

OBJETIVO

- Luego de leído y comprendido este documento el estudiante habrá APRENDIDO a trabajar con la SENTENCIA FETCH y el concepto de CURSOR en COBOL
- Sentencias SQL embebidas en código COBOL cuyo resultado emite más de 1 fila o CURSOR

ESPECIFICACIONES

¿Cuál es el concepto de CURSOR en COBOL?

En COBOL, un cursor es una herramienta que se utiliza para recorrer fila por fila los resultados de ejecutar una consulta SQL (embebida en código COBOL) sobre una base de datos, como DB2.

Es fundamental cuando se quieren procesar múltiples registros de manera secuencial dentro de un programa.

Para hacerlo más gráfico; un CURSOR tiene un tratamiento similar a la utilización de un archivo QSAM dentro del código COBOL. Cada registro del archivo QSAM es equivalente a cada fila del resultado de la ejecución de una query SQL embebida.

Por lo tanto, también habrá que:

- 1) DEFINIR EL CURSOR (mediante sentencia DECLARE CURSOR)
- 2) ABRIR EL CURSOR (mediante sentencia OPEN CURSOR)
- 3) LEER EL CURSOR (mediante sentencia FETCH)
- 4) CERRAR EL CURSOR (mediante sentencia CLOSE CURSOR)

¿Cómo funciona un cursor en COBOL?

1. **DECLARE CURSOR:** Se declara el cursor con una sentencia SQL que define la query SQL a ejecutar.
2. **OPEN CURSOR:** Se abre el cursor para ejecutar la consulta; recién en este momento las filas resultantes quedan disponibles en memoria propia del código COBOL con dicho CURSOR EMBEBIDO.

3. **FETCH**: Se recupera (**lee**) una fila a la vez del conjunto de filas que se guardaron en memoria en el momento de ejecución del OPEN CURSOR.
4. **CLOSE CURSOR**: Se cierra el cursor cuando ya no se necesita. Se recomienda realizar esta ejecución inmediatamente que haya finalizado de ejecutar las funcionalidades previstas dentro de cada sentencia FETCH, a efectos de liberar memoria del programa; **un recurso muy importante** que se comparte con el resto de los usuarios.

Ejemplo básico: dentro del código resuelto en:

KC02788.ALU9999.FUENTE(PGMSIN28)

¿Por qué la sentencia FETCH?

La sentencia **FETCH en DB2** se utiliza para recuperar filas de una tabla de resultados generada por **un cursor**. Es esencial cuando querés procesar múltiples registros uno por uno dentro de un programa, especialmente en lenguajes como COBOL o en procedimientos almacenados.

¿Por qué usar **FETCH**?

- **SELECT** solo recupera una vez: Al hacer un **SELECT** simple (o sea que, solo devuelve una fila), se obtienen todos los datos de una vez. Pero si la resolución de dicho **SELECT** está representada por varias filas y se necesita accederlas (por ejemplo, para mostrarlas en pantalla o hacer cálculos), se necesita un **CURSOR**.
- **FETCH** permite recorrer los resultados: Una vez que se abre un cursor con **OPEN**, se utiliza la sentencia **FETCH** para traer la siguiente fila disponible.
- **Acceso a tablas DB2 totalmente controlado**: Se pueden manejar errores, condiciones especiales (como fin de datos o filas), y aplicar lógica personalizada para cada uno de ellos.

A continuación se brinda un ejemplo: tomado desde el código resuelto en:

KC02788.ALU9999.FUENTE(PGMSIN28)

La ejecución de este código COBOL PGMSIN28 está en: KC02788.ALU9999.FUENTE(EJESIN28)

DECLARE CURSOR:

```

UBA - wc3270
File Options Keypad
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT KC02788.ALU9999.FUENTE(PGMSIN28) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR_
009200 *****
009300 * DEFINE CON DECLARE EL CURSOR
009400 *****
009500 EXEC SQL
009600 DECLARE ITEM_CURSOR CURSOR
009700 FOR
009800 SELECT A.TIPCUEN,
009900        A.NROCUEN,
010000        A.SUCUEN,
010100        A.NROCLI,
010200        B.NOMAPE,
010300        A.SALDO,
010400        A.FECSAL
010500 FROM KC02787.TBCURCTA A
010600      INNER JOIN
010700      KC02787.TBCURCLI B
010800      ON A.NROCLI = B.NROCLI
010900      WHERE A.SALDO > 0
011000
011100 END-EXEC.
011200
011300 77 FILLER PIC X(26) VALUE '* FINAL WORKING-STORAGE *'.
011400 *****
011500 PROCEDURE DIVISION.
011600 *****
011700 *
011800 *
011900 * CUERPO PRINCIPAL DEL PROGRAMA *
012000 *
012100 *****
012200 MAIN-PROGRAM.
012300
012400 PERFORM 1000-I-INICIO THRU
012500 1000-F-INICIO.
F1=Help F2=Split F3=Exit F4=Expand F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Cancel
Display Filter View Print Options Search Help
*EDIT -SDSF
.B TS000008 004/015

```

La ejecución de este código COBOL PGMSIN28 está en: KC02788.ALU9999.FUENTE(EJESIN28)

OPEN CURSOR:

```

UBA - wc3270
File Options Keypad
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT KC02788.ALU9999.FUENTE(PGMSIN28) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR_
015600
015700 *****
015800 * APERTURA DEL CURSOR
015900 * EN ESTE INSTANTE QUEDARÁN LAS FILAS DISPONIBLES PARA EL PROG.
016000 *
016100 *****
016200 EXEC SQL
016300 OPEN ITEM_CURSOR
016400 END-EXEC.
016500
016600 IF SQLCODE NOT EQUAL ZEROS
016700 MOVE SQLCODE TO WS-SQLCODE
016800 DISPLAY '* ERROR OPEN CURSOR = ' WS-SQLCODE
016900 MOVE 9999 TO RETURN-CODE
017000 SET WS-FIN-PROCESO TO TRUE
017100 END-IF.
017200
017300
017400 1000-F-INICIO. EXIT.
017500
017600 *****
017700 *
017800 * CUERPO PRINCIPAL DEL PROGRAMA
017900 *
018000 *****
018100 2000-I-PROCESO.
018200
018300 *****
018400 * SE LEEN LAS FILAS DEL CURSOR
018500 *
018600 *****
018700 EXEC SQL
018800 FETCH ITEM_CURSOR
018900 INTO
F1=Help F2=Split F3=Exit F4=Expand F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Cancel
Display Filter View Print Options Search Help
*EDIT -SDSF
IB TS000008 005/015

```

La ejecución de este código COBOL PGMSIN28 está en: KC02788.ALU9999.FUENTE(EJESIN28)

FETCH CURSOR:

```

UBA - wc3270
Options Keypad
File Edit Edit_Settings Menu Utilities Compilers Test Help
IT      KC02788.ALU9999.FUENTE(PGMSIN28) - 01.00      Columns 00001 00072
Command ==> 2000-I-PROCESO.      Scroll ==> CSR_
8100
8200
8300
8400      *****
8500      * SE LEEN LAS FILAS DEL CURSOR
8600      *
8700      EXEC SQL
8800          FETCH   ITEM_CURSOR
8900              INTO
9000                  :DCLTBCURCTA.WS-SALDO,
9100                  :DCLTBCURCTA.WS-TIPCUEN,
9200                  :DCLTBCURCTA.WS-NROCUEN,
9300                  :DCLTBCURCTA.WS-SUCUEN
9400      END-EXEC.
9500
9600
9700      EVALUATE TRUE
9800          WHEN SQLCODE EQUAL ZEROS
9900              DISPLAY 'ACCESO DB2 OK '
0000              MOVE WS-SALDO    TO REG-SALDO
0100              MOVE WS-TIPCUEN  TO REG-TIPCUEN
0200              MOVE WS-NROCUEN  TO REG-NROCUEN
0300              MOVE WS-SUCUEN   TO REG-SUCUEN
0400              PERFORM 3000-I-GRABAR      THRU
0500                  3000-F-GRABAR
0600
0700      *****
0800      * SQLCODE = +100 INDICA FIN DEL CURSOR
0900      *      NO HAY MÁS FILAS PARA PROCESAR
1000      *****
1100      WHEN SQLCODE EQUAL +100
1200          DISPLAY 'FINAL DEL CURSOR'
1300          SET WS-FIN-PROCESO TO TRUE
1400
1500      F1=Help      F2=Split      F3=Exit      F4=Expand      F5=Rfind      F6=Rchange
1600      F7=Up        F8=Down       F9=Swap      F10=Left     F11=Right    F12=Cancel
Display Filter View Print Options Search Help
*EDIT -SDSF
TS000008      005 /044

```

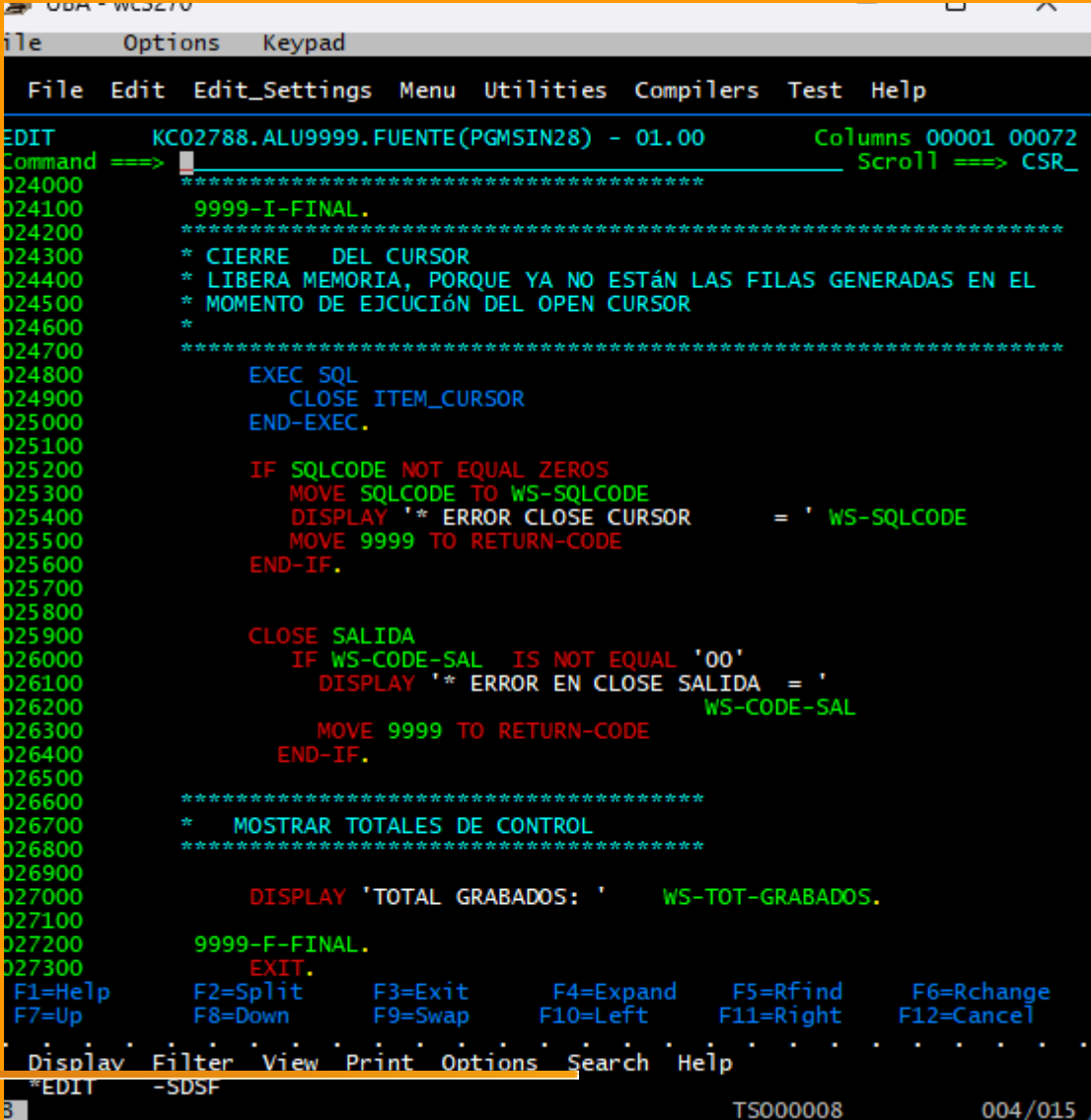
La ejecución de este código COBOL PGMSIN28 está en: KC02788.ALU9999.FUENTE(EJESIN28)

Repetir FETCH en un bucle hasta que no haya más filas; por ese motivo está dentro del rótulo:

PERFORM 2000-I-PROCESO THRU

2000-F-PROCESO UNTIL WS-FIN-PROCESO.

CLOSE CURSOR:



```

024000 *****
024100 9999-I-FINAL.
024200 *****
024300 * CIERRE DEL CURSOR
024400 * LIBERA MEMORIA, PORQUE YA NO ESTÁN LAS FILAS GENERADAS EN EL
024500 * MOMENTO DE EJECUCIÓN DEL OPEN CURSOR
024600 *
024700 *****
024800 EXEC SQL
024900 CLOSE ITEM_CURSOR
025000 END-EXEC.
025100
025200 IF SQLCODE NOT EQUAL ZEROS
025300 MOVE SQLCODE TO WS-SQLCODE
025400 DISPLAY '* ERROR CLOSE CURSOR = ' WS-SQLCODE
025500 MOVE 9999 TO RETURN-CODE
025600 END-IF.
025700
025800
025900 CLOSE SALIDA
026000 IF WS-CODE-SAL IS NOT EQUAL '00'
026100 DISPLAY '* ERROR EN CLOSE SALIDA = '
026200 WS-CODE-SAL
026300 MOVE 9999 TO RETURN-CODE
026400 END-IF.
026500
026600 *****
026700 * MOSTRAR TOTALES DE CONTROL
026800 *****
026900
027000 DISPLAY 'TOTAL GRABADOS: ' WS-TOT-GRABADOS.
027100
027200 9999-F-FINAL.
027300 EXIT.

```

F1=Help F2=Split F3=Exit F4=Expand F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Cancel

Display Filter View Print Options Search Help

*EDIT -SDSF TS000008 004/015

RESUMIENDO, **FETCH** es como pasar las páginas de un libro de resultados:

- permite leer una fila a la vez, con total control sobre el proceso.