(*Data Manipulation Operations*)

By: Ammara Yaseen

Data-Manipulation Language

Data manipulation language is a core part of SQL. When we want to add, update or delete data in the database, we execute a DML statement.

The SQL DML includes statements to perform following operations:-

Statement	Description
INSERT	Enter new rows into tables
UPDATE	To change existing rows
DELETE	To delete existing rows

Adding a new row to a table

We can add new rows to a table by using the INSERT statement. The syntax is

INSERT INTO table [(column [, column …])] VALUES (value [, value …]);

Examples

i. Inserting a new row in the dept table

INSERT INTO dept (deptno, dname, loc) VALUES (50, 'DEVELOPMENT', 'DETROIT');

Note: If the column list is not included, the values must be listed according to the default order of the columns in the table. The order can be seen using the DESCRIBE command.

ii. Inserting rows with Null values

- Implicit Method: Omit the column from the column list.
  INSERT INTO dept (deptno, dname) VALUES (60, 'MIS');
- Explicit Method: Specify the NULL keyword
  INSERT INTO dept VALUES (70, 'FINANCE', NULL);

Note: The oracle server automatically enforces all data types, data ranges and data integrity constraints. Any column that is not listed explicitly obtains a null value in the new row.

iii. Using special values, for example, SYSDATE function, to obtain data for a column when inserting a row in a table
INSERT INTO emp (empno, ename, job, mgr, hiredate, sal, comm, deptno) VALUES (7196, 'GREEN', 'SALESMAN', 7782, SYSDATE, 2000, NULL, 10);

iv. We can produce an INSERT statement that allows the user to add values interactively by using substitution variables.
INSERT INTO dept (deptno, dname, loc) VALUES (&department_id, '&department_name', '&location');

Changing data in a table

We can modify existing rows in a table with the UPDATE statement. The syntax is

UPDATE table SET column = value [, column = value, …] [WHERE condition];

As shown in the above syntax, we can update more than one row at a time depending on a condition.

Examples

i. To transfer an employee with number 7782 to department 20.
```
UPDATE emp
SET deptno = 20
WHERE empno = 7782;
```
ii. All rows in the table are modified if the WHERE clause is omitted.
```
UPDATE emp
SET deptno = 20;
```
iii. Updating with multiple column subquery: Update employee 7698's job and department to match that of employee 7499.
```
UPDATE emp
SET (job, deptno) =
                    (SELECT job, deptno FROM emp WHERE empno = 7499)
WHERE empno = 7698;
```

Removing a row from a table

We can remove existing rows from a table by using the DELETE statement. The syntax is

DELETE [FROM] table [WHERE condition];

Examples

i. Specific rows are deleted from a table by specifying the WHERE clause.
```
DELETE FROM department
 WHERE dname = 'DEVELOPMENT';
```
ii. All rows in the table are deleted if we omit the WHERE clause.
```
DELETE FROM department;
```
iii. Remove all employees who started after January 1, 1997.
```
DELETE FROM employee
WHERE hiredate > TO_DATE('01.01.97', 'DD.MM.YY');
```
iv. Deleting rows based on another table by using subqueries in DELETE statements.
```
DELETE from employee
WHERE deptno =
                    (SELECT deptno FROM dept WHERE dname = 'SALES');
```
v. Delete record of employees in department 30
```
DELETE FROM employee WHERE DEPTNO = 30;
```

EXERCISES

1. Write a transaction to insert following rows in EMP table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7123	RALPH	DESIGNER	7566	21-APR-85	2300		50
7890	GEORGE	CLERK	7566	03-MAY-85	1235		50
7629	BOB	SALESMAN	7698	06-MAR-86	1800	1000	30

2. Write down SQL statements to perform following functions:-
   i. Increase the salary by 250 of all clerks with a salary less than 900
   ii. Transfer the employee with number 7890 to department 20 and increase his salary by 15%.
   iii. Remove all employees who were hired before 1981.