





- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

PRECOMPILACIÓN

- Cuando se debe compilar un programa que contiene sentencias DB2, el job de compilación debe tener un paso más, llamado precompilación, que debe ejecutarse antes de la compilación COBOL.
- La precompilación transforma las sentencias SQL del programa, a calls a DB2. Dichos calls pueden ser compilados sin error por el compilador COBOL.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

PACKAGES - VENTAJAS

- Evita el excesivo re-bind de aplicaciones:
 - Cuando cambia un programa que contiene sentencias DB2, se debe rehacer la estructura de control para la ejecución de las sentencias SQL.
 - Utilizando packages, el cambio puede efectivizarse precompilando nuevamente el programa, y generando un package que será tomado en tiempo de ejecución por el plan.
- Posibilita la ejecución remota de bind:
 - Se puede realizar el bind en forma remota, es decir, desde otro DB2.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

PACKAGES - VENTAJAS

BIND PACKAGE(DESTO)

MEMBER(RTOU001) -

FLAG(I) EXPLAIN(NO) -

OWNER(ALTBAAN) -

QUALIFIER(ALTBAAN) -

ISOLATION(CS) VALIDATE(BIND)



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

PLAN

- Un plan es un objeto DB2 que contiene 1 o ambos de los siguientes elementos :
 - Una lista de nombres de packages
 - La forma original de las sentencias SQL, de uno o más DBRM
- Cada programa que contenga sentencias DB2, requiere un plan.
- Un plan se crea usando el subcomando DB2 BIND PLAN.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

PLAN

- La composición del plan se determina por cláusulas presentes en el subcomando BIND PLAN:
 - Si la cláusula presente es MEMBER, significa que esa instalación no trabaja con packages. La cláusula MEMBER contiene una lista de DBRM, con la estructura de control para la ejecución de las sentencias SQL en el programa correspondiente.
 - Si la cláusula presente es PKLIST, significa que esa instalación trabaja con packages. La cláusula PKLIST determina los packages disponibles para el plan.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

PLAN

BIND PLAN(NTOD0011) -

PKLIST(DESTO.RTOU001, -

ARQ.QC2CAUS, -

ARQ.QC2CAUT, -

ARQ.QC2CSQ1, -

ARQ.QC2CSQ2, -

ARQ.QC2CSQ3, -

ARQ.QC2CSQ4, -

ARQ.QC2CFTO) -

FLAG(I) EXPLAIN(NO) -

OWNER(ALTBAAN) -

QUALIFIER(ALTBAAN) -

ISOLATION(CS) RETAIN



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

BIND

- Es el proceso por el cual se crea un package o un plan.
- La conexión con DB2 es el resultado de un BIND o REBIND.
- Ambos subcomandos permiten la especificación del identificador de autorización (owner), del plan o package.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

BIND

- Si el bind de un package o de un plan es ejecutado con el parámetro VALIDATE(BIND) todos los objetos DB2 referenciados en el plan o package, así como los privilegios necesarios para el acceso a los mismos, deben existir en el momento del BIND.
- Si el bind de un package o de un plan es ejecutado con el parámetro VALIDATE(RUN) todos los objetos DB2 referenciados en el plan o package, así como los privilegios necesarios para el acceso a los mismos, deben existir en el momento de la ejecución.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

UNIDADES DE TRABAJO

- Una unidad de trabajo comienza por el inicio de un proceso o por la terminación de la unidad de trabajo previa.
- Una unidad de trabajo finaliza por una operación COMMIT, por una operación ROLLBACK o por la terminación del proceso.
- Las operaciones de commit o rollback afectan sólo a los resultados de las operaciones ejecutadas en la unidad de trabajo que ellos finalizaron.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

COMMIT

- Los cambios que se realizan en una unidad de trabajo, que afectan a las filas de una tabla (inserts, updates y deletes) se realizan en un área de trabajo de DB2.
- Recién cuando finalice la unidad de trabajo con un COMMIT, estos cambios se verán reflejados físicamente en las tablas.
- Por default, al finalizar un programa con DB2, el mismo realiza un commit automático.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

INCLUDE

- La sentencia include incluye el copy dentro del programa.
- Al ser una sentencia DB2, ésta se resuelve en el tiempo de precompilación.
- Cuando un programa tiene DB2, se debe utilizar INCLUDE en vez de la sentencia COBOL COPY, ya que, en caso que una variable sea utilizada en un comando DB2, el precompilador la tomará como no existente.

EXEC SQL INCLUDE CT001010 END-EXEC.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

SELECT

- La sentencia SELECT dentro de un programa, se utiliza para acceder a un registro puntual de una tabla o vista.
- A la sentencia select que conocemos, debemos agregar el parámetro INTO, donde se debe indicar en qué variable COBOL el DB2 nos va a dejar el registro seleccionado.

```
EXEC SQL

SELECT *

INTO: DCLTT004-COD-EXCEP

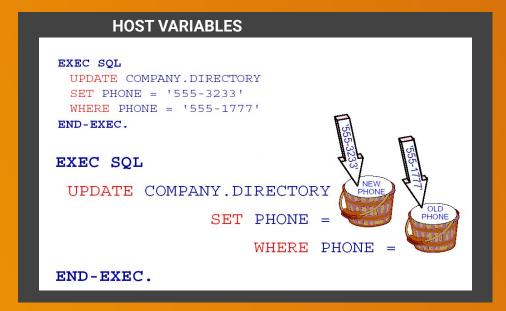
FROM TT004_COD_EXCEP

WHERE COD_EXCEP = :CT004011-COD-EXCEP

AND COD_ADMIN = :CT004011-COD-ADMIN
```

END-EXEC.

- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete





- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

IN COBOL, C

COBOL

```
WORKING-STORAGE SECTION.

01 NEWPHONE PIC X(10)

01 OLDPHONE PIC X(10)

.

PROCEDURE DIVISION.
.

ACCEPT OLDPHONE.
.

ACCEPT NEWPHONE.
.

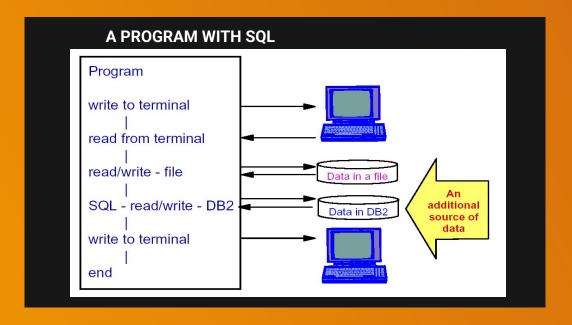
EXEC SQL
UPDATE FRANCE.PARIS
SET PHONE = :NEWPHONE
WHERE PHONE = :OLDPHONE
END-EXEC.
```

```
char newphone [10];
char oldphone [10];
fscanf (stdin, newphone, %s);
fscanf (stdin, oldphone, %s);

EXEC SQL
UPDATE FRANCE.PARIS
SET PHONE = :newphone
WHERE PHONE = :oldphone;
```

- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

SELECT





- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

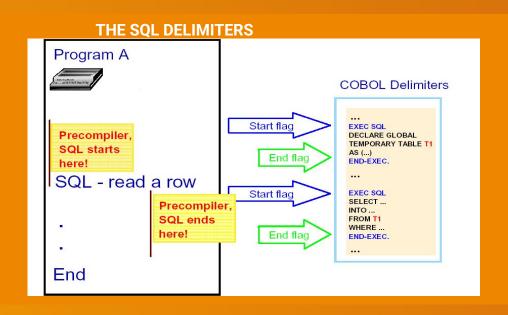
SELECT

Pasos para codificar SQL en un programa:

- Declarar área de comunicaciones(SQLCA)
- Delimitar y codificar sentencias SQL
- Manejar condición de excepciones
- Utilizar variables de host.

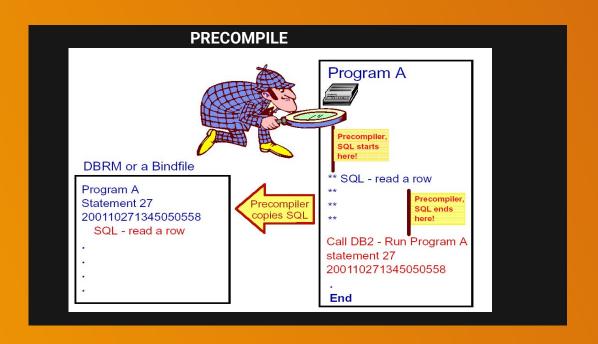


- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

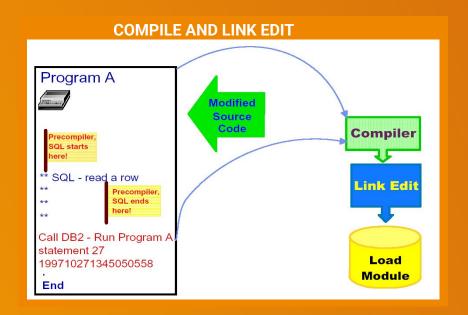




- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

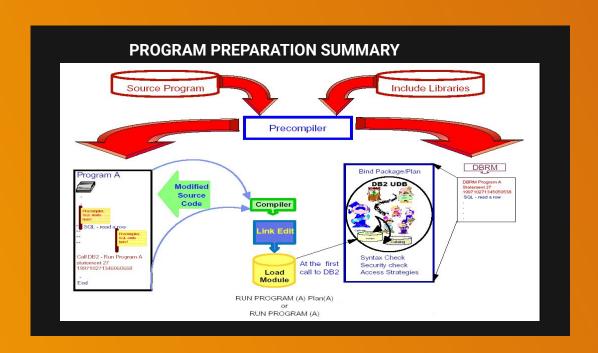


- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete



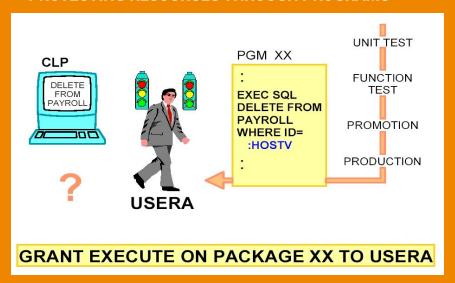


- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete



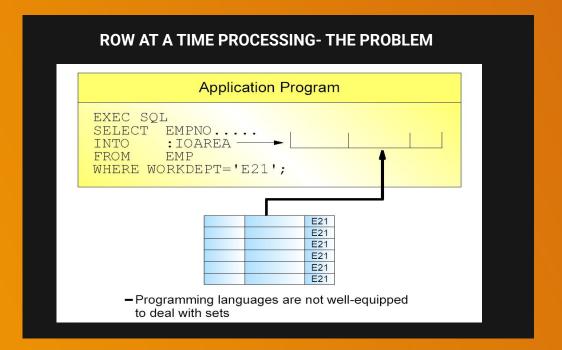
- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

PROTECTING RESOURCES THROUGH PROGRAMS





- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete





- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

SQL CODE

- La respuesta de toda operación que se ejecute sobre objetos DB2, queda en la variable SQLCODE.
- Dentro de los programas, se debe preguntar por el sqlcode después de cualquier operación, para saber si la misma pudo resolverse satisfactoriamente.

```
EXEC SQL SELECT *

INTO:DCLTTO04-COD-EXCEP FROM TTO04_COD_EXCEP
WHERE COD_EXCEP = :CTO04011-COD-EXCEP
END-EXEC.
IF SQLCODE EQUAL ZEROES
NEXT SENTENCE
ELSE
IF SQLCODE = 100
DISPLAY 'NO ENCONTRO EL REGISTRO'
ELSE
PERFORM 9999-ABEND-SQL

THRU 9999-ABEND-SQL-EXIT.
```



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

CURSORES

- Como hemos visto, La sentencia SELECT dentro de un programa, se utiliza para acceder a un registro puntual de una tabla o vista.
- Si queremos acceder a TODOS los registros de una tabla, que cumplan con una condición, debemos construir un cursor.
- Un cursor es un objeto lógico. Se crea cuando realizamos el OPEN del cursor, y deja de existir una vez que hacemos el CLOSE del mismo.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

CURSORES

- Para utilizar un cursor, debemos :
 - Declararlo
 - Abrirlo
 - Leer cada una de sus filas
 - Cerrarlo



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

CURSORES

- CURSOR:
 - La definición del cursor no crea al mismo, sino que sólo define qué datos contendrá una vez que el mismo se abra.
 - Al ser una definición, la misma puede realizarse en la Working Storage Section, no es necesario que la misma se encuentre en la Procedure Division

EXEC SQL DECLARE CUR-AJUSTES CURSOR FOR

SELECT A.TRAMITE, A.FECHA, A.HORA, A.RESPONSABLE,
A.FEULMOD, A.HORULMOD, A.USUARIO, A.NUMTER, B.*

FROM VTK3201_ETAPATRA A, TT001_AJUSTES B

WHERE A.ETAPA = 'ENAJ'

AND A.TRAMITE = B.NRO_AJU

AND B.COD_ADMIN = :CT001011-COD-ADMIN

ORDER BY B.COD_EXCEP

END-EXEC.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

CURSORES

APERTURA DEL CURSOR:

- La apertura del cursor crea al mismo, por lo tanto, si en la definición del mismo hay variables COBOL, las mismas deben tener los valores correspondientes antes de realizar el OPEN.
- Como a toda sentencia DB2, se debe evaluar el SQLCODE.

EXEC SQL
OPEN CUR-AJUSTES
END-EXEC.

IF SQLCODE NOT = ZERO
DISPLAY 'ERROR EN EL OPEN DEL CURSOR DE AJUSTES'
PERFORM 9999-ABEND-SQL
THRU 9999-ABEND-SQL-EXIT
END-IF.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

CURSOR DECLARATION

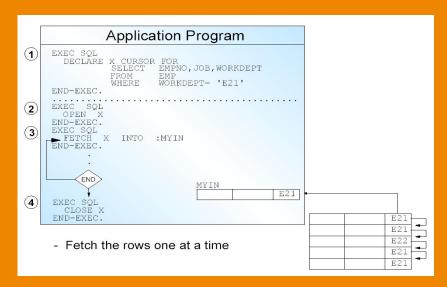
EXEC SQL DECLARE cursor-name CURSOR (WITH HOLD) FOR select-statement END-EXEC.

- A cursor:
- -Is required for SELECT of multiple rows
- -Is never used for INSERT
- -May be reused (CLOSE + new OPEN)
- -Will be closed at COMMIT unless declared with WITH HOLD
- Multiple cursors:
 - -May be defined in a program
 - -May work with the same table
 - -May be open simultaneously



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

ROW AT A TIME PROCESSING- CURSORS





- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

PROGRAM STRUCTURE WITH CURSORS

Select all projects for a given department and list the name of the person responsible for it.

Terminal Input Area:

```
01 INP DEPT PIC X(3).
```

Host Structure:

```
01 DCLPROJ.
02 PROJNO
02 PROJNAME.
49 PROJNAME-LEN PIC X(6).
02 DEPTNO
02 EMPNO
02 LASTNAME
49 LASTNAME-LEN PIC X(3).
04 LASTNAME-LEN PIC X(6).
07 LASTNAME-LEN PIC X(6).
```

Cursor Declaration:

```
EXEC SQL DECLARE DPROJ CURSOR FOR SELECT PROJNO, PROJNAME, DEPTNO, EMPNO, LASTNAME FROM PROJ, EMP WHERE RESPEMP EMPNO AND DEPTNO = :INP_DEPT
```



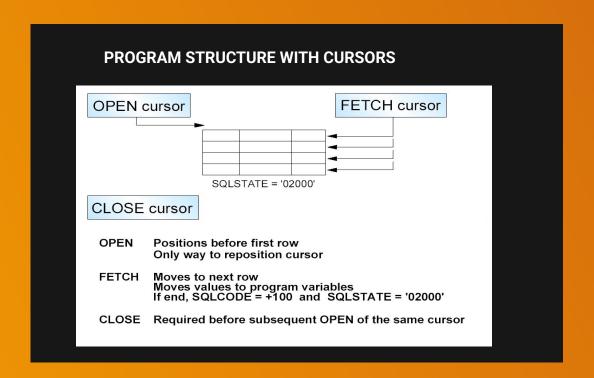
- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

PROGRAM STRUCTURE WITH CURSORS

Read Terminal Input into INP_DEPT: ACCEPT INP_DEPT. Open Cursor: EXEC SQL OPEN DPROJ END-EXEC. Fetch a Row: EXEC SQL FETCH DPROJ INTO : DCLPROJ END-EXEC. Fetch Successful? Go to routine to write output to terminal which will then fetch another row. IF SQLSTATE = '00000' GO TO DISP-DEPT-RESPEMP All Projects Processed ? ELSE IF SQLSTATE = '02000' THEN GO TO FETCH-ROW-EXIT. Close Cursor: EXEC SQL CLOSE DPROJ END-EXEC.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete





- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

CURSORES

CURSORES FOR UPDATE:

- Cuando se desea modificar filas, que se encuentran en un cursor, el mismo debe ser declarado con el parámetro FOR UPDATE.
- Este parámetro permite modificar el dato de la fila que se ha leído, o borrar dicha fila de la tabla.

EXEC SQL DECLARE CUR-AJUSTES CURSOR FOR SELECT *
FROM TT001_AJUSTES
WHERE COD_ADMIN = :CT001011-COD-ADMIN FOR UPDATE OF ESTADO, FEC_ESTADO END-EXEC.

- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

UPDATING/DELETING WITH A CURSOR

*Row-at-a time UPDATE or DELETE (positioned UPDATE/DELETE) -declare

-FOR UPDATE OF ... OPTION

```
EXEC SQL DECLARE CUR1 CURSOR FOR
SELECT EMPNO, LASTNAME, SALARY, HIREDATE
FROM EMP
WHERE WORKDEPT = :DPT
FOR UPDATE OF SALARY
END-EXEC.
```

-and use

-WHERE CURRENT OF ... OPTION

```
MOVE 'A00' TO DPT.
EXEC SQL OPEN CUR1 END-EXEC.
EXEC SQL FETCH CUR1 INTO :EMPNO,:NAME,:SAL,:HD END-EXEC.

IF THEN
EXEC SQL UPDATE EMP
SET SALARY = :SAL *1.10
WHERE CURRENT OF CUR1

END-EXEC.

IF THEN
EXEC SQL DELETE FROM EMP
WHERE CURRENT OF CUR1

END-EXEC.
```

-Branch back to fetch and process next row



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

CURSORES

LECTURA DE UN CURSOR:

- La lectura del cursor se hace mediante la instrucción FETCH.
- El parámetro INTO debe completarse con las variables correspondientes, de acuerdo a las columnas seleccionadas en el DECLARE, en el mismo orden que éste.

EXEC SQL FETCH CUR-AJUSTES INTO:CTK32011-TRAMITE,:CTK32011-FECHA, :CTK32011-USUARIO,:CTK32011-NUMTER, :CT001011-REGISTRO :CT001011-IMPORTE END-EXEC. EVALUATE SQLCODE WHEN +0 ADD CT001011-IMPORTE TO WS-LEIDO-IMP WHEN +100 SET FIN-CURSOR TO TRUE WHEN OTHER PERFORM 9999-ABEND-SQL THRU 9999-FABEND-SQL END-EVALUATE.

SQL embebido



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

READ ONLY CURSORS

EXEC SQL DECLARE cursor-name CURSOR FOR select-statement FOR FETCH/READ ONLY END-EXEC.

- The SELECT statement contains
 - -FOR FETCH ONLY/FOR READ ONLY
 - -ORDER BY
 - -UNION or UNION ALL
- •The first SELECT contains
- -DISTINCT
- -a FUNCTION or EXPRESSION
- The outer subselect contains
 -GROUP BY / HAVING
- •The same table is used in SELECT statement and SUBQUERY
- •Select from multiple tables (join)
- A nested table expression is used in the first FROM
- •Isolation UR is used without FOR UPDATE OF

A read-only cursor cannot be the target of a positioned UPDATE/DELETE



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

CURSORES

CIERRE DE UN CURSOR:

Cuando se cierra un cursor, el mismo deja de existir

EXEC SQL

CLOSE CUR-AJUSTES

END-EXEC.

IF SQLCODE NOT EQUAL ZEROES

PERFORM 9999-ABEND-SQL THRU 9999-ABEND-SQL-EXIT

END-IF.



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

UPDATE

Permite modificar todas las filas de la tabla o vista que coincidan con las condiciones fijadas. Menos performante que el for update.

```
EXEC SQL

UPDATE VTK0501_TRAMITE

SET ESTADO = 'ENVD',

FECENVIO = :WS-FECHA-HOY,

FEULMOD = :WS-FECHA-HOY,

HORULMOD = :WS-TIME-AUX,

USUARIO = 'PTOE100',

NUMTER = 'BTCH'

WHERE TRAMITE = :CTK32011-TRAMITE

END-EXEC.
```



- Introducción al DB2
- Conceptos Básicos
- Sentencias SQL
- SQL Embebido
- Precompilación
- Packages
- Plan
- Bind
- Commit
- Rollback
- Include
- Select
- Sqlcode
- Cursores
- Update
- Delete

DELETE

Permite borrar todas las filas de la tabla o vista que coincidan con las condiciones fijadas

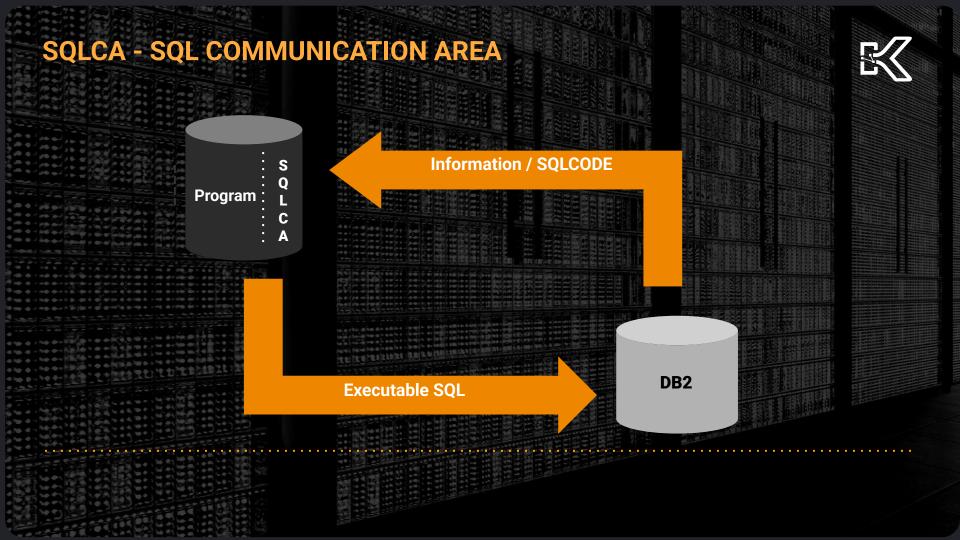
MOVE CTK05011-TRAMITE TO CTK32011-TRAMITE.

EXEC SOL

DELETE FROM VTK3201_ETAPATRA
WHERE TRAMITE = :CTK32011-TRAMITE

END-EXEC.





02 ... CODING AIDS



SQLCODE:

- Detailed
- Possibily platform dependent
- Numeric

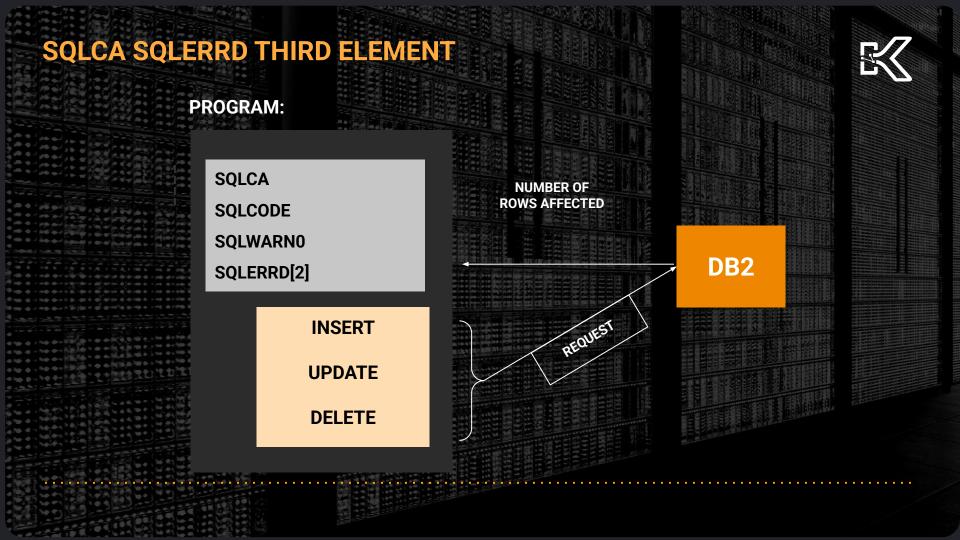
SQLSTATE:

- First two characters general
- All five characters more information
- Not as detailed as SQLCODE but not as platform dependent

02 ... SQLCA - CODES



CONDITION	INTEGER SQLCODE	CHAR(1) SQLWARNO	REQUEST STATUS
ERROR	<0		FAILED
WARNING	>0 & <> 100	or 'W'	SATISFIED WITH SPECIAL CONDITION
NOTFOUND	+ 100		(MORE) DATA NOT FOUND SQLSTATE= "02000"
SUCCESS	0	AND''	SUCCESS



02 ... CODING AIDS



INCLUDE statement and INCLUDE files

to add SQLCA/SQLDA structures to program and to provide function prototypes

¡Muchas gracias!

