Using Manipulating Data

Using Manipulating Data

INSERT

Syntax:

The below syntax can be followed if the values for all the columns in the table is definite and known.

```
INSERT INTO table
VALUES (column1 value, column2 value,  ...);
```

Example:

```
INSERT INTO employees
VALUES (130, 'KEMP','GARNER', 'kemp.garner@xxx.com', '48309290’,
TO_DATE ('01-JAN-2012'), 'SALES', 3800, 0, 110, 10);
```

The below syntax can be used if only few columns from the table have to be populated with a value. Rest of the columns can deduce their values either as NULL or from a different business logic.

```
INSERT INTO table (column1 name, column2 name, . . .)
VALUES (column1 value, column2 value, . . .);
```

Example:

```
INSERT INTO employees (EMPLOYEE_ID, FIRST_NAME, SALARY, DEPARTMENT_ID)
VALUES (130, 'KEMP', 3800, 10);
```

The below syntax can be used if only few columns from the table have to be populated with a value. Rest of the columns can deduce their values either as NULL or from a different business logic.

```
INSERT INTO table (column1 name, column2 name, . . .)
VALUES (column1 value, column2 value, . . .);
```

Example:

```
INSERT INTO employees (EMPLOYEE_ID, FIRST_NAME, SALARY, DEPARTMENT_ID)
VALUES (130, 'KEMP', 3800, 10);
```

INSERT-AS-SELECT (IAS) statement

Data can be populated into the target table from the source table using INSERT..AS..SELECT (IAS) operation. Its a direct path read operation.Its a simple way of creating copy of the data from one table to another or creating a backup copy of the table which the source table operations are online.

For example, data can be copied from EMPLOYEES table to EMP_HISTORY table.

```
INSERT INTO EMP_HISTORY
SELECT EMPLOYEE_ID, EMPLOYEE_NAME, SALARY, DEPARTMENT_ID
FROM employees;
```

Using Manipulating Data

UPDATE

UPDATE statement

The UPDATE command modifies the data stored in a column.It can update single or multiple rows at a time depending on the result set filtered by conditions specified in WHERE clause. Note that Updating columns is different from altering columns. Earlier in this chapter, you studied the ALTER command.The ALTER command changes the table structure, but leaves the table data unaffected.The UPDATE command changes data in the table, not the table structure.

Syntax:

```
UPDATE table
SET column = value [, column = value ...]
[WHERE condition]
```

Examples:

The UPDATE statement below updates the salaries of all the employees in the table.

```
UPDATE employees
SET salary = 5000;
```

The UPDATE statement below updates the salary of employee JOHN to 5000.

```
UPDATE employees
SET salary = 5000
WHERE UPPER (first_name) = 'JOHN';
```

Multiple columns can also be updated by specifying multiple columns in SET clause separated by a comma. For example, if both salary and job role has to be changed to 5000 and SALES respectively for JOHN, the UPDATE statement looks like,

```
UPDATE employees
SET     SALARY = 5000,
        JOB_ID = 'SALES'
WHERE UPPER (first_name) = 'JOHN';
```

Another way of updating multiple columns of the same row shows the usage of subquery.

```
UPDATE employees
SET (SALARY, JOB_ID) = (SELECT 5000, 'SALES' FROM DUAL)
WHERE UPPER (ENAME) = 'JOHN'
```

Using Manipulating Data

DELETE

DELETE statement

The DELETE command is one of the simplest of the SQL statements. It removes one or more rows from a table. Multiple table delete operations are not allowed in SQL.The syntax of the DELETE command is as below.

```
DELETE FROM table_name
   [WHERE condition];
```

Example:

The below DELETE statement would remove EDWIN's details from EMP table.

```
DELETE employees
WHERE UPPER (ENAME) = 'EDWIN'
```

DELETE statement

The DELETE command is one of the simplest of the SQL statements. It removes one or more rows from a table. Multiple table delete operations are not allowed in SQL.The syntax of the DELETE command is as below.

```
DELETE FROM table_name
   [WHERE condition];
```

Example:

The below DELETE statement would remove EDWIN's details from EMP table.

```
DELETE employees
WHERE UPPER (ENAME) = 'EDWIN’
```

Note: DELETE [TABLE NAME] and DELETE FROM [TABLE NAME] hold the same meaning.

Using Manipulating Data

TRUNCATE

TRUNCATE

Truncate is a DDL command which is used to flush out all records from a table but retaining the table structure. It does not supports WHERE condition to remove the selected records.

Syntax:

TRUNCATE [table name]

Resource : https://www.tutorialspoint.com/sql_certificate/manipulating_data.htm