



### Print Row

to print:

id:	2
kind:	TXT_VAL
data.int_val:	
data.dbl_val:	Fred
data.txt_val:	

Instruction: Step 2

### Main

i: 2

db\_data[]:

id:	0
kind:	DBL_VAL
data.int_val:	
data.dbl_val:	3.1415
data.txt_val:	
id:	1
kind:	INT_VAL
data.int_val:	
data.dbl_val:	27
data.txt_val:	
id:	2
kind:	TXT_VAL
data.int_val:	
data.dbl_val:	Fred
data.txt_val:	

Instruction: Step 4

2 This passes the entire Row to the parameter (by value, so the data is copied in)

Procedure: Print Row

Parameters:

1: to print (Row) - the row to print

Steps:

1: Output 'Row with id ', and to print's id to the Terminal

2: select case from to print's kind

3: case is INT\_VAL

4: Output ' has integer ' and int\_val of to print's data

5: case is DBL\_VAL

6: Output ' has double ' and dbl\_val of to print's data

7: case is TXT\_VAL

8: Output ' has text ' and txt\_val of print's data

Step 1 outputs the Row's id

Procedure: Main

Local Variables:

\*: db\_data (array containing 3 Row values)

\*: i (Integer) -

Steps:

2: ...  
3: for i loops over each element in db\_data  
4: call Print Row( db\_data[i] )

Step 2 uses the kind of the row to determine the path to take

Each time Print Row is called it is passed a Row from the array



Terminal — bash — 80x24

Enter value: 27  
Stored in row with id 1  
Enter value: Fred  
Stored in row with id 2  
Row with id 0 has double 3.1415  
Row with id 1 has integer 27  
Row with id 2

