Advantage™ VISION:Report® Interface to DB2®

Reference Guide

16.1

Computer Associates®

ILREF161.PDF/D23-002-011

Contents

Index

Chapter

1

Introduction

Advantage™ VISION:Report® Interface to DB2® is an integrated facility that allows the use of standard, as well as extended, embedded SQL requests in Advantage VISION:Report programs. This interface extends the reporting and data manipulation capabilities of Advantage VISION:Report to DB2 and SQL/DS databases and tables.

In VISION:Report Release 16.1 documentation:

- Advantage VISION:Report will be referred to as VISION:Report.
- Advantage VISION:Report Interface to DB2 will be referred to as VISION:Report Interface to DB2. The previous name for this interface product was VISION:Interface for DB2 with VISION:Report.
- The terms z/OS™, OS/390®, and MVS® are used interchangeably.
- The terms DB2 and SQL/DS™ are used interchangeably, although most users think of DB2 as applicable to MVS and SQL/DS to VSE. See the appropriate IBM® manual for descriptions and exact syntax of DB2 and SQL/DS statements.

New Features and Enhancements

VISION:Report Interface to DB2 for z/OS, Release 16.1 features:

- Support for IBM DB2 Versions 6 and 7.
- Enhanced Attach capability – In addition to the default CALL Attach and TSO Attach (batch and foreground), IMS™ Attach is now supported.

  You can use the same VISION:Report program with CALL Attach, IMS Attach, or TSO Attach, using JCL appropriate to the environment.

  For additional information, see the chapter IMS Attach, TSO Attach, and CALL Attach in this guide.
- Access stored procedures using the CALL command.

  For additional information, see the chapter Calling Stored Procedures in this guide.

Release Dependent Features

Some SQL and DB2 statements, options, and features are release dependent. This guide specifies those restrictions when appropriate. Refer to your systems programmer to determine the release of your operating system and which DB2/SQL statements, options, and features are supported at your site.

Note: VISION:Report Interface to DB2 is not supported under VM SQL/DS.

Supported SQL Requests

The following standard SQL requests are supported in both versions of VISION:Report Interface to DB2:

- SELECT
- DECLARE TABLE (Non-executable statement)
- DECLARE CURSOR (For SELECT only)
- OPEN
- FETCH
- CLOSE
- WHENEVER condition GOTO VISION:Report-label (or sequence number)
- WHENEVER condition CONTINUE
- WHENEVER condition STOP
- BEGIN (Non-executable statement)
- END (Non-executable statement)
- CONNECT (Different for MVS and VSE)

The following statements are not available for VSE.

- CALL
- SET hostvariable = USER
- SET hostvariable = CURRENT TIMEZONE
- SET hostvariable = CURRENT TIME
- SET hostvariable = CURRENT TIMESTAMP
- SET hostvariable = CURRENT SERVER
- SET hostvariable = CURRENT DATE
- SET CURRENT SQLID
- SET CURRENT PACKAGESET

In addition, the following SQL requests are supported for full functions:

- INSERT INTO table-name
- UPDATE table-name (searched update)
- UPDATE ... WHERE CURRENT OF ...
- DELETE FROM table-name (searched update)
- DELETE FROM ... WHERE CURRENT OF ...
- ALTER DATABASE/INDEX/TABLE
- COMMENT ON
- COMMIT (Avoid when using an IMS Attach, use IMS CKPT call instead.)
- CREATE ALIAS/DATABASE/INDEX/STOGROUP/ SYNONYM/TABLE/TABLESPACE/VIEW
- DECLARE ... statement
- DROP TABLE/INDEX/VIEW/SYNONYM
- EXECUTE
- EXECUTE IMMEDIATE
- GRANT
- LABEL
- LOCK
- PUT (VSE only)
- PREPARE
- REVOKE
- ROLLBACK (Avoid when using an IMS Attach, use IMS ROLB statement instead.)

Notes:

- Some statements do not have the same syntax in DB2 and SQL/DS (for example, EXPLAIN and CONNECT). Where the two systems are different, it is best to use the EXECUTE IMMEDIATE for that statement. Where the system does not allow that statement to be executed dynamically, VISION:Report will support it with static calls.
- DESCRIBE is not supported; you can get the equivalent through the PREPARE statement.
- EXPLAIN is not supported, but it could be the object of an EXECUTE IMMEDIATE statement.
- SET CURRENT SERVER is not supported in this release.

In order to make the use of SQL within VISION:Report easier, a simplified SQL request has been added to VISION:Report Interface to DB2. This request is:

```
WHENEVER condition STOP
```

WARNING! This simplified SQL request is not available in any IBM-supported DB2 environment. Simplified SQL statements are discussed in detail in the chapter Simplified SQL.

Note: VISION:Report Interface to DB2 has more complex extensions. These are described in detail in the chapter Data Retrieval Using Embedded SQL in this guide.

Contacting Total License Care (TLC)

TLC is available Monday-Friday 7 AM - 9 PM Eastern time in North America and 7 AM - 7 PM United Kingdom time. Additionally, 24-hour callback service is available for after hours support. Contact TLC for all your licensing requirements.

Be prepared to provide your site ID for product activation.

To activate your product, use one of the following:

Location	Phone	Email
North America:	800-338-6720 (toll free) 631-342-5069	help@licensedesk.cai.com
Europe:	00800-1050-1050	euro.tlc@ca.com

If your company or local phone service does not provide international access, please call your local Computer Associates® office and have them route you to the above number.

Australia:	1-800-224-852	
New Zealand:	0-800-224-852	
Asia Pacific:	800-224-852	
Brazil:	55-11-5503-6100	
Japan:	Not available	jpntlc@ca.com

Contacting Computer Associates

For further technical assistance with this product, contact Computer Associates Technical Support on the Internet at http://supportconnect.ca.com. Technical support is available 24 hours a day, 7 days a week.

Chapter

2

Operational Specifications

This chapter describes the operation specifications and requirements for VISION:Report Interface to DB2.

Required Computer Associates Software

VISION:Report Interface to DB2 requires VISION:Report.

Required IBM Software

z/OS, OS/390, or MVS

VISION:Report Interface to DB2:

- Can be used from TSO, IMS, or from a batch job.
- Detects the environment in which it is running and calls the appropriate DB2 interface.

When running CNTLPREP, you must correctly specify the version and release of DB2 you are using in order to prevent errors due to release dependency at runtime.

At runtime, the DB2 load library must be included in the STEPLIB/JOBLIB or LINKLIB, because the IBM DB2 interface is dynamically loaded.

The Distributed Data Facility (DDF) functions in VISION:Report Interface to DB2 must be run under IBM DB2 Version 5.1 or later (MVS only).

VSE

VISION:Report Interface to DB2 (previously known as VISION:Interface for SQL/DS with VISION:Report) requires the installation of IBM program product SQL/DS Version 3 or later.

At runtime, the SQL/DS phase library must be included in the JCL and LIBDEF statements, or in the VSE concatenation list.

The Distributed Data Facility (DDF) functions in VISION:Report Interface to DB2 are not available under VSE at this time.

The SQL database must be active (in multi-user mode) within its own partition for the interface to work properly.

Product Specifications

The following specifications apply to SQL statements in VISION:Report programs using VISION:Report Interface to DB2:

- VISION:Report data names used as host variables in SQL statements can be up to 34 characters long. However, do not use names containing a comma (,), parentheses open (( ) and close ( )), period (.), single quote ('), asterisk (*), slash (/), or a plus sign (+). These characters could be misinterpreted and cause unpredictable results.
- Up to 18 (default) SQL statements can be active at one time. You can change this installation default up to 999. (See the *VISION:Report Interface to DB2 for z/OS Installation Guide* for more information.)
- A single SQL statement can contain up to 32K characters.

Chapter

3

SQL Concepts

VISION:Report Interface to DB2 uses embedded SQL (Structured Query Language) to access and manipulate data from DB2 tables. To make effective use of VISION:Report Interface to DB2, you must understand the concepts of SQL. This chapter introduces how to use SQL. To learn more about the powerful capabilities of SQL, see the appropriate IBM DB2 or SQL/DS Reference Manual.

Definition of SQL

SQL is the standard for data management and retrieval with relational databases. SQL is a set-oriented language. All information in SQL is handled in terms of sets of records (such as, relations) rather than one record at a time as is the case with traditional languages. The inherent power of SQL, as implemented under DB2, simplifies the process of specifying how data should be managed.

Examples

A simple SQL request looks like this:

```
SELECT EMPNO, LASTNAME, SALARY
FROM DSN8610.EMP
WHERE SALARY > 20000
ORDER BY LASTNAME;
```

This SQL request defines a set of data that can be passed from DB2 to the VISION:Report program using VISION:Report Interface to DB2. In SQL terms (with traditional terms in parentheses), an intermediate table (file) is created from data in the source table DSN8610.EMP, containing a set of rows (records), each consisting of three columns (fields), where each row has a SALARY that exceeds $20,000. In addition, the rows are sequenced in LASTNAME order.

Very complex queries and data manipulations are possible with SQL. Here is an example of a more complex SQL request:

```
UPDATE DSN8610.EMP SET SALARY = SALARY * 1.1
WHERE EMPNO IN (SELECT EMPNO FROM DSN8610.EMP, DSN8610.DEPT
WHERE (DEPTNO = 'A00' OR ADMRDEPT = 'A00')
  AND WORKDEPT = DEPTNO )
```

In this example, two tables are involved: DSN8610.EMP (employee table) and DSN8610.DEPT (department table). The request increases the salaries of all selected employees by 10%. In order to select the desired employees (all those working in division A00), the WHERE clause of the UPDATE statement retrieves a list of employees that match two conditions: where either DEPTNO or ADMRDEPT (in TDEPT) is A00 and the employee work department (in TEMPL) is the same as a DEPTNO (in TDEPT).

There is no need to qualify any of the columns mentioned since the column names are unique between the two tables. If there is a problem with name ambiguity, the names must be fully qualified (such as, DSN8610.EMP.WORKDEPT).

VISION:Report Interface to DB2 supports the complete relational retrieval and data manipulation model of SQL. This allows much of the work involved in selecting, merging, sorting, retrieving, inserting, updating, and deleting to be transferred from VISION:Report to DB2, while preserving the richness of the VISION:Report language for report writing and other tasks.

VISION:Report Interface to DB2 also allows data from other sources (such as IMS/DB, CA-IDMS®/DB databases, or VSAM files) to be merged with DB2 data, using normal VISION:Report facilities.

Chapter 4

Data Retrieval Using Embedded SQL

The basic rules for using SQL inside VISION:Report programs are similar to the rules for embedding SQL in programs written in COBOL or other programming languages.

Embedded SQL Statements

The syntax for SQL statements embedded within a VISION:Report program is:

```
EXEC SQL
  sql statement
END-EXEC [special options]
```

All SQL statements placed inside a VISION:Report program must begin with EXEC SQL.

- This defines the beginning of the SQL statement.
- All source code that follows is considered to be SQL until the SQL terminator END-EXEC is reached.
- The EXEC SQL phrase is identical to the standard phrase used to delimit embedded SQL in all IBM-supported languages.

The terminator END-EXEC is identical to the one used with COBOL.

SQL Statements

Inside an SQL statement, all standard embedded SQL rules apply, as described in the appropriate IBM DB2 or SQL/DS Reference Manual with the following exception: All VISION:Report data names (host variables in DB2 terminology) must be preceded by a colon (:). The colon eliminates confusion between DB2 names and VISION:Report data names.

- All SQL statements can be written completely freeform within the boundaries of an EXEC SQL - END-EXEC pair.
- The EXEC SQL and END-EXEC statements must be on separate lines and each must be the first word(s), except for a possible sequence number.
- If SEQCHK=YES, then sequence numbers can be on all input lines; otherwise, a sequence number should only be on the EXEC SQL line.
- VISION:Report restricts the user to one verb per line.
- The use of commas within numeric values is not permitted inside of an SQL statement.
- Non-SQL statements cannot be written with EXEC SQL or END-EXEC on a single line.

Some examples of EXEC SQL - END-EXEC pairs are:

```
100  EXEC SQL   ... SQL statement
         .....
         .....
     END-EXEC

     IF  condition
         EXEC SQL
             ... SQL statement
             ...
         END-EXEC.
```

SELECT Statement

VISION:Report Interface to DB2 supports the complete SELECT syntax provided for DB2, as defined in the appropriate IBM DB2 or SQL/DS Reference Manual. Since the complete syntax of the SELECT statement can be extremely complex, only the basic syntax is presented here.

The SELECT statement specifies which DB2 tables are used as a source for retrieval, what columns (fields) are retrieved, what selection conditions are applied, in what order the data is sorted, and whether to return summary or detail information.

- The SELECT statement can be very complex and can join two or more DB2 tables into a single view.
- You can select returned rows (records) either in DB2 (using the WHERE clause of the SELECT statement) or in VISION:Report. (It is more efficient to have DB2 do the selection.)

The syntax of a simple SELECT statement is:

```
SELECT column-list
       FROM table-name
       WHERE search-conditions
       ORDER BY column-list
```

The use of commas is required in many parts of SQL. Do not forget to put commas between each column in a list of columns or wherever the standard SQL syntax requires commas.

The SELECT statement itself merely defines the domain of retrieved data. In order to pass data to VISION:Report for reporting purposes, the SELECT statement must be used as an embedded SELECT or used in a DECLARE CURSOR statement.

Search Conditions

Search conditions used in a WHERE clause of a SELECT statement are similar to those used in the VISION:Report IF statement, except that algebraic symbols are used for defining logical comparisons instead of the two-character codes.

SQL Operators

The following is a list of the valid SQL comparison operators (and the corresponding VISION:Report codes):

DB2	Meaning	VISION:Report
>	Greater than	(GT)
<	Less than	(LT)
=	Equal	(EQ)
¬=	Not equal to	(NOT EQ)
¬>	Not greater than	(NOT GT)
¬<	Not less than	(NOT LT)

- All character literals used in search conditions must be enclosed in single quotation marks.
- Numeric values cannot have embedded commas (1,234.5 must be written as 1234.5).
- VISION:Report data names can be used as search values wherever SQL permits the use of host variables. The standard SQL syntax is followed, requiring the use of a colon before the VISION:Report data name.
- When VISION:Report data names are used as search values, the columns or values that are compared must be type compatible.
  - VISION:Report character data can be compared to any valid DB2 character type.
  - Valid VISION:Report numeric data can be compared to any valid DB2 numeric type, and so on.

See the section, Embedded SELECT Statement, in this chapter for more information about type compatibility.

- Complex search conditions can be specified using the words AND or OR between two comparisons.
- Parentheses can be used to group a set of search conditions.
- There are many other operators that can be used to search for information, including IN (list testing), LIKE (pattern matching), BETWEEN (range testing), and EXISTS (test for existing data). These operators are explained fully in the IBM DB2 or SQL/DS Reference Manual.
- There are no restrictions on the use of search conditions in VISION:Report Interface to DB2.

Embedded SELECT Statement

The embedded SELECT statement is used to retrieve a single row of data from a DB2 table. An example of the embedded SELECT statement supported in VISION:Report Interface to DB2 is:

```
EXEC SQL
  SELECT FNAME, LNAME, SALARY
  INTO :FIRST_NAME, :LAST_NAME, :SALARY_AMT INDICATOR :SALARY_IND
  FROM DYLA.TEMPL WHERE EMPNO = :IN_EMPNO
END-EXEC HOLD
```

For each column in the SELECT column list, there must be a corresponding VISION:Report field name in the INTO list (such as, a matched pair). SALARY_IND is a 2-byte binary field used to hold a null indicator. Note that there is no comma between SALARY_AMT and keyword INDICATOR.

The VISION:Report data types must be type-compatible as far as DB2 is concerned. DB2 will not convert from packed to binary, because DB2 does not support decimal points in binary fields.

Extensions to VISION:Report Field Definitions

Variable Length String

With VISION:Report Interface to DB2, there is an extension to the data type. This new data type is in the format of:

:WST4-32-S	Variable length string
or	
:STRING SQLTYPE VARCHAR	Variable length string
:STRING SQLTYPE VARCHAR(32)	Override length of variable string

New Data Types in EXEC SQL or EQU Statements

New data types that are permissible only within an EXEC SQL statement or an EQU statement (which can then only be used within an EXEC SQL statement or another EQU statement):

-S	String, for VARCHAR
-G	Fixed graphic string
-F	Floating point (long — 8-bytes)
-R	Floating point real (short — 4-bytes)
-V	Varying length graphic string

Explicitly specify SQLTYPE Field

In SQL statements, the host variable declaration is extended so you can explicitly specify the SQLTYPE field. The format is:

```
:host-variable [SQLTYPE data-type [(integer [, integer])]]
```

where data type is similar to how you would define it in a DECLARE TABLE statement (with a few extensions). Allowed values are:

INTEGER or INT	Implies the VISION:Report definition is (4)-B. Integer, if used must be 4.
SMALLINT	Implies the VISION:Report definition is (2)-B. Integer, if used must be 2.
REAL	For single precision (4-byte) floating point. Integer, if used should be 4.
DOUBLE PRECISION	For double precision (8-byte) floating point. Integer, if used should be 8.
FLOAT	For floating point. Only integer 4 or 8 should be used. Default is integer of 8. Note this differs from DECLARE TABLE in that you specify the length instead of the number of bits for the precision you want.
CHARACTER or CHAR	For a fixed length string. Integer overrides the length of the field. Integer can be in the 1 to 32K range.

VARCHAR or LONG VARCHAR	For a variable length character string. Implies the VISION:Report definition consists of two consecutive fields. A length field defined as (2)-B followed by a character field defined as (integer). ■ For fields no more than 254 bytes, it is best to use VARCHAR. ■ For fields longer than 254 bytes, use LONG VARCHAR. ■ For WHERE clauses, you cannot refer to columns if the length is greater than 254 bytes.
GRAPHIC	For a fixed length graphic (double byte) string. To VISION:Report, the field needs to be defined as twice as long as the integer. GRAPHIC (5) within DB2 is defined to VISION:Report as a 10-byte field.
VARCHAR-C	For a variable length character string that is null terminated by a X'00' (low-value). This is intended for C programmers, but can also be used by VISION:Report. If you use this method, see the *Advantage VISION:Report Advantage VISION:Forms Reference Guide* for more information about the WHEN statement and @VAL-225-228-B.
VARGRAPHIC or LONG VARGRAPHIC	For a variable length graphic (double byte) string. Describes two consecutive fields. A length field defined as (2)-B followed by a character field twice as long as the integer field. ■ For fields whose integer is no more than 127 double bytes, it is best to use VARGRAPHIC. ■ For fields longer than 127 double bytes, use LONG VARGRAPHIC. ■ For WHERE clauses, you cannot refer to columns where the length is greater than 127 double bytes.
DATE	For a date field. The integer is for 10 bytes of character data. Normally, you would use CHARACTER, but DATE can be used instead and the SQL processor will verify that the column being returned is a date field.
TIME	For a time field. The integer is for 8 bytes of character data. Normally, you would use CHARACTER, but TIME can be used instead and the SQL processor will verify that the column being returned is a time field.

TIMESTAMP	For a TIMESTAMP field. The integer is a 26-byte character field. Normally, you would use CHARACTER, but you can use TIMESTAMP instead for documentation. The SQL processor will verify the column being returned is a TIMESTAMP field.
DECIMAL, DEC, or NUMERIC	For a packed field. The first integer is the total number of packed digits to return; the second integer is the number of digits after the decimal point that are in the field to be returned. VISION:Report DEC(5,2) could be defined as EQU name (3)-P 2. For COBOL, a DEC(5,2) field is the same as saying PIC S999V99 COMP-3.
NUMERIC-COBOL	For the COBOL clause PIC S999V99 SIGN LEADING SEPARATE. For VISION:Report the above should be defined as EQU name (6)2. Note that you add a byte for the leading separate signs (+ or -). It might be easier to break the field into two contiguous fields (the + or - first). Check with your systems programmer to make sure your SQL processor handles this feature.

Supported Data Types

Supported DB2 character data types:

- CHAR 1-254 characters
- VARCHAR 1-254 characters
- LONG VARCHAR 1-32767 characters

Supported DB2 numeric data types:

- SMALLINT 2-byte binary integer
- INTEGER 4-byte binary integer
- DECIMAL 1- to 16-byte packed decimal (VISION:Report only allows destination of 1- to 8-byte packed decimal)
- FLOAT 8-byte double precision floating point
- REAL 4-byte single precision floating point

Supported DB2 graphic data types:

- GRAPHIC — 1-127 double byte characters: define to VISION:Report as 2-254 bytes
- VARGRAPHIC — 1-16383 double byte characters: define to VISION:Report as 2-32766 bytes

Supported VISION:Report character data types:

- n characters (where n = 1-32767)

Supported VISION:Report numeric data types:

- var-B — binary integers (1-, to 4-bytes long)
- var-P — packed decimal (1-, to 8-bytes long)

Indirect Support of Floating Point

VISION:Report does not support floating point directly. You can use CHAR (fixed length character strings) in place of VARCHAR (variable length character strings) and DB2 will make any needed conversions for you, except when you INSERT or UPDATE a column using a character string that is too large. (Some SQL processors will make the conversion from floating point, packed decimal, and binary fields. See your IBM DB2 Reference Manual to see what is allowed.)

For example, to define a VARCHAR to VISION:Report:

```
STRING              WST1-200-S
STRING-LEN          WST1-2-B
STRING-DATA         WST3-200
```

To DB2, STRING was defined as VARCHAR(198), as DB2 does not count the two bytes in front. To move the field to a print field, use the variable form of the MOVE:

```
MOVE STRING-DATA TO PRT1  STRING-LEN
```

- For fixed strings, DB2 fills any unneeded characters with blanks.
- For variable length strings, DB2 sets the string length, but leaves any data after the string unchanged.

If, in the previous example, you wanted to examine the last character of the string for a period (.) and if you did not find one, you could add a period and update the value. The code would be as follows:

```
SET PTA TO STRING-DATA                                  /* START OF DATA
SET PTA UP STRING-LEN                                   /* BUMP UP TO END
SET PTA DOWN 1
IF PTA1-1  NOT  =  C'.'
    MOVE C'.' TO PTA2-2
    ADD C'1'  TO STRING-LEN
    EXEC SQL
        UPDATE  table  SET  column-name = :STRING
        WHERE CURRENT OF cursor-name
    END-EXEC.
```

Cursor-name can be the same as a VISION:Report field name.

Truncation and Padding

When DB2 character data (CHAR and VARCHAR) are passed to VISION:Report character data names, the data is truncated or padded with blanks to the VISION:Report length as necessary.

VARCHAR columns are returned as fixed length (since VISION:Report does not support variable length character data).

SQL Cursors

The embedded SELECT statement can only retrieve a single row for a given set of search conditions. However, a set of search conditions describes a set of rows rather than a single row. In order to support the retrieval of a set of rows into a program, SQL uses a file-like concept called a cursor.

Using a SELECT statement, a program can define a set of data to be retrieved from DB2 tables.

- With a cursor, this data can be retrieved one row at a time, in a manner very much like the READ statements of other programming languages.
- This type of retrieval is often referred to as cursor-based retrieval.
- The SELECT statement, which is associated with a cursor, is called a cursor-based SELECT.

DECLARE CURSOR Statement

The DECLARE CURSOR statement is the heart of a cursor-based SELECT request. It defines the retrieval set with a standard SELECT statement. Since there are theoretically no limits on the size of a SELECT statement (nor on the total number of DB2 tables that can be referenced), a DECLARE CURSOR is a remarkably open-ended tool.

Each DECLARE CURSOR statement is associated with a named cursor.

- The cursor name is used in subsequent OPEN, FETCH, and CLOSE requests to associate a particular DECLARE CURSOR statement (and its set of rows as specified in the SELECT statement) with these other DB2 requests.
- A cursor can be thought of as a temporary file name for use in VISION:Report.
- There are two different cursors which can be declared by VISION:Report Interface to DB2.

The syntax of the DECLARE CURSOR statements is:

```
EXEC SQL
  DECLARE cursor-name CURSOR FOR SELECT fullselect
END-EXEC
```

or

```
EXEC SQL
  DECLARE cursor-name CURSOR WITH HOLD FOR SELECT fullselect
END-EXEC
```

The fullselect statement used in a DECLARE CURSOR request is any valid SQL SELECT statement. Remember that the INTO clause is not used in a DECLARE CURSOR, but only in an embedded SELECT request. The WITH HOLD option is available with DB2 Version 3.2 (MVS) or higher and is described in the IBM DB2 Reference Manual.

OPEN Statement

The OPEN statement activates the specified cursor. The syntax for the OPEN statement is:

```
EXEC SQL
  OPEN cursor-name
END-EXEC
```

A cursor can be repositioned to the beginning of the sequence of retrieved data by closing and reopening the same cursor name. If you try to reopen the cursor before it has been closed, an error occurs.

FETCH Statement

The FETCH statement retrieves a single row (record) of data from the DB2 table defined with the corresponding DECLARE CURSOR statement.

- The data values that are returned to VISION:Report data names must be prefixed with a colon (:).
- Each VISION:Report data name must correspond to one of the DB2 columns being retrieved in the SELECT.
- There must be the same number of VISION:Report data names as there are DB2 columns.

VISION:Report repeats the FETCH at runtime if you supply too many data names. If you supply too few, DB2 supplies only as many columns as data names.

The FETCH statement syntax is:

```
EXEC SQL
  FETCH cursor-name [INTO :qj-var1, :qj-var2,... :qj-varn]
END-EXEC
```

Using VISION:Report data names in the FETCH request is the same as using them in an embedded SELECT request. Type compatibility is enforced between DB2 columns and VISION:Report data names. See the section, Embedded SELECT Statement, in this chapter for more information about type compatibility.

CLOSE Statement

The CLOSE statement terminates use of the cursor and frees up DB2 memory. Although it is not required, it is a good idea to close each cursor before the program ends. It must be done if a cursor is to be reopened (that is, a closed cursor can be reopened and the same data will be returned).

The syntax for the CLOSE statement is:

```
EXEC SQL
  CLOSE cursor-name
END-EXEC
```

Host Variable Substitution (DB2 Extension)

Some DB2 statements do not allow host variables. However, VISION:Report Interface to DB2 allows you to use substitution variables. You cannot substitute the words EXEC, SQL, END-EXEC, or the statement verb (in the example, CREATE).

To identify a substitution variable, use the syntax:

```
:=host-variable
```

The value of the host variable is used wherever the host variable is placed in the statement. For example, suppose you had a program that read a file to create synonyms for DB2 tables. You could code the following:

```
EXEC SQL
    CREATE SYNONYM :=INF1-18  FOR  :=INF21-38
END-EXEC
```

In the previous example, :=INF1-18 is replaced by the first 18 positions of the input, and :=INF21-38 is replaced by whatever is in positions 21 through 38 of the input.

- Substitutions can be made even inside quotation marks.
- Use double colons (::) if you want a single colon.
- Extended substitutions can be made anywhere in the DB2 statements ALTER, COMMENT, CREATE, DROP, GRANT, LABEL, LOCK, and REVOKE.
- Extended substitutions cannot be used in the DB2 statements BEGIN, CLOSE, COMMIT, CONNECT, END, FETCH, OPEN, PUT, ROLLBACK, SET, or WHENEVER.

For the other DB2 statements, the following general rules apply to substitution variables:

- Do not use a substitute variable in place of a host variable.
- Do not use packed or binary fields with digits after the decimal point inside a literal.
- Do not use a substitute for cursor names or statement names.
- Do not use a substitute for the words WHERE CURRENT, USING, DESCRIPTOR, INDICATOR (or IND), IMMEDIATE, INTO, FROM, NAMES, LABELS, ANY, BOTH, STATEMENT, or TABLE.
- Do not use a substitute for the commas in a host variable list.

DECLARE Statement

CURSOR WITH HOLD cannot be placed in a substitution variable (MVS only).

DECLARE ... TABLE is treated as documentation and substitution is meaningless.

EXECUTE IMMEDIATE and PREPARE Statements

The EXECUTE IMMEDIATE command does not have to be in quotation marks.

```
EXEC SQL
    EXECUTE IMMEDIATE 'DROP TABLE :=table-name'
END-EXEC
```

Example — EXPLAIN Statement

The following example illustrates how to execute an EXPLAIN statement. The QUERYNO in WST1-2-B is a binary halfword that you want expanded. The statement you want to explain is in the field WST3-100. The statement does not need to be within quotation marks.

```
EXEC SQL
EXECUTE IMMEDIATE
    EXPLAIN PLAN SET QUERYNO = :=WST1-2-B
        FOR :=WST3-100
END-EXEC
```

Example — SQL Statement in Columns 1-72

The following example shows how to execute the SQL statement contained in columns 1-72. The host variable is allowed, since it contains the entire statement to execute.

```
EXEC SQL
    EXECUTE IMMEDIATE
        :INF1-72        /* statement to execute
END-EXEC
```

Example — Host Markers

The following example illustrates how to use host markers — question marks (?) — in a free-form substitution. The ? is determined by the EXECUTE or OPEN statement.

```
EXEC SQL
    PREPARE  my-stmt                    FROM
    INSERT  :=a-stmt                    VALUES(?,?,NULL)
END-EXEC
EXEC SQL EXECUTE
    my-stmt USING
    :VALUE1, :VALUE2 :IND-2
END-EXEC
```

END-EXEC Statement

The END-EXEC statement has some special options available.

HOLD

Keeps the statement available for reuse without updating the substitution values. You can use HOLD on:

- a PREPARE or EXECUTE statement,
- an INSERT statement (intent is to insert several rows within a loop),
- a CLOSE statement (intent is to reopen the cursor), and
- a searched UPDATE or DELETE.

It is assumed for a positioned DELETE.
For a positioned UPDATE, it should be used only if there is one UPDATE statement for that cursor... WHERE CURRENT OF statement for the cursor in question.

Note: The HOLD option reduces overhead in both VISION:Report and DB2 if the statement is reused. However, not using the HOLD option reduces overhead in both VISION:Report and DB2 if the statement is not reused. If you reuse a statement, use the TOKEN statement to release the statement with the HOLD option.

TOKEN token-name or token-number

Shares the same area with more than one statement.

- Token-name defines a 4-byte area that was reserved for the address of the entry used for the DB2 statement.
- Token-number reserves a 4-byte area internally to keep the address for you to refer to later using that same token-number.

Note: Only one cursor sharing the same token should be opened at a time. It is possible to have two cursors opened at a time if the token-name is PTA1-4, but that method requires many pointer variable manipulations and is not recommended. The TOKEN option can be used with the following statements:

```
DECLARE                 Statement
DECLARE                 CURSOR FOR SELECT
INSERT                  with END-EXEC  HOLD option*
DELETE                  with END-EXEC  HOLD option
UPDATE                  with END-EXEC  HOLD option*
```

You cannot use these statements with the WHERE CURRENT OF version.

Using TOKEN token-name or TOKEN token-number

With the CLOSE, EXECUTE, FETCH, positional DELETE, and positional UPDATE statements, you can replace the cursor-name or SELECT statement-name with TOKEN token-name or TOKEN token-number.

- This allows you to open a choice of cursors (possibly with different ORDER BY or WHERE clauses) and use the same FETCH and CLOSE statements for those cursors.
- It also allows you to release a statement that used the END-EXEC HOLD option.

```
100 GET INF   ATEND  150
    EXEC SQL
        INSERT INTO MY-TABLE
        (ACCOUNT, TRAN-DATE, AMOUNT)
        VALUES (:INF1-8, :INF4-16, :INF20-25-P 2D)
    END-EXEC  HOLD  TOKEN 1
    GOTO 100
150 EXEC SQLCLOSE TOKEN 1                       /* special form of close
    END-EXEC                                    /* frees up SQL area
    EXEC SQL
        DECLARE CURSOR-1 CURSOR
        CURSOR FOR SELECT
          ACCOUNT, TRAN-DATE, AMOUNT
        FROM MY.TABLE
        ORDER BY AMOUNT
    END-EXEC  HOLD  TOKEN 1
    EXEC SQL  OPEN CURSOR-1
    END-EXEC
    EXEC SQL
        WHENEVER NOT FOUND GOTO 250
    END-EXEC
200 EXEC SQL  FETCH TOKEN 1
        INTO :PRT1-8, :PRT10-17, :WST1-6-P
    END-EXEC
    MOVE WST1-6-P 2D   TO  PRT20
    PRINT
    GOTO 200
250 EXEC SQL  CLOSE  TOKEN 1
    END-EXEC                            /* no need to HOLD cursor
    GOTO EOJ
```

Exceptional Condition Processing

On the END-EXEC statement, you can specify exceptional condition branches, such as NOT FOUND, SQLERROR, and SQLWARNING, followed by the sequence number that would be taken if the exceptional condition occurred.

The following is an example of NOT FOUND on an END-EXEC statement:

```
END-EXEC   NOT FOUND 250    /* goto 250 if NOT FOUND
```

The previous statement would replace the WHENEVER NOT FOUND GOTO 250 statement in the program shown in the previous section.

The keyword STOP or CONTINUE can also be used after the exceptional condition instead of sequence number. The following is an example of a STOP after an SQLERROR condition:

```
END-EXEC   SQLERROR   STOP    /* Terminates VISION:Report processing
```

Chapter

5

Data Management Using Embedded SQL

The following SQL statements can be used to manipulate DB2 tables from VISION:Report programs:

- INSERT INTO table-name
- UPDATE table-name (searched update)
- UPDATE ... WHERE CURRENT OF ...
- DELETE FROM table-name (searched update)
- DELETE FROM ... WHERE CURRENT OF ...
- ALTER DATABASE/INDEX/TABLE
- CALL (MVS only)
- COMMENT ON
- COMMIT
- CREATE ALIAS/DATABASE/INDEX/STOGROUP/SYNONYM/TABLE/TABLESPACE/VIEW
- DECLARE ...
- DESCRIBE (Not supported; can get equivalent through PREPARE statement.)
- DROP TABLE/INDEX/VIEW/SYNONYM
- EXECUTE
- EXECUTE IMMEDIATE
- EXPLAIN (Not supported directly; can be object of EXECUTE IMMEDIATE statement.)
- GRANT
- LABEL
- LOCK
- PREPARE
- PUT (VSE only.)
- REVOKE
- ROLLBACK

The syntax of each statement is described fully in the IBM DB2 or SQL/DS Reference Manual. When used within a VISION:Report program, VISION:Report data names can be used wherever host variables are permitted. See the section, Embedded SELECT Statement, in the chapter Data Retrieval Using Embedded SQL for more information about how to enforce type compatibility.

Each execution of these statements results in substitution of the current VISION:Report values for the specified VISION:Report data names.

INDICATOR Keyword

To specify an indicator variable, code the keyword INDICATOR in front of the indicator variable (such as, INDICATOR:FIRST_NAME_IND). Optionally, you can separate the keyword and the variable with a blank space (such as, INDICATOR :FIRST_NAME_IND).

You can also code the host variable name immediately followed by the indicator variable name (such as, :FIRST_NAME_LEN:FIRST_NAME_IND).

```
   EQU  WORK-AREA             WST0-0     /* MY ENTIRE WORKAREA
   EQU  FIRST_NAME_IND       (2)-B
   EQU  FIRST_NAME_LEN       (2)-B
   EQU  FIRST_NAME           (15)   SPACES

100  EXEC SQL   FETCH TOKEN 1
         INTO  :FIRST_NAME_LEN SQLTYPE VARCHAR(8)
               INDICATOR:FIRST_NAME_IND
     END-EXEC   NOTFOUND 150
```

Chapter

6

Simplified SQL

WARNING! *Simplified SQL requests are not available in any IBM-supported DB2 environment, but are used solely to simplify using DB2 in conjunction with VISION:Report. The use of these extensions is not compatible with standard SQL.*

In many ways, the use of SQL to retrieve data from an application program can be complicated. Since VISION:Report is a powerful programmer data management language and, at the same time, a versatile end user tool, a simplified SQL extension to standard SQL has been implemented for use exclusively in the VISION:Report environment. You can use this simplified SQL extension for hiding some of the SQL complexity.

There are two SQL simplifications available with VISION:Report Interface to DB2:

```
WHENEVER condition STOP
WHENEVER NOT FOUND GOTO EOJ
```

WHENEVER Statement

The simplified WHENEVER statement makes end-of-data testing easy. For example:

```
EXEC SQL
  WHENEVER NOT FOUND STOP
END-EXEC
```

You can stack all WHENEVER condition processing into a single EXEC SQL statement:

```
EXEC SQL
     WHENEVER NOT FOUND GOTO 100 SQLERROR STOP
          SQLWARNING CONTINUE
END-EXEC
```

STOP causes VISION:Report to enter its normal end of job processing when the end of data has been reached on an SQL FETCH, without requiring a special label to be coded for STOP.

There is no restriction on its use with the other conditions (SQLWARNING and SQLERROR), but its usefulness in these situations is limited.

For SQLERROR, the message associated with the error is printed before going to EOJ.

Note: The simplified SQL extension WHENEVER condition STOP is almost identical to the syntax supported in SQL/DS for VSE.

VISION:Report does not take any default action on the WHENEVER statement, such as CONTINUE. You must code the action you want to take.

Chapter

7

Dynamic SQL

SQL Descriptor Area (SQLDA)

For dynamic SQL, you will need the use of an SQL Descriptor Area (SQLDA). This could require more than one SQLDA. This example of a VISION:Report description is based on PTA. You can change what the PTA points to, so you can reuse the definition for each SQLDA needed. For example:

```
EQU SQLDA          PTA000-000        /* Group identifier
EQU SQLDAID        (8)  C'SQLDA   '  /* Eye catcher
EQU SQLDABC        (4)-B             /* Length of SQLDA
*                                        = 16 + 44 * SQLN
EQU SQLN           (2)-B             /* Number of occurrences
*                                         of SQLVAR
EQU SQLD           (2)-B             /* Number of occurrences
*                                         actually being used
*****************************************************
* The following repeats itself as needed;           *
* To get to the next entry, SET PTA UP 44,          *
* but note that the prior definitions are no longer  *
* valid.                                            *
*****************************************************
EQU SQLVAR         PTA17-60         /* Group Length
*
EQU SQLTYPE        PTA17-18-B       /* Common Data Type
*****************************************************
* The following shows the fields under TYPE with     *
* their two values. The first value shown is         *
* "With-Null" and the second value, following the    *
* '/', is the "Without-Null-Indicator.               *
*****************************************************
*                                                   *
* TYPE                                  VALUES
*
* Date                                  384/385
* Time                                  388/389
* Timestamp                             392/393
* Varying-length character string       448/449
* Fixed-length character string         452/453
* Long character string                 456/457
* Varying-length, optionally null
*    terminated, character string       460/461
* Varying-length graphic string         464/465
* Fixed-length graphic string           468/469
* Long graphic string                   472/473
* Floating-Point                        480/481
* Decimal                               484/485
* Large Integer (binary field 2 bytes)  496/497
```

```
* Small Integer (binary field 4 bytes)  500/501
*
EQU SQLLEN         (2)-B          /* Length of sending/
*                                    receiving data
*
* For most fixed length fields, it is the length
* of the field. For most variable length fields, it is
* the maximum length of the field (do not count the
* 2 bytes in front of the variable field as part of the
* length). For graphic, double-byte character strings,
* use the number of double bytes or the number of bytes
* divided by two. For packed decimal, see redefinition
* below.
*
EQU  SQLLEN-P1      PTA19-19-B     /* Number of packed
*                                    digits in number
EQU  SQLEN-P2       (1)-B          /* Number of digits
*              after decimal point in packed number
*              i.e., for a COBOL field defined as:
*                   PIC S9(4)V99   COMP-3
*         use:      SQLLEN-P1 = 6  /* 6 digits total
*                   SQLLEN-P2 = 2  /* 2 digits, other
*                                    decimal point
EQU  SQLDATA        (4)-B          /* Addr of column
*                                    in your program
EQU  SQLIND         (4)-B          /* Addr of NULL
*                                    indicator for
*                                    column. If not
*                                    needed, you can
*                                    put LOVALUE here
EQU  SQLNAME-L      (2)-B          /* Length of column
*                                    name (used)
EQU  SQLNAME        (30)           /* Name of column*
```

It might be easier to split the above definitions into two pointers, such as PTA and PTB, with the second pointer starting with SQLVAR.

The sample program, SQL#COPY, creates the VISION:Report EQU statements for the host variables for receiving data from SQL verbs, such as SELECT. Output is in card image and in the format of a COPYBOOK.

See the chapter, Sample Programs and Output, for more information about SQL#COPY.

Placing Addresses in the SQLDA, QJADDR

To put addresses into the SQLDA (for SQLDATA and SQLIND), you need to use the program, QJADDR or use VISION:Report SET statements.

- QJADDR requires an even number of parameters to be passed to it.
- The first of each pair of parameters is the field whose address you want to place in the second 4-byte parameter.
- The SET statements are invoked in pairs.
  - The first SET statement points to a location in storage.
  - The second SET statement inserts the address of the first location into the second location in storage.

The following example uses WST1-80 as a DB2 column address and WST101-102-B as its NULL-indicator address. This places the address of WST1-80 into SQLDATA and the address of WST101-102-B into the field SQLIND.

```
CALL  QJADDR  WST1-80  SQLDATA  WST101-102-B  SQLIND
```

The order of the pairs is immaterial and the two pairs, WST1-80 / SQLDATA, and WST101-102-B / SQLIND could easily have been reversed in the calling statement.

The SET statement to perform the same action as the CALL to QJADDR are:

```
SET  PTR      WST1
SET  PTR      SAVE    SQLDATA
SET  PTR      WST101
SET  PTR      SAVE    SQLIND
```

Storage for SQLDA

To obtain storage for SQLDA for SELECT statements, you need to issue the PREPARE statement. For example:

```
EXEC SQL
    PREPARE statement-name
    SET pointer-name                                    /* such as PTA
    USING NAMES/LABELS/ANY/BOTH                         /* choose 1 of 4
keywords
    FROM :host-variable/literal...
END-EXEC
```

Note: Instead of using the SET pointer-name, you can use INTO :host-variable. VISION:Report copies the SQLDA into your :host-variable. It does not check how large your host-variable is, so if the SQLDA area is not large enough, anything following the area is overlaid.

Neither variation is as described in the IBM DB2 or SQL/DS Reference Manual. The SET option is not available (from IBM), and the INTO option works differently in VISION:Report than the way it is documented in IBM's reference manual.

Refresh SQL Pointers or Address

There will be times when you need to refresh your SQLDA pointers.

```
EXEC SQL
    ADDRESS SQLDA FOR
    { TOKEN  token-name/number  }
    { CURSOR cursor-name        }
    SET PTR pointer-name
END-EXEC
```

The previous example looks up the SQLDA for the SELECT statement currently associated with the given token/cursor and refreshes your SQLDA pointers.

Note: The previous SQL is a VISION:Report extension to standard SQL.

Chapter 8

IMS Attach, TSO Attach, and CALL Attach

There are three methods of using the Attach facility:

1 IMS Attach

2 TSO Attach (batch and foreground)

3 CALL Attach (default)

Notes:

- You do not need to change any of your current VISION:Report programs that use CALL Attach.
- You do not need to change your VISION:Report programs to use the different Attach methods. To use a specific Attach facility, you run the job with the JCL appropriate to the Attach facility.

The DB2SAMP library contains sample programs and sample JCL for VISION:Report Interface to DB2. Review the @SQLINDX member in the DB2SAMP library for JCL appropriate to each Attach facility.

If you have not restored the sample test file, ARFILE, during the installation testing process, you should do so now. In the SAMPLIB, run member ARBUILD.

The following samples demonstrate using the three Attach facilities. Note that these are samples only, and you must modify the JCL and any input statements to your installation requirements.

Using IMS Attach

MEMBER	PURPOSE
DBDGEN	Create IMS DBD for IMS SHISAM database.
PSBGEN	Create IMS PSB for IMS SHISAM database.
VSAMIMS	Copy the VSAM ARFILE file to IMS SHISAM database. If you have not done so, run ARBUILD2 in the SAMPLIB to create the VSAM ARFILE.

SQLDIMSJ	JCL to load the DB2 ACCOUNTSRECEIVABLE table from the IMS SHISAM database using IMS Attach. For a listing of SQLDIMSJ, see the following section, SQLDIMSJ for IMS Attach.

SQLDIMSJ for IMS Attach

Follow the instructions in SQLDIMSJ and modify the JCL to your installation requirements.

```
//SQLDIMSJ JOB  YOUR ACCOUNTING INFORMATION
//* ************************************************************
//*                                                            *
//* SQLDIMSJ:                                                  *
//*                                                            *
//*   JCL TO LOAD DB2 ACCOUNTSRECEIVABLE TABLE FROM            *
//*           IMS DATABASE USING THE IMS ATTACH FACILITY.      *
//*                                                            *
//*             RUNS ONLY WITH 16.1+.                          *
//*                                                            *
//* CHANGE @@@AAA.@@@BBB TO THE APPROPRIATE HLQ'S,             *
//* AS WELL AS THE DB2 LOADLIB.  (SEE <=== BELOW).             *
//*                                                            *
//* NOTE:  SAMPLE ONLY!                                        *
//*        YOU WILL NEED TO CHECK YOUR INSTALLATION'S          *
//*        REQUIREMENTS FOR DB2 AND IMS AND CHANGE             *
//*        THE JCL ACCORDINGLY.                                *
//*                                                            *
//* ************************************************************
//QJ       EXEC PGM=DFSRRC00,
//             PARM='DLI,DSNMTV01,ARFPSB,10,0000,,0,,N,0,,,,N'
//STEPLIB  DD  DISP=SHR,DSN=@@@AAA.@@@BBB.PATCHLD   <===
//         DD  DISP=SHR,DSN=@@@AAA.@@@BBB.LOADLIB   <===
//         DD  DISP=SHR,DSN=IMDIMS.IMSSYS61.RESLIB  <===
//         DD  DISP=SHR,DSN=IMDIMS.IMSSYS61.PGMLIB  <===
//         DD  DISP=SHR,DSN=DB2.DB2710.SDSNLOAD     <===
//QUIKIPDS DD  DISP=SHR,DSN=@@@AAA.@@@BBB.SAMPLIB   <===
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TR#SQL   DD  SYSOUT=*
//SYSPUNCH DD  SYSOUT=*,DCB=(LRECL=81,BLKSIZE=81)
//SYSOUT   DD  SYSOUT=*
//DFSRESLB DD  DISP=SHR,DSN=IMDIMS.IMSSYS61.RESLIB  <===
//DFSVSAMP DD  DISP=SHR,DSN=@@@AAA.ARFILE.IMS.DFSVSAMP  <===
//DDITV02  DD  DISP=SHR,DSN=@@@AAA.ARFILE.IMS.DDITV02  <===
//DDOTV02  DD  DISP=(NEW,PASS,DELETE),SPACE=(TRK,(1,1)),UNIT=SYSDA,
//             DCB=(RECFM=VB,BLKSIZE=4096,LRECL=4092)
//IEFRDER  DD  DUMMY
//IEFRDER2 DD  DUMMY
//IMSMON   DD  DUMMY
//IMS      DD  DISP=SHR,DSN=IMDIMS.IMSSYS61.PSBLIB  <===
//         DD  DISP=SHR,DSN=IMDIMS.IMSSYS61.DBDLIB  <===
//ARFBASE  DD  DISP=SHR,                        IMS ARFILE DATABASE
//             DSN=@@@AAA.ARFILE.IMS                 <===
//SYSIN    DD  DISP=SHR,DSN=@@@AAA.@@@BBB.DB2SAMP(SQLLDDB2)  <===
//
```

Using TSO Attach

You can use TSO Attach in batch using SQLDTSOJ or foreground using RUNDSN.

MEMBER	PURPOSE
SQLDTSOJ	JCL to load the DB2 ACCOUNTSRECEIVABLE table from the ARFILE using batch TSO Attach. For a listing of SQLDTSOJ, see the following section, SQLDTSOJ for TSO Attach.
RUNDSN	This is a CLIST and should be placed accordingly. This CLIST loads the DB2 ACCOUNTSRECEIVABLE table from the ARFILE using foreground TSO Attach.

SQLDTSOJ for TSO Attach

Follow the instructions in SQLDTSOJ and modify the JCL to your installation requirements.

```
//SQLDTSOJ JOB (ACCOUNTING INFO),
//     MSGCLASS=X
/*JOBPARM SYSAFF=CPUC
//* ************************************************************
//* SQLDTSOJ:                                                  *
//*                                                            *
//*   JCL TO LOAD DB2 ACCOUNTSRECEIVABLE TABLE FROM            *
//*           SORTED ARFILE USING BATCH TSO.                   *
//*                                                            *
//* CHANGE @@@AAA.@@@BBB TO THE APPROPRIATE HLQ'S,             *
//* AS WELL AS THE DB2 LOADLIB.  (SEE <=== BELOW).             *
//*                                                            *
//* NOTE:  SAMPLE ONLY!                                        *
//*        YOU WILL NEED TO CHECK YOUR INSTALLATION'S          *
//*        REQUIREMENTS FOR TSO AND DB2, AND CHANGE            *
//*        THE JCL ACCORDINGLY.                                *
//*                                                            *
//* ************************************************************
//QJ       EXEC PGM=IKJEFT01
//STEPLIB  DD  DISP=SHR,DSN=@@@AAA.@@@BBB.PATCHLD  <===
//         DD  DISP=SHR,DSN=@@@AAA.@@@BBB.LOADLIB  <===
//         DD  DISP=SHR,DSN=DB2.DB2710.SDSNLOAD DB2 LIB   <===
//SYSTSIN  DD  *
DSN SYSTEM(DE0G)
RUN PROGRAM(QUIKJOB) LIBRARY('@@@AAA.@@@BBB.LOADLIB') PLAN(QUIKJOB)
END
/*
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TR#SQL   DD  SYSOUT=*
//SYSPUNCH DD  SYSOUT=*,DCB=(LRECL=81,BLKSIZE=81)
//QUIKIPDS DD  DISP=SHR,DSN=@@@AAA.@@@BBB.SAMPLIB  <===
//SYSOUT   DD  SYSOUT=*
//SYSUT1   DD  DISP=SHR,DSN=@@@AAA.@@@BBB.ARFILE   <===
//SYSIN    DD    DISP=SHR,DSN=@@@AAA.@@@BBB.DB2SAMP(SQLLDDB2)  <===
//
```

Using CALL Attach

The CALL Attach facility is the default, and your current programs will continue to run without modifications.

SQLDCALJ is the JCL to load the DB2 ACCOUNTSRECEIVABLE table from the ARFILE using CALL Attach.

SQLDCALJ for CALL Attach

Check your installation DB2 requirements, follow the instructions in SQLDCALJ, and modify the JCL to your installation requirements.

```
//SQLDCALJ JOB (ACCOUNTING INFO),
//     MSGCLASS=X
/*JOBPARM SYSAFF=CPUC
//* ********************************************************************
//* SQLDCALJ:                                                          *
//*                                                                    *
//*   JCL TO LOAD DB2 ACCOUNTSRECEIVABLE TABLE FROM                    *
//*          SORTED ARFILE USING CALL ATTACH.                          *
//*                                                                    *
//* CHANGE @@@AAA.@@@BBB TO THE APPROPRIATE HLQ'S,                     *
//* AS WELL AS THE DB2 LOADLIB.  (SEE <=== BELOW).                     *
//*                                                                    *
//* NOTE:  SAMPLE ONLY!                                                *
//*        YOU WILL NEED TO CHECK YOUR INSTALLATION'S                  *
//*        REQUIREMENTS FOR DB2 AND CHANGE                             *
//*        THE JCL ACCORDINGLY.                                        *
//*                                                                    *
//* ************************************************v*****************
//QJ       EXEC PGM=QUIKJOB
//STEPLIB  DD  DISP=SHR,DSN=@@@AAA.@@@BBB.PATCHLD  <===
//         DD  DISP=SHR,DSN=@@@AAA.@@@BBB.LOADLIB  <===
//         DD  DISP=SHR,DSN=DB2.DB2710.SDSNLOAD    <===
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TR#SQL   DD  SYSOUT=*
//SYSPUNCH DD  SYSOUT=*,DCB=(LRECL=81,BLKSIZE=81)
//QUIKIPDS DD  DISP=SHR,DSN=@@@AAA.@@@BBB.SAMPLIB  <===
//SYSOUT   DD  SYSOUT=*
//SYSUT1   DD  DISP=SHR,DSN=@@@AAA.@@@BBB.ARFILE   <===
//SYSIN    DD    DISP=SHR,DSN=@@@AAA.@@@BBB.DB2SAMP(SQLLDDB2)<===
//
```

Running the Sample Program SQLLDDB2

You can run the same VISION:Report program, without change, to access the Attach facilities by using JCL appropriate to a specific Attach facility: SQLDIMSJ for IMS Attach, SQLDTSOJ for TSO Attach, or SQLDCALJ for CALL Attach.

SQLLDDB2 is the VISION:Report program to load the DB2 ACCOUNTSRECEIVABLE table.

Chapter

9

Calling Stored Procedures

The DB2SAMP library contains sample source and JCL for creating and invoking stored procedures.

The members and their descriptions are:

Member	Description
ACCTTAB6	Stored SQL procedure language source to create ACCOUNTSRECEIVABLE table and associated indexes for DB2 Version 6.1. This source does not include COMMIT statements.
ACCTTAB7	Stored SQL procedure language source to create ACCOUNTSRECEIVABLE table and associated indexes for DB2 Version 7.1. This source includes COMMIT statements.
SQLSTORA	Pre-compile, compile, pre-link, and link the SQL procedure language stored procedure.
SQLSTORB	Bind the stored procedure.
SQLSTORC	Define the stored procedure to the DB2 subsystem.
SQLSTORD	Call the stored procedure to create the ACCOUNTSRECEIVABLE table and associated indexes, then load the DB2 table.

The 'COLLID' parm in the ACCTTAB6 and ACCTTAB7 source must match the 'PACKAGE' name in the SQLSTORB job.

Prior to running a job that calls a stored procedure the user must have DB2 authority to execute the stored procedure. See the section on authorization in the description of the CALL statement in the *SQL Reference*.

For example:

```
GRANT EXECUTE ON PROCEDURE SPSCHEMA.STORPRCA TO JONES;
```

SQLSTORA

Follow the instructions in SQLSTORA and modify the JCL to your installation requirements.

```
//SQLSTORA JOB (ACCTINFO),MSGCLASS=X,REGION=2M
//*********************************************************************
//* FIRST OF 3 JOBS TO ADD A STORED PROCEDURE TO A DB2 SUBSYSTEM.     *
//*                                                                   *
//* THIS JOB COMPILES & LINK-EDITS A SQL STORED PROCEDURE.            *
//*                                                                   *
//* THE RESULTING MEMBER "ACCTTABL" CREATES THE ACCOUNTSRECEIVABLE    *
//* DB2 TABLE AND ASSOCIATED INDEXES.                                 *
//*                                                                   *
//* THIS JOB USES THE IBM SUPPLIED PROC "DSNHSQL" DISTRIBUTED IN      *
//* THE IBM DATASET MEMBER "####.SDSNSAMP(DSNTIJMV)", WHERE "####"    *
//* IS THE HLQ FOR YOUR DB2 PRODUCT DATASETS.                         *
//*                                                                   *
//* FOR A MORE DETAILED EXPLANATION SEE THE CHAPTER "USING STORED     *
//* PROCEDURES FOR CLIENT/SERVER PROCESSING" IN THE IBM MANUAL        *
//* "APPLICATION PROGRAMMING AND SQL GUIDE".                          *
//*                                                                   *
//* CHANGE:                                                           *
//*                                                                   *
//* 1) "????" TO THE SYSTEM NAME OF THE PROCESSOR ASSOCIATED WITH     *
//* YOUR DB2 SUBSYSTEM.                                               *
//*                                                                   *
//* 2) "@@@AAA.@@@BBB" TO THE HLQ'S OF YOUR VISION:REPORT LIBRARY.    *
//*                                                                   *
//* 3) "$$$$" TO THE HLQ OF THE DB2 SUBSYSTEM FOR THE FOLLOWING       *
//* DATASETS:                                                         *
//* //DBRMLIB DD DISP=SHR,DSN=&USER..DBRMLIB.DATA(&MEM)               *
//* //SYSLIB  DD DISP=SHR,DSN=&USER..SRCLIB.DATA                      *
//* //SYSLMOD DD DISP=SHR,DSN=&USER..RUNLIB.LOAD(&MEM)                *
//* IN THE IBM PROC "DSNHSQL".                                        *
//*                                                                   *
//* 4) "ACCTTAB?" TO "ACCTTAB6" OR "ACCTTAB7" DEPENDING ON THE        *
//* VERSION OF DB2 YOU ARE USING.                                     *
//*                                                                   *
//* YOU MAY IGNORE THE RC=04 IN THE EDCPRLK STEP FOR THE MESSAGE:     *
//*                                                                   *
//* "WARNING EDC4052: MODULE NAME TEMPNAME CHOSEN FOR GENERATED       *
//* IMPORT CONTROL STATEMENTS. IMPORT DATA 'TEMPNAME' @@TRT".         *
//*                                                                   *
//*********************************************************************
/*JOBPARM S=????
//SQLSTORA EXEC  DSNHSQL,USER=$$$$,MEM=ACCTTABL,                      -
//             PARM.PC='HOST(SQL),SOURCE',                            -
//             PARM.PCC='HOST(C),MAR(1,80),SOURCE'
//PC.SYSIN   DD    DISP=SHR,DSN=@@@AAA.@@@BBB.DB2SAMP(ACCTTAB?)
//PLKED.SYSDEFSD   DD SYSOUT=*,DCB=(LRECL=80,RECFM=FB)
//LKED.SYSIN DD *
 INCLUDE SYSLIB(DSNALI)
/*
//
```

SQLSTORB

Follow the instructions in SQLSTORB and modify the JCL to your installation requirements.

```
//SQLSTORB JOB (ACCTINFO),MSGCLASS=X,REGION=2M
//*********************************************************************
//* SECOND OF 3 JOBS TO ADD A STORED PROCEDURE TO A DB2 SUBSYSTEM.    *
//* THIS JOB BINDS A SQL STORED PROCEDURE.                            *
//*                                                                   *
//* CHANGE:                                                           *
//*                                                                   *
//* 1) "????" TO THE SYSTEM NAME OF THE PROCESSOR ASSOCIATED WITH     *
//* YOUR DB2 SUBSYSTEM.                                               *
//*                                                                   *
//* 2) "%%%%" TO THE DB2 SUBSYSTEM NAME.                              *
//*                                                                   *
//* 3) "####" TO THE HLQ FOR YOUR DB2 PRODUCT DATASETS.               *
//*                                                                   *
//* 4) "$$$$" TO THE HLQ OF THE DBRM LIBRARY FOR YOUR DB2 SUBSYSTEM.  *
//*                                                                   *
//*********************************************************************
/*JOBPARM S=????
//SQLSTORB EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD   DISP=SHR,DSN=####.SDSNLOAD
//DBRMLIB  DD   DISP=SHR,DSN=$$$$.DBRMLIB.DATA
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
  DSN SYSTEM(%%%%)
  BIND PACKAGE(QUIKJOB) MEMBER(ACCTTABL)
  END
/*
//
```

SQLSTORC

Follow the instructions in SQLSTORC and modify the JCL to your installation requirements.

```
//SQLSTORC JOB (ACCTINFO),MSGCLASS=X,REGION=2M
//*********************************************************************
//* THIRD OF 3 JOBS TO ADD A STORED PROCEDURE TO A DB2 SUBSYSTEM.     *
//* THIS JOB DEFINES A SQL STORED PROCEDURE TO THE DB2 SUBSYSTEM.     *
//*                                                                   *
//* CHANGE:                                                           *
//*                                                                   *
//* 1) "????" TO THE SYSTEM NAME OF THE PROCESSOR ASSOCIATED WITH     *
//* YOUR DB2 SUBSYSTEM.                                               *
//*                                                                   *
//* 2) "@@@AAA.@@@BBB" TO THE HLQ'S OF YOUR VISION:REPORT LIBRARY.    *
//*                                                                   *
//* 3) "%%%%" TO THE DB2 SUBSYSTEM NAME.                              *
//*                                                                   *
//* 4) "####" TO THE HLQ FOR YOUR DB2 PRODUCT DATASETS.               *
//*                                                                   *
//* 5) "$$$$" TO THE HLQ OF THE DBRM AND RUN LIBRARY FOR YOUR DB2     *
//* SUBSYSTEM.                                                        *
//*                                                                   *
//* 6) "ACCTTAB?" TO "ACCTTAB6" OR "ACCTTAB7" DEPENDING ON THE        *
//* VERSION OF DB2 YOU ARE USING.                                     *
//*                                                                   *
//* 7) "&&" TO THE VERSION AND RELEASE OF YOUR DB2 PRODUCT.           *
//*                                                                   *
//* YOU MAY IGNORE THE RC=08 FOR THE DROP OF THE STORED PROCEDURE     *
//* IF IT DOES NOT PREVIOUSLY EXIST.                                  *
//*                                                                   *
//*********************************************************************
/*JOBPARM S=????
//SQLSTORC EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD   DISP=SHR,DSN=####.SDSNLOAD
//DBRMLIB  DD   DISP=SHR,DSN=$$$$.DBRMLIB.DATA
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSTSIN  DD   *
  DSN SYSTEM(%%%%)
  RUN PROGRAM(DSNTEP2) PLAN(DSNTEP&&) -
      PARMS('/ALIGN(LHS) MIXED') -
      LIB('$$$$.RUNLIB.LOAD')
  END
/*
//SYSIN    DD   *
--#SET TERMINATOR #
DROP PROCEDURE ACCTTABL RESTRICT#
/*
//         DD   DISP=SHR,DSN=@@@AAA.@@@BBB.DB2SAMP(ACCTTAB?)
//         DD   *
#
/*
//
```

SQLSTORD

Follow the instructions in SQLSTORD and modify the JCL to your installation requirements.

```
//SQLSTORD JOB (ACCTINFO),MSGCLASS=X,REGION=2M
//*********************************************************************
//* THIS JOB CALLS THE "ACCTTABL" STORED PROCEDURE CREATED IN JOBS    *
//* "SQLSTORA", "SQLSTORB" AND "SQLSTORC" TO CREATE THE               *
//* ACCOUNTSRECEIVABLE TABLE AND ASSOCIATED INDEXES AND THEN LOADS    *
//* THE TABLE.                                                        *
//*                                                                   *
//* PRIOR TO RUNNING THIS JOB THE USER MUST HAVE AUTHORITY TO         *
//* EXECUTE THE STORED PROCEDURE. SEE THE SECTION ON                  *
//* AUTHORIZATION IN THE DESCRIPTION OF THE CALL STATEMENT IN THE     *
//* "SQL REFERENCE".                                                  *
//*                                                                   *
//* CHANGE:                                                           *
//*                                                                   *
//* 1) "????" TO THE SYSTEM NAME OF THE PROCESSOR ASSOCIATED WITH     *
//* YOUR DB2 SUBSYSTEM.                                               *
//*                                                                   *
//* 2) "@@@AAA.@@@BBB" TO THE HLQ'S OF YOUR VISION:REPORT LIBRARY.    *
//*                                                                   *
//* 3) "####" TO THE HLQ FOR YOUR DB2 PRODUCT DATASETS.               *
//*                                                                   *
//*********************************************************************
/*JOBPARM S=????
//SQLSTORD EXEC PGM=QUIKJOB
//STEPLIB  DD  DISP=SHR,DSN=@@@AAA.@@@BBB.PATCHLD
//         DD  DISP=SHR,DSN=@@@AAA.@@@BBB.LOADLIB
//         DD  DISP=SHR,DSN=####.SDSNLOAD
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TR#SQL   DD  SYSOUT=*
//SYSPUNCH DD  SYSOUT=*,DCB=(LRECL=81,BLKSIZE=81)
//QUIKIPDS DD  DISP=SHR,DSN=@@@AAA.@@@BBB.SAMPLIB
//SYSUT1   DD  DISP=SHR,DSN=@@@AAA.@@@BBB.ARFILE
//SYSIN    DD  DISP=SHR,DSN=@@@AAA.@@@BBB.DB2SAMP(SQLLOADC)
//
```

Chapter

10 Sample Programs and Output

The samples described in this chapter are based on the Accounts Receivable (AR) file/table, as distributed on the VISION:Report distribution tape.

- For MVS customers, the DB2 samples are in the DB2 sample library (DB2SAMP).
- For VSE customers, the SQL/DS samples are in the Source Sublib.

Prior to running any of the DB2 samples listed below, ensure that the following steps are completed successfully:

- Restore the regular VISION:Report sample library (SAMPLIB) from the installation tape (MVS only).
- Restore the Accounts Receivable File from the installation tape. If this was not done previously, run ARBUILD from the SAMPLIB (MVS only).
- Restore the VISION:Report Interface to DB2 sample library (DB2SAMP) from the installation tape (MVS only).
- Read the @SQLINDX member for the latest documentation.

The examples in this chapter were tested using the following partial parameter list.

```
SAVAREA=4096, SEQCHK=NO,
STMTEND=71,   STMTLCT=50,   WSTSIZE=4096,
SQLA1=8192,   SQLA2=4096,   SQLA3=1024,   SQLA4=1024,
SQLA5=8192,   SQLPLNM=      SQLSYSN=      SQLVER=04.01
```

Note: The SAVAREA and WSTSIZE parameters may be larger than the size recommended in QJOPTDB2. This is due to the unique testing environment at Computer Associates. SQLPLNM and SQLSYSN also contain values unique to Computer Associates (MVS only).

All of these examples have been extensively tested. To use these examples at your installation, you need to complete the following steps:

- Verify that the SQLPLNM and SQLSYSN (for MVS), or the USER and PASSWORD on the PRESQL macro (for VSE), match your installation's requirements.
- Modify the JCL to fit your installation standards.

- Change the input parameters to IDCAMS (such as, volume serial number).
- Change globally all @@@AAA.@@@BBB to the high level and middle level qualifiers you have chosen to associate with the VISION:Report library and file data names.
- Modify QJTSTDB2 to fit your installation needs. All of the sample job streams invoke QJTSTDB2 to minimize the amount of changes you have to make (MVS only).
  - For MVS, specific jobs can require a REGION size, JOBPARM of SYSAFF=, or COND parameters on the JOB statement; change the STEPLIB to point to the VISION:Report LOADLIB and SLIB2 to point to your DB2 LOADLIB.
  - For VSE, note the SIZE parameter on the EXEC statement; change the LIBDEF and phase library statements to point to the VISION:Report phase library and SQL/DS phase library.

  You might want to move this PROC where it would be available for all to use.

Within the DB2SAMP library, the VISION:Report program is contained in the member name as shown above. The sample JCL to run the job stream is contained in the member name with a suffix of J (that is, LOAD is the program member; LOADJ is the JCL). Required input members are noted in the previous section under Miscellaneous members.

Since most of the JCL required is in the QJTSTDB2 PROC, only the additional JCL required for files is shown in all the samples, and they would already be contained in the ***membernameJ***. The JCL must be modified to fit your installation requirements, as well as the data set/file names.

- For VSE, the VISION:Report program member, as well as any input data, needs to be in your job step so that it can be read by SYSIPT. To do this, punch the output to your ID and then update the job stream accordingly. The mechanism for doing this varies from installation to installation. You will have to remove the asterisk (*) in position 1 on any I/O parameter statements.
- For MVS, a SYSIN DD statement pointing to the member containing VISION:Report program is required, and the membernameJ also contains this already.

Members

The member names and purposes are shown in the following table.

- Samples 1 through 7 should be run in the order of the sample number.
- The JCL for the sample member names is included in the member names with a suffix of J (that is, LOAD is in LOADJ). All JCL must be modified to match your installation requirements. The actual members can vary slightly from the figures shown here.

Note: Samples 8 and 9 are samples only and do not need to be run in any particular order.

Sample #	Member	Purpose
	@SQLINDX	Description of all sample members for DB2 testing.
1	LOAD	Loads a DB2 table from the sequential ARFILE, after input file is sorted.
2	PRINT	Print DB2 table after being loaded.
3		RANDOM1 and RANDOM2 are similar, but use different input.
	RANDOM1	Uses extended host variable as input to SQL DECLARE statement.
	RANDOM2	Uses descriptor as input to SQL OPEN statement.
		Both RANDOM1 and RANDOM2 do the following: ■ Access DB2 randomly, using member RANDOMIx to provide key to DB2 table. ■ Print the input record and the record obtained, including any No Record Found conditions.
4	UPDATE	Update DB2 table randomly, using member UPDINPUT. Some logic in the program: IF AR-BALANCE is over $1,000.00 AND AR-TRANS-DATE is greater than 180 days (from today's date, USING QUIKDATE) ADD 10% to AR-BALANCE ADD 30 days to AR-TRANS-DATE Show AR-BALANCE and AR-TRANS-DATE, before and after update
5	ADD	Add records to DB2 table, using member ADDINPUT. Print updated DB2 record.

Sample #	Member	Purpose
6	DELETE	Delete records from DB2 table, using the same input as ADD. Print updated DB2 record.
7	DUMP	Dump DB2 table to VSAM file using Native VSAM. Print DB2 table.

The following are provided as samples only:

Sample#	Member	Purpose
8	DDF	Distributed Data Facility — get rows from a DB2 table in another DB2 subsystem requires DB2 Version 5.1 or later. This is for MVS only.
9	COMMIT	Program showing commands COMMIT WORK and ROLLBACK WORK.

The following miscellaneous members are all MVS, except for SQL#COPY:

Member	Purpose
QJTSTDB2	PROC for testing VISION:Report Interface to DB2. ■ Modify this PROC to fit your installation requirements. ■ Move this PROC into a PROCLIB where all can use it.
ADDINPUT	Member for records to be added/deleted. For VSE, this will have to be inserted into your job stream.
RANDOMI1	Member for records randomly retrieved in RANDOM1. For VSE, this will have to be inserted into your job stream.
RANDOMI2	Member for records randomly retrieved in RANDOM2. For VSE, this will have to be inserted into your job stream.
SQL#COPY	Sample only — extracts SQL table column information and uses it to build VISION:Report EQU statements for host variables in FETCH, SELECT, or INSERT SQL statements. Creates COPYBOOK.
UPDINPUT	Member for records to be updated. For VSE, this will have to be inserted into your job stream.

Installation samples:

Sample#	Member	Purpose
1	CNTLPREP	Create/code DBRMs. Creates preprocessor input job stream for post-processor. The first step assembles the various options using the PRESQL macro, including plan name, system name, number of cursors, DB2 release/version number, and continues in this manner. For VSE, see the *Advantage VISION:Report Interface to DB2 for VSE Installation Guide* for installing samples and sequence.
2	CNTLINST	Link edit of DB2 interface. This step must run after CNTLPREP. Run a DB2 BIND after CNTLINST.

Run a DB2 bind after CNTLINST.

You need authorization (to create/access the DB2 table) from your DB2 System Administrator before you run the sample programs.

The following is a sample MVS GRANT request that can be used:

```
GRANT ALL ON TABLE DSN8610.EMP, DSN8610.DEPT TO userid
```

If global authorization is not appropriate at your installation, request specific authority using the following GRANT requests:

```
GRANT SELECT ON TABLE DSN8610.EMP, DSN8610.DEPT TO userid
GRANT INSERT, DELETE ON TABLE DSN8610.EMP TO userid
```

The userid specified in the above GRANT requests must be the same as the userid from which the samples are to be run.

For VSE, the following example can be used:

```
CONNECT sqldba IDENTIFIED BY sqldbapw;
GRANT DBA TO DYLA;
```

Sample Programs and Output

Although the samples in this chapter were run under MVS, most of them also apply to VSE, with few or no modifications. There could be slight differences in either the VISION:Report program or the output, due to the operating system differences.

The VISION:Report program output (printouts) shown in this section can vary slightly from the actual program in the SAMPLIB, as installation requirements are different.

Program	Description
LOAD	See the section LOAD Program. The LOAD program creates and loads a DB2 table. There is no report after running the program.
PRINT	See the section PRINT Program and the section PRINT Output. The PRINT program prints a report of each row, sorted by Account Code and Account Number.
RANDOM1	See the section RANDOM1 Program and the section RANDOM1 and RANDOM2 Output. The RANDOM1 program prints a report of each row.
RANDOM2	See the section RANDOM2 Program. The RANDOM2 program prints a report identical to the report for RANDOM1 (see the section RANDOM1 and RANDOM2 Output).
UPDATE	See the section UPDATE ProgramUPDATE Program and the section UPDATE Output. The UPDATE program prints a report of each input record to be updated and any error messages.
ADD	See the section ADD Program and the section ADD Output. The ADD program prints a report of each input record to be added and any error messages.
DELETE	See the sections DELETE Program and the section DELETE Output. The DELETE program prints a report of each input record to be deleted and with any error messages.
DUMP	See the section DUMP Program and the section DUMP Output. The report is identical to the report for the PRINT program (except for those rows that have been updated), sorted by Account Code and Account Number.
DDF	See the section DDF Program. The program is meant only as an example. (MVS only.)
COMMIT	See the section COMMIT. The program is meant only as an example.

LOAD Program

```
    ************************************************************
    *                                                          *
    * LOAD:                                                    *
    *          EXAMPLE OF A VISION:REPORT PROGRAM              *
    *          FOR LOADING A DB2 TABLE.                        *
    *                                                          *
    *   NOTE:  =====> BE SURE TO CHANGE THE FIELD-NAME         *
    *          "USER-DATABASE" CONSTANT OF '????????'          *
    *          TO AN EXISTING USER-DATABASE NAME.              *
    *                                                          *
    ************************************************************
*          QSAM  ACCOUNTS RECEIVABLE  FILE DESCRIPTION
EQU  FILLER             INF
++INCLUDE ARDEFINE
    ************************************************************
    *                                                          *
    * USER DEFINED DATABASE:                                   *
    *  REPLACE QUESTION MARKS WITH THE NAME OF YOUR DATABASE.  *
    *  AS AN EXAMPLE, REPLACE '????????' WITH DYIQDPRG.        *
    *                                                          *
    ************************************************************

*  EQU USER-DATABASE       WST1-8         C'SCRUBJAY'
   EQU USER-DATABASE       WST1-8         C'????????'

* SQL DATE HOST VARIABLES -- MM/DD/YYYY DATE FORMAT
EQU  WS-TRAN-DATE       WST451-460
EQU  WS-TRAN-MONTH      WST451-452
EQU  WS-TRAN-SLASH1     WST453         C'/'
EQU  WS-TRAN-DAY        WST454-455
EQU  WS-TRAN-SLASH2     WST456         C'/'
EQU  WS-TRAN-CENTUR     WST457-458     C'19'
EQU  WS-TRAN-YEAR       WST459-460

EQU  WS-BILL-DATE       WST461-470
EQU  WS-BILL-MONTH      WST461-462
EQU  WS-BILL-SLASH1     WST463         C'/'
EQU  WS-BILL-DAY        WST464-465
EQU  WS-BILL-SLASH2     WST466         C'/'
EQU  WS-BILL-CENTUR     WST467-468     C'19'
EQU  WS-BILL-YEAR       WST469-470

* SQL NULL INDICATORS
EQU  WS-TRAN-NULL       WST471-472-B
EQU  WS-BILL-NULL       WST473-474-B

* RESULT AREA FOR QUIKDATE
EQU  JULIAN-DATE        WST475-479

* FOR CHECKING THE RESULT OF QUIKDATE'S DATE VALIDATION
EQU  QUIKDATE-RC        VAL46-49

     SORT FILE INF ON INF182-183  INF4-10

*  THE FOLLOWING CAN BE "UN-COMMENTED" IF YOU NEED TO TRACE CODE

*  EXEC SQL TRACE
*  END-EXEC

* --------->  START OF EXECUTABLE CODE    -----------*
* DELETE ANY EXISTING SQL TABLE - USUALLY A GOOD IDEA, JUST IN CASE
```

```
*  EXEC SQL DROP
*      DATABASE SCRUBJAY
*  END-EXEC SQLERROR CONTINUE
*
*  EXEC SQL CREATE
*      DATABASE SCRUBJAY
*  END-EXEC SQLERROR CONTINUE
*
   EXEC SQL DROP
       TABLE ACCOUNTSRECEIVABLE
   END-EXEC SQLERROR CONTINUE

   IF @VAL-SQL-CODE = ZERO
     MOVE C'DROP OK' TO PRT1
     PRINT.

   EXEC SQL COMMIT
   END-EXEC

* CREATE NEW SQL TABLE
   EXEC SQL CREATE TABLE ACCOUNTSRECEIVABLE
     (
      AR_ACCOUNT       CHAR(7)       NOT NULL             ,
      AR_TRAN_DATE     DATE                               ,
      AR_BILL_DATE     DATE                               ,
      AR_LAST_NAME     VARCHAR(15)   NOT NULL             ,
      AR_FIRST_NAME    VARCHAR(10)   NOT NULL             ,
      AR_MIDDLE_INIT   CHAR(1)                            ,
      AR_STREET        VARCHAR(25)                        ,
      AR_CITY_ST_ZIP   VARCHAR(25)                        ,
      AR_BALANCE       DECIMAL(9,2)  NOT NULL WITH DEFAULT,
      AR_ACCT_CODE     CHAR(2)                            ,
      AR_INST_BALANCE  DECIMAL(11,2)                      ,
      AR_INST_PAY      DECIMAL(9,2)                       ,
      AR_BAL_PARTPAY   DECIMAL(5,2)                       ,
      AR_INT_PARTPAY   DECIMAL(5,2)                       ,
      AR_NR_PAY        DECIMAL(3,0)                       ,
      AR_KEY           CHAR(9)       NOT NULL             ,
      PRIMARY KEY(AR_KEY)
     )
      IN DATABASE :=USER-DATABASE
   END-EXEC SQLERROR 300

   EXEC SQL COMMIT
   END-EXEC

* CREATE REQUIRED SQL PRIMARY INDEX
   EXEC SQL CREATE UNIQUE INDEX ACCTRECACCOUNT ON ACCOUNTSRECEIVABLE
            (AR_KEY) CLUSTER
   END-EXEC

* CREATE ANY OPTIONAL SQL INDEXES
   EXEC SQL CREATE INDEX ACCTRECFIRSTNAME ON ACCOUNTSRECEIVABLE
            (AR_FIRST_NAME, AR_BALANCE)
   END-EXEC

   EXEC SQL COMMIT
   END-EXEC

100
   GET INF ATEND 300                      /* GET ARFILE
   MOVE AR-ACCT-CODE  TO AR-KEY-ACCT-CD    /* BUILD KEY
   MOVE AR-ACCOUNT    TO AR-KEY-ACCOUNT

* VALIDATE QSAM DATE FIELDS BEFORE PASSING THEM TO SQL
   MOVE  X'FFFF' TO WS-TRAN-NULL
```

```
   CALL QUIKDATE C'02' AR-TRAN-DATE C'MMDDYY ' JULIAN-DATE
   IF   QUIKDATE-RC EQ C'0000'
      MOVE AR-TRAN-MONTH   TO   WS-TRAN-MONTH
      MOVE AR-TRAN-DAY     TO   WS-TRAN-DAY
      MOVE AR-TRAN-YEAR    TO   WS-TRAN-YEAR
      MOVE  X'0000' TO WS-TRAN-NULL.

   MOVE  X'FFFF' TO WS-BILL-NULL
   CALL QUIKDATE C'02' AR-BILL-DATE C'MMDDYY ' JULIAN-DATE
   IF   QUIKDATE-RC EQ C'0000'
      MOVE AR-BILL-MONTH   TO   WS-BILL-MONTH
      MOVE AR-BILL-DAY     TO   WS-BILL-DAY
      MOVE AR-BILL-YEAR    TO   WS-BILL-YEAR
      MOVE  X'0000' TO WS-BILL-NULL.

* INSURE RETURN CODE IS ZERO FROM QUIKDATE FUNCTION
     MOVE C'0000' TO QUIKDATE-RC

* LOAD SQL TABLE WITH VALIDATED QSAM RECORD
   EXEC SQL INSERT
     INTO ACCOUNTSRECEIVABLE
     (
      AR_ACCOUNT       ,
      AR_TRAN_DATE     ,
      AR_BILL_DATE     ,
      AR_LAST_NAME     ,
      AR_FIRST_NAME    ,
      AR_MIDDLE_INIT   ,
      AR_STREET        ,
      AR_CITY_ST_ZIP   ,
      AR_BALANCE       ,
      AR_ACCT_CODE     ,
      AR_INST_BALANCE  ,
      AR_INST_PAY      ,
      AR_BAL_PARTPAY   ,
      AR_INT_PARTPAY   ,
      AR_NR_PAY        ,
      AR_KEY
     )
     VALUES
     (
      :AR-ACCOUNT                    ,
      :WS-TRAN-DATE    :WS-TRAN-NULL ,
      :WS-BILL-DATE    :WS-BILL-NULL ,
      :AR-LAST-NAME                  ,
      :AR-FIRST-NAME                 ,
      :AR-MIDDLE-INIT                ,
      :AR-STREET                     ,
      :AR-CITY-ST-ZIP                ,
      :AR-BALANCE                    ,
      :AR-ACCT-CODE                  ,
      :AR-INSTL-BAL                  ,
      :AR-INSTL-PAY                  ,
      :AR-BAL-PARTPAY                ,
      :AR-INT-PARTPAY                ,
      :AR-NR-PAY                     ,
      :AR-KEY
     )
  END-EXEC TOKEN 1 HOLD SQLERROR 200

     GOTO 100
* SQL REJECTED RECORD BECAUSE OF DUPLICATE PRIMARY KEY--PRINT IT
200
     MOVE AR-KEY        TO PRT1
     MOVE AR-LAST-NAME  TO PRT10
     MOVE AR-FIRST-NAME TO PRT40
```

```
     MOVE C'** REJECTED RECORD **'   TO PRT60
     PRINT
     PERFORM 400 THRU 500
     GOTO 100
300
     PERFORM 400 THRU 500
     GOTO EOJ
400  MOVE VAL440-519 TO PRT1
     PRINT
     MOVE VAL520-599 TO PRT1
     PRINT
     MOVE VAL600-679 TO PRT1
     PRINT
     MOVE VAL680-759 TO PRT1
     PRINT
500  EXIT
9999END
```

PRINT Program

```
*****************************************************************
*                                                               *
* PRINT:   PRINT DB2 TABLE AFTER INITIAL LOAD (PARTIAL).        *
*                                                               *
*****************************************************************
EQU  AR-ENT-REC         WST000-000
++INCLUDE ARDEFINE

* SQL DATE HOST VARIABLES -- YYYY-DD-MM DATE FORMAT
EQU  WS-TRAN-DATE       WST451-460
EQU  WS-TRAN-CENTUR     WST451-452     C'19'
EQU  WS-TRAN-YEAR       WST453-454
EQU  WS-TRAN-DASH1      WST455         C'-'
EQU  WS-TRAN-MONTH      WST456-457
EQU  WS-TRAN-DASH2      WST458         C'-'
EQU  WS-TRAN-DAY        WST459-460

EQU  WS-BILL-DATE       WST461-470
EQU  WS-BILL-CENTUR     WST461-462     C'19'
EQU  WS-BILL-YEAR       WST463-464
EQU  WS-BILL-SLASH1     WST465         C'-'
EQU  WS-BILL-MONTH      WST466-467
EQU  WS-BILL-SLASH2     WST468         C'-'
EQU  WS-BILL-DAY        WST469-470

EQU  WS-TRAN-NULL       WST471-472-B
EQU  WS-BILL-NULL       WST473-474-B

* AR FILE PRINT LINE FORMAT
* LINE 1
EQU  PR-ACCOUNT         PRT1              /* ACCOUNT NUMBER
EQU  PR-LAST-NAME       PRT11             /* LAST NAME
EQU  PR-FIRST-NAME      PRT26             /* FIRST NAME
EQU  PR-TRAN-DATE       PRT37             /* TRANSACTION DATE
EQU  PR-BILL-DATE       PRT49             /* BILLING DATE
EQU  PR-BALANCE         PRT63             /*
EQU  PR-INSTL-BAL       PRT75             /* AMT LEFT ON INSTALLMENT
EQU  PR-INSTL-PAY       PRT91             /* PLANNED INSTALLMENT PAYMT
EQU  PR-BAL-PARTPAY     PRT104            /* ACTUAL PAYMENT
EQU  PR-INT-PARTPAY     PRT120            /* INTEREST, PART PAYMENT
```

```
    HDR 1A 1 $IPLDAT$                                          DB2
    HDR 1B   PRINT A/R FILE EXAMPLE                            PAGE $PG$
    HDR 2A 0ACCOUNT   CUSTOMER LAST, FIRST NAME  TRAN-DATE   BILL-DATE
    HDR 2B   BALANCE     INSTL-BAL     INSTL-PAY   PAYMENT     INTEREST

*   TRACE ALL

* SPECIFY SQL COLUMNS TO BE RETRIEVED
* AND THE ORDER IN WHICH THE ROWS ARE TO APPEAR.
* NOTE THAT THE COMMAS DO NOT HAVE TO BE IMMEDIATELY
* FOLLOWING THE DATANAME.
    EXEC SQL DECLARE BLUEJAY CURSOR
       FOR SELECT
      AR_ACCOUNT,
      AR_TRAN_DATE ,
      AR_BILL_DATE ,
      AR_LAST_NAME ,
      AR_FIRST_NAME  ,
      AR_MIDDLE_INIT  ,
      AR_STREET       ,
      AR_CITY_ST_ZIP ,
      AR_BALANCE ,
      AR_ACCT_CODE     ,
      AR_INST_BALANCE ,
      AR_INST_PAY    ,
      AR_BAL_PARTPAY,
      AR_INT_PARTPAY,
      AR_NR_PAY    ,
      AR_KEY
       FROM ACCOUNTSRECEIVABLE
       ORDER BY AR_KEY
    END-EXEC

    EXEC SQL OPEN
       BLUEJAY
    END-EXEC

* SPECIFY VISION:REPORT STATEMENT TO RECEIVE CONTROL
* WHEN ALL ROWS HAVE BEEN RETRIEVED
    EXEC SQL WHENEVER
       NOT FOUND GOTO 500
    END-EXEC

400
* SPECIFY THE VISION:REPORT AREAS INTO WHICH THE SQL
* COLUMNS ARE TO BE PLACED.
   EXEC SQL FETCH
     BLUEJAY INTO
      :AR-ACCOUNT,
      :WS-TRAN-DATE    :WS-TRAN-NULL  ,
      :WS-BILL-DATE    :WS-BILL-NULL  ,
      :AR-LAST-NAME,
      :AR-FIRST-NAME ,
      :AR-MIDDLE-INIT ,
      :AR-STREET,
      :AR-STATE-ZIP ,
      :AR-BALANCE       ,
      :AR-ACCT-CODE  ,
      :AR-INSTL-BAL   ,
      :AR-INSTL-PAY    ,
      :AR-BAL-PARTPAY   ,
      :AR-INT-PARTPAY,
      :AR-NR-PAY,
      :AR-KEY
   END-EXEC
```

```
   MOVE AR-ACCOUNT       TO PR-ACCOUNT
   MOVE AR-LAST-NAME     TO PR-LAST-NAME
   MOVE AR-FIRST-NAME    TO PR-FIRST-NAME

   IF WS-TRAN-NULL EQ X'FFFF'
      MOVE C'****-**-**' TO PR-TRAN-DATE
      GOTO 420.
   MOVE WS-TRAN-DATE     TO PR-TRAN-DATE

420
   IF WS-BILL-NULL EQ X'FFFF'
      MOVE C'****-**-**' TO PR-BILL-DATE
      GOTO 440.
   MOVE WS-BILL-DATE     TO PR-BILL-DATE

440
   MOVE AR-INT-PARTPAY   TO PR-INT-PARTPAY
   MOVE AR-BAL-PARTPAY   TO PR-BAL-PARTPAY
   MOVE AR-INSTL-PAY     TO PR-INSTL-PAY
   MOVE AR-INSTL-BAL     TO PR-INSTL-BAL
   MOVE AR-BALANCE       TO PR-BALANCE
   PRINT
   GOTO 400
500
   GOTO EOJ
9999END
```

PRINT Output

```
05/14/02                                    DB2              PRINT A/R FILE EXAMPLE                          PAGE    1

ACCOUNT   CUSTOMER LAST, FIRST NAME  TRAN-DATE   BILL-DATE         BALANCE     INSTL-BAL     INSTL-PAY   PAYMENT     INTEREST

8006547   TORRES         ERNESTO    1992-02-15  1992-04-24           44.99         42.41           .00       .00          .00
6002587   HAASE          ELLA       1991-04-07  1992-03-02           15.00      2,230.17        102.40     92.92         9.48
6208657   CHO PYUNG      SUH        1992-02-17  1992-06-12           32.00        261.57         44.64     43.60         1.04
7082509   ORTIZ          DISRAELI   1991-08-21  1991-11-30            5.00        272.78         46.56     45.46         1.10
6024963   HILL           GARY       1991-04-17  1992-02-26            3.80         22.38           .00       .00          .00
6044395   CANO           MICHEAL    1992-02-11  1992-04-06           15.00        195.74         33.41     32.62          .79
6059708   CHAVEZ         RAY        1992-02-09  1992-05-04           15.00         49.39           .00       .00          .00
6095631   FODIPE         MICHAEL    1992-02-12  1992-05-21           45.24        486.21         82.99     81.04         1.95
6107265   GENVARDI       G          1992-02-10  1992-06-05           43.00        419.48         71.60     69.91         1.69
6123228   SILVA          JULIAN     1990-01-05  ****-**-**            .00        273.99         46.76     45.67         1.09
8011508   HUGHES         RAY        1992-02-13  1992-06-08          178.70         47.88           .00       .00          .00
2002299   PLACIDO        ORTEGA     1992-03-16  1992-05-11          413.58        486.21         82.99     81.04         1.95
6009166   LOCKE          JEFFREY    1992-02-12  1992-05-21           15.00        234.91         40.09     39.15          .94
6112536   CHAVEZ         NORMA      1991-09-06  1992-01-01           55.00        496.84         84.80     82.81         1.99
6123317   VASGUEZ        IRENE      1990-01-05  ****-**-**            .00        496.40         84.73     82.73         2.00
6218113   AGUIERA        EMILIO     1992-02-20  1992-06-15          108.44        197.85         33.77     32.98          .79
8012644   MONTEZ         CARMEN     1992-02-07  1992-06-16           89.28        484.75         82.74     80.79         1.95
1014021   WOOD           ROGER D    1991-10-08  1992-08-16        2,059.92        287.33         49.04     47.89         1.15
2005131   WICINSKI       ALEXANDER  1991-04-10  1992-03-19          848.71        261.70         44.67     43.62         1.05
6004695   ABROWN         ELIZABETH  1991-04-11  1992-03-06           32.02        297.25         50.73     49.54         1.19
6016316   VANCE          VERNON     1991-04-20  1992-03-15           40.76      3,243.83        104.29     90.11        14.18
6041272   MARTIN         GAYLORD    1992-02-10  1992-04-05           15.00        361.80         61.75     60.30         1.45
6042325   CHAMBERLINE    FRANCES    1992-02-11  1992-04-06           34.24        118.62         20.25     19.77          .48
6049354   SWARTZ         GEORGE     1992-02-06  1992-05-01           21.25        403.60         68.89     67.27         1.62
6062946   BROCK          ETHEL      1992-02-17  1992-04-26           77.00         75.19           .00       .00          .00
6067956   PORTER         MAY        1992-02-22  ****-**-**           80.24      4,592.59        147.66    127.57        20.09
6080669   WILLIAMS       JAMES      1992-02-15  1992-05-10           24.24        435.56         74.34     72.59         1.75
6088678   TYLER          JAMES      1992-02-20  1992-05-15          305.64      1,426.33         65.49     59.43         6.06
6101291   RACH           MARY       1992-02-21  1992-05-30           35.00        172.68         29.47     28.78          .69
6103375   GARKOW         DOROTHY    1992-02-21  1992-05-30           25.00        153.52         26.20     25.59          .61
6116728   STEINER        ROBERT     1990-01-03  ****-**-**           15.00      1,730.11         79.44     72.09         7.35
6216129   MUNSON         CLARENCE   1992-02-11  1992-06-20           15.00      1,903.48         87.40     79.31         8.09
6218512   MARTIN         GAYLORD    1992-02-18  1992-06-13           55.25      1,415.41         64.99     58.98         6.01
7014996   OLVERA         CLEMENTINE 1991-10-20  1992-06-15          106.74        270.15         46.11     45.03         1.08
7029926   WHITE          HIAWATHA   1991-10-19  1992-06-28           20.44         84.66           .00       .00          .00
7072694   WARTON         LEE        1991-10-20  1992-06-15          224.00      1,268.90         58.26     52.87         5.39
7100035   BROWN          RICHARD    1991-04-22  ****-**-**           50.62        461.39         78.75     76.90         1.85
8033692   MARLETTE       YVETTE     1992-02-10  1992-05-19           14.95         90.75           .00       .00          .00
9005226   WILSON         W C        1991-10-22  1992-05-31           22.40        386.85         66.03     64.48         1.55
6114113   NERY           GENEROSO   1991-09-07  1992-01-02           45.24        298.66         50.98     49.78         1.20
1002775   OWENS          J          1991-10-22  1992-05-31            8.06        417.60         71.28     69.60         1.68
1006231   DE LAUNEY      RAYMOND    1991-10-06  1992-04-01          202.84         87.54           .00       .00          .00
1013637   STALLINS       LEO F      1991-10-16  1992-06-11          843.74        154.53         26.38     25.76          .62
1014617   MORRIS         CLARENCE E 1991-10-17  1992-06-26          855.22        190.82         32.57     31.80          .77
2006286   BALDWIN        HARRY      1991-06-09  1992-07-03        1,341.79      1,296.88         59.55     54.04         5.51
2007177   HART           GRADDIE    1992-02-18  1992-04-13          128.37        144.43         24.65     24.07          .58
2008831   ZERING         WILLIAM    1992-03-10  1992-05-05          124.95        322.87         55.11     53.81         1.30
2009072   JONES          RFUS       1992-03-20  1992-05-15          128.90        432.91         73.89     72.15         1.74
2011719   REED           LOWELL     1992-03-11  1992-06-06          127.22      2,033.34         93.36     84.72         8.64
2013568   CULLENDER      EVERETT    1989-12-26  ****-**-**        3,886.33        135.42         23.11     22.57          .54
```

05/14/02 DB2 PRINT A/R FILE EXAMPLE PAGE 2

ACCOUNT	CUSTOMER LAST,	FIRST NAME	TRAN-DATE	BILL-DATE	BALANCE	INSTL-BAL	INSTL-PAY	PAYMENT	INTEREST
6005489	HALL	JOHN	1991-04-12	1992-03-07	60.00	187.65	32.03	31.28	.75
6013546	KIMURA	MICHI	1991-04-20	1992-03-15	37.60	251.95	43.00	41.99	1.01
6016529	BARRERA	JESUS	1991-04-20	1992-03-15	18.60	59.84	.00	.00	.00
6031714	FENZEE	WILLIAM	1991-04-19	1992-03-28	20.00	428.21	73.09	71.37	1.72
6033873	NEBRENSKY	CARMEN	1991-04-20	1992-03-29	3.40	246.35	42.05	41.06	.99
6034241	MC COY	FRANCES	1991-04-20	1992-03-29	15.00	368.06	62.82	61.34	1.48
6044751	SOARES	LAWERENCE	1992-02-15	1992-04-10	8.15	1,223.96	56.20	51.00	5.20
6046819	YEAGER	CLIFFORD	1992-02-15	1992-04-10	15.04	122.14	20.85	20.36	.49
6049672	YOUNG	ADELE	1992-02-18	1992-04-13	15.00	2,356.24	108.19	98.18	10.01
6058523	CALLEROS	MARCELINO	1992-02-11	1992-04-20	5.05	282.02	48.14	47.00	1.14
6060749	TAYLOR	FLORENCE	1992-02-17	1992-04-26	20.00	359.35	61.33	59.89	1.44
6061931	DIAZ	JOSE	1992-02-16	1992-04-25	3.00	541.91	47.41	45.16	2.25
6063977	WALTON	MARVIN	1992-02-18	1992-04-27	3.00	117.36	20.03	19.56	.47
6067646	MCELENY	JAMES	1992-02-19	1992-04-28	15.00	2,889.87	92.91	80.27	12.64
6068316	SCHMITZ	JOHN	1992-02-19	1992-04-28	15.00	3,496.31	112.41	97.12	15.29
6069495	SEDGWICK	JOWARD	1992-02-07	1992-05-02	57.00	394.09	67.26	65.68	1.58
6081428	JOHNSON	KNUTE	1992-02-14	1992-05-09	4.25	3,854.12	123.91	107.06	16.85
6085547	GRAHAM	LAYRA	1992-02-16	1992-05-11	4.25	10.52	.00	.00	.00
6090753	BEAVER	OPAL	1992-02-07	1992-05-16	15.00	3,288.60	105.73	91.35	14.38
6094333	HOBDY	DOROTHY	1992-02-09	1992-05-18	13.26	488.63	83.40	81.44	1.96
6096093	YEAGER	CLIFFORD	1992-02-13	1992-05-22	3.00	227.16	38.77	37.86	.91
6096859	WILSON	ALEXANDER	1992-02-16	1992-05-25	4.25	229.23	39.12	38.21	.91
6098223	ANGEL	JESUS	1992-02-19	1992-05-28	3.00	159.04	27.14	26.51	.63
6099467	GOULDING	JAHN	1992-02-19	1992-05-28	3.00	2,694.34	86.63	74.84	11.79
6105629	MILLER	JOHN	1990-01-01	****-**-**	30.00	13.79	.00	.00	.00
6106501	ZARATE	JOHN	1992-02-10	1992-06-05	101.20	3,548.10	114.08	98.56	15.52
6108415	HEMPE	RALPH	1992-02-14	1992-05-23	4.25	1,101.76	50.59	45.91	4.68
6113583	BROWN	STUART	1991-09-06	1992-01-01	26.20	9.01	.00	.00	.00
6123031	FLYNN	JOSEPH	1990-01-05	****-**-**	.00	438.40	74.83	73.07	1.76
6201997	IRWIN	HAZEL	1992-02-11	1992-06-06	23.40	8.52	.00	.00	.00
6202616	CONNER	AUBRY	1992-02-16	1992-06-11	16.80	152.15	25.97	25.36	.61
6208924	WHITE	ELMER	1992-02-17	1992-06-12	15.00	286.33	48.87	47.72	1.15
6211151	COX	ELAINE	1992-02-10	1992-06-19	100.20	498.08	85.01	83.01	2.00
6215785	RODEN	HAROLD	1992-02-11	1992-06-20	170.91	446.37	76.19	74.40	1.79
6216412	ROUKE	CURTIS	1992-02-12	1992-06-21	48.00	71.96	.00	.00	.00
6219004	SOOLEY	WILLIAM	1992-02-18	1992-06-13	15.00	64.58	.00	.00	.00
7000227	HAKEEM	N	1991-10-22	1992-07-31	36.00	272.72	46.55	45.45	1.10
7004753	FORBES	JOHN	1991-10-16	1992-07-25	11.20	320.91	54.77	53.49	1.28
7010966	SMITH	JOHN	1991-10-22	1992-07-31	13.66	355.60	60.69	59.27	1.42
7011407	WHITE	ELMER	1991-10-16	1992-07-25	10.24	145.08	24.76	24.18	.58
7012799	RODRIGUEZ	EVARISTO	1991-10-16	1992-07-25	10.58	102.72	17.53	17.12	.41
7012837	WOODS	LOUISE	1991-10-19	1992-06-28	.00	63.87	.00	.00	.00
7023189	SIAS	EDWARDO	1991-10-19	1992-06-28	.00	4,322.92	138.99	120.08	18.91
7033133	DE VAULT	JERRY	1991-10-16	1992-07-25	11.20	93.55	.00	.00	.00
7039085	BLACK	LENORE	1991-10-21	1992-04-30	50.40	804.68	70.40	67.06	3.34
7099657	MALTSBERGER	JOHN A	1991-04-22	****-**-**	6.72	25.03	.00	.00	.00
7102194	TURNER	HAROLD	1991-04-22	****-**-**	5.12	259.44	44.28	43.24	1.04
7106246	SMITH	AUSTIN	1991-04-08	1992-02-17	5.12	142.29	24.29	23.72	.57
9000534	HOPKINS	BARRY P	1991-10-22	1992-07-31	14.00	314.79	53.73	52.47	1.26
9002626	WORRELL	TED	1990-10-22	1991-07-31	13.00	385.78	65.84	64.30	1.54

05/14/02 DB2 PRINT A/R FILE EXAMPLE PAGE 3

ACCOUNT	CUSTOMER LAST,	FIRST NAME	TRAN-DATE	BILL-DATE	BALANCE	INSTL-BAL	INSTL-PAY	PAYMENT	INTEREST
9017828	HILL	MYRTLE G	1991-04-22	****-**-**	11.20	46.67	.00	.00	.00
1013904	HINDERLICH	PAUL	1991-10-10	1992-08-18	1,382.19	136.52	23.30	22.75	.55
1014889	SULLIVAN	JOSEPH C	1991-10-14	1992-06-23	337.70	261.97	44.71	43.66	1.05
4003896	PETERSON	WALTER	1991-04-15	1992-02-10	196.00	470.12	80.24	78.35	1.89
6007678	FARREL	HAZEL	1991-04-13	1992-03-08	53.20	237.67	40.57	39.61	.96
7053657	BARNETT	LESTER	1991-08-21	1991-11-30	25.07	210.10	35.86	35.02	.84
7077289	HICKMAN	VERNON	1991-10-19	1992-02-28	36.95	2,200.98	101.06	91.71	9.35
7107137	VALDEZ	MARY	1991-04-08	1992-02-17	28.00	1,516.24	69.62	63.18	6.44
9014543	MARTINEZ	SIMON	1991-10-22	1992-01-31	17.44	4,538.90	145.93	126.08	19.85
2004739	STEIDLEY	GURTHA	1991-04-19	1992-03-14	136.23	73.98	.00	.00	.00
6008321	YEARGER	CLIFFORD	1991-04-14	1992-03-09	140.00	464.20	79.23	77.37	1.86
2005956	SOTO	RUFINO	1991-05-10	1992-04-05	125.29	178.96	30.54	29.83	.71
6063748	SEDANO	JOSE	1992-02-18	1992-04-27	3.00	59.74	.00	.00	.00
6091407	PINTO	GILBERT	1992-02-08	1992-05-17	5.00	4,475.33	143.89	124.31	19.58
8033684	WHITEHURST	CHARLES	1992-02-10	1992-05-19	25.00	231.77	39.56	38.63	.93
2002922	FLOWERS	ETHEL	1989-12-21	****-**-**	2,541.70	112.23	19.16	18.71	.45
6001327	CARLON	MARIANO	1991-05-06	1992-04-01	118.00	26.71	.00	.00	.00
6007724	HARRINGTON	RUTH	1991-04-13	1992-03-08	37.60	156.76	26.76	26.13	.63
6012132	ZENZOLA	MICHEAL	1991-04-18	1992-03-13	30.20	42.69	.00	.00	.00
6017479	BERSON	ANNA	1991-04-07	1992-03-16	3.00	4,340.28	139.55	120.56	18.99
6039804	MORENO	ELIZABETH	1992-02-10	1992-04-05	.00	325.29	55.52	54.22	1.30
6204384	FREDRICK	EMERY	1992-02-12	1992-06-07	15.00	621.08	54.33	51.76	2.57
8002053	JENNINGS	WILL	1991-04-17	1992-03-12	56.34	450.15	76.83	75.03	1.80
6106897	GASIOR	MARY	1992-02-10	1992-06-05	43.00	4,384.63	140.97	121.80	19.17
2014017	HERNANDEZ	CARLOS	1990-01-04	****-**-**	258.00	313.13	53.44	52.19	1.25
6061087	RAMOS	LORENZO	1992-02-17	1992-04-26	31.93	238.06	40.63	39.68	.95
6091156	SONDOVAL	SUSIE	1992-02-08	1992-05-17	15.00	394.23	67.29	65.71	1.58
6112498	BARRON	BONNIE	1992-02-18	1992-06-27	36.96	430.76	73.52	71.79	1.73
8009279	JARAMILLO	MADGALENA	1992-02-10	1992-05-05	17.24	204.07	34.83	34.01	.82
1014706	SHAW	MABEL	1991-10-16	1992-07-11	406.46	54.69	.00	.00	.00
6105904	CRESS	NELLIE	1991-09-06	1992-01-01	176.00	354.03	60.43	59.01	1.42
6209351	YOUNG	MARVIIE	1992-02-20	1992-06-15	46.59	310.25	52.95	51.71	1.24
0000434			****-**-**	****-**-**	11.20-	482.94	82.43	80.49	1.94
0002623			****-**-**	****-**-**	256.12-	4,413.40	141.90	122.59	19.31
1009681			****-**-**	****-**-**	10,085.55-	3,265.20	104.98	90.70	14.28
6027989			****-**-**	1990-09-21	310.75	285.54	48.74	47.59	1.15
6045154			****-**-**	1990-11-16	64.96	442.32	75.50	73.72	1.78
6090516			****-**-**	1990-12-14	19.54	311.75	53.21	51.96	1.25
7046936			****-**-**	****-**-**	2.46-	128.83	21.99	21.47	.52
7053941			****-**-**	****-**-**	33.60-	3,708.70	119.24	103.02	16.22
7066775			****-**-**	****-**-**	3.32-	2,403.85	110.37	100.16	10.21
7087896			****-**-**	****-**-**	1.32-	286.48	48.90	47.75	1.15
8003173			****-**-**	1990-10-28	35.00	185.46	31.65	30.91	.74
9010033			****-**-**	****-**-**	1.04-	133.44	22.78	22.24	.54
1001191	CARDOZA	MARY	1990-08-13	1990-10-22	177.52-	397.40	67.83	66.23	1.60
1006681	MYERS	EARL	1991-10-15	1992-03-10	123.24	240.10	40.98	40.02	.96
6023185	PORTWOOD	JOSEPHINE	1991-04-12	1992-03-21	15.00	442.25	75.48	73.71	1.77
6099963	ROMO	JESUS	1992-02-20	1992-05-29	3.12	1,328.00	60.97	55.33	5.64
6216846	TORRES	PAUBLINA	1992-02-11	1992-06-20	.00	117.75	20.10	19.63	.47
7030142	HARRIS	JEWEL	1991-10-19	1992-06-28	14.56	276.16	47.13	46.03	1.10

05/14/02 DB2 PRINT A/R FILE EXAMPLE PAGE 4

ACCOUNT	CUSTOMER LAST, FIRST NAME		TRAN-DATE	BILL-DATE	BALANCE	INSTL-BAL	INSTL-PAY	PAYMENT	INTEREST
7053185	SMITH	MARY	1991-10-22	1992-05-31	.00	260.02	44.38	43.34	1.04
7064535	PENNEYWELL	JERMEL	1991-10-22	1992-05-31	40.40	145.39	24.82	24.23	.59
7068794	BACTAD	MERIAN	1991-10-19	1992-06-28	11.20	1.68	.00	.00	.00
7070721	VERA	AGUSTIN C	1991-08-22	1991-10-31	27.45	49.68	.00	.00	.00
7071507	ALLEN	CARRIE A	1990-10-21	1991-09-30	38.08	409.48	69.89	68.25	1.64
7083696	ORACION	ARSENIA G	1991-10-22	1992-01-31	67.15	362.59	61.89	60.43	1.46
7103387	PEREZ	EVELYN	1991-04-22	****-**-**	53.76	389.72	66.52	64.95	1.57
8009155	FLORES	JOSE	1992-02-11	1992-05-06	40.42	379.93	64.85	63.32	1.53
8032084	ADAMS	CLAUDE	1992-02-08	1992-05-17	50.00	29.68	.00	.00	.00
9015884	RIZARDO	HIPOLITO J	1991-10-20	1992-06-15	10.04	492.76	84.10	82.13	1.97
9016643	DE MARCO	VINCENT D	1991-10-19	1992-06-28	33.60	178.67	30.50	29.78	.72
9018239	HUAYLLARA	BLANCA	1991-04-22	****-**-**	89.60	95.35	.00	.00	.00
8031126	RULEY	ANNA	1992-02-11	1992-06-20	65.00	203.55	34.74	33.93	.81
2000504	WILSON	HERMAN	1989-08-16	****-**-**	5,292.24	301.17	51.40	50.20	1.20
2010933	QUINTANAR	ANTONIO	1992-02-09	1992-05-18	1,460.61	372.04	63.50	62.01	1.49
6067557	BROWNING	EVA	1992-02-19	1992-04-28	3.45	286.21	48.85	47.70	1.15
6103723	KITAOKA	GEORGE	1992-02-21	1992-05-30	20.00	3,421.71	110.01	95.05	14.96
6114253	MARTINEZ	RACHEL	1991-09-06	1992-01-01	51.60	171.55	29.28	28.59	.69
7085494	JARAMILLO	BOB	1991-10-19	1992-02-28	20.00	438.88	74.91	73.15	1.76
8031401	SPECTOR	MICHEAL	1991-10-07	1992-01-02	28.00	2,567.75	82.56	71.33	11.23
6089771	HERNSTORM	VICTOR	1992-02-07	1992-05-16	41.91	100.98	17.24	16.83	.41
7109857	CRITTENDEN	CHARLE	1991-04-22	****-**-**	11.20	268.64	45.85	44.77	1.08
6019587	BENAVIDEZ	CESARIO	1991-04-13	1992-03-22	15.00	296.01	50.52	49.34	1.18
6032214	GONZALEZ	RAUL	1991-04-19	1992-03-28	30.40	160.50	27.39	26.75	.64
6044832	ROBBINS	WILLIAM	1992-02-15	1992-04-10	.00	470.29	80.27	78.38	1.89
6066402	ALCARAZ	ANDRES	1992-02-22	****-**-**	32.80	479.07	81.77	79.85	1.92
6074359	ASHLEY	JOHN	1992-02-08	1992-05-03	15.00	254.97	43.52	42.50	1.02
6118577	NAULLS	MARK	1990-01-03	****-**-**	15.00	425.26	72.58	70.88	1.70
6206328	VILLASENOR	TERESA	1992-02-14	1992-06-09	15.00	3,279.08	105.43	91.09	14.34
6208347	HAMELTON	MERDEDES	1992-02-19	1992-06-14	15.00	52.68	.00	.00	.00
6216617	TAUTRIM	RICHARD	1991-09-06	1992-01-01	140.00	156.25	26.67	26.04	.63
7043732	MARTINEZ	AMILCAR	1988-10-21	1989-04-30	108.25	137.16	23.41	22.86	.55
7054645	COWAN	EDWARD	1991-10-22	1992-07-31	22.40	64.37	.00	.00	.00
7059582	HARRIS	ZELMAL	1989-08-22	1989-10-31	44.75	444.71	75.90	74.12	1.78
7071361	BAEZA	REBECCA	1990-10-22	1991-01-31	66.00	170.52	29.10	28.42	.68
7084447	JACKSON	GARLAND	1991-10-22	1992-01-31	121.55	359.98	61.44	60.00	1.44
7090943	GILL	BILLY L	1991-10-18	1992-06-13	11.20	49.28	.00	.00	.00
7093853	CHAVEZ	ROBERT L	1991-10-19	1992-06-28	265.74	396.27	67.64	66.05	1.59
7097166	MORGAN	ROGER H	1991-10-22	1992-07-31	44.45	751.03	65.70	62.59	3.11
7098375	VALDEZ	JACK	1991-10-16	1992-07-25	22.40	45.96	.00	.00	.00
7112947	MARTINEZ	DOMINGO	1991-04-08	1992-02-17	.00	438.52	74.85	73.09	1.76
8001472	NARANJO	DELIA	1991-04-15	1992-03-10	115.60	492.96	84.14	82.16	1.98
8003246	BADILY	NEIL	1991-04-10	1992-03-19	114.23	201.46	34.39	33.58	.81
8010641	MURPHY	MATTHEWS	1992-02-09	1992-06-04	82.56	2,256.23	103.59	94.01	9.58
8011036	ANDERWS	CHARLES	1992-02-07	1992-06-02	114.52	196.87	33.60	32.81	.79
8011699	OBRESON	FRANK	1992-02-10	1992-06-05	57.50	292.24	49.88	48.71	1.17
8031096	ROMERP	ARMENDO	1992-02-08	1992-06-17	81.26	115.05	19.64	19.18	.46
9007156	SANCHO	CARLOS	1991-10-19	1992-06-28	352.80	295.42	50.42	49.24	1.18
9009469	CARMICHAEL	GEORGE	1988-10-22	1989-05-31	72.50	498.63	85.11	83.11	2.00
9020039	LOPEZ	SALVADOR	1991-04-08	1992-02-17	58.24	4,745.73	152.58	131.83	20.75

RANDOM1 Program

```
*****************************************************************
*                                                               *
* RANDOM1: ACCESS DB2 RANDOMLY, USING THE MEMBER "RANDOMI1"     *
*          TO ACCESS RECORDS DESIRED.                           *
*                                                               *
*    ===>  USES EXTENDED HOST VARIABLE AS INPUT TO              *
*          SQL DECLARE STATEMENT.                               *
*                                                               *
*          PRINT THE INPUT RECORDS, AS WELL AS THE              *
*          RECORD OBTAINED.                                     *
*                                                               *
*****************************************************************
*INFCARD                          /* IF VSE, REMOVE * IN POS. 1

EQU  INF-LAST-NAME      INF1-15

* SQL HOST VARIABLES
EQU  AR-ENT-REC         WST000-000
++INCLUDE ARDEFINE

EQU  NAME-COUNT         WST411-412-B   LOVALUE

* SQL DATE HOST VARIABLES -- YYYY-DD-MM DATE FORMAT
EQU  WS-TRAN-DATE       WST451-460
EQU  WS-TRAN-CENTUR     WST451-452     C'19'
EQU  WS-TRAN-YEAR       WST453-454
EQU  WS-TRAN-DASH1      WST455         C'-'
EQU  WS-TRAN-MONTH      WST456-457
EQU  WS-TRAN-DASH2      WST458         C'-'
EQU  WS-TRAN-DAY        WST459-460

EQU  WS-BILL-DATE       WST461-470
EQU  WS-BILL-CENTUR     WST461-462     C'19'
EQU  WS-BILL-YEAR       WST463-464
EQU  WS-BILL-SLASH1     WST465         C'-'
EQU  WS-BILL-MONTH      WST466-467
EQU  WS-BILL-SLASH2     WST468         C'-'
EQU  WS-BILL-DAY        WST469-470

EQU  WS-TRAN-NULL       WST471-472-B
EQU  WS-BILL-NULL       WST473-474-B

* AR FILE PRINT LINE FORMAT
* LINE 1
EQU  PR-ACCOUNT         PRT1             /* ACCOUNT NUMBER
EQU  PR-LAST-NAME       PRT11            /* LAST NAME
EQU  PR-FIRST-NAME      PRT26            /* FIRST NAME
EQU  PR-TRAN-DATE       PRT37            /* TRANSACTION DATE
EQU  PR-BILL-DATE       PRT49            /* BILLING DATE
EQU  PR-BALANCE         PRT63            /*
EQU  PR-INSTL-BAL       PRT75            /* AMT LEFT ON INSTALLMENT
EQU  PR-INSTL-PAY       PRT91            /* PLANNED INSTALLMENT PAYMT
EQU  PR-BAL-PARTPAY     PRT104           /* ACTUAL PAYMENT
EQU  PR-INT-PARTPAY     PRT120           /* INTEREST, PART PAYMENT

    HDR 1A 1 $IPLDAT$                                          DB2
    HDR 1B   RANDOM1 (USES HOST VARIABLE AS INPUT TO SQL)      PAGE $PG$
    HDR 2A 0ACCOUNT   CUSTOMER LAST, FIRST NAME  TRAN-DATE   BILL-DATE
    HDR 2B   BALANCE     INSTL-BAL     INSTL-PAY   PAYMENT     INTEREST


* SPECIFY VISION:REPORT STATEMENT TO RECEIVE CONTROL
```

```
* WHEN ALL ROWS HAVE BEEN RETRIEVED
    EXEC SQL WHENEVER
       NOT FOUND GOTO 500
    END-EXEC

300
    GET INF ATEND 700

* SPECIFY SQL COLUMNS TO BE RETRIEVED
* AND THE ORDER IN WHICH THE ROWS ARE TO APPEAR
    EXEC SQL DECLARE BLUEJAY CURSOR
       FOR SELECT
      AR_ACCOUNT,
      AR_TRAN_DATE ,
      AR_BILL_DATE  ,
      AR_LAST_NAME   ,
      AR_FIRST_NAME   ,
      AR_MIDDLE_INIT  ,
      AR_STREET      ,
      AR_CITY_ST_ZIP,
      AR_BALANCE   ,
      AR_ACCT_CODE,
      AR_INST_BALANCE ,
      AR_INST_PAY    ,
      AR_BAL_PARTPAY,
      AR_INT_PARTPAY ,
      AR_NR_PAY
       FROM ACCOUNTSRECEIVABLE
       WHERE AR_LAST_NAME LIKE :=INF-LAST-NAME
       ORDER BY AR_LAST_NAME, AR_FIRST_NAME
    END-EXEC

    EXEC SQL OPEN
       BLUEJAY
    END-EXEC

400
* SPECIFY THE VISION:REPORT AREAS INTO WHICH THE SQL
* COLUMNS ARE TO BE PLACED.
* NOTE THAT THE COMMAS DO NOT HAVE TO BE IMMEDIATELY
* FOLLOWING THE DATANAME.
   EXEC SQL FETCH
     BLUEJAY INTO
      :AR-ACCOUNT ,
      :WS-TRAN-DATE    :WS-TRAN-NULL  ,
      :WS-BILL-DATE    :WS-BILL-NULL  ,
      :AR-LAST-NAME  ,
      :AR-FIRST-NAME  ,
      :AR-MIDDLE-INIT  ,
      :AR-STREET        ,
      :AR-CITY-ST-ZIP    ,
      :AR-BALANCE         ,
      :AR-ACCT-CODE        ,
      :AR-INSTL-BAL         ,
      :AR-INSTL-PAY          ,
      :AR-BAL-PARTPAY         ,
      :AR-INT-PARTPAY        ,
      :AR-NR-PAY
   END-EXEC

   IF  NAME-COUNT EQ LOVALUE
       MOVE C'ROWS LIKE NAME:' TO PRT1
       MOVE INF-LAST-NAME TO PRT16
       PRINT DOUBLE.

   MOVE AR-ACCOUNT       TO PR-ACCOUNT
```

```
   MOVE AR-LAST-NAME     TO PR-LAST-NAME
   MOVE AR-FIRST-NAME    TO PR-FIRST-NAME

   IF WS-TRAN-NULL EQ X'FFFF'
      MOVE C'****-**-**' TO PR-TRAN-DATE
      GOTO 420.
   MOVE WS-TRAN-DATE     TO PR-TRAN-DATE

420
   IF WS-BILL-NULL EQ X'FFFF'
      MOVE C'****-**-**' TO PR-BILL-DATE
      GOTO 440.
   MOVE WS-BILL-DATE     TO PR-BILL-DATE

440
   MOVE AR-INT-PARTPAY   TO PR-INT-PARTPAY
   MOVE AR-BAL-PARTPAY   TO PR-BAL-PARTPAY
   MOVE AR-INSTL-PAY     TO PR-INSTL-PAY
   MOVE AR-INSTL-BAL     TO PR-INSTL-BAL
   MOVE AR-BALANCE       TO PR-BALANCE
   PRINT

   ADD X'0001' TO NAME-COUNT

   GOTO 400

500
    EXEC SQL CLOSE
       BLUEJAY
    END-EXEC

   IF  NAME-COUNT EQ LOVALUE
       MOVE C'NO ROWS LIKE NAME:' TO PRT1
       MOVE INF-LAST-NAME TO PRT19
       PRINT DOUBLE.
   MOVE LOVALUE TO NAME-COUNT
   GOTO 300

700
   GOTO EOJ

9999END               /* IF VSE, PLACE "INF" CARDS NEXT
```

RANDOM1 and RANDOM2 Output

```
05/14/02                                    DB2              RANDOM1 (USES HOST VARIABLE AS INPUT TO SQL)     PAGE    1

ACCOUNT   CUSTOMER LAST, FIRST NAME  TRAN-DATE   BILL-DATE          BALANCE     INSTL-BAL     INSTL-PAY   PAYMENT     INTEREST

NO ROWS LIKE NAME:'E%'

ROWS LIKE NAME:'K%'
6013546   KIMURA         MICHI      1991-04-20  1992-03-15           37.60        251.95         43.00     41.99         1.01
6103723   KITAOKA        GEORGE     1992-02-21  1992-05-30           20.00      3,421.71        110.01     95.05        14.96

ROWS LIKE NAME:'MA%'
7099657   MALTSBERGER    JOHN A     1991-04-22  ****-**-**            6.72         25.03          .00       .00          .00
8033692   MARLETTE       YVETTE     1992-02-10  1992-05-19           14.95         90.75          .00       .00          .00
6041272   MARTIN         GAYLORD    1992-02-10  1992-04-05           15.00        361.80        61.75     60.30         1.45
6218512   MARTIN         GAYLORD    1992-02-18  1992-06-13           55.25      1,415.41        64.99     58.98         6.01
7043732   MARTINEZ       AMILCAR    1988-10-21  1989-04-30          108.25        137.16        23.41     22.86          .55
7112947   MARTINEZ       DOMINGO    1991-04-08  1992-02-17             .00        438.52        74.85     73.09         1.76
6114253   MARTINEZ       RACHEL     1991-09-06  1992-01-01           51.60        171.55        29.28     28.59          .69
9014543   MARTINEZ       SIMON      1991-10-22  1992-01-31           17.44      4,538.90       145.93    126.08        19.85

ROWS LIKE NAME:'NA%'
8001472   NARANJO        DELIA      1991-04-15  1992-03-10          115.60        492.96        84.14     82.16         1.98
6118577   NAULLS         MARK       1990-01-03  ****-**-**           15.00        425.26        72.58     70.88         1.70

ROWS LIKE NAME:'PO%'
6067956   PORTER         MAY        1992-02-22  ****-**-**           80.24      4,592.59       147.66    127.57        20.09
6023185   PORTWOOD       JOSEPHINE  1991-04-12  1992-03-21           15.00        442.25        75.48     73.71         1.77

ROWS LIKE NAME:'SM%'
7106246   SMITH          AUSTIN     1991-04-08  1992-02-17            5.12        142.29        24.29     23.72          .57
7010966   SMITH          JOHN       1991-10-22  1992-07-31           13.66        355.60        60.69     59.27         1.42
7053185   SMITH          MARY       1991-10-22  1992-05-31             .00        260.02        44.38     43.34         1.04

ROWS LIKE NAME:'V%'
7098375   VALDEZ         JACK       1991-10-16  1992-07-25           22.40         45.96          .00       .00          .00
7107137   VALDEZ         MARY       1991-04-08  1992-02-17           28.00      1,516.24        69.62     63.18         6.44
6016316   VANCE          VERNON     1991-04-20  1992-03-15           40.76      3,243.83       104.29     90.11        14.18
6123317   VASGUEZ        IRENE      1990-01-05  ****-**-**             .00        496.40        84.73     82.73         2.00
7070721   VERA           AGUSTIN C  1991-08-22  1991-10-31           27.45         49.68          .00       .00          .00
6206328   VILLASENOR     TERESA     1992-02-14  1992-06-09           15.00      3,279.08       105.43     91.09        14.34

ROWS LIKE NAME:'WA%'
6063977   WALTON         MARVIN     1992-02-18  1992-04-27            3.00        117.36        20.03     19.56          .47
7072694   WARTON         LEE        1991-10-20  1992-06-15          224.00      1,268.90        58.26     52.87         5.39

NO ROWS LIKE NAME:'WE%'
```

RANDOM2 Program

```
*****************************************************************
*                                                               *
* RANDOM2: ACCESS DB2 RANDOMLY, USING THE MEMBER "RANDOMI2"     *
*          TO ACCESS RECORDS DESIRED.                           *
*                                                               *
*   ===>   USES DESCRIPTOR AS INPUT TO SQL OPEN STATEMENT.      *
*                                                               *
*          PRINT THE INPUT RECORDS, AS WELL AS THE              *
*          RECORD OBTAINED.                                     *
*                                                               *
*****************************************************************
*INFCARD                          /* IF VSE, REMOVE * IN POS. 1
EQU  INF-LAST-NAME      INF1-15

* SQL HOST VARIABLES
EQU  AR-ENT-REC         WST000-000
++INCLUDE ARDEFINE

* SQL DATE HOST VARIABLES -- YYYY-DD-MM DATE FORMAT
EQU  WS-TRAN-DATE       WST451-460
EQU  WS-TRAN-CENTUR     WST451-452     C'19'
EQU  WS-TRAN-YEAR       WST453-454
EQU  WS-TRAN-DASH1      WST455         C'-'
EQU  WS-TRAN-MONTH      WST456-457
EQU  WS-TRAN-DASH2      WST458         C'-'
EQU  WS-TRAN-DAY        WST459-460

EQU  WS-BILL-DATE       WST461-470
EQU  WS-BILL-CENTUR     WST461-462     C'19'
EQU  WS-BILL-YEAR       WST463-464
EQU  WS-BILL-DASH1      WST465         C'-'
EQU  WS-BILL-MONTH      WST466-467
EQU  WS-BILL-DASH2      WST468         C'-'
EQU  WS-BILL-DAY        WST469-470

EQU  WS-TRAN-NULL       WST471-472-B
EQU  WS-BILL-NULL       WST473-474-B

* SQL DESCRIPTOR AREA
EQU  OPEN-SQLDA         WST501-561

EQU  OPEN-SQLDA /* EXAMPLE OF COBOL SYSTEM REDEFINE FOR WST301-361

EQU  SQLDAID            (8)          C'SQLDA   '  /* WST301-308
EQU  SQLDABC            (4)-B        X'0000003C'
EQU  SQLN               (2)-B        X'0001'
EQU  SQLD               (2)-B        X'0001'

* SQLVAR-1
EQU  SV1-SQLTYPE        (2)-B        X'01C4'
EQU  SV1-SQLLEN         (2)-B
EQU  SV1-SQLDATA        (4)-B
EQU  SV1-SQLIND         (4)-B
EQU  SV1-SQLNAME-LN     (2)-B
EQU  SV1-SQLNAME        (30)

EQU  NAME-COUNT         WST611-612-B   LOVALUE

EQU  LIKE-STRING        WST613-628

* AR FILE PRINT LINE FORMAT
* LINE 1
EQU  PR-ACCOUNT         PRT1              /* ACCOUNT NUMBER
```

```
EQU  PR-LAST-NAME       PRT11             /* LAST NAME
EQU  PR-FIRST-NAME      PRT26             /* FIRST NAME
EQU  PR-TRAN-DATE       PRT37             /* TRANSACTION DATE
EQU  PR-BILL-DATE       PRT49             /* BILLING DATE
EQU  PR-BALANCE         PRT63             /*
EQU  PR-INSTL-BAL       PRT75             /* AMT LEFT ON INSTALLMENT
EQU  PR-INSTL-PAY       PRT91             /* PLANNED INSTALLMENT PAYMT
EQU  PR-BAL-PARTPAY     PRT104            /* ACTUAL PAYMENT
EQU  PR-INT-PARTPAY     PRT120            /* INTEREST, PART PAYMENT

    HDR 1A 1 $IPLDAT$                                          DB2
    HDR 1B   RANDOM2 (USES DESCRIPTOR AS INPUT TO SQL OPEN)   PAGE $PG$
    HDR 2A 0ACCOUNT   CUSTOMER LAST, FIRST NAME  TRAN-DATE   BILL-DATE
    HDR 2B   BALANCE     INSTL-BAL     INSTL-PAY   PAYMENT     INTEREST


* INITIALIZE OPEN SQLDA
    CALL QJADDR LIKE-STRING SV1-SQLDATA

* SPECIFY VISION:REPORT STATEMENT TO RECEIVE CONTROL
* WHEN ALL ROWS HAVE BEEN RETRIEVED
    EXEC SQL WHENEVER
       NOT FOUND GOTO 500
    END-EXEC

300
    GET INF ATEND 700

* DETERMINE LENGTH OF INPUT & INSERT VALUE INTO SQLDA
    WHEN INF-LAST-NAME INCLUDES NONBLANKS REVERSE
      MOVE C'15' TO SV1-SQLLEN
      SUB  VAL225-228-B FR SV1-SQLLEN
      MOVE INF-LAST-NAME TO LIKE-STRING SV1-SQLLEN.

* SPECIFY SQL COLUMNS TO BE RETRIEVED
* AND THE ORDER IN WHICH THE ROWS ARE TO APPEAR
    EXEC SQL DECLARE BLUEJAY CURSOR
       FOR SELECT
      AR_ACCOUNT       ,
      AR_TRAN_DATE     ,
      AR_BILL_DATE     ,
      AR_LAST_NAME     ,
      AR_FIRST_NAME    ,
      AR_MIDDLE_INIT   ,
      AR_STREET        ,
      AR_CITY_ST_ZIP   ,
      AR_BALANCE       ,
      AR_ACCT_CODE     ,
      AR_INST_BALANCE  ,
      AR_INST_PAY      ,
      AR_BAL_PARTPAY   ,
      AR_INT_PARTPAY   ,
      AR_NR_PAY
       FROM ACCOUNTSRECEIVABLE
       WHERE AR_LAST_NAME LIKE ?
       ORDER BY AR_LAST_NAME, AR_FIRST_NAME
    END-EXEC

    EXEC SQL OPEN
       BLUEJAY
       USING DESCRIPTOR :OPEN-SQLDA
    END-EXEC
400
* SPECIFY THE VISION:REPORT AREAS INTO WHICH THE SQL
* COLUMNS ARE TO BE PLACED
   EXEC SQL FETCH
```

```
     BLUEJAY INTO
      :AR-ACCOUNT                    ,
      :WS-TRAN-DATE    :WS-TRAN-NULL  ,
      :WS-BILL-DATE    :WS-BILL-NULL  ,
      :AR-LAST-NAME                  ,
      :AR-FIRST-NAME                 ,
      :AR-MIDDLE-INIT                ,
      :AR-STREET                     ,
      :AR-CITY-ST-ZIP                ,
      :AR-BALANCE                    ,
      :AR-ACCT-CODE                  ,
      :AR-INSTL-BAL                  ,
      :AR-INSTL-PAY                  ,
      :AR-BAL-PARTPAY                ,
      :AR-INT-PARTPAY                ,
      :AR-NR-PAY
   END-EXEC

   IF  NAME-COUNT EQ LOVALUE
       MOVE C'ROWS LIKE NAME:' TO PRT1
       MOVE LIKE-STRING TO PRT16 SV1-SQLLEN
       PRINT DOUBLE.

   MOVE AR-ACCOUNT       TO PR-ACCOUNT
   MOVE AR-LAST-NAME     TO PR-LAST-NAME
   MOVE AR-FIRST-NAME    TO PR-FIRST-NAME

   IF WS-TRAN-NULL EQ X'FFFF'
      MOVE C'****-**-**' TO PR-TRAN-DATE
      GOTO 420.
   MOVE WS-TRAN-DATE     TO PR-TRAN-DATE

420
   IF WS-BILL-NULL EQ X'FFFF'
      MOVE C'****-**-**' TO PR-BILL-DATE
      GOTO 440.
   MOVE WS-BILL-DATE     TO PR-BILL-DATE

440
   MOVE AR-INT-PARTPAY   TO PR-INT-PARTPAY
   MOVE AR-BAL-PARTPAY   TO PR-BAL-PARTPAY
   MOVE AR-INSTL-PAY     TO PR-INSTL-PAY
   MOVE AR-INSTL-BAL     TO PR-INSTL-BAL
   MOVE AR-BALANCE       TO PR-BALANCE
   PRINT
   ADD X'0001' TO NAME-COUNT

   GOTO 400

500
    EXEC SQL CLOSE
       BLUEJAY
    END-EXEC

   IF  NAME-COUNT EQ LOVALUE
       MOVE C'NO ROWS LIKE NAME:' TO PRT1
       MOVE LIKE-STRING TO PRT19 SV1-SQLLEN
       PRINT DOUBLE.
   MOVE LOVALUE TO NAME-COUNT
   GOTO 300
700
   GOTO EOJ
9999END               /* IF VSE, PLACE "INF" CARDS NEXT
```

UPDATE Program

```
*****************************************************************
*                                                               *
* UPDATE:  UPDATE DB2 RANDOMLY, USING THE MEMBER "UPDINPUT"     *
*          TO ACCESS RECORDS DESIRED.                           *
*                                                               *
*          LOGIC:                                               *
*                                                               *
*            IF AR-BALANCE   IS OVER $1,000.00                  *
*               AND AR-TRAN-DATE IS GREATER THAN 180 DAYS       *
*               FROM TODAY'S DATE USING QUIKDATE,               *
*                  ADD 10% TO AR-BALANCE,                       *
*                  ADD 30 DAYS TO AR-BILL-DATE.                 *
*            SHOW AR-BALANCE   AND AR-BILL-DATE,                *
*            BEFORE AND AFTER UPDATE                            *
*                                                               *
*          PRINT THE INPUT RECORDS, AS WELL AS THE              *
*          RECORD OBTAINED.                                     *
*                                                               *
*****************************************************************
*INFCARD                          /* IF VSE, REMOVE * IN POS. 1
* SQL HOST VARIABLES
EQU  AR-ENT-REC         WST000-000
++INCLUDE ARDEFINE

EQU  DIFF               WST400-402-P

* SQL DATE HOST VARIABLES -- YYYY-DD-MM DATE FORMAT
EQU  WS-TRAN-DATE       WST451-460
EQU  WS-TRAN-CENTUR     WST451-452     C'19'
EQU  WS-TRAN-YEAR       WST453-454
EQU  WS-TRAN-DASH1      WST455         C'-'
EQU  WS-TRAN-MONTH      WST456-457
EQU  WS-TRAN-DASH2      WST458         C'-'
EQU  WS-TRAN-DAY        WST459-460

EQU  WS-BILL-DATE       WST461-470
EQU  WS-BILL-CENTUR     WST461-462     C'19'
EQU  WS-BILL-YEAR       WST463-464
EQU  WS-BILL-SLASH1     WST465         C'-'
EQU  WS-BILL-MONTH      WST466-467
EQU  WS-BILL-SLASH2     WST468         C'-'
EQU  WS-BILL-DAY        WST469-470

EQU  WS-TRAN-NULL       WST471-472-B
EQU  WS-BILL-NULL       WST473-474-B

* QUIKDATE WORKAREA -- MM/DD/YY DATE FORMAT

EQU  QD-TRAN-DATE       WST501-506
EQU  QD-TRAN-MONTH      WST501-502
EQU  QD-TRAN-DAY        WST503-504
EQU  QD-TRAN-YEAR       WST505-506

EQU  QD-BILL-DATE       WST511-516
EQU  QD-BILL-MONTH      WST511-512
EQU  QD-BILL-DAY        WST513-514
EQU  QD-BILL-YEAR       WST515-516

EQU  QUIKDATE-RC        VAL46-49

* INPUT DESCRIPTION
EQU  TR-ACCOUNT         INF1-9      /* INCLUDES SINGLE QUOTES
```

```
* LINE 1
EQU  PR-ACCOUNT         PRT1              /* ACCOUNT NUMBER
EQU  PR-LAST-NAME       PRT11             /* LAST NAME
EQU  PR-FIRST-NAME      PRT26             /* FIRST NAME
EQU  PR-TRAN-DATE       PRT37             /* TRANSACTION DATE
EQU  PR-BILL-DATE       PRT49             /* BILLING DATE
EQU  PR-BALANCE         PRT63             /*
EQU  PR-INSTL-BAL       PRT75             /* AMT LEFT ON INSTALLMENT
EQU  PR-INSTL-PAY       PRT91             /* PLANNED INSTALLMENT PAYMT
EQU  PR-BAL-PARTPAY     PRT104            /* ACTUAL PAYMENT
EQU  PR-INT-PARTPAY     PRT120            /* INTEREST, PART PAYMENT

    HDR 1A 1 $IPLDAT$                                          DB2
    HDR 1B   UPDATE A/R FILE EXAMPLE                           PAGE $PG$
    HDR 2A 0ACCOUNT   CUSTOMER LAST, FIRST NAME  TRAN-DATE   BILL-DATE
    HDR 2B   BALANCE     INSTL-BAL     INSTL-PAY   PAYMENT     INTEREST

010  GET INF ATEND 900               /* GET INPUT RECORD
     MOVE  INF1-30             TO   PRT1
     MOVE  C' **INPUT RECORD'  TO   PRT31
     PRINT                          /* PRINT INPUT RECORD

* GET SQL RECORD RANDOMLY
* SPECIFY SQL COLUMNS TO BE RETRIEVED.
* NOTE THAT THE COMMAS DO NOT HAVE TO BE IMMEDIATELY
* FOLLOWING THE DATANAME.
    EXEC SQL SELECT
      AR_ACCOUNT ,
      AR_TRAN_DATE,
      AR_BILL_DATE ,
      AR_LAST_NAME  ,
      AR_FIRST_NAME  ,
      AR_MIDDLE_INIT  ,
      AR_STREET        ,
      AR_CITY_ST_ZIP  ,
      AR_BALANCE      ,
      AR_ACCT_CODE    ,
      AR_INST_BALANCE,
      AR_INST_PAY   ,
      AR_BAL_PARTPAY,
      AR_INT_PARTPAY,
      AR_NR_PAY     ,
      AR_KEY
     INTO
      :AR-ACCOUNT,
      :WS-TRAN-DATE    :WS-TRAN-NULL,
      :WS-BILL-DATE    :WS-BILL-NULL,
      :AR-LAST-NAME                 ,
      :AR-FIRST-NAME ,
      :AR-MIDDLE-INIT ,
      :AR-STREET       ,
      :AR-CITY-ST-ZIP   ,
      :AR-BALANCE        ,
      :AR-ACCT-CODE       ,
      :AR-INSTL-BAL        ,
      :AR-INSTL-PAY         ,
      :AR-BAL-PARTPAY        ,
      :AR-INT-PARTPAY         ,
      :AR-NR-PAY            ,
      :AR-KEY
       FROM ACCOUNTSRECEIVABLE
       WHERE AR_ACCOUNT = :=TR-ACCOUNT
    END-EXEC  NOTFOUND 600

* CHECK MINIMUM BALANCE
     IF AR-BALANCE IS LE P'100000'
```

```
         MOVE AR-BALANCE TO PRT1
         MOVE C'BALANCE' TO PRT20
         PRINT DOUBLE SPACED
         GOTO 10.

      IF WS-TRAN-NULL EQ X'FFFF'
         MOVE C'RECORD CANNOT BE UPDATED' TO PRT1
         MOVE C'-- NO TRANSACTION DATE.' TO PRT40
         PRINT DOUBLE SPACED
         GOTO 10.

      MOVE WS-TRAN-MONTH TO QD-TRAN-MONTH
      MOVE WS-TRAN-DAY   TO QD-TRAN-DAY
      MOVE WS-TRAN-YEAR  TO QD-TRAN-YEAR

* COMPUTE NUMBER OF DAYS DIFFERENCE
      CALL  QUIKDATE C'03'  QD-TRAN-DATE  C'MMDDYY '  C'070492'
            C'MMDDYY '   DIFF

      IF QUIKDATE-RC NOT EQ C'0000'
         MOVE C'BAD DATE' TO PRT1
         PRINT DOUBLE SPACED
         GOTO 10.

* CHECK MINIMUM DIFFERENCE
      IF DIFF IS LT P'180'
         MOVE DIFF TO PRT1
         MOVE C'DIFF' TO PRT20
         PRINT DOUBLE SPACED
         GOTO 10.

      IF WS-BILL-NULL EQ X'FFFF'
         MOVE C'RECORD CANNOT BE UPDATED' TO PRT1
         MOVE C'-- NO BILLING DATE.' TO PRT40
         PRINT DOUBLE SPACED
         GOTO 10.

      MOVE WS-BILL-MONTH TO QD-BILL-MONTH
      MOVE WS-BILL-DAY   TO QD-BILL-DAY
      MOVE WS-BILL-YEAR  TO QD-BILL-YEAR

* ADD 30 DAYS
      CALL  QUIKDATE C'08'  QD-BILL-DATE   C'MMDDYY '  C'00030'
            QD-BILL-DATE   C'MMDDYY '

      MOVE QD-BILL-MONTH TO WS-BILL-MONTH
      MOVE QD-BILL-DAY   TO WS-BILL-DAY
      MOVE QD-BILL-YEAR  TO WS-BILL-YEAR

* ADD 10% TO BALANCE
      MULT AR-BALANCE 2D  BY  C'11'  1D  GIVING  AR-BALANCE  2D

   MOVE AR-ACCOUNT       TO PR-ACCOUNT
   MOVE AR-LAST-NAME     TO PR-LAST-NAME
   MOVE AR-FIRST-NAME    TO PR-FIRST-NAME

   IF WS-TRAN-NULL EQ X'FFFF'
      MOVE C'****-**-**' TO PR-TRAN-DATE
      GOTO 420.
   MOVE WS-TRAN-DATE     TO PR-TRAN-DATE

420
   IF WS-BILL-NULL EQ X'FFFF'
      MOVE C'****-**-**' TO PR-BILL-DATE
      GOTO 440.
   MOVE WS-BILL-DATE     TO PR-BILL-DATE
```

```
440
   MOVE AR-INT-PARTPAY   TO PR-INT-PARTPAY
   MOVE AR-BAL-PARTPAY   TO PR-BAL-PARTPAY
   MOVE AR-INSTL-PAY     TO PR-INSTL-PAY
   MOVE AR-INSTL-BAL     TO PR-INSTL-BAL
   MOVE AR-BALANCE       TO PR-BALANCE
   PRINT

* UPDATE SQL ROW
    EXEC SQL
       UPDATE ACCOUNTSRECEIVABLE
       SET AR_BALANCE   = :AR-BALANCE,
           AR_BILL_DATE = :WS-BILL-DATE
       WHERE AR_ACCOUNT = :=TR-ACCOUNT
    END-EXEC
    GOTO 10
600
    MOVE C'RECORD NOT FOUND' TO PRT1
    PRINT DOUBLE SPACED
    GOTO 10
900
    GOTO EOJ
9999END               /* IF VSE, PLACE "INF" CARDS NEXT
```

UPDATE Output

```
05/14/02                                    DB2              UPDATE A/R FILE EXAMPLE                          PAGE    1

 ACCOUNT   CUSTOMER LAST, FIRST NAME  TRAN-DATE   BILL-DATE          BALANCE     INSTL-BAL     INSTL-PAY   PAYMENT     INTEREST

 '1014021'      1702.42   KO    **INPUT RECORD
 1014021   WOOD           ROGER D    1991-10-08  1992-07-17         1,872.66        287.33         49.04     47.89         1.15
 '2002299'       413.58   IO    **INPUT RECORD

       413.58      BALANCE
 '1013904'      1142.31   MF    **INPUT RECORD
 1013904   HINDERLICH     PAUL       1991-10-10  1992-07-19         1,256.54        136.52         23.30     22.75          .55
 '9020039'        58.24   WO    **INPUT RECORD

        58.24      BALANCE
 '9003333'         RNF          **INPUT RECORD

 RECORD NOT FOUND
 '2002922'      2541.70   MO    **INPUT RECORD

 RECORD CANNOT BE UPDATED             -- NO BILLING DATE.
 '2006286'      1108.92   MA    **INPUT RECORD
 2006286   BALDWIN        HARRY      1991-06-09  1992-06-03         1,219.81      1,296.88         59.55     54.04         5.51
 '6084444'         RNF          **INPUT RECORD

 RECORD NOT FOUND
 '6202222'         RNF          **INPUT RECORD

 RECORD NOT FOUND
 '2013568'      3886.33   MA    **INPUT RECORD

 RECORD CANNOT BE UPDATED             -- NO BILLING DATE.



 00000010 RECORDS FOR INF  FILE
```

ADD Program

```
************************************************************
*                                                          *
* ADD:     ADD RECORDS TO DB2 TABLE, USING MEMBER "ADDINPUT". *
*                                                          *
* A PERFORMANCE ADVANTAGE CAN BE GAINED BY USING THE       *
* 'TOKEN TOKEN# HOLD' OPTION WHENEVER POSSIBLE.            *
*                                                          *
* TO VIEW THE DIFFERENCE, RUN THIS TEST JOB AGAINST THE    *
* DELETE TEST JOB, USING THE OPTION AND THEN NOT USING THE *
* OPTION.  BE SURE TO EXECUTE THE FOLLOWING STATEMENT IN   *
* THE 'PROCEDURE' SECTION OF YOUR VISION:REPORT PROGRAM    *
* FIRST:                                                   *
*                                                          *
*       EXEC SQL                                           *
*            TRACE ALL                                     *
*       END-SQL                                            *
*                                                          *
* NOTICE THE DIFFERENCE IN SIZE OF THE TR#SQL TRACE DATASET. *
* THE DIFFERENCE REFLECTS THE ADDITIONAL SQL PREPARE STEPS  *
* THAT MUST BE DONE WHEN THE OPTION IS NOT USED.           *
*                                                          *
************************************************************
*INFCARD                          /* IF VSE, REMOVE * IN POS. 1
*
* TRANSACTION FILE DESCRIPTION
*
EQU  TR-KEY             INF1-9          /* KEY: ACCT-CD, ACCT #
EQU  TR-ACCT-CD         INF1-2          /*      ACCT-CD
EQU  TR-ACCOUNT         INF3-9          /*      ACCOUNT NUMBER
EQU  TR-LAST-NAME       INF10-18        /* LAST NAME
EQU  TR-FIRST-NAME      INF19-28        /* FIRST NAME
EQU  TR-STREET          INF29-47        /* CUSTOMER STREET ADDR
EQU  TR-CITY-ST-ZIP     INF48-66        /* CUSTOMER ADDR, STATE, ZIP
EQU  TR-BALANCE         INF67-71        /* ACCOUNT BALANCE

* DEFAULT DATA
EQU  WS-MIDDLE-INIT     WST1            BLANKS
EQU  WS-ACCT-CODE       WST1
EQU  WS-INSTL-BAL       WST2-P          ZERO
EQU  WS-INSTL-PAY       WST2-P
EQU  WS-BAL-PARTPAY     WST2-P
EQU  WS-INT-PARTPAY     WST2-P
EQU  WS-NR-PAY          WST2-P

* FOR CONVERSION OF TRANSACTION CHARACTER DATA TO PACKED
EQU  WS-BALANCE         WST3-5-P

* DATE VALIDATION FIELDS
EQU  WS-TRAN-DATE       WST451-460
EQU  WS-TRAN-MONTH      WST451-452
EQU  WS-TRAN-SLASH1     WST453          C'/'
EQU  WS-TRAN-DAY        WST454-455
EQU  WS-TRAN-SLASH2     WST456          C'/'
EQU  WS-TRAN-CENTUR     WST457-458      C'19'
EQU  WS-TRAN-YEAR       WST459-460

EQU  WS-BILL-DATE       WST461-470
EQU  WS-BILL-MONTH      WST461-462
EQU  WS-BILL-SLASH1     WST463          C'/'
EQU  WS-BILL-DAY        WST464-465
EQU  WS-BILL-SLASH2     WST466          C'/'
EQU  WS-BILL-CENTUR     WST467-468      C'19'
EQU  WS-BILL-YEAR       WST469-470
```

```
EQU  WS-TRAN-NULL       WST471-472-B
EQU  WS-BILL-NULL       WST473-474-B

EQU  TODAY-MONTH        VAL50-51
EQU  TODAY-DAY          VAL52-53
EQU  TODAY-YEAR         VAL54-55

* INITIALIZE SQL DATE HOST VARIABLES WITH TODAY'S DATE
   MOVE TODAY-MONTH   TO   WS-TRAN-MONTH
   MOVE TODAY-DAY     TO   WS-TRAN-DAY
   MOVE TODAY-YEAR    TO   WS-TRAN-YEAR
   MOVE  X'0000' TO WS-TRAN-NULL

   MOVE TODAY-MONTH   TO   WS-BILL-MONTH
   MOVE TODAY-DAY     TO   WS-BILL-DAY
   MOVE TODAY-YEAR    TO   WS-BILL-YEAR
   MOVE  X'0000' TO WS-BILL-NULL

   MOVE C'INPUT RECORDS (FOR ADD) FOLLOW:'   TO PRT1
   PRINT DOUBLE SPACED

100
   GET INF ATEND 900

   MOVE INF1-80       TO   PRT1
   PRINT

   MOVE TR-BALANCE TO WS-BALANCE

   EXEC SQL INSERT
     INTO ACCOUNTSRECEIVABLE
     (
      AR_ACCOUNT       ,
      AR_TRAN_DATE     ,
      AR_BILL_DATE     ,
      AR_LAST_NAME     ,
      AR_FIRST_NAME    ,
      AR_MIDDLE_INIT   ,
      AR_STREET        ,
      AR_CITY_ST_ZIP   ,
      AR_BALANCE       ,
      AR_ACCT_CODE     ,
      AR_INST_BALANCE  ,
      AR_INST_PAY      ,
      AR_BAL_PARTPAY   ,
      AR_INT_PARTPAY   ,
      AR_NR_PAY        ,
      AR_KEY
     )
     VALUES
     (
      :TR-ACCOUNT                    ,
      :WS-TRAN-DATE    :WS-TRAN-NULL ,
      :WS-BILL-DATE    :WS-BILL-NULL ,
      :TR-LAST-NAME                  ,
      :TR-FIRST-NAME                 ,
      :WS-MIDDLE-INIT                ,
      :TR-STREET                     ,
      :TR-CITY-ST-ZIP                ,
      :WS-BALANCE                    ,
      :WS-ACCT-CODE                  ,
      :WS-INSTL-BAL                  ,
      :WS-INSTL-PAY                  ,
      :WS-BAL-PARTPAY                ,
      :WS-INT-PARTPAY                ,
```

```
      :WS-NR-PAY                      ,
      :TR-KEY
     )
* END-EXEC TOKEN 1 HOLD SQLERROR 200
  END-EXEC SQLERROR 200

* TO OBSERVE PERFORMANCE ADVANTAGE OF THE 'TOKEN TOKEN# HOLD' OPTION
* USE ONE OR THE OTHER OF THE 'END-EXEC' STATEMENTS ABOVE.
* THE 'TOKEN TOKEN# HOLD' OPTION IS PREFERED.

     GOTO 100

200
* CHECK SQL ERROR CODE FOR DUPLICATE VALUE IN UNIQUE INDEX COLUMN
   IF @VAL-SQL-CODE EQ X'FFFFFCDD' /* -803 = DUPLICATE KEY
     MOVE  TR-KEY                           TO PRT1
     MOVE  C'<==== DUPLICATE ACCOUNT NUMBER' TO PRT12
     PRINT
     GOTO 100.

* UNEXPECTED SQL ERROR -- OUTPUT SQL ERROR MESSAGE
   PRINTHEX @VAL-SQL-MALL @VAL-SQL-MLEN
   MOVE C'0008' TO @VAL-RETURN-CD

900
   GOTO EOJ

9999END               /* IF VSE, PLACE "INF" CARDS NEXT
```

ADD Output

```
INPUT RECORDS (FOR ADD) FOLLOW:
 KO8039876SMITH    JOHNNY    123 MAIN ST        BANGOR, ME 00202   90025
 PO9009876SMYTHE   JOHN      87 COPPER AVENUE   DENVER, CO 82345   28431
 VO6089876SCHMIDT  JOHN      456 RIVERSIDE WAY  BILOXI, MI 23456   45678
 MA6209876DOVER    BENJAMIN  1 WAY STREET       CHICAGO, IL 60606  39803
 MA6209876VEE      SAM       2 KEY DRIVE        MIAMI, FL 12345    11112
 6209876    <<==== DUPLICATE ACCOUNT NUMBER
 WO7059876BEE      SAM       63 BASIC WAY       NEW YORK, NY 22222 11834
 WE7109876QU       SAM       22 SEQUENT ST      SEATTLE, WA 81405  11834
 00000007 RECORDS FOR INF  FILE
```

DELETE Program

```
*************************************************************
*                                                           *
* DELETE:  DELETE RECORDS FROM DB2 TABLE, USING MEMBER      *
*          "ADDINPUT"  (REVERSE OUT THE ADDED RECORDS FROM  *
*          RUNNING THE 'ADD' JOBSTEP PREVIOUSLY).           *
*                                                           *
* A PERFORMANCE ADVANTAGE CAN BE GAINED BY USING THE        *
* 'TOKEN TOKEN# HOLD' OPTION WHENEVER POSSIBLE.             *
*                                                           *
* TO VIEW THE DIFFERENCE, RUN THE ADD TEST JOB AGAINST THE  *
* DELETE TEST JOB, USING THE OPTION AND THEN NOT USING THE  *
* OPTION.  BE SURE TO EXECUTE THE FOLLOWING STATEMENT IN    *
* THE 'PROCEDURE' SECTION OF YOUR VISION:REPORT PROGRAM     *
* FIRST:                                                    *
*       EXEC SQL                                            *
*            TRACE ALL                                      *
*       END-EXEC                                            *
*                                                           *
* NOTICE THE DIFFERENCE IN SIZE OF THE TR#SQL TRACE DATASET.*
* THE DIFFERENCE REFLECTS THE ADDITIONAL SQL PREPARE STEPS  *
* THAT MUST BE DONE WHEN THE OPTION IS NOT USED.            *
*                                                           *
*************************************************************
*
*INFCARD                          /* IF VSE, REMOVE * IN POS. 1
*
* TRANSACTION FILE DESCRIPTION
*
EQU  TR-KEY             INF1-9            /* KEY:  ACCT-CD, ACCT #
EQU  TR-ACCT-CD         INF1-2            /*       ACCT-CD
EQU  TR-ACCOUNT         INF3-9            /*       ACCOUNT NUMBER

    HDR 1A 1 $IPLDAT$                                       DB2
    HDR 1B   DELETE FROM A/R FILE                           PAGE $PG$
    HDR 2A 0INPUT RECORDS TO DELETE FOLLOW:

010  GET  INF ATEND 900              /* GET DETAIL INPUT FILE

     MOVE  INF1-80             TO   PRT1
     PRINT

* DELETE DB2 RECORD
   EXEC SQL
       DELETE FROM ACCOUNTSRECEIVABLE
       WHERE AR_KEY     = :TR-KEY
*  END-EXEC TOKEN 1 HOLD SQLERROR 800
   END-EXEC SQLERROR 800

* TO OBSERVE PERFORMANCE ADVANTAGE OF THE 'TOKEN TOKEN# HOLD' OPTION
* USE ONE OR THE OTHER OF THE 'END-EXEC' STATEMENTS ABOVE.
* THE 'TOKEN TOKEN# HOLD' OPTION IS PREFERED.

   GOTO 010

* ERROR ROUTINE
800  MOVE  TR-KEY                          TO PRT1
     MOVE  C'<==== NRF OR ERROR'           TO PRT12
     PRINT
     GOTO 10.
900
   GOTO EOJ
9999END               /* IF VSE, PLACE "INF" CARDS NEXT
```

DELETE Output

```
05/14/02                                DB2              DELETE FROM A/R FILE                          PAGE    1

 INPUT RECORDS TO DELETE FOLLOW:

 KO8039876SMITH    JOHNNY    123 MAIN ST        BANGOR, ME 00202   90025
 PO9009876SMYTHE   JOHN      87 COPPER AVENUE   DENVER, CO 82345   28431
 VO6089876SCHMIDT  JOHN      456 RIVERSIDE WAY  BILOXI, MI 23456   45678
 MA6209876DOVER    BENJAMIN  1 WAY STREET       CHICAGO, IL 60606  39803
 MA6209876VEE      SAM       2 KEY DRIVE        MIAMI, FL 12345    11112
 WO7059876BEE      SAM       63 BASIC WAY       NEW YORK, NY 22222 11834
 WE7109876QU       SAM       22 SEQUENT ST      SEATTLE, WA 81405  11834



 00000007 RECORDS FOR INF  FILE
```

DUMP Program

```
*****************************************************************
*                                                               *
* DUMP:    DUMP DB2 TABLE TO A VSAM FILE, PRINT FILE.           *
*                                                               *
*  ===>    WE NEED TO RUN IDCAMS FIRST; CHANGE 'VOLSER'.        *
*                                                               *
*****************************************************************
*INFKSDS    0352                  /* IF VSE, REMOVE * IN POS. 1
*
*   VSAM  ACCOUNTS RECEIVABLE  FILE DESCRIPTION
*
EQU  VS-RECORD          OFA1-352          /* ACCOUNT RECORD
EQU  VS-ACCOUNT         OFA4-10           /* ACCOUNT NUMBER
EQU  VS-TRAN-DATE       OFA38-43       D  /* TRANSACTION DATE
EQU  VS-TRAN-MONTH      OFA38-39          /* TRANSACTION MONTH
EQU  VS-TRAN-DAY        OFA40-41          /* TRANSACTION DAY
EQU  VS-TRAN-YEAR       OFA42-43          /* TRANSACTION YEAR
EQU  VS-BILL-DATE       OFA44-49       D  /* BILLING DATE
EQU  VS-BILL-MONTH      OFA44-45          /* BILLING MONTH
EQU  VS-BILL-DAY        OFA46-47          /* BILLING DAY
EQU  VS-BILL-YEAR       OFA48-49          /* BILLING YEAR
EQU  VS-LAST-NAME       OFA11-25          /* LAST NAME
EQU  VS-FIRST-NAME      OFA26-35          /* FIRST NAME
EQU  VS-MIDDLE-INIT     OFA36             /* MIDDLE INITIAL
EQU  VS-STREET          OFA110-134        /* CUSTOMER STREET ADDR
EQU  VS-CITY-ST-ZIP     OFA135-159        /* CUSTOMER ADDR, STATE, ZIP
EQU  VS-BALANCE         OFA170-174-P  2C  /* ACCOUNT BALANCE
EQU  VS-ACCT-CODE       OFA182-183        /*
EQU  VS-INSTL-BAL       OFA191-196-P  2C  /* AMT LEFT ON INSTALLMENT
EQU  VS-INSTL-PAY       OFA197-201-P  2C  /* PLANNED INSTALLMENT PAYMT
EQU  VS-BAL-PARTPAY     OFA202-205-P  2C  /* ACTUAL PAYMENT
EQU  VS-INT-PARTPAY     OFA206-208-P  2C  /* INTEREST, PART PAYMENT
EQU  VS-NR-PAY          OFA209-210-P  2C
EQU  VS-KEY             OFA211-219        /* KEY

* SQL DATE HOST VARIABLES -- YYYY-DD-MM DATE FORMAT
EQU  WS-TRAN-DATE       WST451-460
EQU  WS-TRAN-CENTUR     WST451-452     C'19'
EQU  WS-TRAN-YEAR       WST453-454
EQU  WS-TRAN-DASH1      WST455         C'-'
EQU  WS-TRAN-MONTH      WST456-457
```

```
EQU  WS-TRAN-DASH2      WST458         C'-'
EQU  WS-TRAN-DAY        WST459-460

EQU  WS-BILL-DATE       WST461-470
EQU  WS-BILL-CENTUR     WST461-462     C'19'
EQU  WS-BILL-YEAR       WST463-464
EQU  WS-BILL-SLASH1     WST465         C'-'
EQU  WS-BILL-MONTH      WST466-467
EQU  WS-BILL-SLASH2     WST468         C'-'
EQU  WS-BILL-DAY        WST469-470

EQU  WS-TRAN-NULL       WST471-472-B
EQU  WS-BILL-NULL       WST473-474-B

* LINE 1
EQU  PR-ACCOUNT         PRT1              /* ACCOUNT NUMBER
EQU  PR-LAST-NAME       PRT11             /* LAST NAME
EQU  PR-FIRST-NAME      PRT26             /* FIRST NAME
EQU  PR-TRAN-DATE       PRT37             /* TRANSACTION DATE
EQU  PR-BILL-DATE       PRT49             /* BILLING DATE
EQU  PR-BALANCE         PRT63             /*
EQU  PR-INSTL-BAL       PRT75             /* AMT LEFT ON INSTALLMENT
EQU  PR-INSTL-PAY       PRT91             /* PLANNED INSTALLMENT PAYMT
EQU  PR-BAL-PARTPAY     PRT104            /* ACTUAL PAYMENT
EQU  PR-INT-PARTPAY     PRT120            /* INTEREST, PART PAYMENT

    HDR 1A 1 $IPLDAT$                                          DB2
    HDR 1B   DUMP  A/R FILE TO VSAM AND PRINT OUT              PAGE $PG$
    HDR 2A 0ACCOUNT   CUSTOMER LAST, FIRST NAME  TRAN-DATE   BILL-DATE
    HDR 2B   BALANCE     INSTL-BAL     INSTL-PAY   PAYMENT     INTEREST

* SET OUTPUT BUFFER SIZE
     MOVE C'352' TO @VAL-VSAMLRECL
     OPEN OFA
* SET RECORD LENGTH
     SET PTA OFA1
     SET PTA DOWN 2
     MOVE P'352'         TO  PTA1-2-B
     MOVE  SPACES        TO  OFA1-352

* SPECIFY SQL COLUMNS TO BE RETRIEVED
* AND THE ORDER IN WHICH THE ROWS ARE TO APPEAR.
    EXEC SQL DECLARE BLUEJAY CURSOR
       FOR SELECT
      AR_ACCOUNT,
      AR_TRAN_DATE     ,
      AR_BILL_DATE     ,
      AR_LAST_NAME     ,
      AR_FIRST_NAME    ,
      AR_MIDDLE_INIT   ,
      AR_STREET        ,
      AR_CITY_ST_ZIP   ,
      AR_BALANCE       ,
      AR_ACCT_CODE     ,
      AR_INST_BALANCE  ,
      AR_INST_PAY      ,
      AR_BAL_PARTPAY   ,
      AR_INT_PARTPAY   ,
      AR_NR_PAY ,
      AR_KEY
       FROM ACCOUNTSRECEIVABLE
       ORDER BY AR_KEY
    END-EXEC

    EXEC SQL OPEN
       BLUEJAY
```

```
    END-EXEC

* SPECIFY VISION:REPORT STATEMENT TO RECEIVE CONTROL
* WHEN ALL ROWS HAVE BEEN RETREIVED

    EXEC SQL WHENEVER
       NOT FOUND GOTO 500
    END-EXEC

400

* SPECIFY THE VISION:REPORT AREAS INTO WHICH THE SQL
* COLUMNS ARE TO BE PLACED

   EXEC SQL FETCH
     BLUEJAY INTO
      :VS-ACCOUNT,
      :WS-TRAN-DATE    :WS-TRAN-NULL,
      :WS-BILL-DATE    :WS-BILL-NULL  ,
      :VS-LAST-NAME                   ,
      :VS-FIRST-NAME                  ,
      :VS-MIDDLE-INIT                 ,
      :VS-STREET                      ,
      :VS-CITY-ST-ZIP                 ,
      :VS-BALANCE                     ,
      :VS-ACCT-CODE                   ,
      :VS-INSTL-BAL                   ,
      :VS-INSTL-PAY                   ,
      :VS-BAL-PARTPAY                 ,
      :VS-INT-PARTPAY,
      :VS-NR-PAY ,
      :VS-KEY
   END-EXEC

      MOVE C'######' TO VS-TRAN-DATE
   IF WS-TRAN-NULL EQ X'0000'
      MOVE WS-TRAN-MONTH TO VS-TRAN-MONTH
      MOVE WS-TRAN-DAY   TO VS-TRAN-DAY
      MOVE WS-TRAN-YEAR  TO VS-TRAN-YEAR.

      MOVE C'######' TO VS-BILL-DATE
   IF WS-BILL-NULL EQ X'0000'
      MOVE WS-BILL-MONTH TO VS-BILL-MONTH
      MOVE WS-BILL-DAY   TO VS-BILL-DAY
      MOVE WS-BILL-YEAR  TO VS-BILL-YEAR.

   MOVE VS-KEY           TO PR-ACCOUNT
   MOVE VS-LAST-NAME     TO PR-LAST-NAME
   MOVE VS-FIRST-NAME    TO PR-FIRST-NAME
   IF VS-TRAN-MONTH   IS NOT EQ  C'##'
      MOVE VS-TRAN-DATE     TO PR-TRAN-DATE.
   IF VS-BILL-MONTH   IS NOT EQ  C'##'
      MOVE VS-BILL-DATE     TO PR-BILL-DATE.
   MOVE VS-INT-PARTPAY   TO PR-INT-PARTPAY
   MOVE VS-BAL-PARTPAY   TO PR-BAL-PARTPAY
   MOVE VS-INSTL-PAY     TO PR-INSTL-PAY
   MOVE VS-INSTL-BAL     TO PR-INSTL-BAL
   MOVE VS-BALANCE       TO PR-BALANCE
   PRINT
   WRITE OFA
   GOTO 400
500
   GOTO EOJ
9999END
```

DUMP Output

```
05/14/02                                        DB2              DUMP  A/R FILE TO VSAM AND PRINT OUT            PAGE    1
ACCOUNT   CUSTOMER LAST, FIRST NAME  TRAN-DATE   BILL-DATE          BALANCE     INSTL-BAL      NSTL-PAY   PAYMENT     INTEREST

8006547   TORRES         ERNESTO     02/15/92    04/24/92             44.99         42.41           .00       .00          .00
6002587   HAASE          ELLA        04/07/91    03/02/92             15.00      2,230.17        102.40     92.92         9.48
6208657   CHO PYUNG      SUH         02/17/92    06/12/92             32.00        261.57         44.64     43.60         1.04
7082509   ORTIZ          DISRAELI    08/21/91    11/30/91              5.00        272.78         46.56     45.46         1.10
6024963   HILL           GARY        04/17/91    02/26/92              3.80         22.38           .00       .00          .00
6044395   CANO           MICHEAL     02/11/92    04/06/92             15.00        195.74         33.41     32.62          .79
6059708   CHAVEZ         RAY         02/09/92    05/04/92             15.00         49.39           .00       .00          .00
6095631   FODIPE         MICHAEL     02/12/92    05/21/92             45.24        486.21         82.99     81.04         1.95
6107265   GENVARDI       G           02/10/92    06/05/92             43.00        419.48         71.60     69.91         1.69
6123228   SILVA          JULIAN      01/05/90                           .00        273.99         46.76     45.67         1.09
8011508   HUGHES         RAY         02/13/92    06/08/92            178.70         47.88           .00       .00          .00
2002299   PLACIDO        ORTEGA      03/16/92    05/11/92            413.58        486.21         82.99     81.04         1.95
6009166   LOCKE          JEFFREY     02/12/92    05/21/92             15.00        234.91         40.09     39.15          .94
6112536   CHAVEZ         NORMA       09/06/91    01/01/92             55.00        496.84         84.80     82.81         1.99
6123317   VASGUEZ        IRENE       01/05/90                           .00        496.40         84.73     82.73         2.00
6218113   AGUIERA        EMILIO      02/20/92    06/15/92            108.44        197.85         33.77     32.98          .79
8012644   MONTEZ         CARMEN      02/07/92    06/16/92             89.28        484.75         82.74     80.79         1.95
1014021   WOOD           ROGER D     10/08/91    07/17/92          1,872.66        287.33         49.04     47.89         1.15
2005131   WICINSKI       ALEXANDER   04/10/91    03/19/92            848.71        261.70         44.67     43.62          .05
6004695   ABROWN         ELIZABETH   04/11/91    03/06/92             32.02        297.25         50.73     49.54         1.19
6016316   VANCE          VERNON      04/20/91    03/15/92             40.76      3,243.83        104.29     90.11        14.18
6041272   MARTIN         GAYLORD     02/10/92    04/05/92             15.00        361.80         61.75     60.30         1.45
6042325   CHAMBERLINE    FRANCES     02/11/92    04/06/92             34.24        118.62         20.25     19.77           48
6049354   SWARTZ         GEORGE      02/06/92    05/01/92             21.25        403.60         68.89     67.27         1.62
6062946   BROCK          ETHEL       02/17/92    04/26/92             77.00         75.19           .00       .00          .00
6067956   PORTER         MAY         02/22/92                         80.24      4,592.59        147.66    127.57        20.09
6080669   WILLIAMS       JAMES       02/15/92    05/10/92             24.24        435.56         74.34     72.59         1.75
6088678   TYLER          JAMES       02/20/92    05/15/92            305.64      1,426.33         65.49     59.43         6.06
6101291   RACH           MARY        02/21/92    05/30/92             35.00        172.68         29.47     28.78          .69
6103375   GARKOW         DOROTHY     02/21/92    05/30/92             25.00        153.52         26.20     25.59          .61
6116728   STEINER        ROBERT      01/03/90                         15.00      1,730.11         79.44     72.09         7.35
6216129   MUNSON         CLARENCE    02/11/92    06/20/92             15.00      1,903.48         87.40     79.31         8.09
6218512   MARTIN         GAYLORD     02/18/92    06/13/92             55.25      1,415.41         64.99     58.98         6.01
7014996   OLVERA         CLEMENTINE  10/20/91    06/15/92            106.74        270.15         46.11     45.03          .08
7029926   WHITE          HIAWATHA    10/19/91    06/28/92             20.44         84.66           .00       .00          .00
7072694   WARTON         LEE         10/20/91    06/15/92            224.00      1,268.90         58.26     52.87         5.39
7100035   BROWN          RICHARD     04/22/91                         50.62        461.39         78.75     76.90         1.85
8033692   MARLETTE       YVETTE      02/10/92    05/19/92             14.95         90.75           .00       .00          .00
9005226   WILSON         W C         10/22/91    05/31/92             22.40        386.85         66.03     64.48         1.55
6114113   NERY           GENEROSO    09/07/91    01/02/92             45.24        298.66         50.98     49.78         1.20
1002775   OWENS          J           10/22/91    05/31/92              8.06        417.60         71.28     69.60         1.68
1006231   DE LAUNEY      RAYMOND     10/06/91    04/01/92            202.84         87.54           .00       .00          .00
1013637   STALLINS       LEO F       10/16/91    06/11/92            843.74        154.53         26.38     25.76          .62
1014617   MORRIS         CLARENCE E  10/17/91    06/26/92            855.22        190.82         32.57     31.80          .77
2006286   BALDWIN        HARRY       06/09/91    06/03/92          1,219.81      1,296.88         59.55     54.04         5.51
2007177   HART           GRADDIE     02/18/92    04/13/92            128.37        144.43         24.65     24.07          .58
2008831   ZERING         WILLIAM     03/10/92    05/05/92            124.95        322.87         55.11     53.81         1.30
2009072   JONES          RFUS        03/20/92    05/15/92            128.90        432.91         73.89     72.15         1.74
2011719   REED           LOWELL      03/11/92    06/06/92            127.22      2,033.34         93.36     84.72         8.64
2013568   CULLENDER      EVERETT     12/26/89                       3,886.33        135.42         23.11     22.57          .54
```

05/14/02 DB2 DUMP A/R FILE TO VSAM AND PRINT UT PAGE 2

ACCOUNT	CUSTOMER LAST,	FIRST NAME	TRAN-DATE	BILL-DATE	BALANCE	INSTL-BAL	NSTL-PAY	PAYMENT	INTEREST
6005489	HALL	JOHN	04/12/91	03/07/92	60.00	187.65	32.03	31.28	.75
6013546	KIMURA	MICHI	04/20/91	03/15/92	37.60	251.95	43.00	41.99	1.01
6016529	BARRERA	JESUS	04/20/91	03/15/92	18.60	59.84	.00	.00	.00
6031714	FENZEE	WILLIAM	04/19/91	03/28/92	20.00	428.21	73.09	71.37	1.72
6033873	NEBRENSKY	CARMEN	04/20/91	03/29/92	3.40	246.35	42.05	41.06	.99
6034241	MC COY	FRANCES	04/20/91	03/29/92	15.00	368.06	62.82	61.34	1.48
6044751	SOARES	LAWERENCE	02/15/92	04/10/92	8.15	1,223.96	56.20	51.00	5.20
6046819	YEAGER	CLIFFORD	02/15/92	04/10/92	15.04	122.14	20.85	20.36	.49
6049672	YOUNG	ADELE	02/18/92	04/13/92	15.00	2,356.24	108.19	98.18	10.01
6058523	CALLEROS	MARCELINO	02/11/92	04/20/92	5.05	282.02	48.14	47.00	.14
6060749	TAYLOR	FLORENCE	02/17/92	04/26/92	20.00	359.35	61.33	59.89	1.44
6061931	DIAZ	JOSE	02/16/92	04/25/92	3.00	541.91	47.41	45.16	2.25
6063977	WALTON	MARVIN	02/18/92	04/27/92	3.00	117.36	20.03	19.56	.47
6067646	MCELENY	JAMES	02/19/92	04/28/92	15.00	2,889.87	92.91	80.27	12.64
6068316	SCHMITZ	JOHN	02/19/92	04/28/92	15.00	3,496.31	112.41	97.12	15.29
6069495	SEDGWICK	JOWARD	02/07/92	05/02/92	57.00	394.09	67.26	65.68	1.58
6081428	JOHNSON	KNUTE	02/14/92	05/09/92	4.25	3,854.12	123.91	107.06	16.85
6085547	GRAHAM	LAYRA	02/16/92	05/11/92	4.25	10.52	.00	.00	.00
6090753	BEAVER	OPAL	02/07/92	05/16/92	15.00	3,288.60	105.73	91.35	14.38
6094333	HOBDY	DOROTHY	02/09/92	05/18/92	13.26	488.63	83.40	81.44	1.96
6096093	YEAGER	CLIFFORD	02/13/92	05/22/92	3.00	227.16	38.77	37.86	.91
6096859	WILSON	ALEXANDER	02/16/92	05/25/92	4.25	229.23	39.12	38.21	.91
6098223	ANGEL	JESUS	02/19/92	05/28/92	3.00	159.04	27.14	26.51	.63
6099467	GOULDING	JAHN	02/19/92	05/28/92	3.00	2,694.34	86.63	74.84	11.79
6105629	MILLER	JOHN	01/01/90		30.00	13.79	.00	.00	.00
6106501	ZARATE	JOHN	02/10/92	06/05/92	101.20	3,548.10	114.08	98.56	15.52
6108415	HEMPE	RALPH	02/14/92	05/23/92	4.25	1,101.76	50.59	45.91	4.68
6113583	BROWN	STUART	09/06/91	01/01/92	26.20	9.01	.00	.00	.00
6123031	FLYNN	JOSEPH	01/05/90		.00	438.40	74.83	73.07	1.76
6201997	IRWIN	HAZEL	02/11/92	06/06/92	23.40	8.52	.00	.00	.00
6202616	CONNER	AUBRY	02/16/92	06/11/92	16.80	152.15	25.97	25.36	.61
6208924	WHITE	ELMER	02/17/92	06/12/92	15.00	286.33	48.87	47.72	1.15
6211151	COX	ELAINE	02/10/92	06/19/92	100.20	498.08	85.01	83.01	2.00
6215785	RODEN	HAROLD	02/11/92	06/20/92	170.91	446.37	76.19	74.40	1.79
6216412	ROUKE	CURTIS	02/12/92	06/21/92	48.00	71.96	.00	.00	.00
6219004	SOOLEY	WILLIAM	02/18/92	06/13/92	15.00	64.58	.00	.00	.00
7000227	HAKEEM	N	10/22/91	07/31/92	36.00	272.72	46.55	45.45	1.10
7004753	FORBES	JOHN	10/16/91	07/25/92	11.20	320.91	54.77	53.49	1.28
7010966	SMITH	JOHN	10/22/91	07/31/92	13.66	355.60	60.69	59.27	1.42
7011407	WHITE	ELMER	10/16/91	07/25/92	10.24	145.08	24.76	24.18	.58
7012799	RODRIGUEZ	EVARISTO	10/16/91	07/25/92	10.58	102.72	17.53	17.12	.41
7012837	WOODS	LOUISE	10/19/91	06/28/92	.00	63.87	.00	.00	.00
7023189	SIAS	EDWARDO	10/19/91	06/28/92	.00	4,322.92	138.99	120.08	18.91
7033133	DE VAULT	JERRY	10/16/91	07/25/92	11.20	93.55	.00	.00	.00
7039085	BLACK	LENORE	10/21/91	04/30/92	50.40	804.68	70.40	67.06	3.34
7099657	MALTSBERGER	JOHN A	04/22/91		6.72	25.03	.00	.00	.00
7102194	TURNER	HAROLD	04/22/91		5.12	259.44	44.28	43.24	1.04
7106246	SMITH	AUSTIN	04/08/91	02/17/92	5.12	142.29	24.29	23.72	.57
9000534	HOPKINS	BARRY P	10/22/91	07/31/92	14.00	314.79	53.73	52.47	1.26
9002626	WORRELL	TED	10/22/90	07/31/91	13.00	385.78	65.84	64.30	1.54

05/14/02 DB2 DUMP A/R FILE TO VSAM AND PRINT UT PAGE 3

ACCOUNT	CUSTOMER LAST,	FIRST NAME	TRAN-DATE	BILL-DATE	BALANCE	INSTL-BAL	NSTL-PAY	PAYMENT	INTEREST
9017828	HILL	MYRTLE G	04/22/91		11.20	46.67	.00	.00	.00
1013904	HINDERLICH	PAUL	10/10/91	07/19/92	1,256.54	136.52	23.30	22.75	.55
1014889	SULLIVAN	JOSEPH C	10/14/91	06/23/92	337.70	261.97	44.71	43.66	1.05
4003896	PETERSON	WALTER	04/15/91	02/10/92	196.00	470.12	80.24	78.35	1.89
6007678	FARREL	HAZEL	04/13/91	03/08/92	53.20	237.67	40.57	39.61	.96
7053657	BARNETT	LESTER	08/21/91	11/30/91	25.07	210.10	35.86	35.02	.84
7077289	HICKMAN	VERNON	10/19/91	02/28/92	36.95	2,200.98	101.06	91.71	9.35
7107137	VALDEZ	MARY	04/08/91	02/17/92	28.00	1,516.24	69.62	63.18	6.44
9014543	MARTINEZ	SIMON	10/22/91	01/31/92	17.44	4,538.90	145.93	126.08	19.85
2004739	STEIDLEY	GURTHA	04/19/91	03/14/92	136.23	73.98	.00	.00	.00
6008321	YEARGER	CLIFFORD	04/14/91	03/09/92	140.00	464.20	79.23	77.37	1.86
2005956	SOTO	RUFINO	05/10/91	04/05/92	125.29	178.96	30.54	29.83	.71
6063748	SEDANO	JOSE	02/18/92	04/27/92	3.00	59.74	.00	.00	.00
6091407	PINTO	GILBERT	02/08/92	05/17/92	5.00	4,475.33	143.89	124.31	19.58
8033684	WHITEHURST	CHARLES	02/10/92	05/19/92	25.00	231.77	39.56	38.63	93
2002922	FLOWERS	ETHEL	12/21/89		2,541.70	112.23	19.16	18.71	.45
6001327	CARLON	MARIANO	05/06/91	04/01/92	118.00	26.71	.00	.00	.00
6007724	HARRINGTON	RUTH	04/13/91	03/08/92	37.60	156.76	26.76	26.13	.63
6012132	ZENZOLA	MICHEAL	04/18/91	03/13/92	30.20	42.69	.00	.00	.00
6017479	BERSON	ANNA	04/07/91	03/16/92	3.00	4,340.28	139.55	120.56	18.99
6039804	MORENO	ELIZABETH	02/10/92	04/05/92	.00	325.29	55.52	54.22	1.30
6204384	FREDRICK	EMERY	02/12/92	06/07/92	15.00	621.08	54.33	51.76	2.57
8002053	JENNINGS	WILL	04/17/91	03/12/92	56.34	450.15	76.83	75.03	1.80
6106897	GASIOR	MARY	02/10/92	06/05/92	43.00	4,384.63	140.97	121.80	19.17
2014017	HERNANDEZ	CARLOS	01/04/90		258.00	313.13	53.44	52.19	1.25
6061087	RAMOS	LORENZO	02/17/92	04/26/92	31.93	238.06	40.63	39.68	.95
6091156	SONDOVAL	SUSIE	02/08/92	05/17/92	15.00	394.23	67.29	65.71	1.58
6112498	BARRON	BONNIE	02/18/92	06/27/92	36.96	430.76	73.52	71.79	1.73
8009279	JARAMILLO	MADGALENA	02/10/92	05/05/92	17.24	204.07	34.83	34.01	82
1014706	SHAW	MABEL	10/16/91	07/11/92	406.46	54.69	.00	.00	.00
6105904	CRESS	NELLIE	09/06/91	01/01/92	176.00	354.03	60.43	59.01	1.42
6209351	YOUNG	MARVIIE	02/20/92	06/15/92	46.59	310.25	52.95	51.71	1.24
0000434					11.20-	482.94	82.43	80.49	1.94
0002623					256.12-	4,413.40	141.90	122.59	19.31
1009681					10,085.55-	3,265.20	104.98	90.70	14.28
6027989				09/21/90	310.75	285.54	48.74	47.59	1.15
6045154				11/16/90	64.96	442.32	75.50	73.72	1.78
6090516				12/14/90	19.54	311.75	53.21	51.96	1.25
7046936					2.46-	128.83	21.99	21.47	.52
7053941					33.60-	3,708.70	119.24	103.02	16.22
7066775					3.32-	2,403.85	110.37	100.16	10.21
7087896					1.32-	286.48	48.90	47.75	1.15
8003173				10/28/90	35.00	185.46	31.65	30.91	.74
9010033					1.04-	133.44	22.78	22.24	.54
1001191	CARDOZA	MARY	08/13/90	10/22/90	177.52-	397.40	67.83	66.23	1.60
1006681	MYERS	EARL	10/15/91	03/10/92	123.24	240.10	40.98	40.02	.96
6023185	PORTWOOD	JOSEPHINE	04/12/91	03/21/92	15.00	442.25	75.48	73.71	.77
6099963	ROMO	JESUS	02/20/92	05/29/92	3.12	1,328.00	60.97	55.33	5.64
6216846	TORRES	PAUBLINA	02/11/92	06/20/92	.00	117.75	20.10	19.63	.47
7030142	HARRIS	JEWEL	10/19/91	06/28/92	14.56	276.16	47.13	46.03	1.10

05/14/02 DB2 DUMP A/R FILE TO VSAM AND PRINT UT PAGE 4

ACCOUNT	CUSTOMER LAST,	FIRST NAME	TRAN-DATE	BILL-DATE	BALANCE	INSTL-BAL	NSTL-PAY	PAYMENT	INTEREST
7053185	SMITH	MARY	10/22/91	05/31/92	.00	260.02	44.38	43.34	1.04
7064535	PENNEYWELL	JERMEL	10/22/91	05/31/92	40.40	145.39	24.82	24.23	.59
7068794	BACTAD	MERIAN	10/19/91	06/28/92	11.20	1.68	.00	.00	.00
7070721	VERA	AGUSTIN C	08/22/91	10/31/91	27.45	49.68	.00	.00	.00
7071507	ALLEN	CARRIE A	10/21/90	09/30/91	38.08	409.48	69.89	68.25	1.64
7083696	ORACION	ARSENIA G	10/22/91	01/31/92	67.15	362.59	61.89	60.43	1.46
7103387	PEREZ	EVELYN	04/22/91		53.76	389.72	66.52	64.95	1.57
8009155	FLORES	JOSE	02/11/92	05/06/92	40.42	379.93	64.85	63.32	1.53
8032084	ADAMS	CLAUDE	02/08/92	05/17/92	50.00	29.68	.00	.00	.00
9015884	RIZARDO	HIPOLITO J	10/20/91	06/15/92	10.04	492.76	84.10	82.13	1.97
9016643	DE MARCO	VINCENT D	10/19/91	06/28/92	33.60	178.67	30.50	29.78	.72
9018239	HUAYLLARA	BLANCA	04/22/91		89.60	95.35	.00	.00	.00
8031126	RULEY	ANNA	02/11/92	06/20/92	65.00	203.55	34.74	33.93	.81
2000504	WILSON	HERMAN	08/16/89		5,292.24	301.17	51.40	50.20	1.20
2010933	QUINTANAR	ANTONIO	02/09/92	05/18/92	1,460.61	372.04	63.50	62.01	.49
6067557	BROWNING	EVA	02/19/92	04/28/92	3.45	286.21	48.85	47.70	1.15
6103723	KITAOKA	GEORGE	02/21/92	05/30/92	20.00	3,421.71	110.01	95.05	14.96
6114253	MARTINEZ	RACHEL	09/06/91	01/01/92	51.60	171.55	29.28	28.59	.69
7085494	JARAMILLO	BOB	10/19/91	02/28/92	20.00	438.88	74.91	73.15	1.76
8031401	SPECTOR	MICHEAL	10/07/91	01/02/92	28.00	2,567.75	82.56	71.33	11.23
6089771	HERNSTORM	VICTOR	02/07/92	05/16/92	41.91	100.98	17.24	16.83	.41
7109857	CRITTENDEN	CHARLE	04/22/91		11.20	268.64	45.85	44.77	1.08
6019587	BENAVIDEZ	CESARIO	04/13/91	03/22/92	15.00	296.01	50.52	49.34	1.18
6032214	GONZALEZ	RAUL	04/19/91	03/28/92	30.40	160.50	27.39	26.75	.64
6044832	ROBBINS	WILLIAM	02/15/92	04/10/92	.00	470.29	80.27	78.38	1.89
6066402	ALCARAZ	ANDRES	02/22/92		32.80	479.07	81.77	79.85	1.92
6074359	ASHLEY	JOHN	02/08/92	05/03/92	15.00	254.97	43.52	42.50	1.02
6118577	NAULLS	MARK	01/03/90		15.00	425.26	72.58	70.88	1.70
6206328	VILLASENOR	TERESA	02/14/92	06/09/92	15.00	3,279.08	105.43	91.09	4.34
6208347	HAMELTON	MERDEDES	02/19/92	06/14/92	15.00	52.68	.00	.00	.00
6216617	TAUTRIM	RICHARD	09/06/91	01/01/92	140.00	156.25	26.67	26.04	.63
7043732	MARTINEZ	AMILCAR	10/21/88	04/30/89	108.25	137.16	23.41	22.86	.55
7054645	COWAN	EDWARD	10/22/91	07/31/92	22.40	64.37	.00	.00	.00
7059582	HARRIS	ZELMAL	08/22/89	10/31/89	44.75	444.71	75.90	74.12	1.78
7071361	BAEZA	REBECCA	10/22/90	01/31/91	66.00	170.52	29.10	28.42	.68
7084447	JACKSON	GARLAND	10/22/91	01/31/92	121.55	359.98	61.44	60.00	1.44
7090943	GILL	BILLY L	10/18/91	06/13/92	11.20	49.28	.00	.00	.00
7093853	CHAVEZ	ROBERT L	10/19/91	06/28/92	265.74	396.27	67.64	66.05	1.59
7097166	MORGAN	ROGER H	10/22/91	07/31/92	44.45	751.03	65.70	62.59	3.11
7098375	VALDEZ	JACK	10/16/91	07/25/92	22.40	45.96	.00	.00	.00
7112947	MARTINEZ	DOMINGO	04/08/91	02/17/92	.00	438.52	74.85	73.09	1.76
8001472	NARANJO	DELIA	04/15/91	03/10/92	115.60	492.96	84.14	82.16	1.98
8003246	BADILY	NEIL	04/10/91	03/19/92	114.23	201.46	34.39	33.58	.81
8010641	MURPHY	MATTHEWS	02/09/92	06/04/92	82.56	2,256.23	103.59	94.01	.58
8011036	ANDERWS	CHARLES	02/07/92	06/02/92	114.52	196.87	33.60	32.81	.79
8011699	OBRESON	FRANK	02/10/92	06/05/92	57.50	292.24	49.88	48.71	1.17
8031096	ROMERP	ARMENDO	02/08/92	06/17/92	81.26	115.05	19.64	19.18	.46
9007156	SANCHO	CARLOS	10/19/91	06/28/92	352.80	295.42	50.42	49.24	1.18
9009469	CARMICHAEL	GEORGE	10/22/88	05/31/89	72.50	498.63	85.11	83.11	.00
9020039	LOPEZ	SALVADOR	04/08/91	02/17/92	58.24	4,745.73	152.58	131.83	20.75

00000008 RECORDS FOR OFA FILE

DDF Program

The DDF is a sample program only. It cannot run in your installation without extensive modifications. It is only meant to demonstrate the basics of running a DDF job stream. DDF is for MVS only.

```
*****************************************************************
*                                                               *
* DDF:     SAMPLE DISTRIBUTED DATA FACILITY (DDF)               *
*          THIS IS A DB2-TO-DB2 SAMPLE ONLY, AND IS FOR         *
*          MVS ONLY.    <=====                                  *
*                                                               *
*   ===>   THIS SAMPLE CANNOT RUN IN YOUR INSTALLATION          *
*          WITHOUT A LOT OF EXTENSIVE MODIFICATION.             *
*                                                               *
*   ===>   THIS SAMPLE IS ONLY MEANT TO DEMONSTRATE THE         *
*          BASICS OF RUNNING A DDF JOBSTREAM USING              *
*          VISION:Report Interface to DB2.                      *
*                                                               *
*          READ RECORDS FROM ANOTHER SYSTEMS DB2 TABLE          *
*                                                               *
*****************************************************************
OPTION  SQLPLNM=REPORT2,SQLSYSN=DB2T
EQU      INPUT-AREA          INF1-80          /* MY ENTIRE WORKAREA
  EQU    IN-NAME             INF1-10
  EQU    IN-DATE             INF12-21
  EQU    IN-TIME             INF23-30
  EQU    IN-TIMESTAMP        INF32-57

EQU      DATA-AREA           WST0-0
*     * DB2 COPYBOOK FOR TABLE ISPHCM1.TEST02
  EQU  KEY_NAME_LEN          (2)-B
  EQU  KEY_NAME              (10)
  EQU  DATA_DATE             (10)      /* TIME
  EQU  DATA_TIME             (8)       /* TIME
  EQU  DATA_TIMESTAMP        (26)      /* TIMESTAMP

    TITLE  'DDF: SAMPLE 9 '
    TITLE2 'DATE, TIME, TIMESTAMP TEST'

    EXEC SQL CONNECT TO 'DB2U_LCS_WH'
    END-EXEC
    EXEC SQL SET CURRENT PACKAGESET = 'REPORTP1'
    END-EXEC

     EXEC SQL
       DECLARE TIME-C CURSOR FOR
       SELECT KEY_NAME, DATA_DATE, DATA_TIME, DATA_TIMESTAMP
              FROM ISPHCM1.TEST02
     END-EXEC TOKEN 1

     REPORT KEY_NAME DATA_DATE DATA_TIME DATA_TIMESTAMP

     EXEC SQL OPEN TIME-C
     END-EXEC

100  MOVE SPACES TO KEY_NAME
     EXEC SQL FETCH TIME-C INTO :KEY_NAME_LEN SQLTYPE VARCHAR(10),
                    :DATA_DATE SQLTYPE DATE, :DATA_TIME SQLTYPE TIME,
                    :DATA_TIMESTAMP SQLTYPE TIMESTAMP
       END-EXEC NOTFOUND 200
     PRINT REPORT
```

```
     GOTO 100

200  MOVE C'CURRENT PACKAGESET =' TO PRT1
     EXEC SQL SET :PRT23-40 = CURRENT PACKAGESET
        END-EXEC
     PRINT DOUBLESPACED
     MOVE C'CURRENT SERVER = ' TO PRT1
     EXEC SQL SET :PRT23-40 = CURRENT SERVER
        END-EXEC
     PRINT DOUBLESPACED
209  EXIT
     EXEC SQL COMMIT
       END-EXEC
     EXEC SQL CONNECT RESET
       END-EXEC
     PERFORM 200 THRU 209
     GOTO EOJ
9999END
```

COMMIT Program

```
*****************************************************************
*                                                               *
* COMMIT:  SAMPLE ONLY OF HOW TO USE THE SQL COMMANDS,          *
*          "COMMIT WORK" AND "ROLLBACK WORK"                    *
*                                                               *
*****************************************************************
 TITLE1 'SAMPLE 10 - COMMIT'
     EXEC SQL
        COMMIT WORK
     END-EXEC

     EXEC SQL
        ROLLBACK WORK
     END-EXEC

     GOTO EOJ
9999END
```

Chapter

11 SQL and SQLCA Error Handling

Error Handling Using WHENEVER Statement

As with standard embedded SQL, VISION:Report Interface to DB2 supports the WHENEVER statement for error handling. The WHENEVER statement specifies a location for the program to branch to when a particular kind of DB2 error occurs.

WHENEVER Statement Syntax

The syntax of the WHENEVER statement is:

```
EXEC SQL
 WHENEVER condition GOTO qj-label (or sequence number)
END-EXEC
```

or

```
EXEC SQL
 WHENEVER condition CONTINUE
END-EXEC
```

or

```
EXEC SQL
 WHENEVER condition STOP
END-EXEC
```

Where:

`condition`	Is one of the following conditions:	
	NOT FOUND	After the last row has been retrieved
	SQLWARNING	When a non-fatal DB2 error occurs
	SQLERROR	When a fatal DB2 error occurs
`qj-label` `or` `sequence number`	Must be defined at some point in the VISION:Report program.	

`CONTINUE`	For a particular condition, resets error-handling for the condition so that processing does not branch to the previously specified label in the event a subsequent error occurs.
`STOP`	Indicates go to normal EOJ. However, for SQLERROR, STOP prints the SQL messages associated with the error.

Understanding How the WHENEVER Statement Works

WHENEVER statements are not executed. These statements define the condition under which processing continues if any subsequent SQL statement causes an error. More importantly, the normal flow of control through the program during execution has no effect on which WHENEVER statement is used for a particular SQL statement.

WHENEVER statements are processed when the VISION:Report program is compiled and are applied to all SQL statements following the particular WHENEVER statement without regard to other VISION:Report logic (unless temporarily overridden by an END-EXEC option).

A single occurrence of a WHENEVER statement at the top of a VISION:Report program provides error handling for all subsequent EXEC SQL statements if not modified later in the program.

When There are no WHENEVER Statments

If no WHENEVER statements are used, VISION:Report provides automatic job termination when SQLERROR conditions are encountered. A four-line message is printed and the program halts with a return code of 3337.

An SQLERROR STOP statement also prints these four lines. If you provide your own SQLERROR handling, the defaults do not occur, and you must manually close the cursor and terminate processing.

Error Handling Using Status Variables

Instead of using WHENEVER statements for error handling, you can choose to inspect the status variable @VAL-SQL-CODE.

- @VAL-SQL-CODE contains the same value as SQLCODE in a standard embedded SQL program written in another language.
- @VAL-SQL-CODE is assigned one of the following values after each SQL statement:

Code	Explanation
0	Valid completion of SQL statement
100	End of cursor data or no data found
> 0	Warning condition (except 100)
< 0	Fatal error condition

A complete list of standard SQL codes (in the range -32K to +32K) can be found in the IBM DB2 or SQL/DS Reference Manual.

Additional VISION:Report Interface to DB2 codes, which are returned in @VAL-SQL-CODE, can be found in *VISION:Report Messages and Codes*.

Debugging Aids

For execution-time problems involving VISION:Report programs (having embedded SQL statements), make the following changes and rerun the program, before contacting Computer Associates Technical Support.

Adding OPTION LISTOPT

Place the following statement at the beginning of your VISION:Report program:

```
OPTION LISTOPT=YES,SPIE=NO                                    MVS
OPTION LISTOPT=YES,STXITPC=NO                                 VSE
```

Adding TRACE ALL

Place the following statement in the Procedure Division of your VISION:Report program, preferably closest to where the problem occurs:

```
TRACE ALL
```

Wherever you think the error starts to occur, place the following statement in your VISION:Report program:

```
EXEC SQL
    TRACE ALL
END-EXEC
```

Adding TR#SQL

For MVS, add the following statement to your execution JCL:

```
//TR#SQL DD SYSOUT=*
```

Overriding QJTSTDB2

If you are using the QJTSTDB2 procedure, you can override it as follows:

```
//QJ.TR#SQL DD SYSOUT=*
```

See the *Advantage VISION:Report Advantage VISION:Forms Reference Guide* for more information about debugging.

Access to the SQLCA

The SQL communications area, (SQLCA) is updated by DB2/SQL after the SQL request is passed to the database.

- For severe VISION:Report Interface to DB2 errors, the only meaningful information is the return code.
- Otherwise, the information is meaningful to the specific request.

For more information, use the VAL names listed below. The SQLCA fields and their meanings are outlined as follows:

VAL Name	SQL Name	LNG	Type	Description
@VAL-SQLCA	SQLCA	136	CH	Entire SQL communications area, detailed as follows:
@VAL-SQL-CAID	SQLCAID	8	CH	Eye catcher (such as, SQLCA)
@VAL-SQL-LEN	SQLCABC	4	BI	Length of SQLCA (such as, 136)
@VAL-SQL-CODE	SQLCODE	4	BI	SQL return code
@VAL-SQL-ERRM	N/A	72	CH	Contains both of the following fields:
@VAL-SQL-ERRML	SQLERRML	2	CH	Length of error message that follows
@VAL-SQL-ERRMC	SQLERRMC	70	CH	Tokenized error message separated by X"FF"
@VAL-SQL-ERRP	SQLERRP	8	CH	DB2/SQL internal diagnostic information such as module name
@VAL-SQL-ERRD1	SQLERRD1	4	BI	Diagnostic information
@VAL-SQL-ERRD2	SQLERRD2	4	BI	Diagnostic information
@VAL-SQL-ERRD3	SQLERRD3	4	BI	Diagnostic information most useful to an applications program. After an INSERT, UPDATE, or DELETE operation, it indicates the number of rows added, changed, or deleted
@VAL-SQL-ERRD4	SQLERRD4	4	BI	Diagnostic information

VAL Name	SQL Name	LNG	Type	Description
@VAL-SQL-ERRD5	SQLERRD5	4	BI	Diagnostic information
@VAL-SQL-ERRD6	SQLERRD6	4	BI	Diagnostic information
@VAL-SQL-WARN	N/A	8	CH	Contains first eight of warning flags as follows:
@VAL-SQL-WARN0	SQLWARN0	1	CH	If set to “W”, at least one of the following flags is set to “W”:
@VAL-SQL-WARN1	SQLWARN1	1	CH	DB2/SQL warning flag 1
@VAL-SQL-WARN2	SQLWARN2	1	CH	DB2/SQL warning flag 2
@VAL-SQL-WARN3	SQLWARN3	1	CH	DB2/SQL warning flag 3
@VAL-SQL-WARN4	SQLWARN4	1	CH	DB2/SQL warning flag 4
@VAL-SQL-WARN5	SQLWARN5	1	CH	DB2/SQL warning flag 5
@VAL-SQL-WARN6	SQLWARN6	1	CH	DB2/SQL warning flag 6
@VAL-SQL-WARN7	SQLWARN7	1	CH	DB2/SQL warning flag 7
@VAL-SQL-WARN8	SQLWARN8	1	CH	DB2/SQL warning flag 8
@VAL-SQL-WARN9	SQLWARN9	1	CH	DB2/SQL warning flag 9
@VAL-SQL-WARNA	SQLWARNA	1	CH	DB2/SQL warning flag A
@VAL-SQL-EXT	SQLSTATE	5	CH	Return code for the outcome of the most recent execution of an SQL statement
@VAL-SQL-LSTFN	SQLLFN	2	BI	End of IBM SQL communication area

Note: CH is character type and BI is binary type.

Index

:

@

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Z