1. **Data Type**
2. **Variable Declaration using Data type**
3. **Operators**
4. **Assigning Values to Variable using Assignment , Expression and Operator**
5. **Conditional Statement**
6. **Repetition Statement**
7. **Functions**
8. **Built-in Useful Functions**
   1. **Searching Record**
   2. GET , FIND , FINDFIRST, FINDLAST,FINDSET, NEXT,
   3. **Sorting and Filtering Functions**
   4. Setcurrentkey , setrange, setfilter, getrangemin , getrangemax
   5. **Inserting , Updating, deleting**
   6. Insert , modify , modifyall, delete,deleteall
   7. **Transaction**
   8. Locktable
   9. **Field Function**
   10. Calcfield , calcsums, fielderror, fieldname, init , testfield , validate
   11. **User Communication Function**
   12. Message , Confirm, Strmenu, error
   13. **String Function**
   14. Strops, copystr, padstr, strlen,maxstrlen, lowercase, uppercase , convertstr,delchar,incstr,selectstr,strchecksum
   15. **System Function**
   16. Userid , companyname , today, time, workdate, getlasterrortext, clearlasterror, applicationpath, temporarypath
   17. **Date Fucntion**
   18. Date2dmy, Date2dwy, CalcDate, NormalDate , Closingdate,
   19. **Number Function**
   20. Abs, power, round , randomize
   21. **Array Functions**
   22. Arraylen , compressarray, Copyarray
   23. **Other function**
   24. EXIT, Clear, ClearAll, Evaluate , Format

**Development**

**Table**

Table is the most fundamental part of NAV 2009. Table is used to store data in database.

1. **Table Description**

Table Properties

Table Triggers

**Fields**

Fields Properties

Fields Triggers

**Keys**

Keys Properties

1. **Table Relationship**

Table Relation is setup using Table relationship property.

**Table Relationship with filter**

Salesperson table is primary table and Sales Transaction is Foreign Table

Select the foreign key in table 🡪 Properties 🡪Select the primary table in table column and set the filter in table filter column

**Table Relationship with Condition**

Open the table relationship property 🡪 table condition column set the conditionAnd table column select the required table.

1. **Special Table Field**

**Flow Field**

Flow Field is not part of table but it calculate values from other table or tables. To update flowfield we are using CALCFIELDS Function { record.calcfields() }

To create special field in primary table, create one column, set the column flowclass property to flowfield. In Calculation Formula choose sum and in table select foreign table and its column

In table Filter option choose salesperson from foreign table and code value from primary key

Not create sumindexfield in foreign tableIn keys create key which is primary key in primary table and enter field in sumindexfiled from secondary table

**Flow Filter**

Put the column in Key from source table (sales transaction) , then create one column in destination table with fieldclass to flow filter and in option string type the required value. Then put the value in flowfield value in culc field in next tine

**Practical Chapter No. 1**

1. **How to Create Table**

Open Object Designer 🡪 Click Table🡪 Click New 🡪

Enter Field No. (**10**) 🡪 Field Name (**No.**)🡪 Data Type (Code)

Enter Field No. (**20**) 🡪 Field Name (**Description**) 🡪Data Type (**Text**) 🡪 Length (**50**)

Enter Field No. (**30**) 🡪 Field Name (**Amount**)🡪 Data Type (**Decimal**)

Enter Field No. (**40**) 🡪 Field Name (**Type**)🡪 Data Type (**Option**)🡪 Property 🡪 String Option property (**G/L Account, Item, Resource**)

Enter Field No. (**50**) 🡪 Field Name (**Sales Person Code**) 🡪 Data Type (**Code**)

Save the Table with No. (90001) Name ( Sales Transactions ).

In this table Sales Person Code is Foreign Key and have relationship with Salesperson / Purchasers Table

**Setting Up Primary Key**

Select the Field No. ( **No.** ) 🡪 View Menu 🡪 Keys 🡪 Enable (**Yes**) 🡪 Key ( **No.** )

**Setting UpForeign Key**

Select the Field No. (Type )🡪 View Menu 🡪 Keys 🡪on second line 🡪 Enable (**Yes**) 🡪 Key ( **Type** )

1. **Setting Up Relationship between table**

**4.1 Filter Type of Relationship**

Select the Column for example ( **Sales Person Code**) 🡪 Property 🡪 Table Relationship 🡪F6 🡪in Table Column select table i.e. ( **Salesperson / Purchasers**) 🡪in Table Filter column🡪 F6 🡪 Field (**Commission %** ) 🡪 Type (**Filter** ) 🡪 Value ( **>0** ) 🡪 Save this

**4.2 Conditional Type of Relationship**

Select the Column for example (No**.**) 🡪 Property 🡪 Table Relationship 🡪F6 🡪in Condition Column 🡪 Field ( **Type** ) 🡪 Type (**CONST** ) 🡪 Value ( **G/L Account** ) 🡪OK 🡪in Table Column ( **G/L Account** )🡪repeat for all the condition 🡪Save this

1. **Creating Special Field in Table like Flowfield and FlowFilter**

**FlowField Creation**

Create a new field in Primary table like Saleperson / Purchaser table 🡪 Field No. ( 50000 ) 🡪 Name ( Sales ) 🡪 Property 🡪select Field Class property ( FlowField ) 🡪CalcFormula🡪 F6 🡪 Method ( SUM ) 🡪Table ( Sales Transcations ) 🡪 Field ( Amount ) 🡪 Table Filter 🡪 F6 🡪Field ( Salesperson Code ) 🡪 Type ( Field ) 🡪 Value ( Code )

Open the Sales Transaction Table in Design 🡪 View 🡪 Key 🡪 Against Salesperson Code Foreign Key 🡪SumIndexfields( Amount )

**FlowFilter Creation**

Create a new field in Primary table like Saleperson / Purchaser table 🡪 Field No. ( 50001 ) 🡪 Name ( Sales Type ) 🡪 Property 🡪 Field Class ( FlowFilter ) 🡪open OptionString Property 🡪 Enter the Option ( G/L Account , Item, Resource ) 🡪 Close property 🡪 Open Sales Field (50000) Property 🡪CalcFormula🡪 F6 🡪 Method ( SUM ) 🡪Table ( Sales Transactions ) 🡪 Field ( Amount ) 🡪 Table Filter 🡪 F6 🡪 Leave the old entries 🡪 Enter new line 🡪Field ( Type ) 🡪 Type ( Field ) 🡪 Value (Type Filter (50001))

Open the Sales Transaction Table in Design 🡪 View 🡪 Key 🡪 Against Type Foreign Key 🡪SumIndexfields( Amount )

**CHAPTER NO. 2**

**FORM CREATION**

**FORMS**

Form Properties

Form Trigger

Controls

Controls Properties

Controls Triggers

1. **Simple Form Creation**

Open Object Designer 🡪 Click form -> Click New 🡪 Create a blank form 🡪 in form property in source table property assign table Name Like ( ITEM )🡪 Save the form with ID ( 90001 ) 🡪 Name ( Sales Transactions ).

1. **Creating Text Box on Form and Assign to Description Field**

Open the Tool Box 🡪 Select Text Box 🡪 Place on On Form 🡪 Open property for control 🡪 Source Expr🡪 F6 🡪 Rec 🡪Field Name 🡪”No.”🡪 OK

1. **Adding Label Control to Form**

Open the Tool Box 🡪 Select Label 🡪 then click on Form 🡪 Select Lable🡪 SHIFT+F4 🡪 Set Caption Property to proper

1. **Adding Check Box Control to Form**

Open the Tool Box 🡪 Select Check Box🡪 then click on Form 🡪 Select Check Box Control 🡪 SHIFT+F4 🡪 Set Show Caption (Yes )🡪SourceExpr ( Print Statement) 🡪 Close Property

1. **Adding Tab Control to Form**

Open the Tool Box 🡪 Select Tab Control🡪 then Place on Form 🡪 Open Control Properties 🡪 Set PageNames(Enter Tab name here like General,Test,Frame)

1. **Adding Menu Button & Menu Item to Form**

Open the Tool Box 🡪 Select Menu Button 🡪 Place on Form 🡪 set the button Caption

To Add Menu Item to Button 🡪 Right Click Button🡪 Menu Items 🡪 Caption ( Card ) 🡪 Action ( RunObject) 🡪RunObject ( Form My Customer Card) 🡪 Open Menu Item Property 🡪 set RunFormLink ( No. Field(No. ) 🡪RunFormLinkType( Update )

1. **Adding Menu Button & Menu Item Lookup Form to Form**

Open the Tool Box 🡪 Select Menu Button 🡪 Place on Form 🡪 set the button Caption

To Add Menu Item to Button 🡪 Right Click Button🡪 Menu Items 🡪 Caption ( List ) 🡪 Action (LookupTable) 🡪 then open the Customer Table in Designer and change the lookup form from table properties.

1. **Creating Form and Sub Form**

Open the Tool Box 🡪 Select Sub Form 🡪 Put on the Form 🡪 Note the Size 🡪 Make new form based on another table 🡪 set the new form size to sub form 🡪 Then link the new form to sub form 🡪 Open Sub Form Proferty🡪 Set SubFormID (90004 ) 🡪SubFormLink (Table Name=CONST(Customer),No.=FIELD(No.)🡪 virtualization

1. IF amount > 10 THEN

Condition is true;

1. IFamount > 10 THEN

Condition is true ( no semicolon )

ELSE

Condition is false;

**C/AL Programming**

1. **Data Types**

**Simple Data Types**

**Numeric Data Type**

Integer

BigInteger( L is used to identify BigInteger)

Decimal

Char ( 0-255) (‘B’,’30’)

**String Data Type**

Text (0-1024) ( ‘123’,’Hello’) (‘Here”s string constant’)

Code (‘HELLO’) ( Right Justified)

**Boolean Data Type**

TRUE

FALSE

**Data and Time**

Date (Rang 01/03/0001 to 12/31/9999) (123197D)(mmddyyyyD)

**Complex Datatype**

BLOB ,RECORD, FORM , REPORT, CODEUNIT, RecordRef , etc

**Identifier, Variable, Syntax**

**Variable Scope**

Local

Global

**System Define Variable**

Rec pointing to current record

XRec pointing to previous record

**Variable Initialized Values**

Integer variable are initialized with zero

String variable are initialized with “

Boolean variable are initialized with FALSE

Date variable are initialized with 0D

After declaring variable using data type we will assign value to it using assignment statement and operators.

**Chapter No. 2 Assignment Statement**

1. **Assignment Statements**

Assign Value to a variable is called assignment. Type of value must match the type of variable.

**Expression, Term, Operator**

Operator work on operand and make term and term make expression

1. **Operators**

Unary Operators ( +,-,NOT)

String Operators +

Arithmetic Operators +,-,/,\*,DIV,MOD

Relational Operators >, < ,>=, <= , <>,IN

Logical Operators NOT,OR,AND,XOR

1. **Statements**

**There are many statements in NAV.**

1. **Condition Statement**
2. **Repetitive Statement**
3. **Compound Statement**

**4.1 Conditional Statements**

**CASE No. 1**

**IF** condition **THEN IF** Amount = 5 **THEN**

Statement; Amount: = -Amount;

**CASE No. 2**

**IF** Condition **THEN IF Quantity <> 0 THEN**

Statement UnitPrice :=TotalPrice / Quantity

**ELSE ELSE**

Statement; UnitPrice := 0;

**CASE No. 3**

**i := 2;**

**CASE** Expression **OF CASE i OF** ExpressionValue : 1:

Statement 1;Description := ‘January’;

ExpressionValue : 2:

Statement 2; Description := ‘February’;

**ELSE ELSE**

Statement n + 1; Message(‘i value is different’);

**END; END;**

**4.2 Repetitive Statements**

**FOR** control variable: = 1 **TO** 9 **DO FOR i = 1 TO 9 DO**

Statement; Message(‘%1’,i);

**FOR** control variable: = 9 **DOWNTO** 1 **DO FOR i = 9 DOWNTO 1 DO**

BEGIN BEGIN

Statements; Message(‘%1’,i);

END; END;

**WHILE** expression **DO WHILE i > 9 DO**

**Statements; BEGIN**

1. **I++;**

**REPEAT END;**

Statements;

**UNTILL** Condition**;**

1. **WITH** record variable **DO**

**Statements;**

**4.3 Compound Statements**

**BEGIN**

Statement 1;

Statement2;

end so on;

**END;**

**Comments**

There are two type comments

// single line comments

Multiline Comments

{

Line1;

Line2;

Line3;

}

**CUSTOM FUNCTIONS**

**PARAMETERS :** Piece of Information passed to Function is called Parameters.

**Two types of Parameter Passing**

**By Value**

Pass only variable Value

**By Reference**

Pass Variable actual location

**Some Use Full Functions**

**Searching Records**

Get / Find / Next Function

Get Function retrieve one record

Find

Find First , Find Last , FindSet

**Sorting & Filtering Function**

**SetCurrentKey**

Use to select a key for a record. Set the sorting order for table.

**SetRange**

Select the Range in TableRecord.SetRange(“No.”,’100’,’150’);

**SetFilter**

Work Same way like SetRange

**GetRangeMin**

Get the minimum value in Range

**GetRangeMax**

Get the maximum value in Range

**Inserting , Modifying , Deleting Record**

Insert

Modify

ModifyAll

Delete

DeleteAll

**Transaction related Function**

**LockTable**

**Table Field Related Function**

**CALCFIELDS**

Calcfields functions is used to update FlowFields

**CALCSUMS**

Calculate the sum of field which is used in sumindexfield

**FIELDEROR**

Check the field for error

**FIELDNAME**

Return the Field Name

**INIT**

Initialized the records

**TESTFIELD**

Test the field against some values

**VALIDATE**

Validate the values of field on validate trigger

**USER COMMUNICATION FUNCTIONS**

**MESSAGE**

**CONFIRM**

**STRMENU**

**ERROR**

**CUSTOM FUNCTIONS**

**Reports**

Report Print information from database

**Report Description**

Properties

Trigger

**Data Items**

Properties

Trigger

**Sections**

Properties

Triggers

**Controls**

Properties

**Request Form**

Property

Trigger

**Control**

Property

Trigger

**Request Page**

Property

Trigger

**Control**

Property

Trigger

**RDL Data**

**Report Design Process**

Logical Structure (Data Model)

Visual Structure

**What Happen when report run**

Complete Flow Chart on 636 pages.

1. On Init Report Trigger
2. Request.Form
3. Pre-Report Trigger
4. Get Next Data Item
5. Data Item. Run
6. Get Next Data Item
7. Data Item.Run
8. Post Report Trigger

Example

Create a report to display customer by salesperson

1. Open Customer table (13) and add salesperson code in key

**DataPort**

Dataport are used to import and export data to NAV

Dataport Description

Properties  
Trigger

Data Item

Properties

Triggers

Dataport Field

Properties

Trigger

Request Form

Properties

Trigger

Control

Properties

Triggers

**Integration Option**

**Integration Option**

* Web Services ( Use SOAP Protocol ) ( Web Services Description Language). VS 2008 is use to build Web Services.
* NAV ODBC
* C/FRONT
* OCX
* Automation

**Web Services Example**

**Step No. 1 Create a Code Unit**

Tool🡪 Object Designer🡪 Code Unit 🡪 New 🡪 Create a function with name ( Capitalize) 🡪 Assign Parameters Name ( InputString ) Type ( Text ) Length ( 250) 🡪 return value ( outputstring) type ( text) Length (250)

* Enter following code in function

🡪 outputstring := UPPERCASE(inputstring)

* Save Code Unit

**Step No. 2 Register Code Unit and a Page web service and publish them.**

Tool 🡪 Object Designer 🡪 Form 🡪 Web Services ( 810 ) 🡪 Run 🡪

Object Type ( Code Unit ) 🡪 No ( 90001 ) 🡪 Name ( Name ) 🡪 Publish ( Yes)

Object Type ( Page ) 🡪 No ( 21) 🡪 Name ( Customer ) 🡪 Publish ( Yes)

**Step No. 3 Check that Web Services are running**

Make Sure that Web Server services is running from web services 🡪 Open Explorer 🡪 Type URL 🡪 <http://localhost:7047/DynamicsNAV/WS/CRONUS%20International%20Ltd./services>

http://<Server>:<WebServicePort>/<ServerInstance>/WS/<CompanyName>/services.

**Step No. 4 Create Application and consume the two Web Services.**

**Open VS 2005 🡪**

**ERP General Functions**

* **ERP** Stand for Enterprise Resource Planning
* **ERP** integrate all the department across the company in single system using common database
* **ERP consist on following department**
  + **Finance**
  + **Warehouse**
  + **Purchasing**
  + **Sales**
  + **HR**
  + **IT**
  + **Manufacturing**
  + **Distribution**
* **Posting Group** create link between master data and G/L Account.
  + **General Business Posting Group** Used for to identify income statement accounts.
    - Customer
    - Vendor
    - G/L Account
  + **Specific Posting Group** Used for to identify balance sheet accounts
    - Customer
    - Vendor
    - Item
    - Fixed Asset
    - Bank Account
  + **VAT Posting Group**

**INTEGRATION OPTION IN NAVISION 2009**

Navision providing integration with following technologies

1. Web Services
   1. Use Web Services Description Language (WSDL)
2. ODBC
3. C/FORNT
4. OCX
5. Automation
6. Web Services

SOAP & Web Services Descriptive Language

When page, codeunit is added to web services table then it available to access.

Computer Name: NAV\_Server1

Web Services Port: 7047

NAV Server Instance :DynamicsNAV

Company Name : CRONUS International Ltd

Publish Page :MyCustomer

FOR i:= 1 TO arraylen(arr)

DO BEGIN

Arr[i] := I;

Message ( ‘display array values’);

END;

**How to declare and Assign Value to variable in Navision 2009**

Open Codeunit in designer 🡪 Click View menu then Global Variable 🡪 Name ( var ) 🡪 Data type ( integer ) 🡪 Close window and on onRun() Function 🡪**var := 10 ;**

Magrudy held seminar. Seminar has fixed duration and minimum & maximum no of participant. Each seminar is define as job.

Instructor : Each instructor is define as resource. Contact is associative with vendor

Particient : each participant is define as contact . contact is associative with customer.

****

**History**

Three Friend make a company ( PC& C ) Personal Computer & Consultancy and develop PC Plus Application. PC Plus was released in 1885. 1987 new version of PC Plus ( Navigator ) then in 1995 Navision Financial was released. In 2002 Microsoft bought the company.

**Project**

We will develop a project **International Community and Neighbors (ICAN).**

**Purpose:**

Collect Food from rich people and distribute to poor people

Track Donation ( Item, money, services ). If item then make inventory.

Track Donor , Volunteer , and Needy people

**Development**

Create Donor Table

Create Donor Card Page

Crate Donor List Page

Create Donor List Report

**Note:**

In Classic we are creating Card Form and related List Form to from Table Property

In Role Tailored we are creating List Page and related with Card Page in List Page Property.

To Assign Default Value Set Init Value Property option

**FOR LOOP Syntax Example**

**Steps:**

Declare Two Variables

**Variable Name Data Type**

i Integer

IntArray Integer ( Array of 10)

**FOR i := 1 TO ArrayLen(IntArray) DO**

**BEGIN**

**CODE GOES HERE**

**END;**

**FOR i := 1 TO ARRAYLEN(IntArray) DO**

**BEGIN**

**IntArray[i] := i;**

**MESSAGE('IntArray[%1] = %2',i,IntArray[i]);**

**END;**

**WHILE LOOP Syntax Example**

**Steps:**

Declare Two Variables

**Variable Name Data Type**

i Integer

IntArray Integer ( Array of 10)

**i := 1;**

**WHILE i <= ARRAYLEN(IntArray) DO**

**BEGIN**

**IntArray[i] := i;**

**MESSAGE('IntArray[i] = %2',i,IntArray[i]);**

**i := i+1;**

**END;**

**CASE Syntax Example**

**Steps:**

Declare Variables with Option Value

**Variable Name Data Type**

Color Option ( None, Red, Green, Blue)

**Color := Color::Green;**

**CASE Color OF**

**Color::None: MESSAGE('You Choose Red');**

**Color::Red: MESSAGE('You Choose Red');**

**Color::Green: MESSAGE('You Choose Green');**

**Color::Blue: MESSAGE('You Choose Blue');**

**END;**

**Searching for Customer and Update Name and Blocking Status (Option Value)**

// First set sorting order

CustomerRec.SETCURRENTKEY(Name);

// Filter for that customer

CustomerRec.SETRANGE(CustomerRec.Name,'Mia Santiago');

// if any record found then

IF CustomerRec.FINDFIRST THEN

BEGIN

// update customer name

CustomerRec.Name := 'Mia Santiago Updated';

// change the blocking status

CustomerRec.Blocked := 1;

CustomerRec.MODIFY;

MESSAGE('Customer record update');

END

ELSE

MESSAGE('NO Customer found');

**SEARCHING**

How to search for particular record

**GET**

We can use GET function to search for particular record

The GET function retrieves one record, based on the value of the primary key.

**Customer.GET(’4711’);**

**FIND**

We can also used FIND function to search for specific record. FIND function take care of filter.

**Set up first the sorting by using SETCURRENTKEY Function.**

**Setup some filter using SETRANGE Function**

**Then use FIND function with IF statement**

FIND(‘-‘) and FINDFIRST have same functionality

FIND(‘+’) and FINDLAST have same functionality.

**FINDFIRST, FINDLAST, FINDSET**

FINDSET is use to select group of record then loop through using REPEAT – UNTILL loop.

**NEXT**

NEXT Function is used with FIND function to go to next record. When there is no record NEXT function return zero value

**SORTING AND FILTERING**

How to sort record and for particular record

**SETCURRENTKEY**

SETCURRENTKEY function is used to set the sorting of record in table based on Key.

**SETRANGE**

SETRANGE function set filter on table record

CustomerRec.SETRANGE (“No.”,’10000’,’50000’);

SETRANGE function will remove all previous filter if it is using without first or last value. If last value is not mention then first value is consider last value as well.

**SETFILTER**

SETFILTER is more flexible then SETRANGE function. Syntex of SETFILTER is

CustomerRec.SETFILTER(“No.”,’>1000 & <> 50000’);