUNCTION’ \* <OPTION1>’, <OPTION2>

- **Push Button** - Push button को हम Command button कह कर भी Define कर सकते हैं। इस Button का कार्य काफी Important हो जाता है, क्योंकि यह Click होते ही Active हो जाता है।

**(ii) FoxDoc** - FoxDoc, FoxPro की एक Built in Unit है जिसका Use Program Files and Files के द्वारा Use की जाने वाली other files को Document करने में किया जा सकता है। यह Automatically DO WHILE \_ END DO, IF \_ END IF, SCAN \_ END SCAN आदि को Indent कर देता है तथा Diagram में प्रदर्शित lines को Draw कर देता है। अतः यह आपको छूटे हुये या गलत जगह पर लगे हुये Loops को Search करने में Help करता है।

EMPLOYEE.PRG Program का Document, FoxDoc का Help से निम्न प्रकार करेंगे –

**(1)** FoxPro Menu System में Program Menu को Open करें तथा उसमें से FoxDoc को Select करें। FoxPro, FoxDoc के लिए Title Screen Display कर देगा तथा आपसे Enter Key को Press करने के लिये कहेगा।

**(2)** Enter Key Press करने पर FoxDoc की Opening Screen Display होगी।

**(3)** इस Screen पर आपको Program से Related सभी Detail देने हैं जो आप Document में चाहते हैं।

**(4)** आगे आपको Main Program /Project नाम को निर्धारित करना चाहिए।

**(5)** FoxDoc का Last Part Program and Data Files और इसके द्वारा बनाई गई Stored Files को Specify करना होता है।

**(6)** FoxPro की Document Process Start करने के लिए Generate को Select करें। FoxPro Application के द्वारा Use की जाने वाली Files को Check करता है, यदि कोई गलती है तो उसे Screen पर प्रदर्शित करता है। फिर आपसे जारी

रखने के लिए Spacebar Press करने को कहता है तथा Spacebar Press करने पर हर Files को Disk पर लिखना Start कर देता है। आप इन Files की Printing भी ले सकते हैं।

FoxPro सारी Files को Modifies करता है तथा Program Files को Modify करने के अलावा FoxDoc एक .ACT File को भी बनाता है, जिसमें line number सम्मिलित होते हैं और Loops को Add करने के लिए Line Draw करता है।

FoxDoc एक बहुत ही Useful Built in utility है। यह Document Purpose के लिए बहुत Useful है।

**(iii) UDF:** - UDF का Full Form User Defined Function है। ये Function User के द्वारा Define किये जाते हैं। FoxPro में लगभग 200 से अधिक Built in function है। कभी-कभी आपको अपने Program के लिए कुछ Functions बनाने की आवश्यकता होती है, इन्हीं Functions को UDF कहते हैं। प्रत्येक Function एक Value Return करता है। Function Defined करने का **Syntax** निम्न है –

```
FUNCTION <function name>
PARAMETER<parameter list>
Command 1
Command 2
Command 3
RETURN<exp>
```

**Ex:** - हम CUBE नाम से एक UDF बनाना चाहते हैं जो कि संख्या का CUBE Calculate करता है।

```
FUNCTION CUBE
PARAMETER NUM
RETURN NUM*NUM*NUM
```

उपरोक्त Function किसी संख्या का Cube Calculate करेगा।

? CUBE (3)

**(iv) SQL:** - SQL का Full Form Structural Query Language है। SQL एक Standard व Developed Language है जिसका Use Main Frame Computers में Relational Data Base पर कार्य करने के लिए होता है। जो लोग SQL से परिचित होते हैं, वे Necessary Selected Instructions को सीधे ही Insert करके, RQBE Window में जाये बिना सीधे ही Result प्राप्त कर लेते हैं। इस Language में Mainly SELECT Keywords वाली Command का प्रयोग होता है। RQBE एक SQL Select Instruction का निर्माण करती है। जिसका Use FoxPro हमें Result देने में कराती है। यह आपको Selected Data Base Files में Necessary Area का Selection, Result या Output file के लिए करने में Help करती है। इससे बढ़कर यह Output Records को Data Base Files में लिखने से पहले एक Browse Window में देखने की सुविधा प्रदान करती है।

**Ex:** - SELECT \* FROM EMP WHERE EMP \_ CODE = "D10"

**SQL** - निम्न Command सभी RDBMS में दी जाती है –

- (i) CREATE, DROP, ALTER Table
- (ii) INSERT, UODATE, DELETE Records के लिये।
- (iii) GRANT, REVOKE Order के लिये।
- (iv) SELECT QUARY के लिये।

**SQL के लिये Rules :-**

- (i) SQL एक Verb के साथ Start होता है। **Ex:** - SELECT, FROM
- (ii) प्रत्येक Verb, Clause की संख्या को दर्शाता है।
- (iii) SQL Word Semi Colon (;) के लिये Use होते हैं।

**SQL के गुण (Nature): -**

- (i) यह एक Non-Procedure language है।
- (ii) यह English की तरह एक Language है।
- (iii) System के रख-रखाव व निर्माण में लगने वाले समय में कमी रहती है।

(iv) इस Language का Use वे भी कर सकते हैं, जिन्हे Programming Expressions नहीं होता है।

**SQL \* PLUS:** - यह SQL एक Verb Language है, इसमें ज्यादा से ज्यादा 100 SQL Command दे सकते है। SQL एक SQL \* PLUS का Sub-Set है।

**Q.:** - Explain the following: -

**(A) @ ..... EDIT:** - @ ..... Edit Command @ ..... GET Command की तरह ही है। यदि आप Long Character Field/Variable या Memory Field में Data Enter करना चाहते हैं तो @ ..... GET के स्थान पर @ .....EDIT Command का Use valid माना जाता है। यह Command एक Rectangular Editing Box Editing के लिए screen पर Display करती हैं।

Text Editing Region (Height, Weight) का Size, SIZE Clause के साथ Specified होता है।

**Ex: -**

USE PRODUCT

@ 5, 6 EDIT DESC SIZE 3, 20

दूसरी Command Screen पर एक Text Editing Window को Create करती है। Screen पर 5, 6 Coordinate पर Editing Region बनता है। Window PRODUCT .dbf के DESC Field के Content को Screen पर Display करती है।

**(B) @ .....SAY .....GET .....READ:** - @ .....SAY ..... GET और READ के साथ हम कई Options का Use कर सकते हैं। @ .... SAY ..... GET and READ दोनों Commands सभी Users द्वारा FoxPro के Interface के लिए प्रयोग की जाती है।

@ .....SAY .....GET Command के साथ आप Field या Variable name specified कर सकते है। GET Command Variable की Current Value को Display करता है। Command GET Variable को Active करता है और आप इसमें New Data Enter कर सकते हैं।

**Ex: -**

CLEAR

STORE "Prakash Dwivedi" TO NAME

@ 5, 5 SAY "Enter New Name" GET NAME

उपरोक्त Command के Active होने पर आपको Screen की 5<sup>th</sup> Row and Column पर निम्नलिखित message Display होगा -

Enter New Name Prakash Dwivedi

यदि आप GET Variable में Data Enter कराना चाहते हैं तो आपको READ Command का प्रयोग करना होगा।

**Ex: -**

STORE "Prakash Dwivedi" TO NAME

CLEAR

@ 5, 5 SAY "Enter Your Name"

@ 5, 20 GET NAME

READ

? NAME

FoxPro Command Window और FoxPro Menu System को Remove कर देता है और निम्नलिखित Information को Screen पर Display करता है -

Enter First Number: 5

Enter Second Number: 10

15

**(C) @ .....GET:** - @ ..... GET Command का प्रयोग User से Data Get करने के लिए किया जाता है। ये Command READ Command के साथ Fields, Memory Variable या Array Element में Data Input करती है।

@ ... SAY की तरह @ ...GET भी कुछ PICTURE Clause और Code का प्रयोग करता है।

Ex: -

```
SAORE SPACE (10) TO N1, N2
@ 5, 0 SAY "Enter Name:" GET N1 FUNCTION "!"
@ 7, 0 SAY "Enter Name:" GET N2 PICTURE: "@"
READ
```

“!” function Code और “@!” PICTURE CODE आपके द्वारा Enter किए गए Text को Upper Case में Change कर देती है।

आप @ ....GET Variable के साथ RANGE Clause का प्रयोग Acceptable Value की Range को Specify करने के लिए भी प्रयोग कर सकते हैं।

```
CLEAR
NUMBER = 1
@ 5, 0 SAY "Enter Number (1 - 500):" GET NUMBER
RANGE 1,500
READ
```

FoxPro आपको Value Enter करने के लिए Prompts देता है। यदि आप Range के बाहर Value Enter करते हैं, जैसे कि 600 तो FoxPro निम्न Message Display करता है –

RANGE 1 TO 500

यदि आप चाहें तो GET Variable के लिए DEFAULT Value भी Assign कर सकते हैं –

```
CLEAR MEMORY
@ 5, 10 SAY "Enter Any Number" GET N3 DEFAULT 10
READ
```

@ ....GET Command साथ प्रयोग होने वाले कुछ FUNCTION CODES निम्न हैं –

Code	Purpose
A	Allow alphabetic character only (no space no symbol)
B	Left Justifies Numeric Data within the output field
C	Display Numeric Data in a Current Format
D	Use the Current SET DATA Format

@ ....GET Command साथ प्रयोग होने वाले कुछ PICTURE CODES निम्न हैं –

Code	Purpose
a	Allow Only Digits in character Data
!	Converts lower case letter to upper case
.	Specifies the Decimal position
#	Allow Digits blanks and signs.
X	Allow any character

(D) @.... **CLEAR**: - @..... Clear या @... CLEAR To Command के द्वारा आप Screen के Specified भाग को Clear कर सकते हैं। इस Command का प्रयोग करने से पहले आपको Screen को EMPLOYEE.dbf File (अथवा कोई भी file) के Record से भर लेना चाहिये।

```
USE EMPLOYEE
LIST FIELDS EMP CODE, NAME, CITY
LIST FIELDS EMP CODE, NAME, CITY
```

**Guided by:** *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

अब @ 15, 25 CLEAR Command को Enter करते है।

Fox Pro Screen पर 15<sup>th</sup> Row से Right Side 25<sup>th</sup> Column तक सभी Text को Clear कर देता है।

@ 15, 0 CLEAR

@ 5,5 CLEAR TO 9,15

पहली Command 14<sup>th</sup> Row के नीचे के सभी Row के Text को Clear कर देती है। दूसरी Command में Clear To का Use करके Specified Region का Text Clear कर देती है।

### ➤ **RQBE: -**

इसका Full form Relational Query By Example है। FoxPro में Query एक Important Property होती है। Data Base में Store Data के बारे में FoxPro से पूछे गये प्रश्नों को Query कहते हैं। जिनका Answer FoxPro Records को Present करके देता है। FoxPro में Query RQBE द्वारा बनायी जाती है।

RQBE का उद्देश्य Data Base Files से चुने गये Records and Fields को आवश्यकतानुसार प्रस्तुत करता है। RQBE की सहायता से Query करना अत्यन्त सरल होता है।

### Need of QUERY: -

LIST, DISPLAY अथवा BROWSE Command किसी भी Data Base File के सभी Contents को Screen पर प्रदर्शित करती है, परन्तु Query के द्वारा Records को Filter किया जा सकता है। तथा आवश्यकतानुसार जिन Records की जरूरत है उन्हें Query करके Screen पर प्रदर्शित किया जाता है। इसके अलावा Data Analysis के लिये Query को Graph के रूप में भी प्रस्तुत किया जा सकता है।

**Query तैयार करना: -** FoxPro में Query तैयार करने की निम्नलिखित दो विधियाँ है –

(1) File Menu के New Command द्वारा अथवा

(2) Command Window के Create Query Command के द्वारा।

File Menu के New Command पर Click करने पर Dialog Box Display होता है जिसमें Query Option पर Click करके New Button पर Click करते है।

Command Window में CREATE QUERY Command द्वारा Query तैयार की जाती है। FoxPro में तैयार Query एक File के रूप में Store होती है, जिसे Query File कहते है। इसका Extension .QPR होता है।

FoxPro में जिस Data Base File की Query बनानी है उसे Open कर निम्न Command को Command window में देते है –

CREATE QUERY <filename>

उपरोक्त Command देने पर एक Dialog Box Display होता है। जब RQBE Window Active रहती है, तब Main Menu bar पर एक New Menu RQBE Display होता है।

RQBE Window में विभिन्न भाग होते है जिनमें से Table वाले भाग में प्रयोग की गई Data Base Files की list Display होती है। Output fields वाले भाग में Data Base File के सभी Fields Display होते हैं तथा इनके साथ बने हुये चारों Check Box में से Field वाला Check Box चुना हुआ होता है। यदि हम इस Check Box पर Click करते हैं तो RQBE Select field नामक Dialog Box Window पर प्रदर्शित होता है। इस Dialog Box में यह निर्धारित किया जाता है कि Query किन Field के आधार पर की जानी है।

Order by Check Box पर Click करके Query का क्रम निर्धारित किया जाता है। किसी भी field के आधार पर Records को Ascending अथवा Descending Order में Arrange किया जाता है।

Selection Criteria वाले भाग में विभिन्न fields को शर्त दी जाती है जिस पर कि Query की जानी है। Output list Box में Query का Output किस रूप (Browse, Report, Table/DBF, Graph, Cursor) में चाहिए, दिया जाता है।

Last में Do Query Button पर Click करके Query को पूर्ण किया जाता है।

**Guided by: Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)**



**Q: -Write a Program in FoxPro to Design a Mark sheet of a student using Custom Screen.**

**Ans: -**

FoxPro की Command Window में Modify Command Mark Sheet .fmt को Type करते हैं इसके पश्चात् निम्नलिखित Code को लिखते हैं –

Clear

@ 2, 10 Say "Student Roll No." GET Roll \_ No

@ 4, 10 Say "Hindi Marks" GET H

@ 6, 10 Say "Math Marks" GET M

@ 8, 10 Say "Science Marks" GET S

@ 10, 10 Say "English Marks" GET E

@ 12, 10 Say "Computer Marks" GET C

Total = H+M+S+E+C

AVG = Total /5

If AVG > = 75

? "First Division"

Else if AVG > = 65 and AVG < 75

? "Second Division"

Else if AVG > = 50 and AVG < 65

? "Third Division"

Else

? "Fail"

End if

**R S I T Computer College  
Karkeli, Umaria (M.P.)**

**PGDCA, DCA, ADCP, CPCT  
And other 8<sup>th</sup> pass Certificate  
Courses**

प्रकाश द्विवेदी (8982505087)

अभिलाष पाठक (8517906324)

**Website: [www.rudrasoftech.in](http://www.rudrasoftech.in)**