M26CDE

Database Systems

Data Manipulation Language

Introduction to Data Manipulation Language (DML)

- Insert statement
- Select statement
- Update statement
- Delete statement
- Use of the DUAL table

SQL – INSERT INTO

To populate a table with INSERT INTO statement

- Generic form

  INSERT INTO *table_name*
  VALUES (*value_1*, *value_2*, …, *value_n*);

  (where values MUST be entered in the order in which the columns are created in the table.)

- The second form specifies both the column names and the values to be inserted:

  INSERT INTO *table_name (column1, column2, ...)*
  VALUES *(value1, value2, value3,...)*

SQL – INSERT (1)

- Character string or number values
- **Note:**
  1. To insert a record that contains a character string value to a table, the value MUST be surrounded with a single quotation mark, ‘’. For example, ‘Adam’ or ‘London’. These are usually defined as VARCHAR2(n).
  2. To insert a numeric value, the single quotation mark is NOT required. For example, a numeric value 20 (defined as number) is entered as 20
  - **Example:** Student (s_id, s_name, loan)

    INSERT INTO Student VALUES (‘000001’, ‘John Smith’, 3000);

    or

    INSERT INTO Student (s_id, s_name, loan)
    VALUES (‘000001’, ‘John Smith’, 3000);

SQL – INSERT (2)

Storing Date Values

Example: Journey (dept_time)

dept_time has type DATE

To insert date values with a default format.

INSERT INTO Journey VALUES (‘01-FEB-04’);

Note: Oracle converts months to uppercase letters. Thus, the above date will be displayed as 01-FEB-04.

To insert date values with a specific date format:

```
INSERT INTO Journey
        VALUES (TO_DATE('01-10-02', 'dd-mm-yy')) ;

INSERT INTO Journey
        VALUES (TO_DATE(‘01/10/2002’, ‘DD/MM/YYYY’)) ;
```

TO_DATE converts a character string into a date type.

SQL – INSERT (3)

Storing dates with time values

Example: Journey (dept_time, arr_time)

1. INSERT INTO Journey VALUES (
   TO_DATE('2004/01/15:07:25:00AM', 'yyyy/mm/dd:hh:mi:ssam'),
   TO_DATE('2004-01-15:05:10:00PM', 'yyyy-mm-dd hh:mi:sspm'));

2. INSERT INTO Journey VALUES (
   TO_DATE('2004/01/15:07:25:00', 'yyyy/mm/dd:hh24:mi:ss'),
   TO_DATE('2004/01/15:20:40:00', 'yyyy/mm/dd:hh24:mi:ss'));

3. INSERT INTO Journey VALUES (
   TO_DATE('05:54', 'HH24:MI'), TO_DATE('16:30', 'HH24:MI'));

DATE values can be compared (<, >, =). It is also possible to take the lowest or highest date value from a list of date values (MIN, MAX aggrggate functions).

UPDATE statement

The UPDATE statement allows you to update a single record or multiple records in a table.

– Update the loan of all students in the Student table:

```
UPDATE Student
SET loan = loan + 3000;
```

- Update the load of one student with a specific s_id:

```
UPDATE Student
SET loan = 6000
WHERE s_id = ‘000002’;
```

SQL – SELECT (1)

- **Syntax of the Select statement**

  SELECT *column_list* FROM *table-name*
  [WHERE Clause]
  [GROUP BY clause]
  [HAVING clause]
  [ORDER BY clause];

- To select and display all data (to select all the rows of all columns).

  SELECT *
  FROM *table_name*

  Example: Consider Student (s_id, s_name, date_of_birth, loan)

  List the data of all students:

  SELECT *
  FROM Student;

SQL – SELECT (2)

- To select the rows of specified columns
  - **Syntax**:

    SELECT *column_list*
    FROM *table_name*

  - **Example**: Student (s_id, s_name, date_of_birth, loan)
    List the id and names of all students.

    SELECT s_id, s_name
    FROM Student;

SQL – SELECT (3)

- To remove duplicate rows
  - **Syntax**:

    SELECT **DISTINCT** *column_name*

    FROM *table_name*
  - **Example**: Student (s_id, s_name, m_id).

    List the modules taken by the students without duplicates.

    SELECT DISTINCT m_id FROM Student;

    Result: M44IS

    M26CDE

    M64COM

Without DISTINCT we could have the following list:

M44IS, M26CDE, M64COM, M26CDE, M44IS.

SQL – SELECT (4)

Select statement with the WHERE clause.

- Select one specific student:

```
SELECT s_name FROM student
WHERE s_id = '0004';
```

- Select students according to some condition:

```
SELECT s_id, s_name FROM student
WHERE loan > 4000;
```

SQL – SELECT (5)

To display information of dates or time.

- TO_CHAR converts date type to character string.

  SELECT TO_CHAR(*column_name*, *specified_format*)
  FROM *table_name*

  – **Example**: Journey (j_id, dept_time, arr_time)
  Assume that a value is stored in dept_time as
  15-MAR-02:14:01. The following displays the default values.

  SELECT dept_time
  FROM Journey;

Comparison of dates:

TO_DATE('12-JUN-60', 'dd-mm-yy') *is less than (<)*
TO_DATE('10-FEB-80', 'dd/mm/yy')

SQL – SELECT (6)

- **To display date and time in a specified format:**

1. SELECT TO_CHAR(dept_time, ‘DD/MM/YY’) FROM Journey
   Result: 15/03/02

2. SELECT TO_CHAR(dept_time, ‘DY DD/MM/YY’) FROM Journey
   Result: FRI 15/03/02
   (Note: the day of week will be given by Oracle.)

4. SELECT TO_CHAR(dept_time, ‘YYYY’) FROM Journey
   Result: 2002

5. SELECT TO_CHAR(dept_time, ‘YY’) FROM Journey
   Result: 02

6. SELECT TO_CHAR(dept_time, ‘HH:MI’) FROM Journey
   Result: ?

SQL - INSERT and SELECT

Inserting data in a table through a select statement.

```
INSERT INTO table_name
[(column1, column2, ... columnN)]
    SELECT column1, column2, ...columnN
    FROM table_name [WHERE condition];
```

Example: Consider the table employee with following schema:

employee (id, name, dept, age, salary, location)

It was decided to create a table empDept with the following schema:

empDept(emp_id, emp_name, emp_dept)

It was also decided to populate it with the data of the employee table. Only the *id*, *name* and *dept* attributes are required. This can achieved by extracting the values from the employee table:

```
INSERT INTO empDept (emp_id, emp_name, emp_dept)
SELECT  id, name, dept FROM employee;
```

SQL - DELETE

To delete data from a table:

DELETE

FROM <*table_name*> WHERE <*condition_expression*>

Example: Consider Student(s_id, s_name).

- Delete the record of a specific student whose id is ‘000005’.

  DELETE

  FROM Student WHERE s_id = ‘000005’;

- To delete all the data in a table:

  DELETE

  FROM Student;

This can also be achieved by the TRUNCATE statement:

TRUNCATE TABLE student ;

TRUNCATE deletes all the rows from the table and frees the space containing the table.

- This contrasts with DROP TABLE which removes the table from the database.
- DROP Table is a DDL statement. DELETE and TRUNCATE TABLE are DML statements.

SQL - Scalar and Vector values

- **There are two types of values that can be returned by a SELECT statement:**
- **Scalar (single value):**

  SELECT s_name FROM Student

  WHERE s_id = ‘005’;

  Returns one single value. (use = operatorfor comparison)

- **Vector (multiple values):**

  SELECT s_name FROM Student;

  Returns many values. (use IN operator for comparison)

SQL - DUAL Table

- The DUAL table is a utility one-row, one-column table present by default in all Oracle database installations. It is a pseudo table which is often used for calculations or checking system variables such as SYSDATE or USER.

- DESC DUAL;
- SELECT * FROM DUAL;
- SELECT 3+1 FROM DUAL;
- SELECT 1 FROM DUAL;
- SELECT user FROM DUAL;
- SELECT sysdate FROM DUAL;
- SELECT to_char(sysdate, ‘dd/mm’) FROM DUAL;
- SELECT to_char(sysdate, ‘yyyy’) FROM DUAL;
- SELECT to_char(sysdate, ‘hh24:mi’) FROM DUAL;
- SELECT to_char(sysdate, ‘year’) FROM DUAL;
- SELECT to_char(sysdate, ‘month’) FROM DUAL;
- SELECT to_char(sysdate, ‘day’) FROM DUAL;
- SELECT to_char(sysdate, ‘hh24’) FROM DUAL;