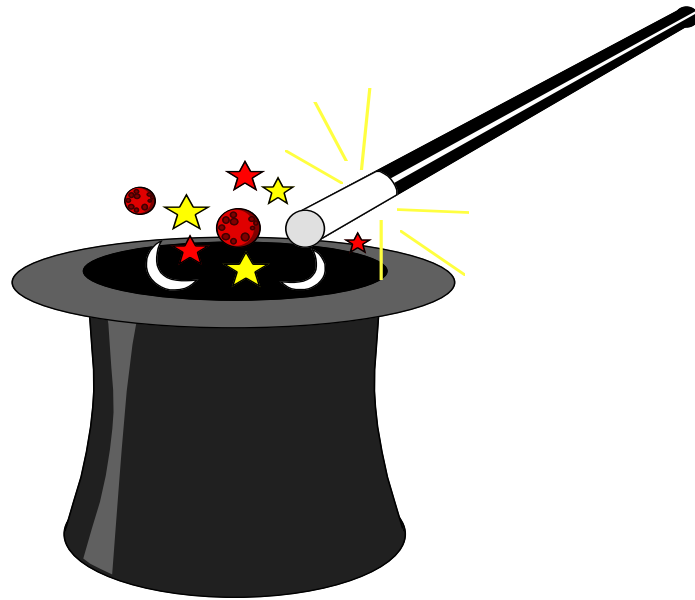


# Introduction to IBM i

## Creating Data Physical Files



# Objectives

- Explain EBCDIC
- Describe Hexadecimal representation
- Explain the different methods of numeric data storage
- Explain the difference between logical and physical organization
- Introduce the tools available for data definition
- Use DDS to define data

# Physical Storage

- Bit - holds a 0 or a 1
- Nibble - four bits
- Byte - eight bits or two nibbles

# Hexadecimal

- Represents each nibble with a single hexadecimal character

<u>Nibble Value</u>	<u>Hex Character</u>
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

# Character Representation

- EBCDIC - midrange and mainframe
- ASCII - microcomputers
- Represent a character with 1 byte

# EBCDIC

**1100 0001 = A**

**1100 0010 = B**

**1100 0011 = C**

**etc.**

**Or in HEX**

**C1 = A**

**C2 = B**

**C3 = C**

**etc.**

# Numeric Representation

- Zoned Decimal
- Packed Decimal
- Binary

# Zoned Decimal

- Uses 1 byte for each digit
- First nibble set to 1111 or F
- Second nibble contains the digit value in binary
- Negative numbers denoted by a 1101 or D in the second to last nibble

# Packed Decimal

- Uses 1 nibble for each digit
- Each nibble contains the digit value in binary
- The last nibble denotes the sign of the number:
  - ◆ Negative numbers have a 1101 or D
  - ◆ Positive numbers have a 1111 or F

# Binary

- Positive numbers stored in “straight” binary
- Negative numbers stored in two’s compliment

# Binary, Zoned & Packed

<u>Hex</u>	<u>Binary</u>	<u>Zoned</u>	<u>Packed</u>	<u>Decimal</u>
1	0001	F1	1F	1
2	0010	F2	2F	2
3	0011	F3	3F	3
4	0100	F4	4F	4
5	0101	F5	5F	5
6	0110	F6	6F	6
7	0111	F7	7F	7
8	1000	F8	8F	8
9	1001	F9	9F	9
A	1010	F1F0	10F	10
B	1011	F1F1	11F	11
C	1100	F1F2	12F	12
D	1101	F1F3	13F	13
E	1110	F1F4	14F	14
F	1111	F1F5	15F	15

# Comparison

	<u>Space</u>	<u>Readability</u>
<u>Binary</u>	Best	Worst
<u>Packed</u>	Medium	Best
<u>Zone</u>	Worst	Medium

# IBM i Field Space Allocation

<u>Number of Digits</u>	<u>Number of Bytes Binary</u>	<u>Number of Bytes Packed Decimal</u>
1	2	1
2	2	2
3	2	2
4	2	3
5	4	3
6	4	4
7	4	4
8	4	5
9	4	5

# Physical vs. Logical Organization

- Physical
  - ◆ bit, nibble, byte
  - ◆ disk, track, byte offset
- Logical
  - ◆ field, record, file
  - ◆ library, object, member
  - ◆ key, access path

# DSPPFM

- Display physical file member
- Gives a physical view of the data
- Initially shows the EBCDIC translation of the stored data
- F10, F11 will display the hexadecimal representation of each EBCDIC character

# DSPPFM

## Display Physical File Member

```
File . . . . . : CHARDATA          Library . . . . . : YOURLIBXX
Member . . . . . : CHARDATA          Record . . . . . : 1
Control . . . . . :                   Column . . . . . : 1
Find . . . . . :
*...+...1...+...2...+...3...+...4...+...5
ABC abc 123 * ) @
XYZ xyz 789 : / =
```

\*\*\*\*\* END OF DATA \*\*\*\*\*

Bottom

F3=Exit    F12=Cancel    F19=Left    F20=Right    F24=More keys

# DSPPFM

## Display Physical File Member

```
File . . . . . : CHARDATA          Library . . . . . : YOURLIBXX
Member . . . . . : CHARDATA          Record . . . . . : 1
Control . . . . . : _____      Column . . . . . : 1
Find . . . . . : _____
* . . . + . . . . 1 . . . + . . . . 2      *...+....1....+....2
C1C2C340 81828340 F1F2F340 5C405D40 7C404040  *ABC abc 123 * ) @  *
E7E8E940 A7A8A940 F7F8F940 7A406140 7E404040  *XYZ xyz 789 : / =  *
```

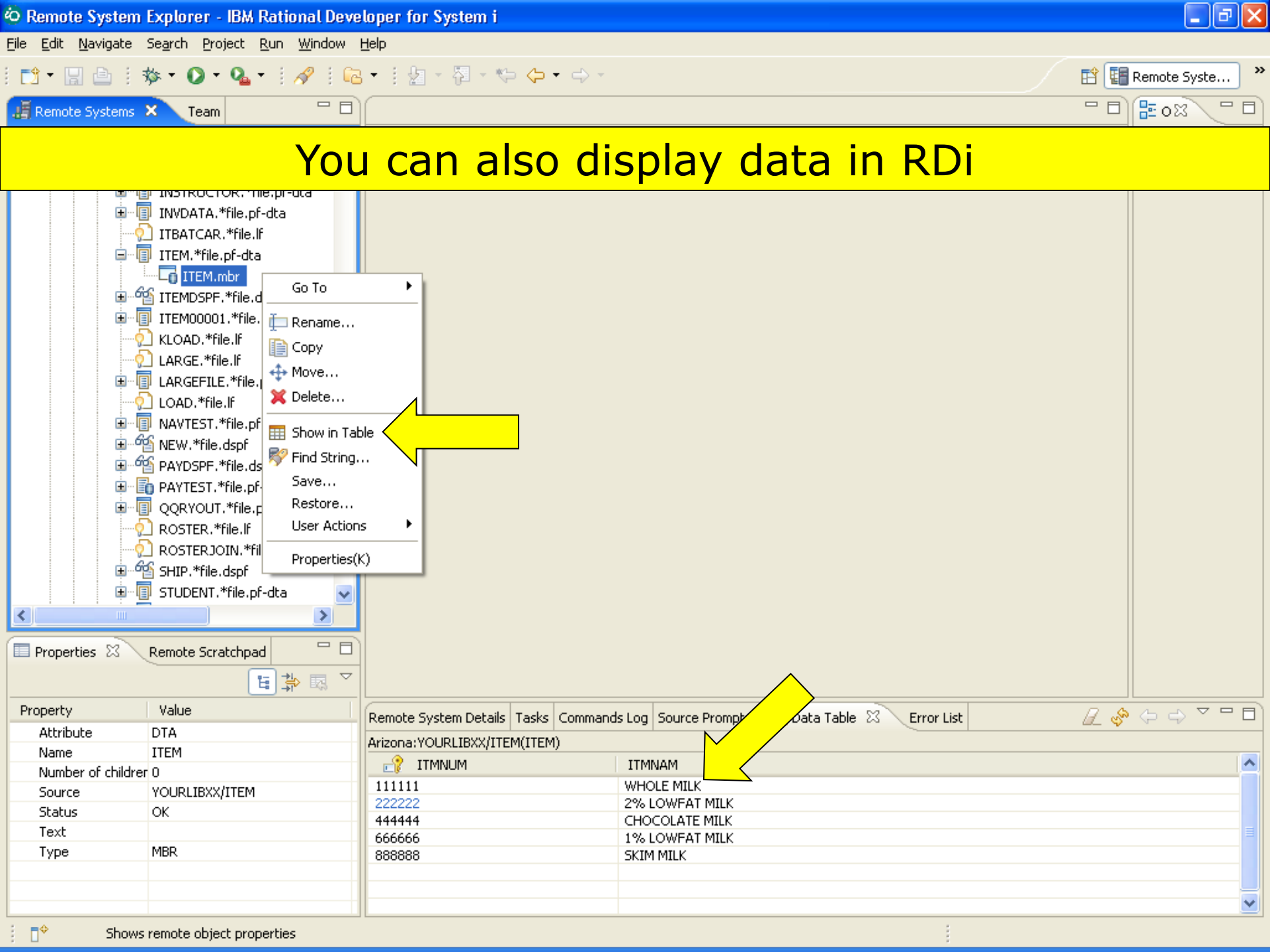
**F10 displays the hexadecimal representations of the EBCDIC characters to the left of each record**

\*\*\*\*\* END OF DATA \*\*\*\*\*

F3=Exit    F12=Cancel    F19=Left    F20=Right    F24=More keys

Bottom





# You can also display data in RDi

- Go To
- Rename...
- Copy
- Move...
- Delete...
- Show in Table
- Find String...
- Save...
- Restore...
- User Actions
- Properties(K)

Remote System Details | Tasks | Commands Log | Source Promph | Data Table | Error List

Arizona:YOURLIBXX/ITEM(ITEM)

ITMNUM	ITMNAM
111111	WHOLE MILK
222222	2% LOWFAT MILK
444444	CHOCOLATE MILK
666666	1% LOWFAT MILK
888888	SKIM MILK

System i Navigator

File Edit View Help

Environment: My Connections

Tables Database: D105098d Schema: YOURLIBXX

SQL Name	Partitioned	Owner	Last Changed	Short Name	Text
"Copy of ITEM"	No	USFL1AD001	6/26/08 1:13:40 PM	ITEM00001	
"Copy of STUDENT"	No	USFL1AD001	7/7/08 11:02:50 AM	STUDE00001	
BATCH	No	USFL1AD001	7/1/08 9:45:24 AM	BATCH	
CARTON	No	USFL1AD001	7/1/08 11:03:58 AM	CARTON	

2 minutes old

# Navigator can show data also but has a lot of other data capabilities

All Objects  
Aliases  
Constraints  
Distinct Types  
Functions  
Indexes  
Journal Receivers  
Journals  
Procedures  
Sequences  
SQL Packages  
Tables

DDSSRC	No	USFL1AD001	6/7/08 1:42:15 PM	DDSSRC	
DEPFILES	No	USFL1AD001	7/7/08 1:59:58 PM	DEPFILES	Output file for DSPDBR
FRF	No	USFL1AD001	7/11/08 11:23:49 AM	FRF	
GRADE	No	USFL1AD001	7/8/08 1:25:08 PM	GRADE	
INSTRUCTOR	No	USFL1AD001	7/8/08 10:46:25 AM	INSTRUCTOR	
INVDATA	No	USFL1AD001	6/20/08 10:39:56 AM	INVDATA	
ITEM	No	USFL1AD001	7/1/08 9:37:13 AM	ITEM	
LARGEFILE	No	USFL1AD001	7/14/08 11:17:59 AM	LARGEFILE	
NAVTEST	No	USFL1AD001	6/23/08 8:51:16 AM	NAVTEST	
PAYTEST	No	USFL1AD001	8/21/08 11:24:27 AM	PAYTEST	
QIXAPROF	No	USFL1AD001	5/20/09 10:01:00 AM	QIXAPROF	
AD001			7/14/08 9:16:42 AM	QQRYP	
AD001			7/11/08 11:23:49 AM	STUDENT	
AD001			7/10/08 9:13:09 AM		
AD001			6/20/08 10:42:29 AM	SUPPLIER	
AD001			6/30/08 1:20:50 PM	TEST	

**Contents of YOURLIBXX.ITEM**

ITMNUM	ITMNAM
1	111111 WHOLE MILK
2	888888 SKIM MILK
3	666666 1% LOWFAT MILK
4	444444 CHOCOLATE MILK
5	222222 2% LOWFAT MILK

**Edit Contents**

**View Contents**

**Data**

**Definition**

Generate SQL...

Index Advisor

Journaling

Lock Holders

Locked Rows

Permissions

Reset Usage Counts...

Show Indexes

Show Materialized Query Tables

Show Related

Statistic Data

Comments...

Cut

Copy

Delete...

Rename...

New

Description

My Tasks: MyIPS

Add a connection

Databases tasks

- Select schemas to display
- Run an SQL script
- Map your database
- Create a new SQL performance monitor
- Import data into a table
- Export data from a table or view
- Help for related tasks

Display a preview of the selected items.

# Defining Data in IBM i

- SQL/400
- IDDU
- DDS

# DDS

- Semi-positional
  - ◆ Columns 1 thru 44 are positional
  - ◆ Columns 45 thru 80 are free format
- Columns 45 thru 80 are called the keyword or function area

# Defining Field & Record Formats

Columns	Purpose	Examples
7	Comments	* makes the line a comment
17	Name type	R = record blank = field
19-28	Name	INVREC ITMNUM
29	Reference	R
30-34	Field Length	22
35	Data type/ Storage Method	blank or A = character blank and a decimal specified = packed P = packed S = zoned B = binary
36-37	Decimal length	2
45-80	Keyword/ Function area	TEXT('Student Record')

# Defining Data with DDS

**Comments      Name type      Name      Keywords**

```

FMT A* .....A* . 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
***** Beginning of data *****
0001.00 *
0002.00 * This is the DDS for the STUDENT file
0003.00 *
0004.00 *
0005.00 R STUDREC TEXT ('STUDENT RECORD')
0006.00 LNAME 10 TEXT ('STUDENT LAST NAME')
0007.00 FNAME 8 TEXT ('STUDENT FIRST NAME')
0008.00 STREET 15 TEXT ('STUDENT STREET ADDR')
0009.00 CITY 10 TEXT ('STUDENT CITY')
0010.00 STATE 2 TEXT ('STUDENT STATE')
0011.00 ZIP 5 TEXT ('STUDENT ZIP CODE')
0012.00 STUDID# 9 TEXT ('STUDENT ID NUMBER')
0013.00 CREDITS 3P TEXT ('CREDITS TAKEN')
0014.00 GRADEPT 3B 2 TEXT ('GRADE POINT AVG')
***** End of data *****

```

**Field Length      Data Type      Decimal Length**

# Defining Data with DDS

Comments	Name	type	Name	Keywords
* * This is the DDS for the STUDENT file * *	R	STUDREC		TEXT('STUDENT RECORD')
	LNAME		10	TEXT('STUDENT LAST NAME')
	FNAME		8	TEXT('STUDENT FIRST NAME')
	STREET		15	TEXT('STUDENT STREET ADDR')
	CITY		10	TEXT('STUDENT CITY')
	STATE		2	TEXT('STUDENT STATE')
	ZIP		5	TEXT('STUDENT ZIP CODE')
	STUDID#		9	TEXT('STUDENT ID NUMBER')
	CREDITS		3P	TEXT('CREDITS TAKEN')
	GRADEPT		3B 2	TEXT('GRADE POINT AVG')

Field Length	Data Type	Decimal Length
10		
8		
15		
10		
2		
5		
9		
3	P	
3	B	2

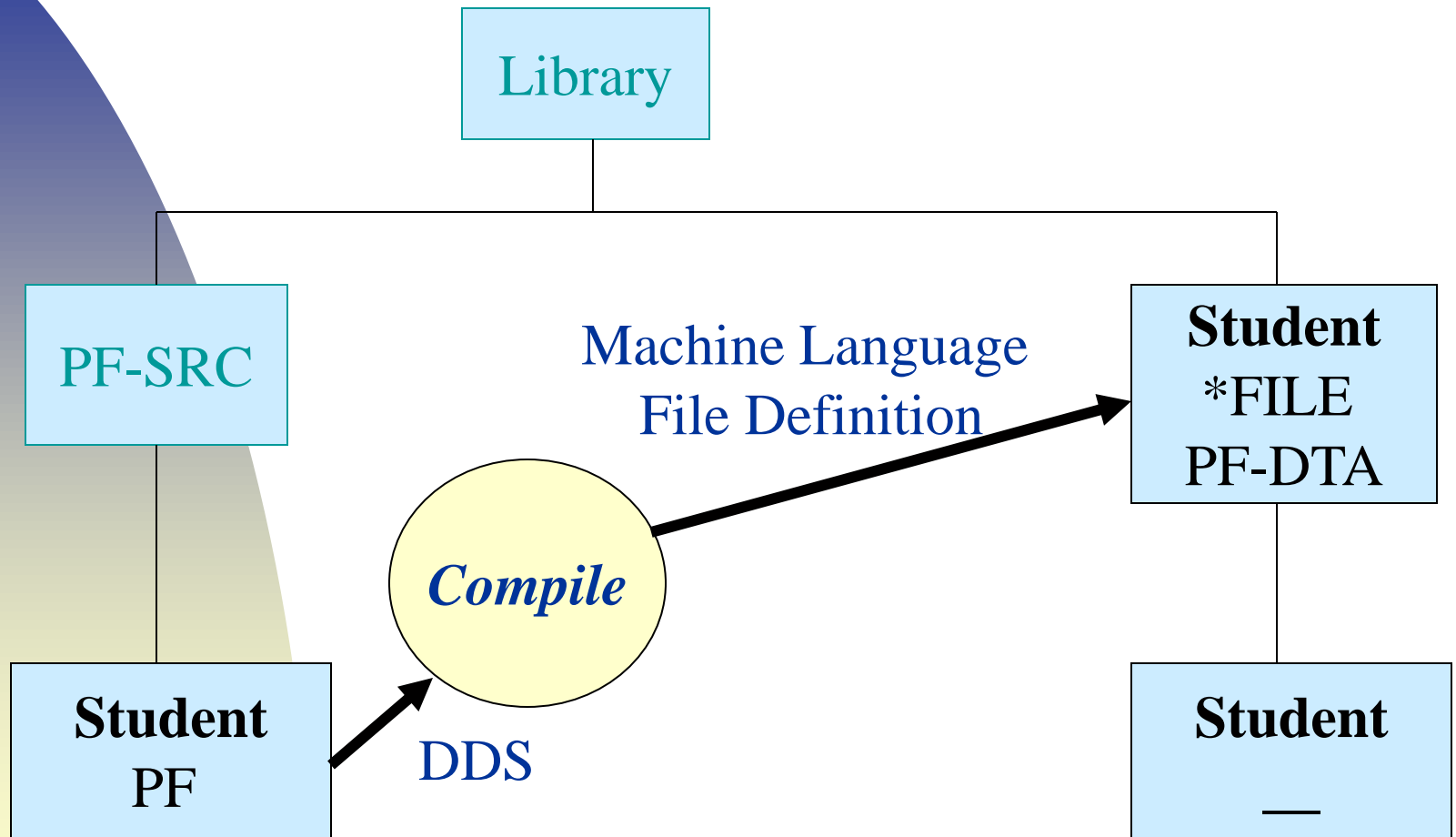
# Compiling DDS File Definitions

- CL command: CRTPF
- In RDi:
  - ◆ Right click source member
  - ◆ Select compile
- In PDM:
  - ◆ Option 14 from the work with members screen

# Compiling DDS File Definitions

- **Compiling DDS:**
  - ◆ Translates the source code definition into machine language
  - ◆ Creates a data physical file
  - ◆ Loads the machine language file definition into the data physical file object
  - ◆ Creates a data physical file member (which will hold raw data)

# Creating a Data Physical File



# Field Reference Files

- Purpose:
  - ◆ Single source of all field definitions
- Advantages:
  - ◆ Less coding
  - ◆ Uniform field definition across all files
  - ◆ Easier to modify fields
- REF keyword used to identify the source (FRF) of field definitions

# Ref Keyword

**Eliminates the need to define fields in each data file  
Instead field definitions are referred to**

**File level  
specification**

**Record level  
specification**

**Field level  
specifications**

```
1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7  
***** Beginning of data *****  
  
This is the DDS for the STUDENT file  
  
REF (MYLIBXX/FRF)  
  
R STUDREC  
  LNAME      R  
  FNAME      R  
  STREET     R  
  CITY       R  
  STATE      R  
  ZIP        R  
  STUDID#    R  
  CREDITS    R  
  GRADEPT    R  
  
0014.00  
***** End of data
```

**REF identifies the source  
of the field definitions**

**R in column 29 means  
that the fields' definition  
is referenced**

# Ref Keyword

**Eliminates the need to define fields in each data file  
Instead field definitions are referred to**

The screenshot shows a DDS editor window for a file named \*STUDENT.PF. The window title bar includes a pencil icon, the filename, and a close button. The editor displays a header line with column positions: "Line 1", "Column 2", "Insert", and "39 changes". Below this is a line of dots representing column positions: "1.....2.....3.....4.....5.....6.....7.....". A comment line reads "THIS is the DDS for the STUDENT file". The main area contains a list of field specifications, each with a red 'R' in column 29, indicating they are referenced. The fields are: STUDREC, LNAME, FNAME, STREET, CITY, STATE, ZIP, STUDID#, and CREDITS. A purple 'REF (MYLIBXX/FRF)' keyword is shown in column 29, with a blue arrow pointing from the 'File level specification' label to it. Another blue arrow points from the 'Record level specification' label to the 'R' in the first field specification. A third blue arrow points from the 'Field level specifications' label to the 'R' in the 'STUDID#' field specification. A yellow box on the right contains text explaining the REF keyword.

**File level specification**

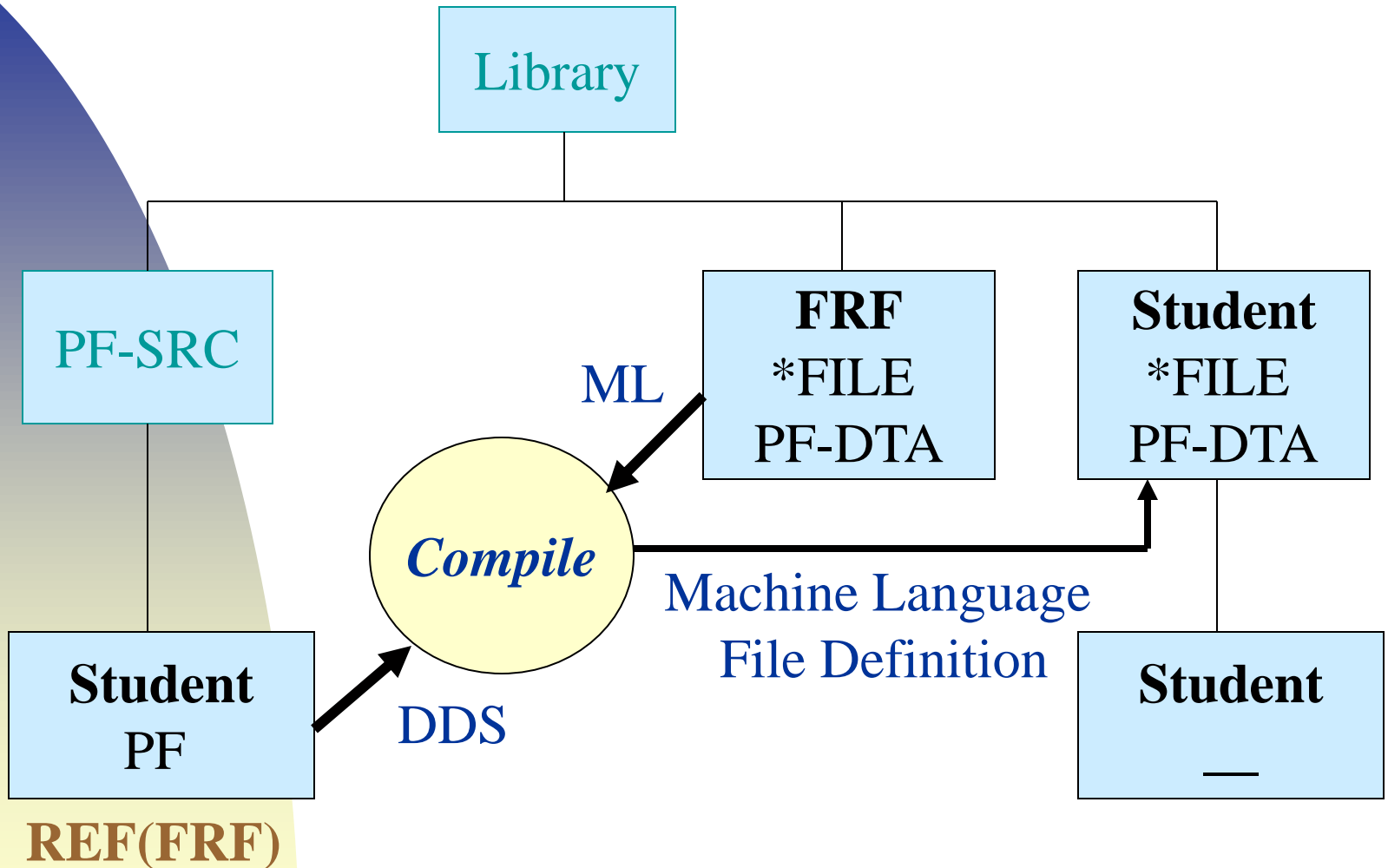
**Record level specification**

**Field level specifications**

REF (MYLIBXX/FRF)

**REF identifies the source of the field definitions  
R in column 29 means that the fields' definition is referenced**

# Ref Keyword



# Points to Remember

- Data in IBM i can be defined:
  - ◆ Internally using a programming language
  - ◆ Externally using DDS, SQL/400 or IDDU
- Character data is stored in EBCDIC
- DDS allows you to specify the logical and physical organization of data