**Chapter objectives**

**1. WORK WITH OBJECTS AND MEMBERS USING CLIENT ACCESS AND RDI**

**2. APPLY AN EDIT Word FOR STUDENT NUMBER AND PHONE NUMBER IN STUDENTS**

**3. CREATE A MARKS FILE USING DDS**

**4. COPY MARKS DATA USING NATIVE AND SQL COMMANDS**

**5. ENTER ASSIGNED DATA INTO STUDENTS FILE**

**6. CREATE A JOIN OF STUDENTS AND MARKS USING SQL**

**7. CREATE A LOGICAL FILE SHOWING FEES OWED**

**8. CREATE A VIEW SHOWING FINES OWED**

**9. VIEW FIELD AND FILE INFORMATION IN ‘GREEN SCREEN’ AND RDi**

**Lab Requirements:**

**Printouts of STDNTMARKS and FINESOWED views and FEESOWED compiled listing**

**You may be asked to demonstrate your newly created objects**

**Questions answered on this handout**

Start an RDi Session.

Start a Client Access Session.

**Part A**

**Objectives:**

* **Working with Objects in RDi and Client Access**
* **Edit Codes and Edit Words**
* **Create a Data File called MARKS using DDS**
* **Copy Data from STUDMARKS into MARKS**
* **Enter Student data into the STUDENTS physical file using DFU**

At the command line in Client Access type the following command

WRKOBJPDM YOURLIB (use your library name here)

|  |
| --- |
|  |

There are several things you can do at this screen. You can use PDM shortcuts beside any object type SP (which is a shortcut for WRKSPLF) and press enter. Exit back to the WRKOBJPDM screen and look for one of your working programs. Type a C (which is a shortcut for Call) beside it.

What command does WS provide a shortcut to? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

On the WRKOBJPDM screen, change the Library at the top from your library to IBC233LIB and press enter.

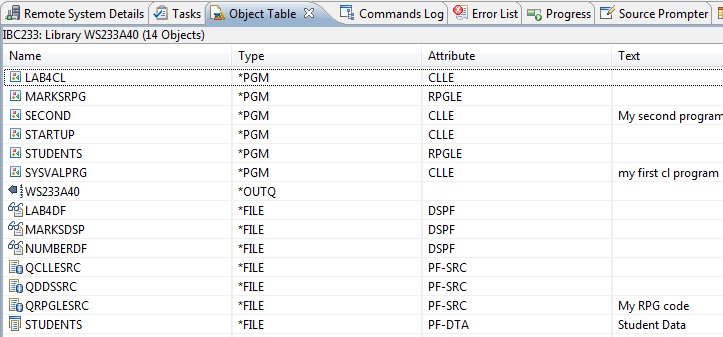
Press F9 to find your WRKOBJPDM command and prompt it. Can you limit your objects to source physical and data files?

How? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Programmers liked the PDM interface that allows WRKOBJPDM and WRKMBRPDM. They asked the RDi developers to come up with the same look and feel.

When trying to distinguish between STUDENTS \*PGM and STUDENTS \*FILE the Remote Systems view adds a period and object type. We can get more separation between these with a WRKOBJPDM like command.

Right Click on your library name and select the Show in Table option and then double click on the Tab called Object Table. On this screen you can easily add text comments for various objects similar to the ones showing below – do it.



Click on the Text column title. Your objects should be sorted by Text Comment. Try sorting according to Type and Name by clicking on the appropriate column title.

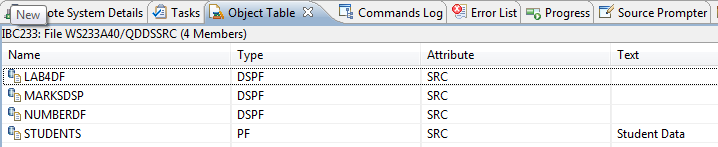
What is the distinction between a \*File that holds source code for compiling and a \*File that holds data like names and addresses?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Go back to your Client Access session. Run the WRKOBJPDM command to look at your library and type a 12 beside QDDSSRC.

What command does option 12 run? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Switch to RDi and use the Show in Table feature on QDDSSRC to look at your source members.



Double click on the STUDENT member to view the DDS code that was used to produce the STUDENTS data file (PF-DTA).

What is the difference between a column heading of COLHDG(‘Fines’ ‘Owed’) and COLHDG(‘Fines Owed’)

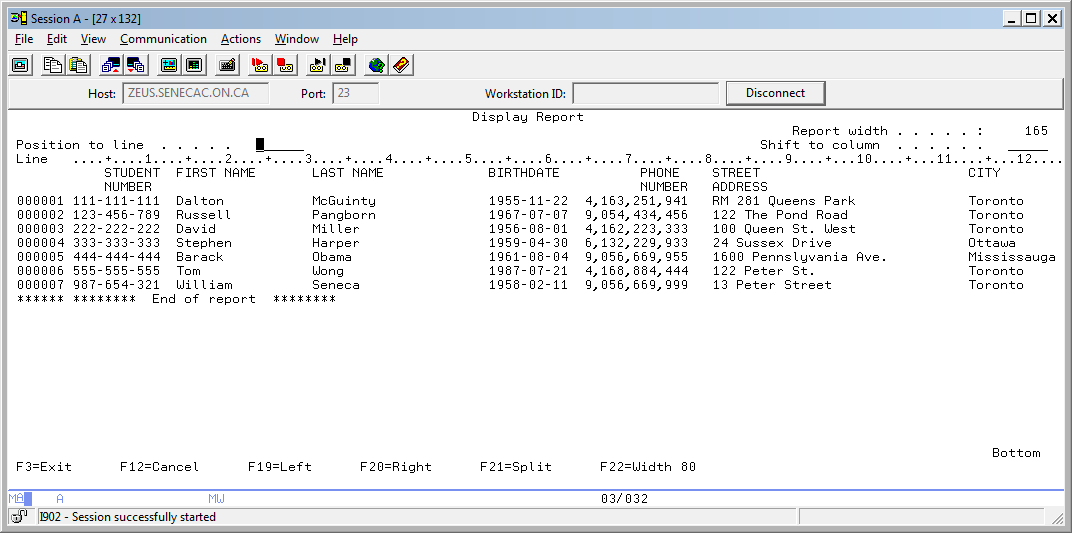
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

How is a key field denoted in DDS? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

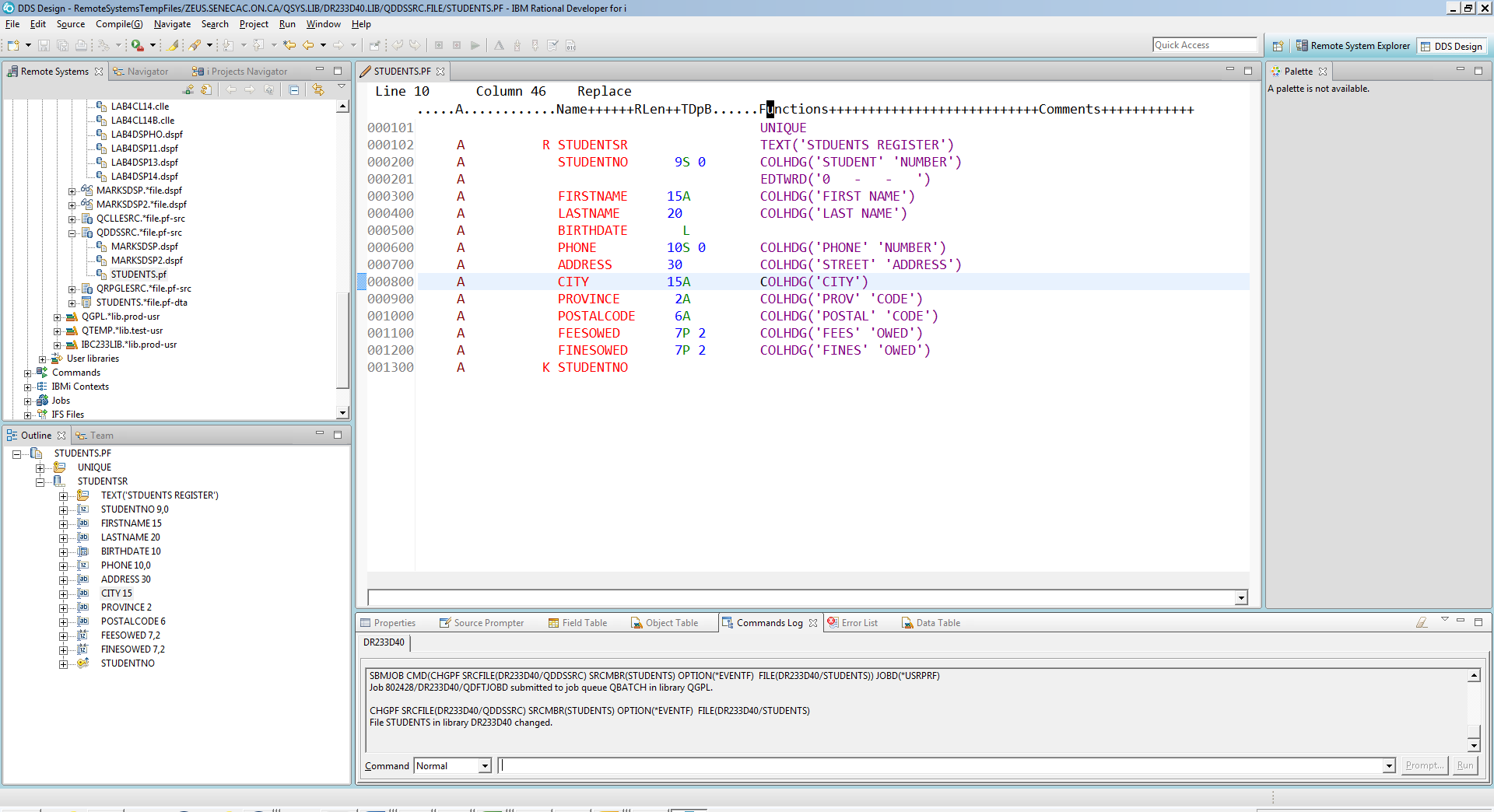
What is the field type for the following:

Date field \_\_\_\_\_\_ Zoned Decimal \_\_\_\_\_\_ Packed \_\_\_\_\_\_

We want the student number to appear with dashes making it more readable 999-999-999.



This is achieved by using an edit word.



A STUDENTNO 9S 0 COLHDG('STUDENT' 'NUMBER')

A EDTWRD('0 - - ')

Note that a zero is used to turn off leading zero suppression. This allows a student number of 000000001 to have all the zeroes print. Usually with numbers we want to supress leading zeroes so that is the default with an edit word.

In our edit word a blank is used to for each digit. It is hard to see the blanks so if we used a b you would see ‘0bbb-bbb-bbb’ Just count the blanks using your cursor and the space bar or arrow key – do not type the b’s.

The dash is literally printed so we could have put an ‘a’ there and the number would show as 999a999a999.

When you have added the line to show the student number with dashes you can do one of two things.

1. Run the CHGPF (command) in RDi – Change Physical File.

Confirm you did not have a syntax error.

Confirm the STUDENTS file was replaced by checking the command log. Sometimes your compile works but the commands log will tell you that the file was not changed!

This technique allows a change without loss of data already in the STUDENTS file.

1. In Client Access remove the file by running DLTF STUDENTS at the command line.

Then in RDi run the CRTPF command to recreate your STUDENTS file.

Your properly edited student number will show up in Client Access with the following command:

===>RUNQRY \*N STUDENTS

The Show in Table feature in RDi does not pick up the editing, so you will see a student number with commas inserted.

Since you are using blanks to represent each digit, you need a way to represent an actual blank. If you want a blank to appear in an edit word, use the ampersand symbol “&”.

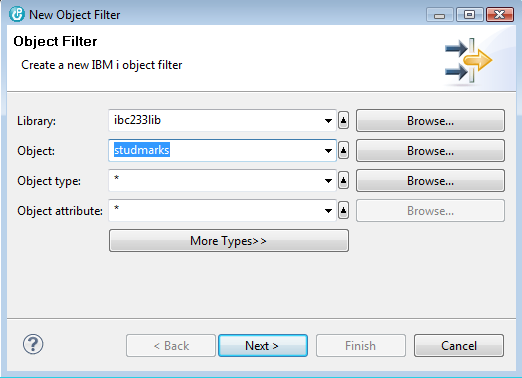
Change the phone number to appear as (999) 999-9999 with an edit word. What compile command will make the adjustment without losing the original data?

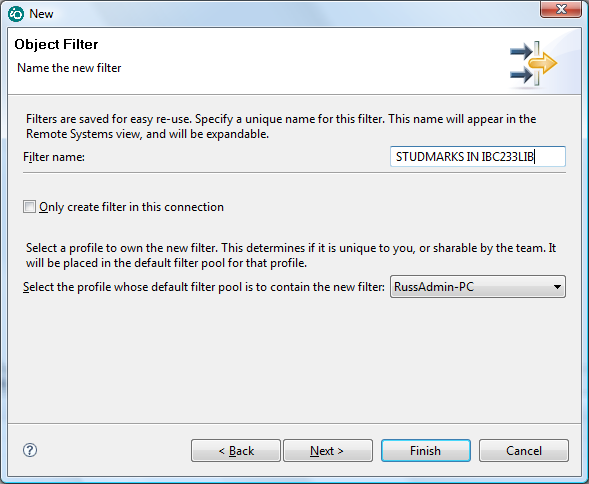
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Hint. Each blank is used for each telephone digit. You need to start your edit word with a 0 to turn off the default of zero suppression. If a telephone number needed an insertion character to start it off like a “(“ it would be suppressed unless you turned off zero suppression. Since a blank is used for each digit you use an ampersand “&” to represent any blank insertion characters.

Test the phone number change with the RUNQRY command in Client Access.

Set up an object filter called STUDMARKS IN IBC233LIB:





Use the show in table option to look at the data and then to look at the fields.

What is the field information?

NAME RECORD TYPE LENGTH TEXT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

Use this as a reference to produce your own file called MARKS by coding the appropriate DDS in QDDSSRC.

Everything should match except the following changes:

Record Format Name is MARKSR

Your STUDENTID field is Packed Decimal

Add a DATE field called DATERUN (Intended as the date the subject was run - it will pick up a default current

date during this lab)

Allow for column headings to show as follows with the RUNQRY command:

Student Course Final Date

ID Code Mark Run

222,222,222 IBC233 79 2013-02-10

222,222,222 DBS201 66 2013-02-10

The file should be sorted by Course code within Student ID.

(note the RUNQRY command will not pick up the sorting order – the UPDDTA command will get the specified

key field sorting order)

Use the CPYF command to copy over data from STUDMARKS in IBC233LIB. Prompt CPYF and pay attention to the following parameters:

The from file and to file names and respective libraries

Replace or Add records should be set to REPLACE to get the data

You need record field mapping since there is one additional field. - Use F10 and page down to find this

Try both the RUNQRY and UPDDTA command to check out the sorting order of the records.

Remove the data in your file with the CLRPFM (Clear Physical File Member) command.

Using the STRSQL command to get to the interactive SQL interface type the following command and press the prompt key:

INSERT INTO MARKS

You want to select Y for “**Select fields to insert into**” and 2 for “**Subselect**”

On the next screen you would specify the appropriate fields. On the Subselect screen you need to refer to the qualified name for locating STUDMARKS and you can put your cursor in Select Fields and prompt to specify them in the same order as previously indicated.

See if you can get the data into your file with this technique.

You will need to type some data into the STUDENTS file to match the MARKS file.

This will give you some practice with DFU. If you have not done so, exit from the interactive SQL interface and enter data into the STUDENTS file using DFU.

The function keys you need to take note of are the following:

F23 - Delete a record

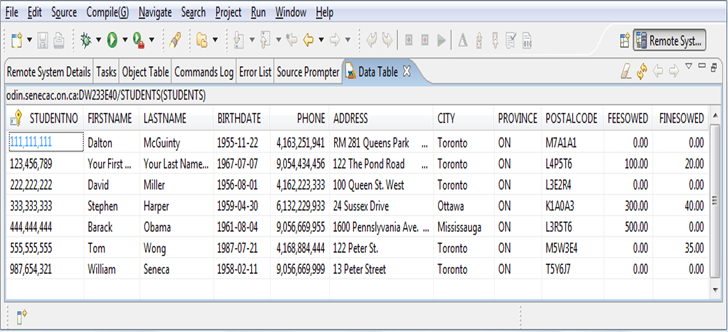
F9 - Insert a new record

F10 - Enter a new record

F11 - Change existing records

Remember UPDDTA does not show a decimal point. Your Fines Owed and Fees Owed should agree with the data that is included here. If it doesn’t, the printouts you hand in will be rejected.

Match the following data and make sure your name is in the appropriate record.



**Part B**

**Objectives:**

* **Create a join of Students and Marks using SQL**
* **Create a Data File called MARKS using DDS**

Using ‘Green Screen’ and interactive SQL

At the command prompt, type SELECT and press F4 to prompt it.

Enter the file names separated by a comma for the File prompt. STUDENTS, MARKS

Put your cursor on the Fields prompt and press F4. You need to select the student number, First Name and Last Name fields from the STUDENTS file and the Course Code and Final Mark fields from the Marks file. Use the numbers 10, 20 , 30 , 40 and 50 beside the fields you want and then press enter.

The select statement for a join is now partially built. We need to link the two files.

Put your cursor on the Where condition and press F4. Select the student number field from the Students table and the Student number field from the Marks table. Press enter.

The qualified field names show with a comma separating them. You need to change this to an expression. Remove the comma and put in an equals sign. This will be the basis for a join of the two tables, i.e. Where the student numbers from both tables are the same.

Press enter. You should see data presented from two different tables. After viewing the data, prompt your select statement again and add the clause to sequence the data in Last Name order.

Here is some sample output:

STUDENT FIRST NAME LAST NAME Course Final Date

NUMBER Code Mark Run

333-333-333 Stephen Harper DBS201 76 02/10/13

333-333-333 Stephen Harper EAC397 69 02/10/13

333-333-333 Stephen Harper IBC233 69 02/10/13

333-333-333 Stephen Harper INT222 79 02/10/13

333-333-333 Stephen Harper OOP244 75 02/10/13

111-111-111 Dalton McGuinty DBS201 90 02/10/13

111-111-111 Dalton McGuinty EAC397 85 02/10/13

111-111-111 Dalton McGuinty IBC233 85 02/10/13

111-111-111 Dalton McGuinty INT222 75 02/10/13

111-111-111 Dalton McGuinty OOP244 75 02/10/13

222-222-222 David Miller DBS201 66 02/10/13

222-222-222 David Miller EAC397 88 02/10/13

222-222-222 David Miller IBC233 79 02/10/13

222-222-222 David Miller INT222 85 02/10/13

222-222-222 David Miller OOP244 69 02/10/13

444-444-444 Barack Obama DBS201 60 02/10/13

Come back to your interactive SQL session.   
Let's make this SELECT statement permanently available by creating a view.

Recall the statement into the editing area, make sure your cursor is on the **S** of Select and press the function key that will split the line. (To find out what that function key is, press F24.) Press the Split Line function key. You should have a new line open up at the top of your SELECT statement. Type the following on this line:

**CREATE VIEW STDNTMARKS AS**

Press enter. Oops, you should get an error: **Keyword ORDER not expected.** Unfortunately, SQL does not support the sequencing of records in a view. **Rerun the command with the Order By clause removed.**

You now have a view that provides the join. The SQL statement to get the join data is now simpler.

Run SELECT \* FROM STDNTMARKS. To get the data in order, you have to add the ORDER BY clause when you run a SELECT statement.

In RDi, Remote Systems Explorer, right click on your library and select Refresh.

Look for an object called STDNTMARKS. What type of file is it? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Its attribute? \_\_\_\_\_\_\_\_\_\_\_\_

The logical file object we created called STDNTMARKS does not store an access path. You can use the ORDER BY clause when looking at STDNTMARKS in SQL but if you wanted to use the STDNTMARKS table elsewhere (e.g. in an RPG or COBOL program or DFU) you have to take the records as they come which is more or less in "arrival sequence", i.e. the order in which the records exist in the physical file.

Exit SQL. At the ‘Green Screen’ command line, RUNQRY \*n STDNTMARKS  
Notice the output is identical to running a simple SELECT statement in SQL. To get a printout of this file, run RUNQRY \*n STDNTMARKS press F4. Look for the input field:

Report output type . . . . . . . \*RUNOPT \*RUNOPT, \*DISPLAY...

and change the output to \*PRINTER

**That printout is to be handed in.**

Views and logical files are very similar. A program can process a view or a logical file. A logical file can also specify an access path (i.e. key) to order records. A view cannot. See member STDNTMARKS in IBC233LIB/QDDSSRC. It is a join file specified using DDS. The code is not as elegant as SQL – that's why we didn't make you do it using DDS. But it does add a key to order the records.

We are going to create a simple logical file over a physical file. It's like a view of one table that uses a key.

In RDi, right click on your QDDSSRC file and add a new member. The member name will be FEESOWED. It is a **Logical** **File**, so the member type is **LF**. Include the following text: Students owing Fees in last name order

The following code should be entered using the prompter:

A R STUDENTSR PFILE(STUDENTS)

A K LASTNAME

A S FEESOWED COMP(GT 0 )

The first line says this logical file is based on Physical File, STUDENTS. It references the same record format name as STUDENTS which means that all fields from STUDENTS are included in the logical file. The second line specifies the Key field. Additional key fields such as FIRSTNAME could also be specified on following lines. Finally, a Selection line specifies that only records where it is true that the FEESOWED field compares to \*NE 0 (or FEESOWED \*NE 0 ) are included in the logical file. Compile this and check in the Remote Systems pane for your new logical \*FILE object.

Switch to ‘Green Screen’. **Print the FEESOWED logical file compile listing for handing in.**

The UPDDTA DFU utility will use the logical file access path over the STUDENTS physical file data.

===> UPDDTA FEESOWED

You should see two students with fees owing in last name order by paging up and paging down. Don’t change any data. Exit from DFU.

**Create an SQL view called FINESOWED that includes students with fines owing.**

**Then you can use: SELECT \* from FINESOWED.**

TheFINESOWED object created by SQL is a Logical file like FEESOWED and can be used in a similar way outside SQL. Try UPDDTA FINESOWED. Notice that Logical files defined with DDS can include a key. Unfortunately, SQL views cannot include an ORDER BY clause.

Print out the contents of the FINESOWED view using  
**===> RUNQRY \*N FINESOWED … change the output to \*PRINTER and hand it in.**

**Part C**

**Objectives:**

* **Working with Objects Subset and View File fields in an RDiSession**
* **View File fields in ‘Green Screen’**
* **View File Description in ‘Green Screen’**

Use the show in table option on your library name and then double click on the Object Table tab.

There is a downward pointing arrow in the upper right of the Objects Table View. Use this to show all columns.

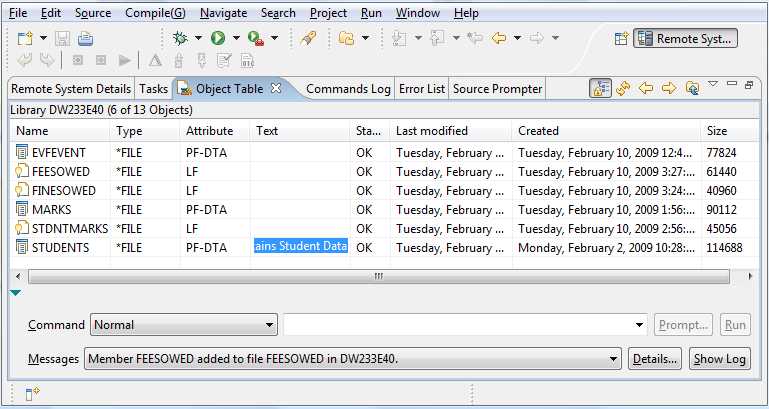
(if you have trouble locating this, it is circled on the next screen shot)

We have now included additional information like the Date Created, the Date Modified, Size etc.

If you like, this information can be sent to a PC file that can be opened with Notepad. What option allows this?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Use the subset option to just show database file objects.

****

Note the Attributes for the different type of database objects

Students \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ STDNTMARKS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ FinesOwed \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Marks \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ FeesOwed \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Remember two different techniques were used to produce FeesOwed and FinesOwed.

In order to see the fields used in a file try the following:

Right click on Students and select Show in Table and select Fields. Try the same thing with the other database objects.

Does this work the same with the View/Logical File called STDNTMARKS? \_\_\_\_\_\_\_\_\_\_\_

In the Objects Table view right click on Students again and this time select Properties.

There is even more information that is not being displayed. This object was created on the System ODIN.

What was the system level when this object was created? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What do you need to click on to find the source file name for this data file and the library that it is located in?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Click on the advanced Info and record the operating system level when your file was created \_\_\_\_\_\_\_\_\_\_\_\_\_\_

In ‘Green Screen’, Try the DSPFFD command with the Students File. – DSPFFD STUDENTS

Can you see the column headings for the fields? \_\_\_\_\_\_\_\_\_

Switch to RDi and use the Show in Table Data option, Do the column headings show in the table? \_\_\_\_\_\_\_\_

Change to the Show in Table Fields option and right Click on the STUDENTNO field and select field properties. Do the column headings show now? \_\_\_\_\_\_\_\_\_

Switch to ‘Green Screen’, Run the DSPFD command with the Students file.

==> DSPFD STUDENTS

There is a lot of information that shows here. Look carefully at the Database File Attributes. This is an externally described file with a file level identifier and a recorded creation date.

Try the same command with the MARKS file. Look at the Database File Attributes. There is something different here that relates to how you constructed these files. What is it?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Although this difference is recorded, you can use both files with any utilities like DFU or any programming language like CL.

**Hand-in instructions for a system printout:**

Printout of view with STDNTMARKS – QPQUPRFIL

Your name was entered in the file and should show in your printout

Compile listing of FEESOWED logical file.

Printout of view called FINESOWED - QPQUPRFIL

All printouts should have your name printed at the bottom of each page.

Note that you may be asked to demonstrate your newly created objects

Hand in the reports on time to get credit for the lab:

* Please remove the page with the "Class/Host/Job/User/etc." This is known as the header or separator page. The printer operator needs it. Your professor does not.
* Please remove any blank trailing pages from the end of the printout
* You may need to refold all the pages so the first page is facing up. Then fold it in half.
* Print your name, Learn ID, and IBC233 Section letter near the top left if your name does not appear at the bottom.
* Thanks. This makes it much easier to handle the paper and record your mark.