A Shallow Dive into DB Modernization

OCEAN User Group – May 17, 2016

Presented by Patrick Behr

A Shallow Dive into DB Modernization

- DDS to SQL Conversion

- RCAC (Field Masking)

- FIELDPROC (Encryption)

- Adopted Authority

A Shallow Dive into DB Modernization

Before we begin...a disclaimer (or two)

This is a **SHALLOW** dive.

This will be a fairly thorough, yet simple example. There are many important nuances that will not be discussed. The specific details of your environment will require your vigilance and lots 'o testing.

There are many regulations (HIPPA, SOX, PCI) that you need to understand.

Do not use example programs in production.

A Shallow Dive into DB Modernization

There's lots of help out there...
be sure to R.T.F.M.

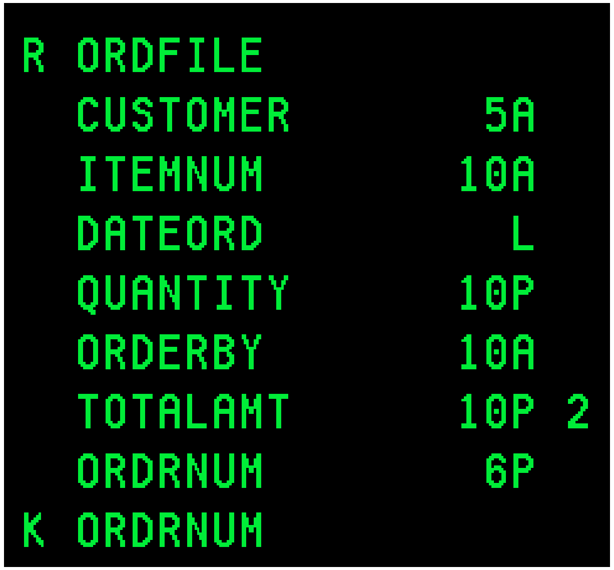
Read The Free Manual

A Shallow Dive into DB Modernization

A journey of 1,000 miles begins with…

A green screen

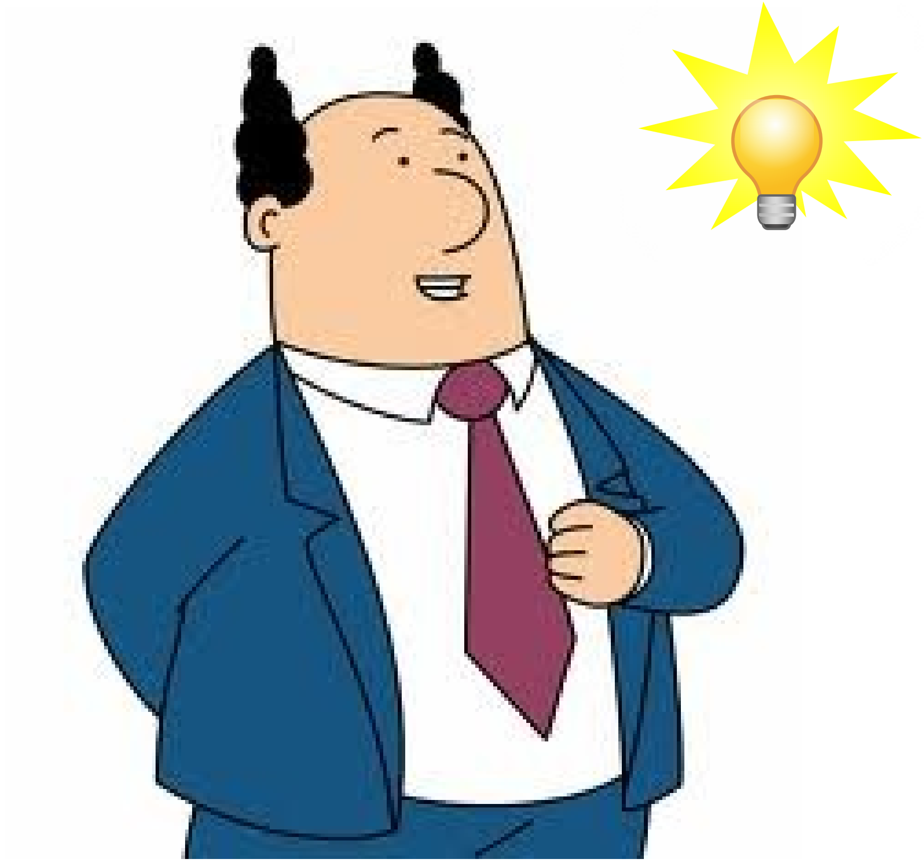
A DDS file



```
R ORDFILE
  CUSTOMER       5A
  ITEMNUM       10A
  DATEORD         L
  QUANTITY      10P
  ORDERBY       10A
  TOTALAMT      10P 2
  ORDRNUM        6P
K ORDRNUM
```

A Shallow Dive into DB Modernization

A journey of 1,000 miles begins with…



Let's just add a field to that table…

We need to mask that data…

We need encryption...

A Shallow Dive into DB Modernization

How we've done it in the past:

Create extension file (or two)

Recompile all your programs

Change the printer files

A Shallow Dive into DB Modernization

Why we don't want to do it that way:

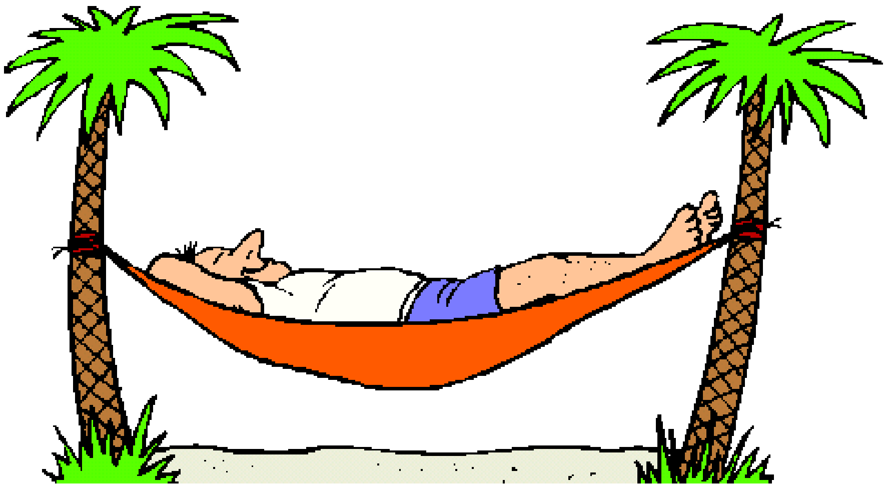
It's a LOT of work.

Quality of the system.

A Shallow Dive into DB Modernization

There's a better way…

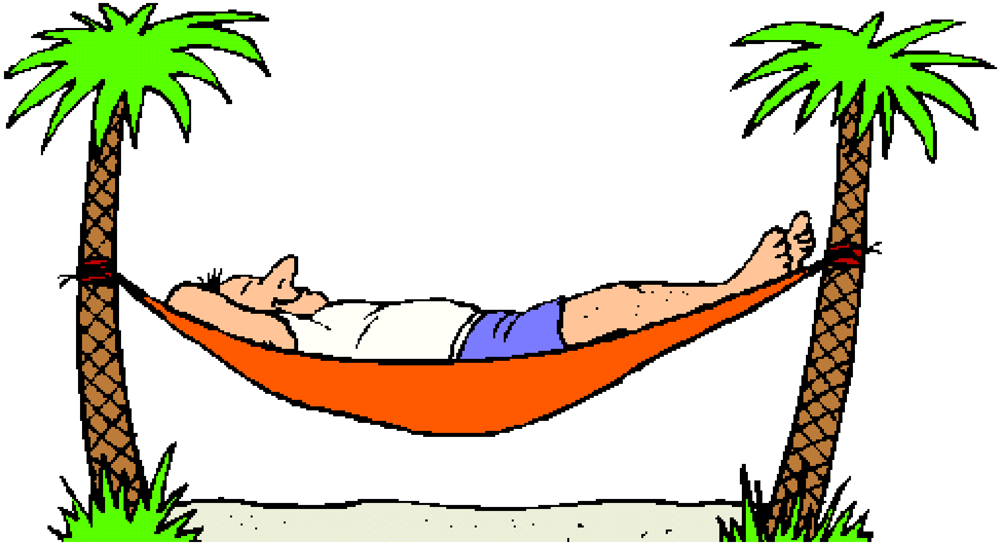
Less work,
Better system



A Shallow Dive into DB Modernization

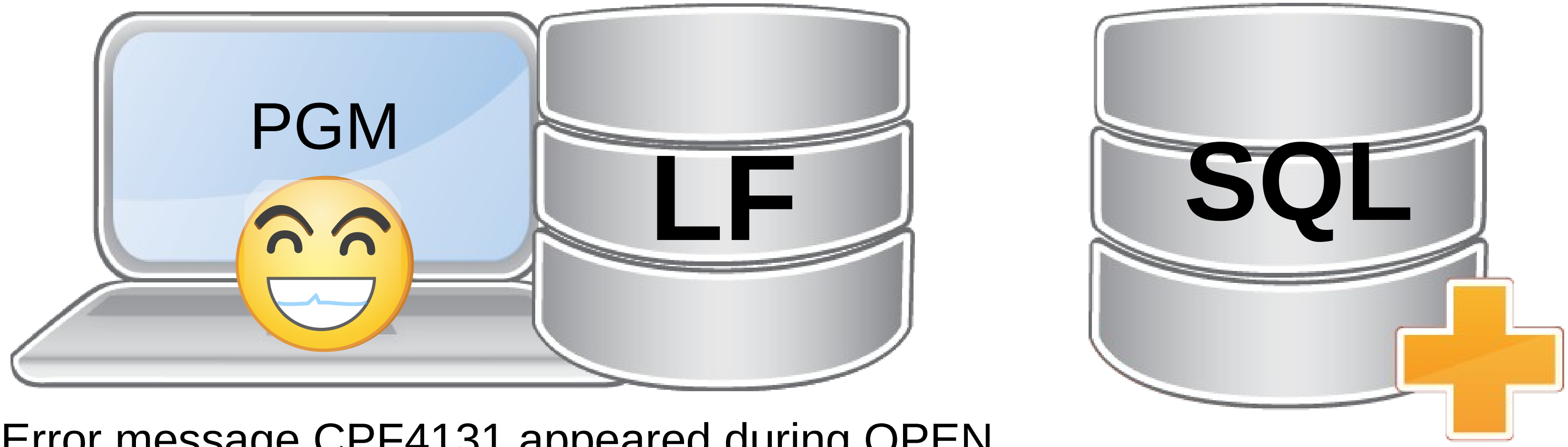
DDS to SQL Conversion

- Create a new SQL table
- Create a logical file



A Shallow Dive into DB Modernization

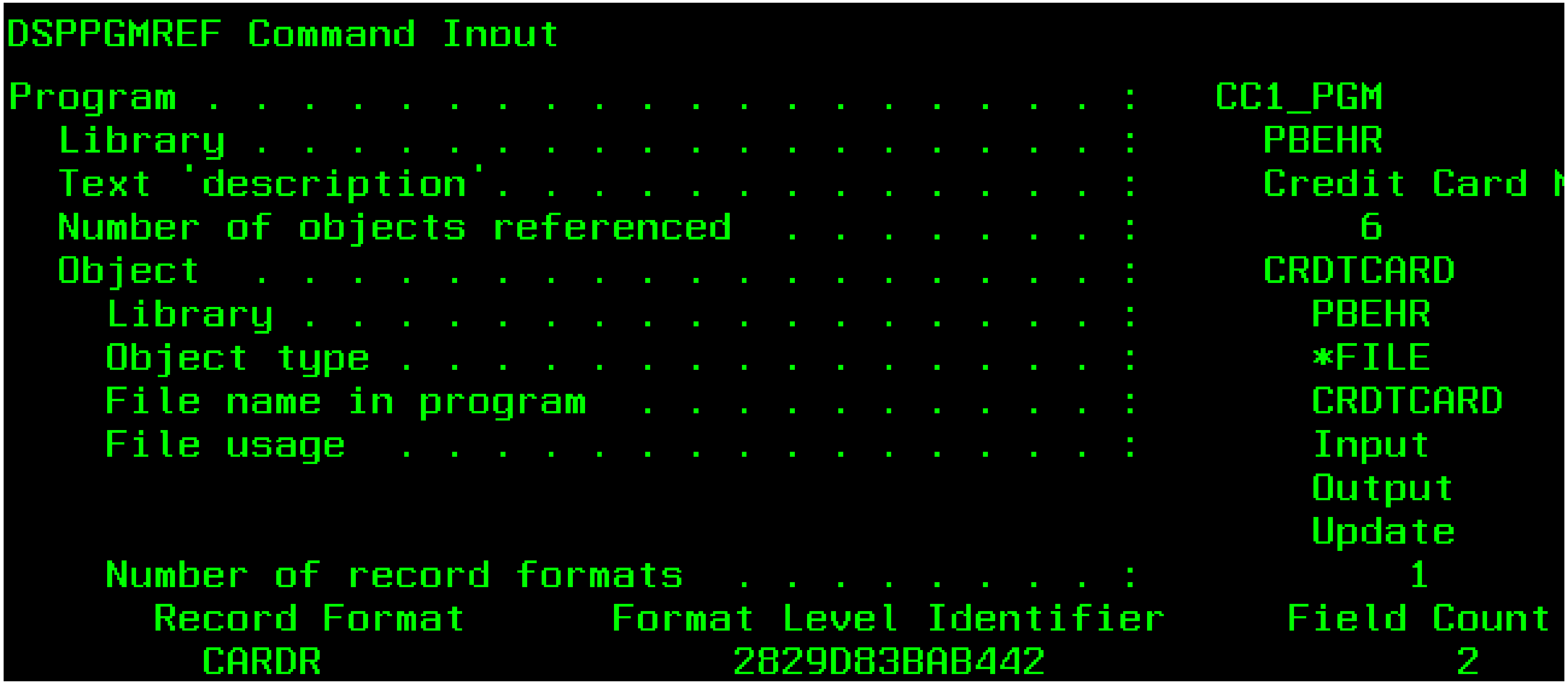
DDS to SQL Conversion



Error message CPF4131 appeared during OPEN

A Shallow Dive into DB Modernization

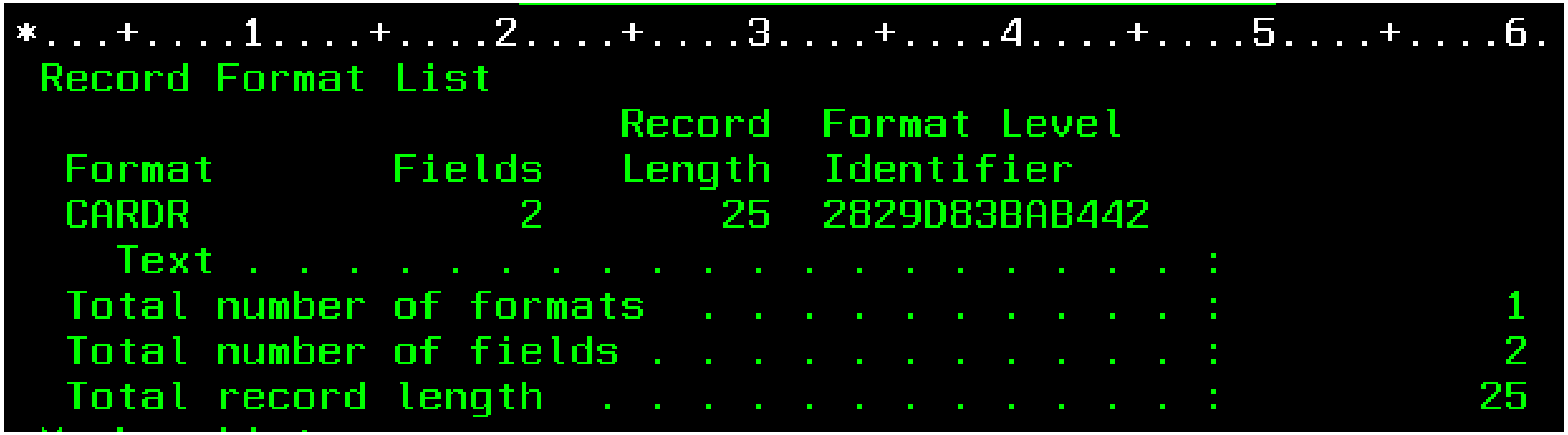
DDS to SQL Conversion



```
DSPPGMREF Command Input

Program . . . . . . . . . . . . . . . . . . . :   CC1_PGM
  Library . . . . . . . . . . . . . . . . . . :     PBEHR
  Text 'description'. . . . . . . . . . . . . :     Credit Card
  Number of objects referenced  . . . . . . . :         6
  Object  . . . . . . . . . . . . . . . . . . :     CRDTCARD
    Library . . . . . . . . . . . . . . . . . :       PBEHR
    Object type . . . . . . . . . . . . . . . :       *FILE
    File name in program  . . . . . . . . . . :       CRDTCARD
    File usage  . . . . . . . . . . . . . . . :       Input
                                                      Output
                                                      Update
    Number of record formats  . . . . . . . . :           1
      Record Format      Format Level Identifier     Field Count
        CARDR                 2829D83BAB442                 2
```

A Shallow Dive into DB Modernization

DDS to SQL Conversion

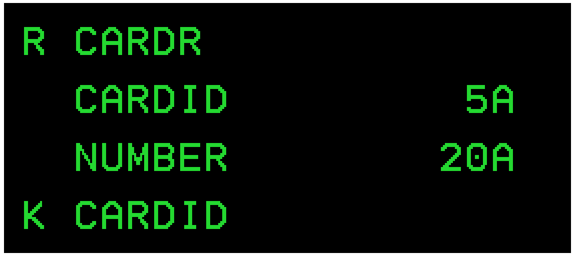


```
*...+....1....+....2....+....3....+....4....+....5....+....6.
 Record Format List
                       Record  Format Level
  Format        Fields  Length  Identifier
  CARDR              2      25  2829D83BAB442
    Text . . . . . . . . . . . . . . . . . . . :
  Total number of formats . . . . . . . . . . :             1
  Total number of fields . . . . . . . . . . . :             2
  Total record length . . . . . . . . . . . . :            25
```

2829D83BAB442 is the “magic” number!

A Shallow Dive into DB Modernization

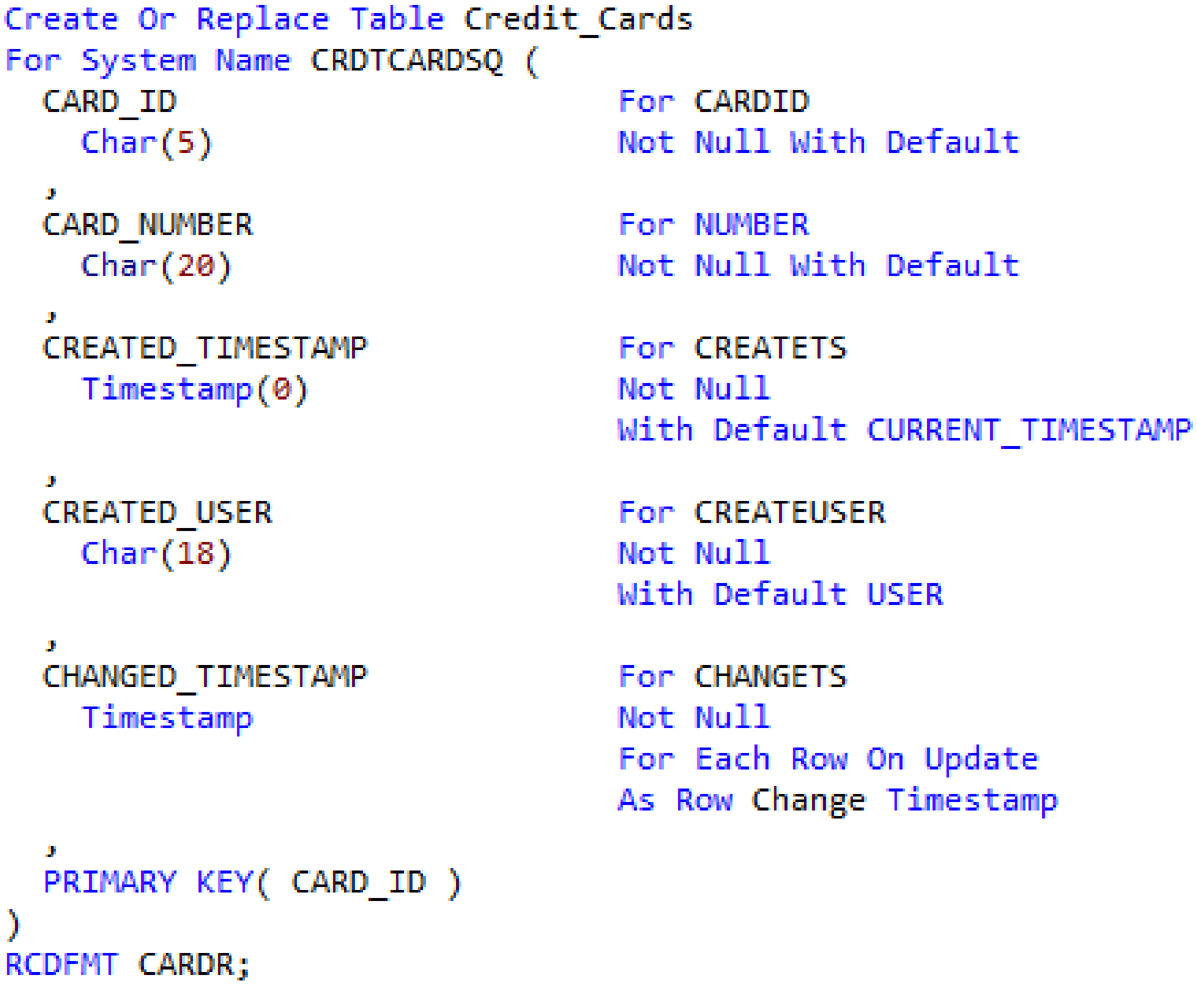
Step 1 Create SQL Table



```
R CARDR
  CARDID             5A
  NUMBER            20A
K CARDID
```

DDS

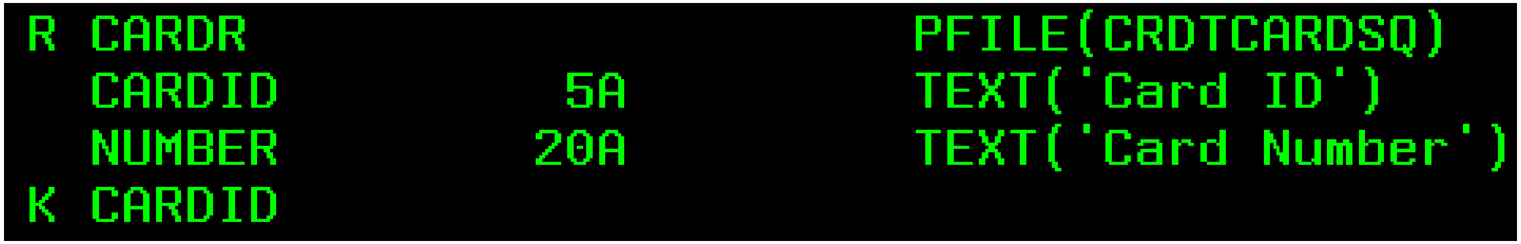
SQL



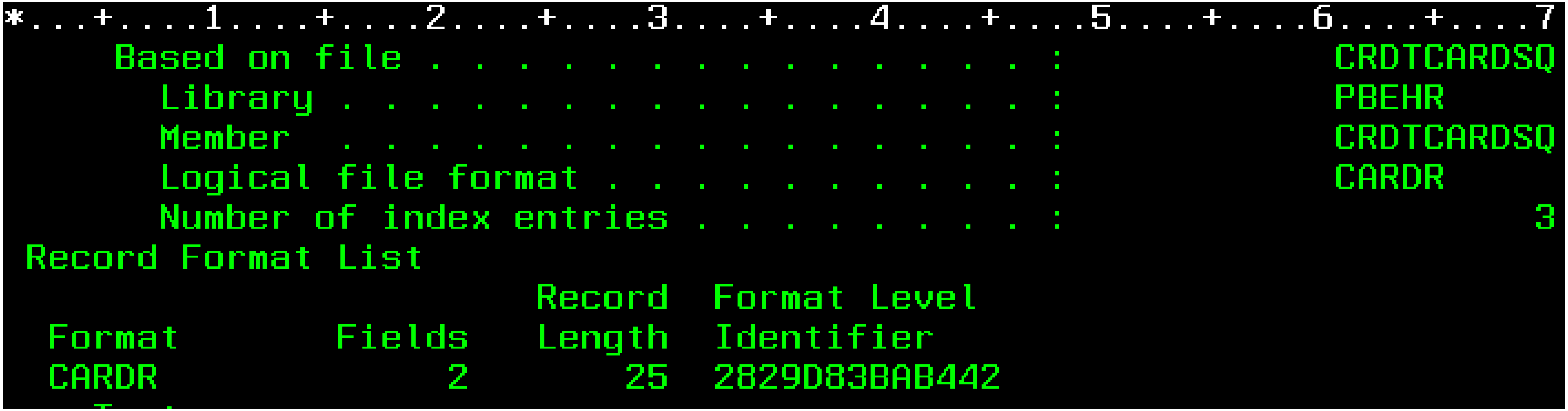
```
Create Or Replace Table Credit_Cards
For System Name CRDTCARDSQ (
  CARD_ID                         For CARDID
    Char(5)                       Not Null With Default
  ,
  CARD_NUMBER                     For NUMBER
    Char(20)                      Not Null With Default
  ,
  CREATED_TIMESTAMP               For CREATETS
    Timestamp(0)                  Not Null
                                  With Default CURRENT_TIMESTAMP

  ,
  CREATED_USER                    For CREATEUSER
    Char(18)                      Not Null
                                  With Default USER

  ,
  CHANGED_TIMESTAMP               For CHANGETS
    Timestamp                     Not Null
                                  For Each Row On Update
                                  As Row Change Timestamp

  ,
  PRIMARY KEY( CARD_ID )
)
RCDFMT CARDR;
```

A Shallow Dive into DB Modernization

Step 2 Create a logical file



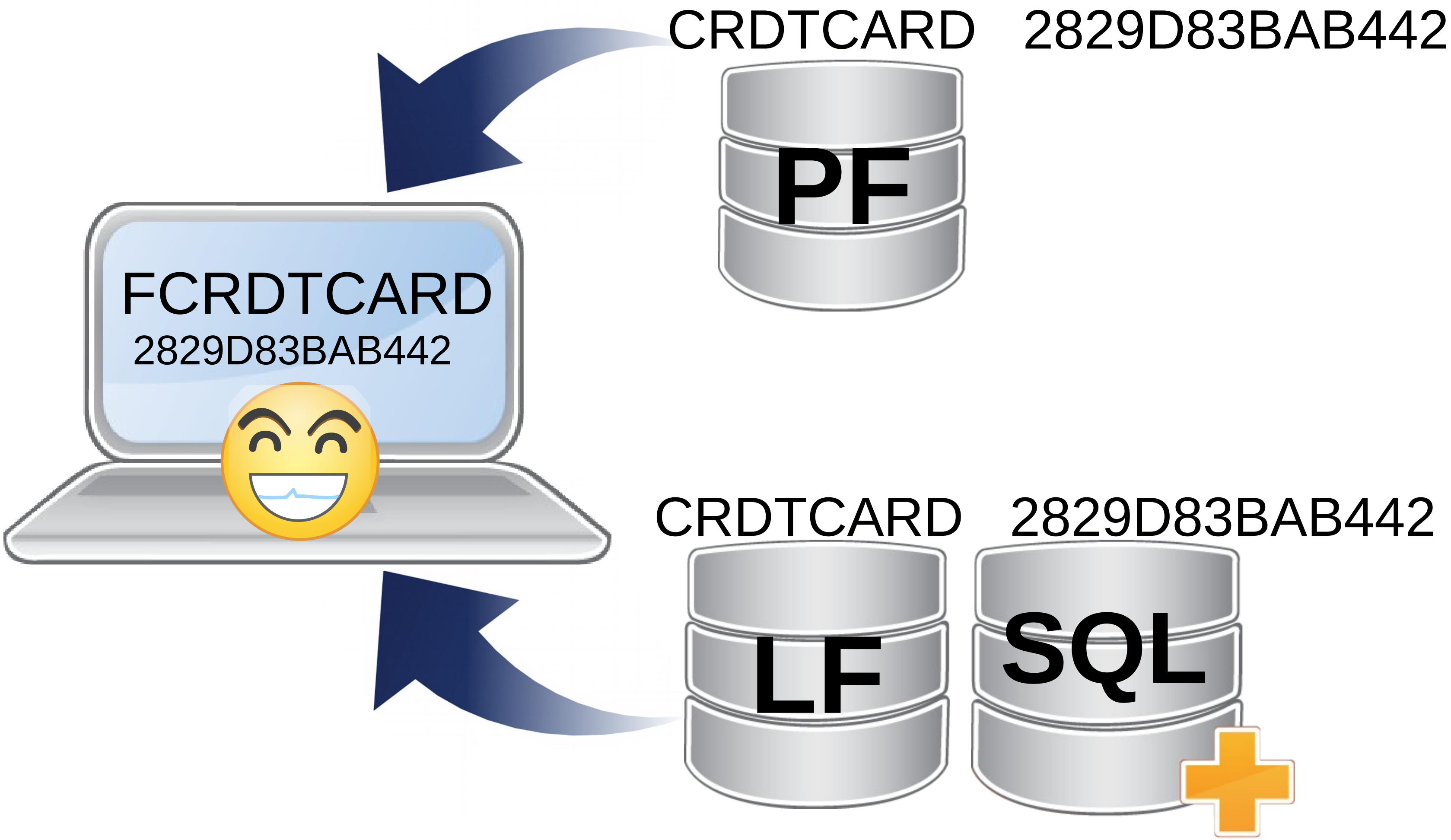
```
R CARDR                      PFILE(CRDTCARDSQ)
  CARDID          5A         TEXT('Card ID')
  NUMBER         20A         TEXT('Card Number')
K CARDID
```



```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7
     Based on file . . . . . . . . . . . . . . :            CRDTCARDSQ
       Library . . . . . . . . . . . . . . . . :            PBEHR
       Member  . . . . . . . . . . . . . . . . :            CRDTCARDSQ
       Logical file format . . . . . . . . . . :            CARDR
       Number of index entries . . . . . . . . :                     3
 Record Format List
                         Record  Format Level
  Format       Fields    Length  Identifier
  CARDR             2        25  2829D83BAB442
```

A Shallow Dive into DB Modernization

DDS to SQL Conversion



CRDTCARD 2829D83BAB442

FCRDTCARD

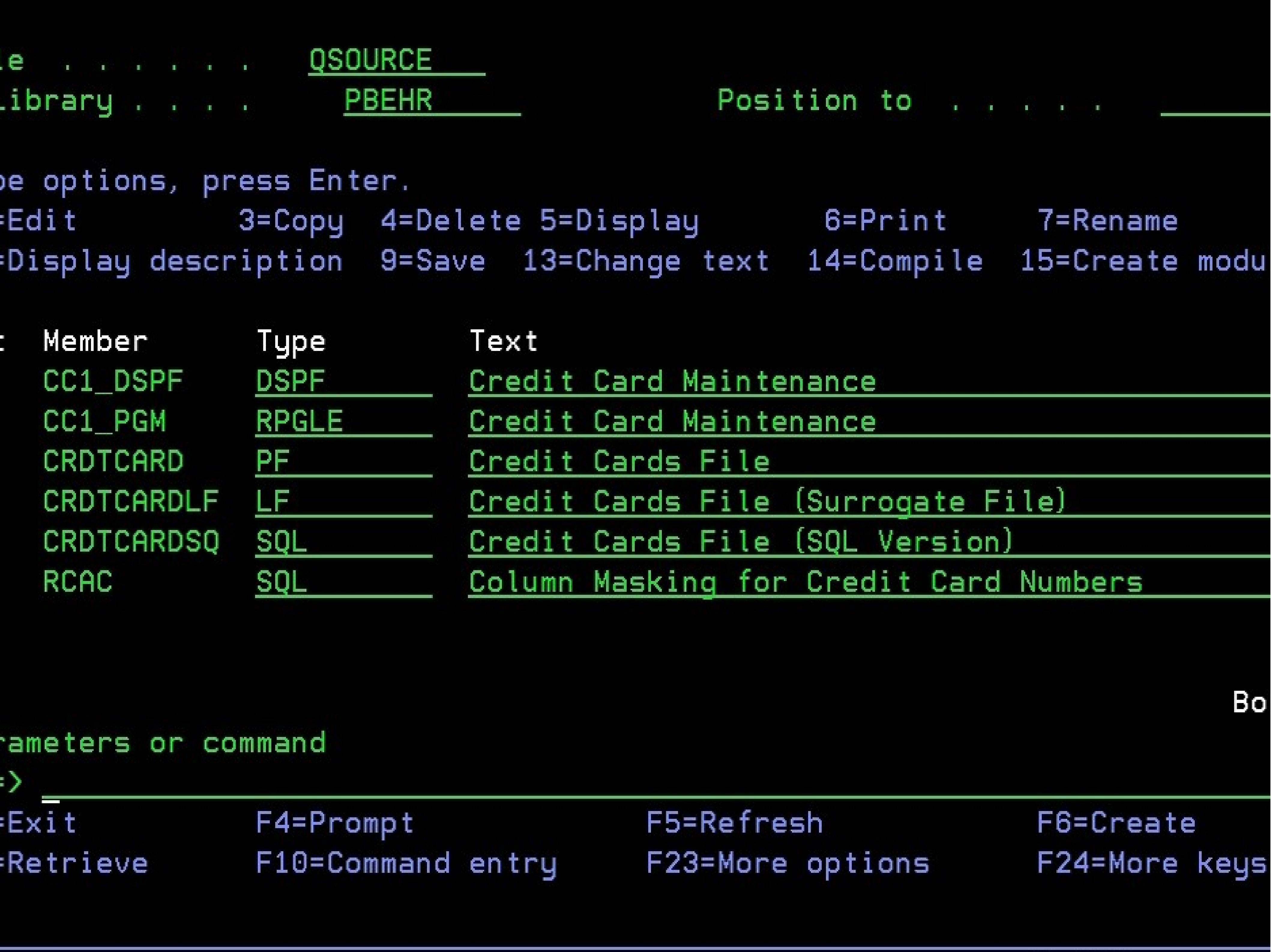
2829D83BAB442

CRDTCARD 2829D83BAB442

A Shallow Dive into DB Modernization

DDS to SQL Conversion

DDS to SQL Conversion
in less than 5 minutes...



```
le . . . . . . .   QSOURCE
Library . . . .     PBEHR                Position to . . . . .

pe options, press Enter.
=Edit         3=Copy  4=Delete 5=Display      6=Print     7=Rename
=Display description  9=Save  13=Change text  14=Compile  15=Create modu

t  Member       Type         Text
   CC1_DSPF     DSPF         Credit Card Maintenance
   CC1_PGM      RPGLE        Credit Card Maintenance
   CRDTCARD     PF           Credit Cards File
   CRDTCARDLF   LF           Credit Cards File (Surrogate File)
   CRDTCARDSQ   SQL          Credit Cards File (SQL Version)
   RCAC         SQL          Column Masking for Credit Card Numbers



                                                                     Bo
rameters or command
=>
=Exit          F4=Prompt              F5=Refresh            F6=Create
=Retrieve      F10=Command entry      F23=More options      F24=More keys
```

A Shallow Dive into DB Modernization

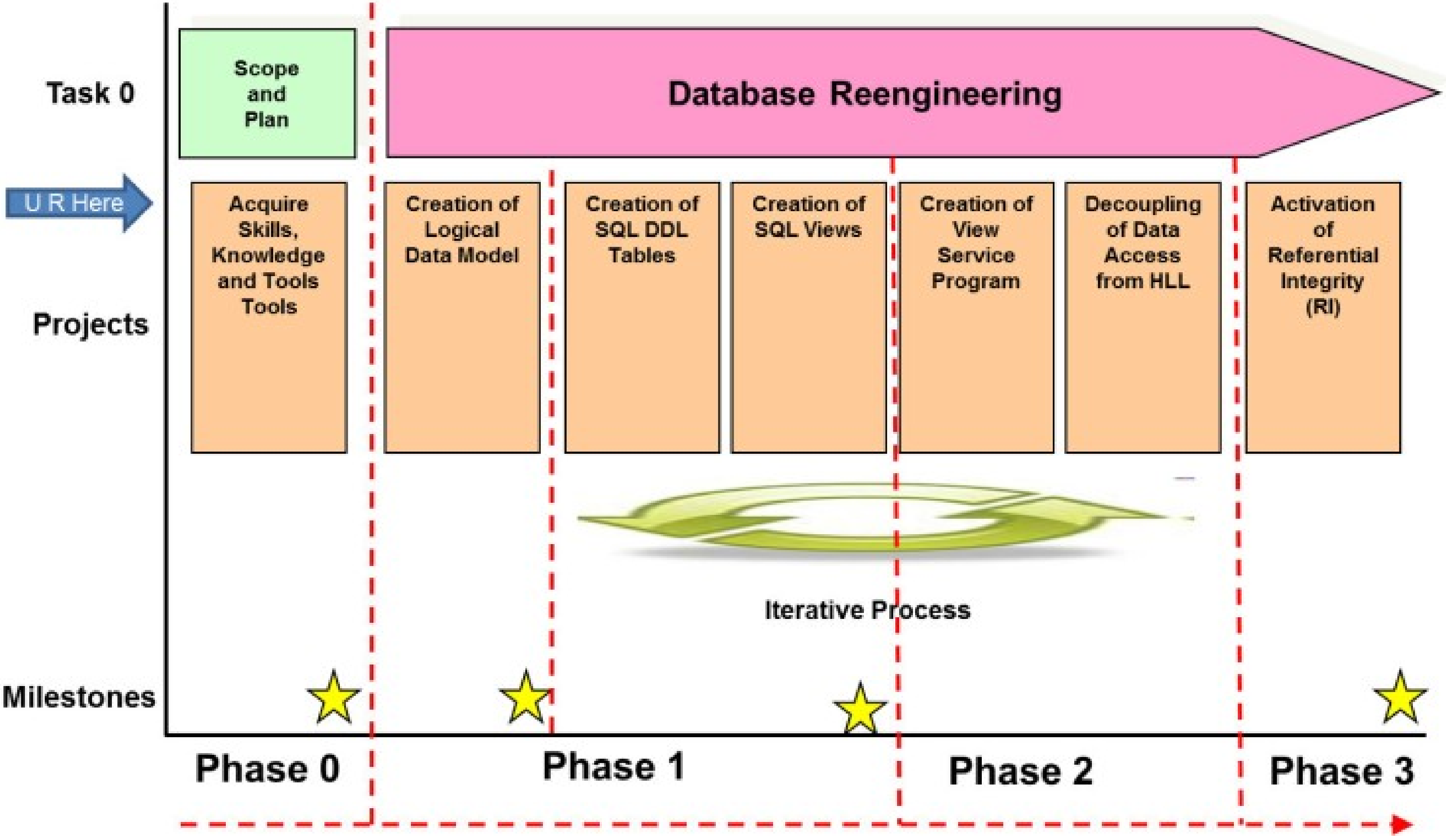
Let's just add a field to that table…

Ok, it will be done in a few minutes…

A Shallow Dive into DB Modernization

DDS to SQL Conversion

This is actually only Stage 1 of the DB Modernization roadmap…


Task 0
Scope and Plan
Database Reengineering
U R Here
Projects
Acquire Skills, Knowledge and Tools Tools
Creation of Logical Data Model
Creation of SQL DDL Tables
Creation of SQL Views
Creation of View Service Program
Decoupling of Data Access from HLL
Activation of Referential Integrity (RI)
Iterative Process
Milestones
Phase 0
Phase 1
Phase 2
Phase 3

A Shallow Dive into DB Modernization

There's lots of help out there...
be sure to R.T.F.M.

Modernizing Database Access; The Madness Behind the Methods
By Dan Cruikshank

Modernizing IBM eServer iSeries Application Data Access
IBM Redbook

Modernizing IBM i Applications from the Database up to the User Interface and Everything in Between
IBM Redbook

A Shallow Dive into DB Modernization

Field Masking with RCAC

Credit Card: ************1234

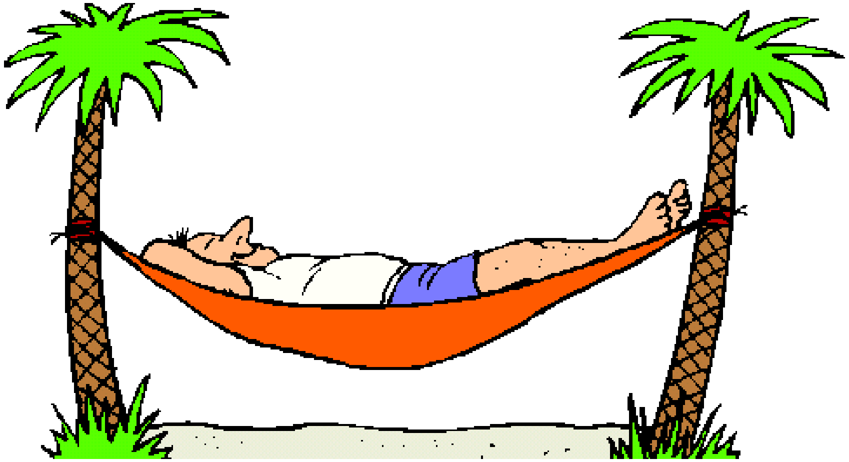
Date Of Birth: 09 / 21 / ####

SSN: xxx-xx-8723

A Shallow Dive into DB Modernization

Field Masking with RCAC

- Register with QIBM_DB_SECADM function
- Create mask function
- Activate the mask function



A Shallow Dive into DB Modernization

Field Masking with RCAC

Works, even if user has *ALLOBJ

Separation of Duties:
- Authority to use RCAC
- Authority to access data

A Shallow Dive into DB Modernization

Field Masking with RCAC

Only users with QIBM_DB_SECADM function can administer and manage RCAC rules.

```
CHGFCNUSG   FCNID(QIBM_DB_SECADM)
            USER(QSECOFR)
            USAGE(*ALLOWED)
```

Work Function Usage (WRKFCNUSG)
Change Function Usage (CHGFCNUSG)
Display Function Usage (DSPFCNUSG)

A Shallow Dive into DB Modernization

Field Masking with RCAC

```
Create Or Replace Mask  mask_name
On FILE
For Column   FIELD
Return
  Case
     When SOME_CONDITION = TRUE
         Then FIELD
     Else
         MASKED_VALUE
     End
Enable;
```

A Shallow Dive into DB Modernization

Field Masking with RCAC

```
Create Or Replace Mask  Credit_Card_Number_Mask
On CRDTCARD
For Column  CARD_NUMBER
Return
  Case When
     Verify_Group_For_User(Current_User, 'SOMEGROUP') = 1
         Then CARD_NUMBER
     Else
         '*********' || Right(CARD_NUMBER, 4)
     End
Enable;
```

A Shallow Dive into DB Modernization

Field Masking with RCAC

```
Alter Table CRDTCARD
Activate Column Access Control;
```

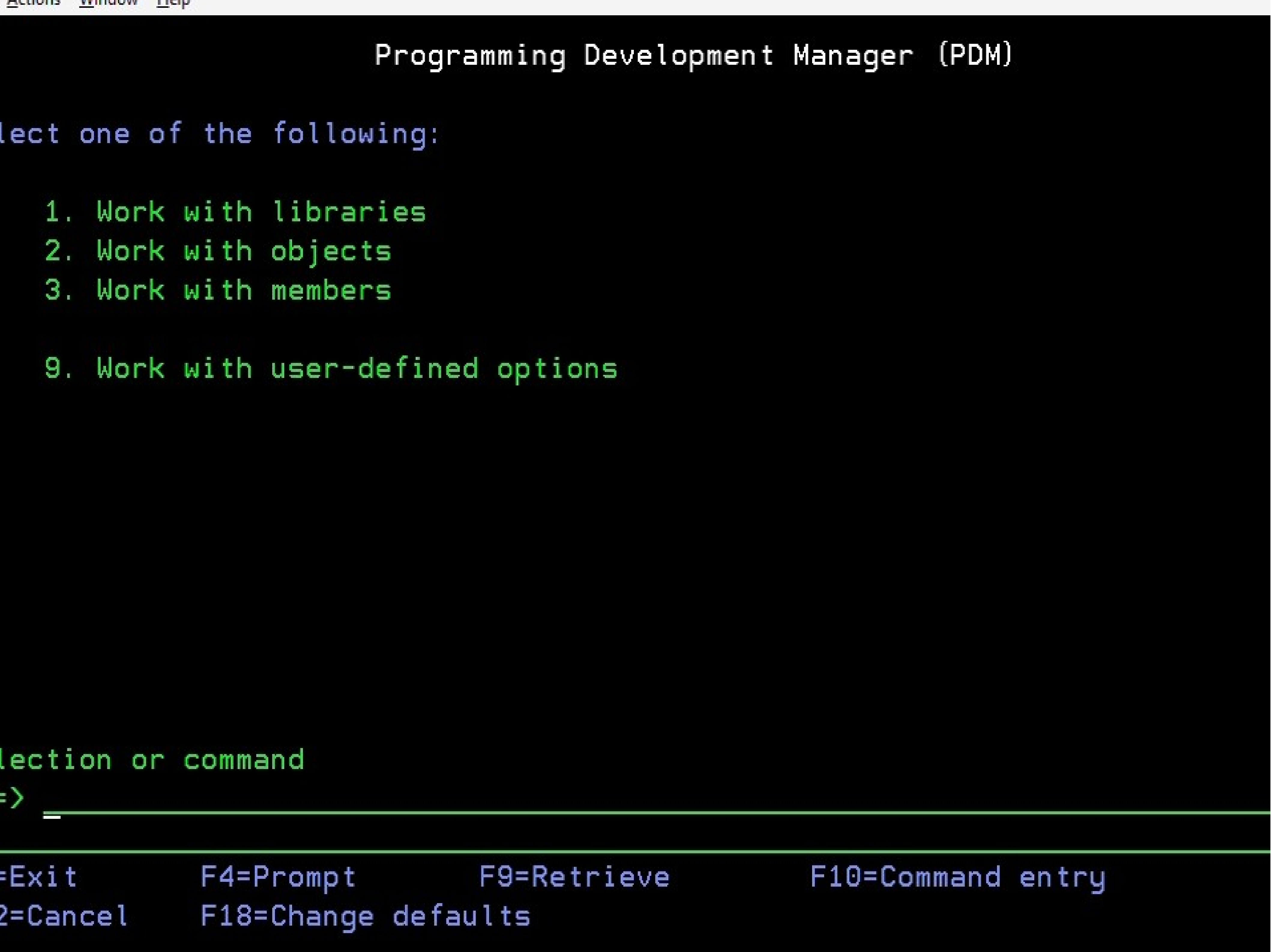
```
Alter Table CRDTCARD
Deactivate Column Access Control;
```

```
Drop Mask Credit_Card_Number_Mask;
```

A Shallow Dive into DB Modernization

Field Masking with RCAC

Field Masking in 2 minutes...



```
                     Programming Development Manager (PDM)

lect one of the following:

  1. Work with libraries
  2. Work with objects
  3. Work with members

  9. Work with user-defined options









lection or command
=>
_______________________________________________________________________

=Exit         F4=Prompt        F9=Retrieve        F10=Command entry
2=Cancel      F18=Change defaults
```

A Shallow Dive into DB Modernization

Field Masking with RCAC

- Requires 7.2 and IBM Advanced Data Security for i (5770SSI option 47)
- RCAC will affect CPYF, CRTDUPOBJ, etc.
  Make sure that your HA/Backup solution will work.
  (RCAC is not applied to the journal receiver process)
- Triggers have access to data outside of RCAC,
  So they must be defined as “SECURE”
- Masking is applied to the final result set.
  Selection, grouping, ordering based on unmasked values
- BE CAREFUL WITH UPDATES!!!

A Shallow Dive into DB Modernization

There's lots of help out there...
be sure to R.T.F.M.

Row and Column Access Control Support in IBM DB2 for i
IBM Redbook

RCAC in DB2 For i, Part 2: Column Masks
by Michael Sansoterra, ITJungle

A Shallow Dive into DB Modernization

We need to mask that data…

Okay…

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

Credit Card: 0xde015724b081ea7003d

Date Of Birth: 0xfd8b695b39e0

SSN: 0x96a45cbcf9ca9425cd

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

A field procedure is a user-written exit routine designed to transform values in a single column.

DB2 will call your field procedure whenever data is written/retrieved from the database.

You are responsible for writing the procedure.

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

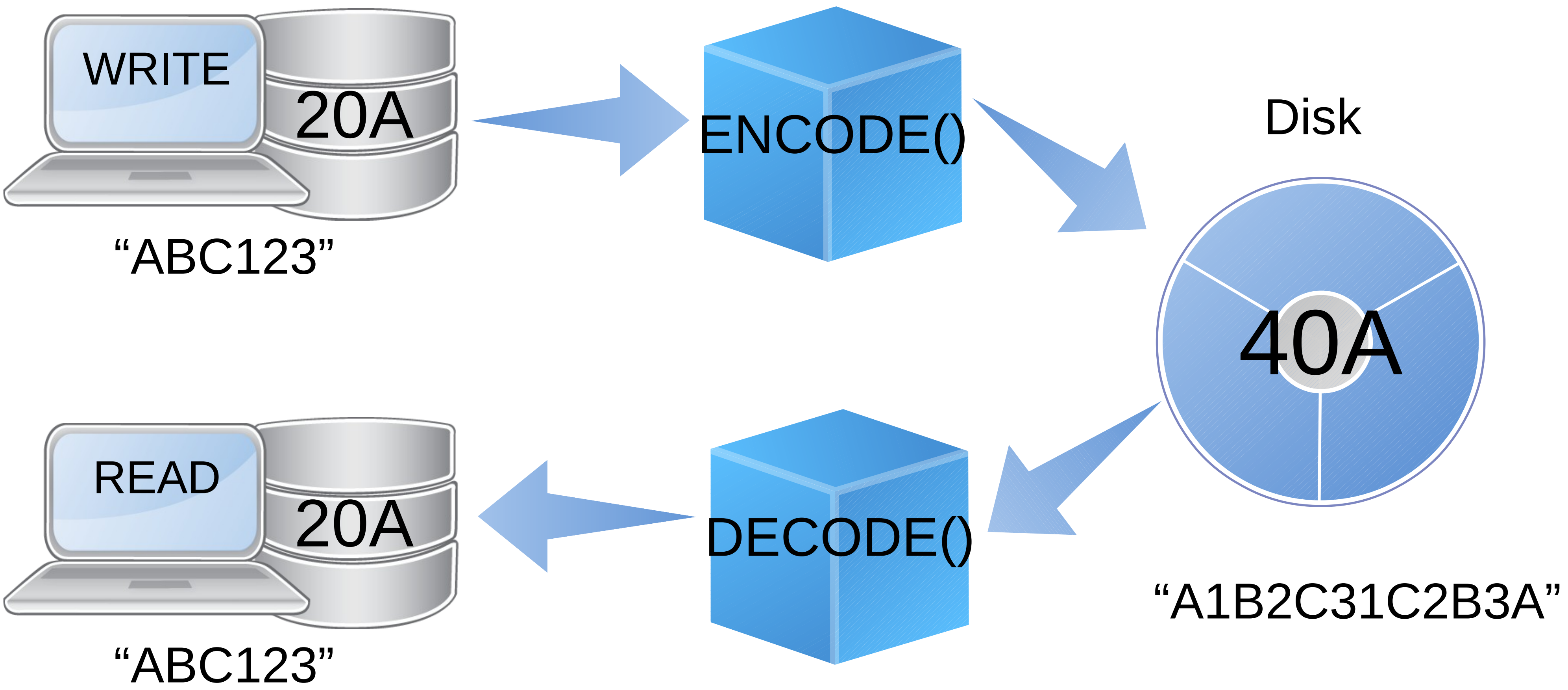
Data, index, and journals stored on hard disks or tapes are transformed. No one can get the decrypted data without the FieldProc program.

Just need to write a FieldProc program and register it.

No change to the original table definition is needed (read as: “no recompiles”).

A Shallow Dive into DB Modernization

Encryption with FIELDPROC



WRITE

20A

“ABC123”

ENCODE()

Disk

40A

“A1B2C31C2B3A”

DECODE()

READ

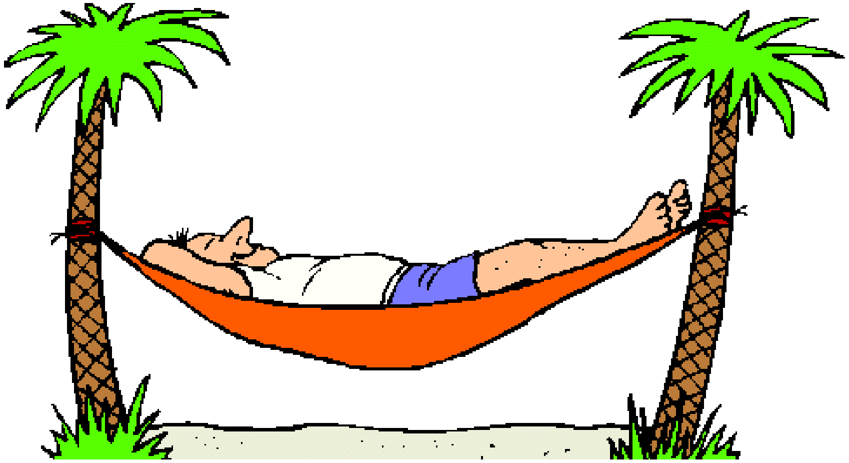
20A

“ABC123”

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

- Define the encoded field definition
- Procedure to encode the data
- Procedure to decode the data
-
-
-
-
-



A Shallow Dive into DB Modernization

Encryption with FIELDPROC

```
SQLFPD (Field Data Type):
  SQLFST = SQL Data Type
  SQLFBL = Length in bytes
  SQLFL  = Length in characters
  SQLFP  = Field precision
  SQLFS  = Scale
  SQLFC  = CCSID
  SQLFAL = Allocated Length
```

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

```
Parms:
  functionCode    int(5) const;
  optParms        likeds(SQLFOPVD);
  decodeType      likeds(SQLFPD);
  decodeData      char(20);
  encodeType      likeds(SQLFPD);
  encodeData      char(40);
  sqlState        char(5);
  sqlMsgText      likeds(SQLFMT);
```

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

```
// Initialization
when functionCode = 8;
    Populate the “encodeType” parm

// Field encoding
when functionCode = 0;
    Transform “decodeData” into “encodeData”

// Field decoding
when functionCode = 4;
    Transform “encodeData” into “decodeData”
```

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

```
// Initialization
when functionCode = INITIALIZE;

    // Make encoded value same as decoded...
      encodeType = decodeType;

    // Change the length to 40 characters
      encodeType.SQLFL = 40;   length
      encodeType.SQLFBL = 40; bytes
```

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

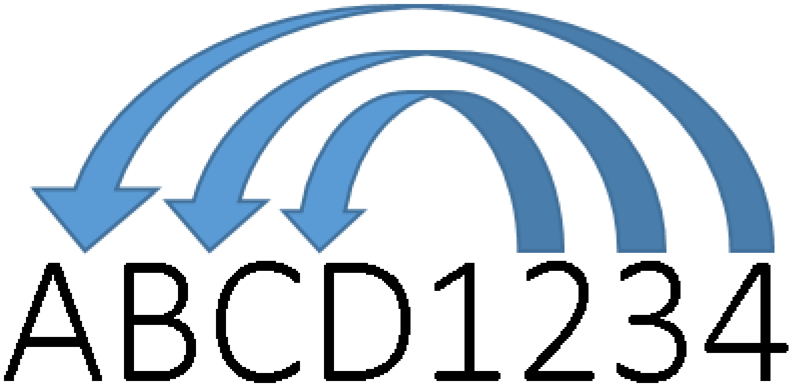
```
// ENCODE
// Called on write/update to encode the field.
when functionCode = ENCODE;

    // your logic to encrypt the data goes here...
      encodeData = EncodeCard(decodeData);
      sqlState = '00000';
```

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

// ENCODE

Take characters from the end of the string and insert them between the existing characters…



A4B3C2D11D2C3B4A

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

```
// DECODE
// Called on read to decode the field.
when functionCode = DECODE;

    // your logic to decrypt the data goes here...
      decodeData = DecodeCard(encodeData);
      sqlState = '00000';
```

(just returns every other character from encodeData)

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

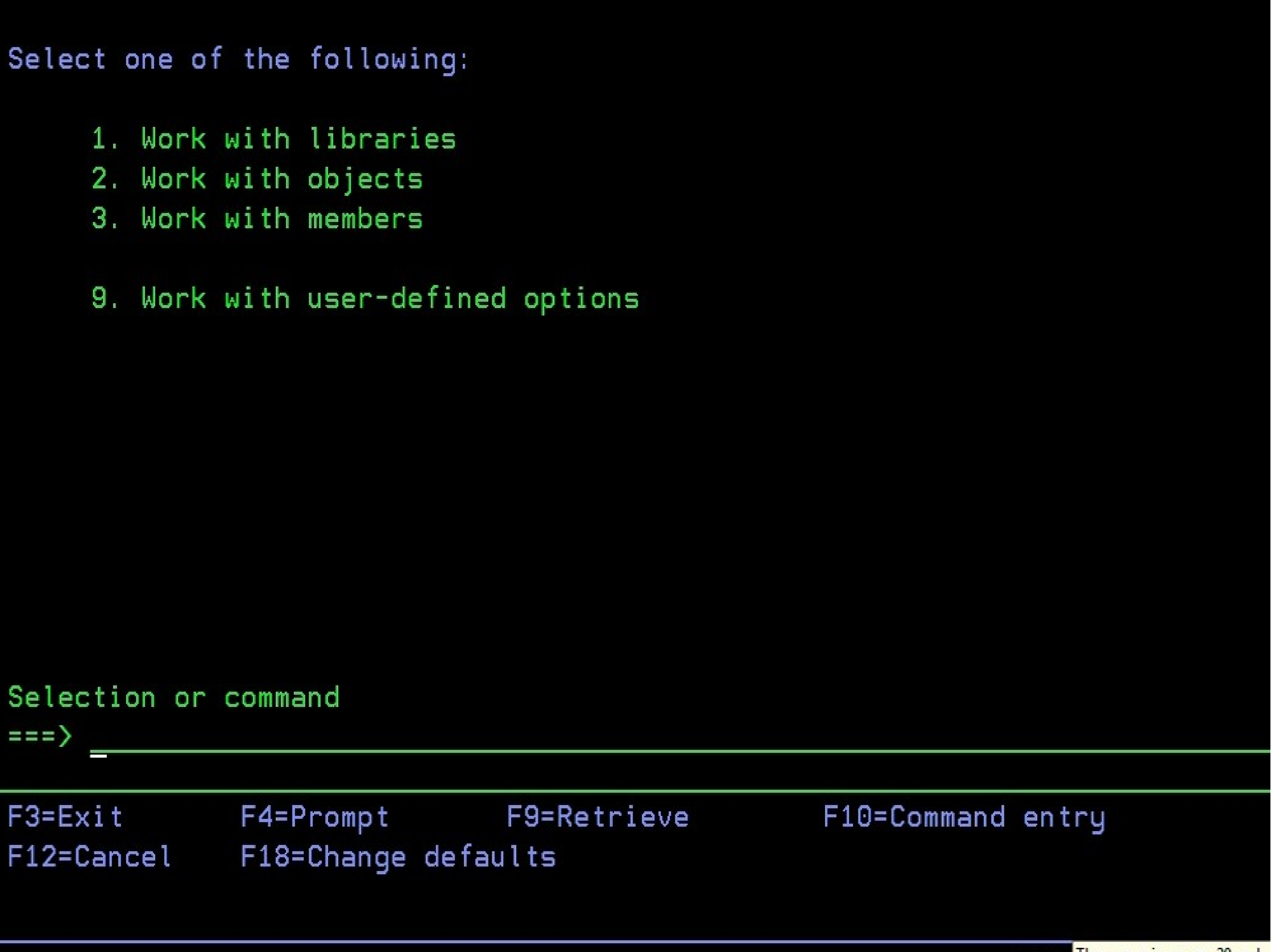
Associate the field procedure with the column:

```
Alter Table CRDTCARDSQ
Alter Column CARD_NUMBER
Set FieldProc  CC1_FLDPRC;
```

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

Field encryption in 3 minutes...



```
Select one of the following:

     1. Work with libraries
     2. Work with objects
     3. Work with members

     9. Work with user-defined options









Selection or command
===> _

F3=Exit        F4=Prompt        F9=Retrieve        F10=Command entry
F12=Cancel     F18=Change defaults
```

A Shallow Dive into DB Modernization

Encryption with FIELDPROC

Index will be built using the ENCODED value.

Be sure you understand the impact of encrypting key fields...some operations (i.e. SETLL + READ) may not work as expected.

If you are using an encrypted field in a selection the database may try to encrypt values

  WHERE credit_card = :userInput

QAQQINI   “FIELDPROC_ENCODED_COMPARISON”

A Shallow Dive into DB Modernization

There's lots of help out there...
be sure to R.T.F.M.

Security Guide for IBM i V6.1
IBM Redbook

IBM System i Security: Protecting i5/OS Data with Encryption
IBM Redbook

A Shallow Dive into DB Modernization

We need encryption…

Yes, we do… Give me a few minutes.

A Shallow Dive into DB Modernization

Adopted Authority

We need object-level authority; our credit card file should not be accessible to the public...at all.

But *some* users still need to have access to the full credit card number…*sometimes*.

A Shallow Dive into DB Modernization

Adopted Authority

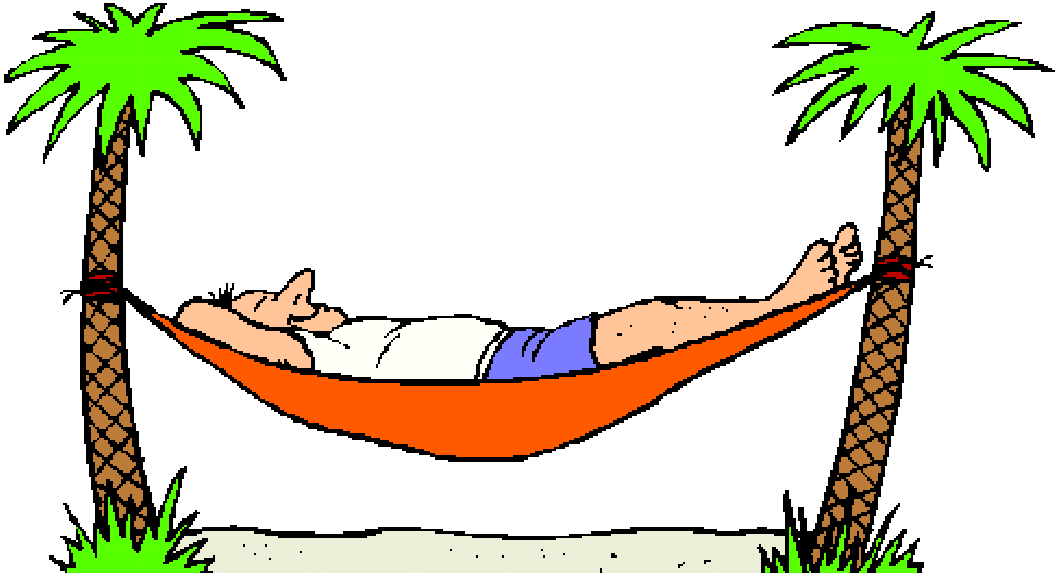
How can we give authority to a user only when they really need it??

Grant authority to the program instead of the user!

A Shallow Dive into DB Modernization

Adopted Authority

- Change the object owner to the group profile
- Change the program to run as the owner



A Shallow Dive into DB Modernization

Adopted Authority

Change the object owner to be the group profile that has the required authority:

```
CHGOBJOWN  OBJ(CC1_PGM)
           OBJTYPE(*PGM)
           NEWOWN(SOMEGROUP)
```

A Shallow Dive into DB Modernization

Adopted Authority

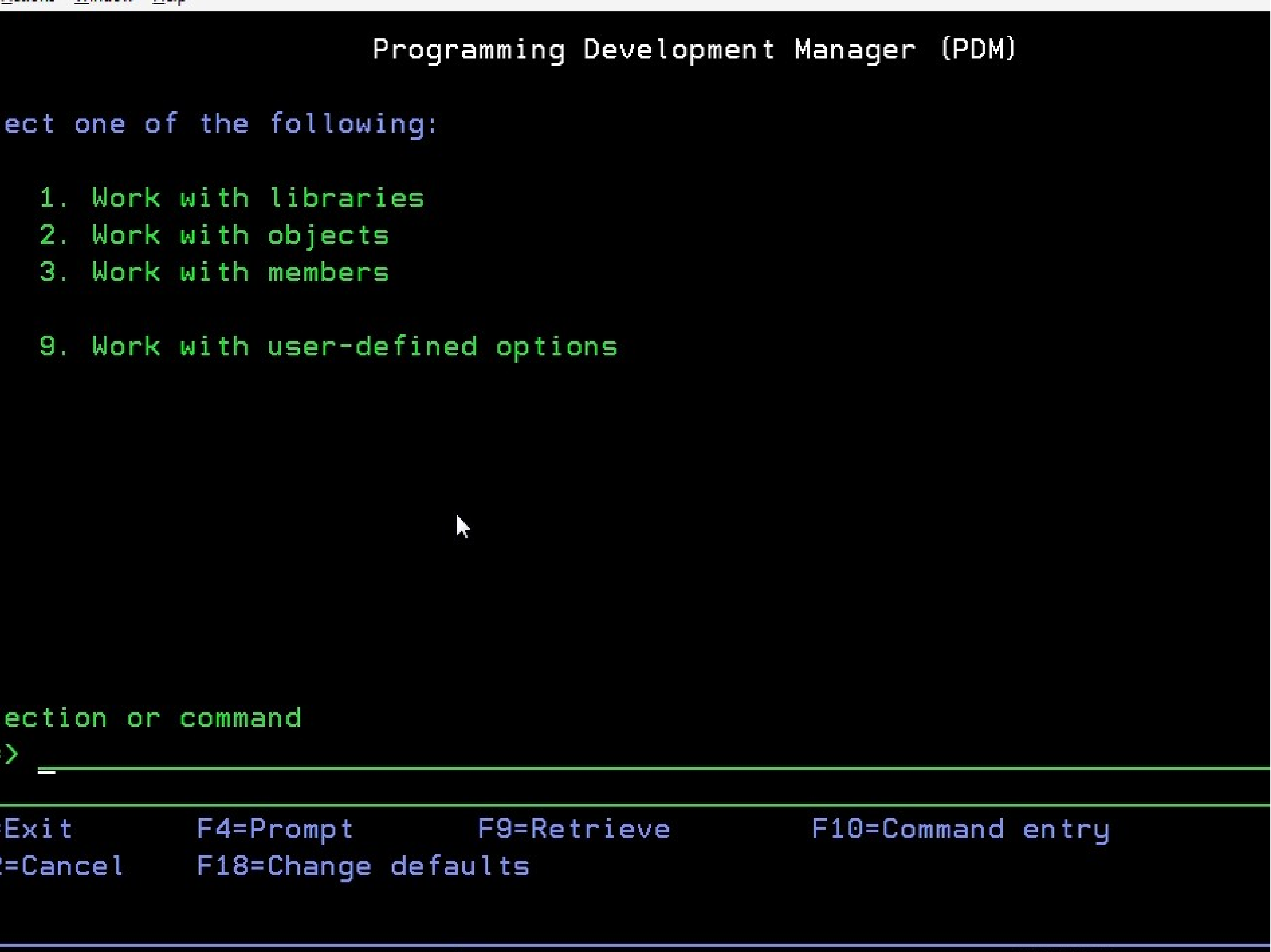
Change the program to run as *OWNER:

```
CHGPGM    PGM(CC1_PGM)
          USRPRF(*OWNER)
```

A Shallow Dive into DB Modernization

Adopted Authority

Adopted authority in 2 minutes...



```
                          Programming Development Manager (PDM)

ect one of the following:

  1. Work with libraries
  2. Work with objects
  3. Work with members

  9. Work with user-defined options




ection or command
>  _______________________________________________________________

Exit        F4=Prompt         F9=Retrieve         F10=Command entry
=Cancel     F18=Change defaults
```

A Shallow Dive into DB Modernization

There's lots of help out there...
be sure to R.T.F.M.

This was a shallow dive; there's lots that wasn't covered.

Be sure you understand YOUR requirements and YOUR environment.

There are lots of articles, white papers, Redbooks, blogs, and websites out there which can help you along the way.

There are also lots of vendors who have already RTFM and know what they're doing and can set you up right.

A Shallow Dive into DB Modernization

Questions?