```
SCR #49
  0 ( Introduction and load for DEMO data file.           MVP-FORTH)
  1 PAGE CR CR CR CR
  2 ."       ELEMENTS OF DATA BASE DESIGN " CR
  3 ."                    by"  CR
  4 ."             Glen B. Haydon" CR CR
  5 ." This demonstration data system provides a pattern for the "
  6 CR ." further development of any type of data base. "  CR
  7
  8 : PROCEED
  9    CR CR 9 SPACES  ." Enter Y to load screens "
 10    KEY  89 =  IF 50 58  DDUP INDEX CR  THRU THEN ;
 11
 12 PROCEED
 13
 14 EXIT
 15
```

```
SCR #50
  0 ( File development   l of 2                           MVP-FORTH)
  1 VARIABLE REC#  0 REC# !  ( holds the current record number )
  2 VARIABLE OPEN  0 OPEN !  ( points to the current file descript)
  3
  4 : LAYOUT  ( --- bytes/rec-2, bytes/block-1 )
  5    OPEN @ 4 + D@ ;
  6
  7 : READ  ( n --- )  ( n-th record is made current )
  8    0 MAX DUP OPEN @ 2+ @ < 0=
  9    IF ." FILE ERROR " QUIT THEN  REC# ! ;
 10
 11 : RECORD  ( n --- address of n-th record )
 12    LAYOUT  */MOD  OPEN  @  @  + BLOCK  + UPDATE ;
 13
 14 : ADDRESS  ( --- address of the current record )
 15    REC# @ RECORD  ;
```

MOUNTAIN VIEW PRESS FORTH VERSION 1.01.03

```
SCR #51
  0 ( File development    2 of 2                          MVP-FORTH)
  1 : DFIELD
  2    CREATE OVER , +    ( Create data field and leave count )
  3    DOES>  @ ADDRESS + ;    ( Leave address )
  4
  5 : TFIELD
  6    CREATE OVER , DUP , +    ( Create text field and leave count  )
  7    DOES>  D@ ADDRESS + SWAP ;   ( Leave addr and count )
  8
  9 : FILE                  ( Create a named storage allocation)
 10    CREATE  ,            ( Origin block                     )
 11    1+ ,                 ( Number of records in file        )
 12    DUP 1024 OVER / * ,  ( # nunber of bytes per block      )
 13    ,                    ( # bytes per record               )
 14    DOES> OPEN ! ;       ( When file name used,  point  to  )
 15                         ( its descriptor parameters.       )
```

```
SCR #52
  0 ( Serial Day   1 of 3                                    MVP-FORTH)
  l : D/   ( d, u --- d )
  2    SWAP OVER /MOD >R SWAP U/MOD SWAP DROP R> ;
  3 : D*   ( d, u --- d )
  4    DUP ROT * ROT ROT U* ROT + ;
  5 : $-N   ( c --- d )
  6    WORD 0 0 ROT CONVERT DDROP ;
  7
  8 : TO.SERIAL.DAY   ( d, d, d, --- u )
  9     ROT DUP 3 < IF 13 + SWAP 1 -
 10                 ELSE 1 + SWAP THEN
 11     52 - 365.25 ROT D* 100 D/ DROP
 12     SWAP 30.6001 ROT D* 10000 D/ DROP + + ;
 13
 14 : ?DATE  ." ( MM/DD/YY ) "
 15    QUERY 47 $-N  47 $-N  BL $-N  TO.SERIAL.DAY ;



SCR #53
  0 ( Serial Day   2 of 3                                    MVP-FORTH)
  1
  2 : YEARS   ( serial-day --- test-year )
  3    0 100 D* 36525 D/ DROP ;
  4
  5 : DAYS/YEARS   ( year --- days )
  6    0 36525 D* 100 D/ DROP ;
  7
  8 : TEST.YEARS  ( serial-day, test-year --- year, days )
  9    DDUP DAYS/YEARS - DUP 123 <
 10      IF DROP 1- SWAP OVER DAYS/YEARS -
 11      ELSE ROT DROP
 12      THEN SWAP 52 + SWAP ;
 13
 14 : MONTHS   ( days --- days, test-month )
 15    DUP 3267963. ROT D* 10000 D/ 10000 D/ DROP ;

          MOUNTAIN VIEW PRESS FORTH  VERSION 1.01.03


SCR #54
  0 ( Serial Day   3 of 3                                        MVP-FORTH)
  l : DAYS.TO.M/D/Y    ( years, days ---  years, days, months )
  2    MONTHS SWAP OVER 30.6001 ROT D* 10000
  3    D/ DROP - SWAP DUP 13 >
  4    IF 13 - ROT 1+ ROT ROT ELSE 1- THEN ;
  5    ?DUP
  6 : OUT.DATE   ( years, days, months --- )
  7    100 * + 0 100 D* ROT 0 D+
  8    <# # # 47 HOLD # # 47 HOLD # # #> TYPE ;
  9
 10 : CONV.SERIAL    ( serial-day --- years, days, months )
 11    DUP  YEARS  TEST.YEARS  DAYS.TO.M/D/Y ;
 12
 13 : .DATE    ( serial-day --- )
 14    ?DUP
```

```
 15    IF CONV.SERIAL OUT.DATE  ELSE ." 00/00/00" THEN ; EXIT


SCR #55
  0 ( Factors for ?$AMOUNT & .$AMOUNT                      MVP-FORTH)
  1
  2 : 0SCALE   ( u --- )  100 D* ;
  3
  4 : 1SCALE   ( u --- )  10  D* ;
  5
  6 : 2SCALE   ( u --- )         ;
  7
  8 : 3SCALE   ( u --- ) ." Input error " ;
  9
 10 CREATE NSCALE
 11 ' 0SCALE CFA ,   ' 1SCALE CFA ,  ' 2SCALE CFA ,  ' 3SCALE ,
 12
 13
 14
 15


SCR #56
  0 ( ?$AMOUNT  &  .$AMOUNT                                MVP-FORTH)
  1
  2 : SCALE    ( d --- )
  3    DPL @ 3 MIN 2 * NSCALE + @ EXECUTE ;
  4
  5 : ?$AMOUNT   ( --- double-cents )
  6    QUERY BL WORD NUMBER DPL @ 0<
  7    ABORT" INPUT ERROR"  SCALE ;
  8
  9 8 CONSTANT $SIZE
 10
 11 : .$AMOUNT   ( double-cents --- )
 12   ( Print $ amount right justified in #SIZE spaces )
 13   SWAP OVER DUP D+-  <# # # 46 HOLD #S ROT SIGN #>
 14   36 EMIT DUP $SIZE SWAP - SPACES TYPE ;
 15 EXIT

          MOUNTAIN VIEW PRESS FORTH VERSION 1.01.03

SCR #57
  0 ( DEMO File   -  Record Generation                     MVP-FORTH)
  1 0  2 DFIELD   TAG   ( a tag         )
  2   30 TFIELD   NAME    ( item name     )
  3    2 DFIELD   DAY     ( the date      )
  4    4 DFIELD   DOLLAR  ( a dollar  amount)
  5 200 ( number of records)   0 ( starting block <1024>)
  6 FILE DEMO
  7 : !NAME  ( wait for name then store it in record )
  8    NAME 32 FILL QUERY 1 TEXT PAD COUNT
  9    NAME ROT MIN CMOVE UPDATE ;
 10 : .NAME   ( print name field )  NAME TYPE ;
 11     ( The rest follow in the same way. )
 12 : !DAY  ?DATE  DAY ! UPDATE ;  : .DAY   DAY @ .DATE ;
 13 : !DOLLAR   ?$AMOUNT DOLLAR D! UPDATE ;
```

```
 14 : .DOLLAR   DOLLAR D@  .$AMOUNT ;
 15 : .REC CR REC# @ 3 .R 2 SPACES .NAME .DAY 2 SPACES .DOLLAR ;



SCR #58
  0 ( DEMO  File  -   CLEAR.DATA, INPUT, OUTPUT           MVP -FORTH)
  1
  2 ( Clear especiall tag in the 0 record in file )
  3 : CLEAR.DATA   0 READ TAG 1024 0 FILL UPDATE ;
  4
  5 ( Example of formatting for input   )
  6 : INPUT   0 READ TAG @ 1+ UPDATE DUP TAG ! READ
  7    CR CR ." ENTER NAME                      --> " !NAME
  8       CR ." ENTER DATE         -->"  !DAY ( has a format prompt)
  9       CR ." ENTER AMOUNT                    --> " !DOLLAR
 10    .REC UPDATE FLUSH ;  ( Save this record on disk )
 11
 12 ( List files 1 through the number in TAG of 0 record )
 13 : OUTPUT   0 READ TAG @ DUP 0= IF CR CR ." Empty file "
 14    DROP ELSE  1+ 1 DO FORTH I READ .REC LOOP THEN  CR CR ;
 15 EXIT

       MOUNTAIN VIEW PRESS FORTH  VERSION 1.01.03
   COLD MVP-FORTH   VERSION 1.01.03
    DR1   OK
   49 LOAD
```

ELEMENTS OF DATA BASE DESIGN

by

Glen B. Haydon

This demonstration data system provides a pattern for the further development of any type of data base.

Enter Y to load screens

```
    50  ( File development   1 of 2                          MVP-FORTH
    51  ( File development   2 of 2                          MVP-FORTH
    52  ( Serial Day   l of 3                                MVP-FORTH
    53  ( Serial Day   2 of 3                                MVP-FORTH
    54  ( Serial Day   3 of 3                                MVP-FORTH
    55  ( Factors for ?$AMOUNT & .$AMOUNT                    MVP-FORTH
    56  ( ?$AMOUNT  &  .$AMOUNT                              MVP-FORTH
    57  ( DEMO File   -  Record Generation                   MVP-FORTH
    58  ( DEMO File   -   CLEAR.DATA, INPUT, OUTPUT          MVP-FORTH
  50 51 52 53 54 55 56 57 58 OK
  DEMO OK
  CLEAR.DATA OK
  OUTPUT

  Empty file

  OK
  INPUT
```

```
   ENTER NAME                        --> EZNITH
   ENTER DATE          --> ( MM/DD/YY ) 4/21/81
   ENTER AMOUNT                      --> 18.50
     1  EZNITH                          04/21/81  $  18.50OK
   !NAME ZENITH OK
   .REC
     1  ZENITH                          04/21/81  $  18.50OK
   UPDATE FLUSH OK
   INPUT


   ENTER NAME                        --> IB
   ENTER DATE          --> ( MM/DD/YY ) 04/21/81
   ENTER AMOUNT                      --> 60.
     2  IBM                             04/21/81  $  60.00OK
   INPUT

   ENTER NAME                        --> DEC
   ENTER DATE          --> ( MM/DD/YY ) 4/21/81
   ENTER AMOUNT                      --> 103.5
     3  DEC                             04/21/81  $   103.50OK



DECIMAL OK
0 READ ADDRESS 144 DUMP DECIMAL
  C7FA    3  0  0  0  0  0  0  0   0  0  0  0  0  0  0   0  ..............
  C80A    0  0  0  0  0  0  0  0   0  0  0  0  0  0  0   0  ..............
  C81A    0  0  0  0  0  0  0  0  5A 45 4E 49 54 48 20  20  .........ZENI
  C82A   20 20 20 20 20 20 20 20  20 20 20 20 20 20 20  20
  C83A   20 20 20 20 20 20  E 2A   0  0 3A  7  0  0 49  42          .*..:.
  C84A   4D 20 20 20 20 20 20 20  20 20 20 20 20 20 20  20  M
  C85A   20 20 20 20 20 20 20 20  20 20 20 20  E 2A  0   0                .
  C86A   70 17  0  0 44 45 43 20  20 20 20 20 20 20 20  20  p...DEC
  C87A   20 20 20 20 20 20 20 20  20 20 20 20 20 20 20  20
OK
 OK
OUTPUT
  1  ZENITH                        04/21/81   $   18.50
  2  IBM                           04/21/81   $   60.00
  3  DEC                           04/21/81   $  103.50

OK
 OK
 OK
: STATEMENT   CR CR 20 SPACES ." STATEMENT" CR CR OUTPUT
   CR CR ." TOTAL VALUE " 33 SPACES  0 0 0 READ TAG @ 1+ 1
   DO _ READ DOLLAR D@ D+ LOOP .$AMOUNT CR CR CR ;  OK
 OK
 OK
STATEMENT


                     STATEMENT


  1  ZENITH                        04/21/81   $   18.50
  2  IBM                           04/21/81   $   60.00
  3  DEC                           04/21/81   $  103.50
```

TOTAL VALUE $ 182.00

OK