Modifying Data

Chapter 2:

Objectives

You will learn:

- Views and expressions.
- NULL and DEFAULT keywords.
- How to utilize the different inserts - single row, from another table, and inserting data into a ROWID Column.
- Inserting using expressions and self-referencing SELECT.
- Updates using SELECT.
- DELETE statement.
Views and Expressions

- A view, nickname, or SQL expression can be inserted into a table.
- For views and SQL expressions, the following rules apply:
  - The list of columns selected cannot include a column function such as MIN.
  - There must be no GROUP BY or HAVING acting on the select list.
  - The list of columns selected must include all those needed to insert a new row.
  - The list of columns selected cannot be from a constant, expression, or scalar function.

Direct INSERT Statement - Example

- In order to insert a single row, where all of the columns are populated, the values to be input need to be in the same order as the columns defined in the table:

```sql
INSERT INTO EMP_ACT_COPY
VALUES('100000' , 'ABC' , 10 , 1.4 ,
       '2003-10-22', '2003-11-24');
```
Multiple Row
INSERT Statement - Example

• If multiple rows are inserted in one statement and single row violates a unique index check, all of the rows are rejected.
• In order to insert multiple rows in one statement, separate the row values using a comma:

```sql
INSERT INTO EMP_ACT_COPY VALUES
, ('200000', 'DEF', 10, 1.4, '2003-10-22', '2003-11-24')
```

NULL and DEFAULT Keywords

• The NULL and DEFAULT keywords can be used for assigning these values to columns.
• It also is possible to reference special registers, such as the current date and current time:

```sql
INSERT INTO EMP_ACT_COPY VALUES
('400000', 'ABC', 10, NULL, DEFAULT, CURRENT_DATE);
```
Explicitly Listing Columns

• In order to leave some columns out of the INSERT statement, the columns to be included need to be explicitly listed.
• When this is done, the columns can be referred to in any order:

```
INSERT INTO EMP_ACT_COPY
(PROJNO, EMENDATE, ACTNO, EMPNO)
VALUES ('ABC', DATE(CURRENT TIMESTAMP), 123, '500000');
```

INSERT from SELECT - Example

• This code snippet inserts a set of rows which are the result of a query INSERT INTO emp_act_copy.

```
SELECT LTRIM(CHAR(ID + 600000))
  ,SUBSTR(UCASE(NAME),1,6)
  ,SALARY / 229
  ,123
  ,CURRENT DATE
  ,'2003-11-11'
FROM STAFF
WHERE ID < 50;
```
Modifying Data
DB2 Advanced SQL for JAVA Developers

Duplicates - Example

- Having unique indexes in a table will prevent poorly coded SQL statements from doubling the number of rows in the table.

```sql
INSERT INTO EMP_ACT_COPY
SELECT * FROM EMP_ACT_COPY;
```

INSERT Defaults

- In each case, for every row which has been inserted, a value must be provided for any column that does not have a default value.
- DEFAULT can be specified to have DB2 insert the default value for a column that meets one of the following conditions:
  - Is nullable.
  - Is defined with a default value.
  - Has data type ROWID.
    - ROWID columns always have default values.
  - Is an identity column.
    - Identity columns always have default values.
Inserting a Single Row

- The VALUES clause of the INSERT statement can be used for inserting a single row of column values into a table.
  - Either all of the columns can be named for which values are being provided or the list of column names can be omitted.
  - If the column name list is provided, values must be specified for all of the columns.

Inserting a Single Row

- For static INSERT statements, name all of the columns for which values are being provided.
- This is done for the following reasons:
  - The INSERT statement is independent of the table format.
    - It is not necessary to change the statement when a column is added to the table.
  - Verification can be performed on the order of the values.
  - Source statements are more self-descriptive.
Inserting from Another Table - Example

- Data can be copied from one table into another table.
- A fullselect can be used within an INSERT statement to select rows from one table for insertion into another table.

```
INSERT INTO TELE
SELECT LASTNAME, FIRSTNME, PHONENO
FROM EMP
WHERE WORKDEPT = 'D21';
```

Inserting Data into a ROWID Column

- A ROWID column is a column that is defined with a ROWID data type.
  - It is a requirement to have a column with a ROWID data type in a table that contains a LOB column.
- The ROWID column is stored in the base table and is used to look up the actual LOB data in the LOB table space.
- A ROWID column provides the capability for writing queries which navigate directly to a row in a table.
ROWID Column

- Prior to inserting data into a ROWID column, it is necessary to know how the ROWID column is defined.
- ROWID columns can be defined as GENERATED ALWAYS or GENERATED BY DEFAULT.
  - GENERATED ALWAYS will have DB2 generate a value for the column;
    - Data can not be inserted into that column.
  - GENERATED BY DEFAULT allows a value to be inserted into a table.
    - DB2 provides a default value if it has not been supplied.

ROWID Column - Example

- Consider an environment in which tables T1 and T2 have two columns:
  - integer column
  - ROWID column
- For this statement to run successfully, ROWIDCOL2 must be defined as GENERATED BY DEFAULT.

```
INSERT INTO T2 (INTCOL2,ROWIDCOL2)
SELECT * FROM T1;
```
DEFAULT Keyword in VALUES Clause

• The DEFAULT keyword in the VALUES clause of an INSERT statement is practical when inserting with dynamic SQL.
  – All the columns that the user does not provide a value for can be inserted using DEFAULT.
• If there has not been a specification in the table definition, the default value to be utilized are the values which have been defined on the table or the system default for each data type.
  – The column must be defined with DEFAULT.

DEFAULT Keyword in VALUES Clause - Example

```
INSERT INTO ITEMS
VALUES (440,
  'HAMMER',
  50,
  1.25,
  DEFAULT) ;
```
Expressions - List of Values: 
INSERT Statement

- Any expression can be specified in the list of values of an INSERT statement.
  - This can be useful with user defined functions, cast functions on user-defined distinct data types, values based on arithmetic, and new datetime functions.

Inserting with a Self-referencing SELECT

- Prior to DB2 V6, in order to INSERT rows into a table based on a selection of rows from that same table, it was a requirement to have had implemented views on that table.
  - This was necessary in order to provide DB2 with sufficient information about the two different tables being used in the statement.
  - This no longer is needed.
- The fullselect that is specified in the INSERT statement can now be a SELECT from the same table that returns more than a single row.
- This is a practical way for creating more rows based on the rows that already are in the table.
**FETCH FIRST n ROWS ONLY Clause**

- DB2 has a FETCH FIRST n ROWS ONLY clause, which provides the capability for specifying a limit on the number of rows returned into the result set.
  - This can be used in an INSERT statement to select rows.

```
SELECT T1.CREATOR, T1.NAME
   FROM SYSIBM.SYSTABLES T1
   WHERE T1.CREATOR = 'SYSIBM'
   AND T1.NAME LIKE 'SYS%'
ORDER BY T1.CREATOR, T1.NAME
FETCH FIRST 5 ROWS ONLY;
```

**UPDATE Statement**

- The UPDATE statement is used to change one or more columns and rows in a table, view, or fullselect.
- Each column that is to be updated has to specified.

```
UPDATE EMP_ACT_COPY
SET EMPTIME = NULL,
    EMENDATE = DEFAULT,
    EMSTDATE = CURRENT DATE + 2 DAYS,
    ACTNO = ACTNO / 2,
    PROJNO = 'ABC'
WHERE EMPNO = '100000';
```
**UPDATE Statement**

- Rows can be updated in a table, view, or fullselect.
  - If the object is not a table, then it must be updateable.
  - The correlation name is optional, and is only needed when there is an expression or predicate that references another table.
- The SET statement lists the columns to be updated and the new values that will be received.
- Predicates are optional.
  - If none are provided, all rows in the table are updated.
  - Typically all matching rows are updated.

**Mass UPDATE - Example**

- In order to update all rows in a table, all predicates need to be left off:

  ```sql
  UPDATE EMP_ACT_COPY
  SET ACTNO = ACTNO / 2;
  ```
Two Columns Get Same Value

• In this code snippet, both target columns get the same values.
  – This occurs because the result for both columns is calculated before the first column is updated.

```sql
UPDATE EMP_ACT_COPY AC1
SET ACTNO = ACTNO * 2,
    EMPTIME = ACTNO * 2
WHERE EMPNO LIKE '910%';
```

UPDATE Using SELECT

• It also is possible to have an update refer to the output of a SELECT statement as long as the result of the select is a single row.

```sql
UPDATE EMP_ACT_COPY
SET ACTNO = (SELECT MAX(SALARY) / 10
             FROM STAFF)
WHERE EMPNO = '200000';
```
Modifying Data
DB2 Advanced SQL for JAVA Developers

Multiple Column UPDATE: Using a Single SELECT

• This code updates multiple columns using a single select.

```
UPDATE EMP_ACT_COPY
SET (ACTNO,
    EMSTDATE,
    PROJNO) = (SELECT MAX(SALARY) / 10
                ,CURRENT DATE + 2 DAYS
                ,MIN(CHAR(ID))
            FROM STAFF
            WHERE ID != 33)
WHERE EMPNO LIKE '600%';
```

DELETE Statement

• The DELETE statement is used to remove rows from a table, view, or fullselect.
  • The set of rows deleted depends on the scope of the predicates used.
• This code snippet deletes a single row from the EMP_ACT sample table.

```
DELETE FROM emp_act_copy
    WHERE empno = '000010'
    AND projno = 'MA2100'
    AND actno = 10;
```
DELETE Statement - Predicates are Optional

- Predicates are optional.
  - If none are provided, all rows are deleted.
  - Typically all matching rows are deleted.

Correlated DELETE - Example 1

Example 1:

```
DELETE FROM STAFF S1
WHERE ID NOT IN
  (SELECT MAX(ID)
   FROM STAFF S2
   WHERE S1.DEPT = S2.DEPT);
```

- This code snippet deletes all the rows in the STAFF table; except those ID’s which have the highest ID in their respective department.
Correlated DELETE - Example 2

Example 2:

DELETE FROM STAFF S1
WHERE EXISTS
  (SELECT *
    FROM STAFF S2
    WHERE S2.DEPT = S1.DEPT
    AND S2.ID > S1.ID);

• This code snippet provides an alternative way for realizing the identical output.