UNIT III

Working with Tables: Data Management and Retrieval:

DML – Adding a New Row/Record – Customized Prompts – Updating and Deleting Existing Rows/Records–Retrieving Data from a Table–Arithmetic Operations– Restricting Data with WHERE clause–Sorting– Revisiting Substitution Variables–DEFINE Command–CASE Structure. Functions and Grouping: Built-in Functions – Grouping Data. Multiple Tables: Joins and Set Operations: Join–Set Operators

Text Book: "DATABASE SYSTEMS USING ORACLE"-NILESH SHAH,2nd Edition, PHI

Prepared by Dr.P.Radha

Working with Tables: Data Management and Retrieval:

DML – Adding a New Row/Record – Customized Prompts – Updating and Deleting Existing Rows/Records–Retrieving Data from a Table– Arithmetic Operations– Restricting Data with WHERE clause– Sorting– Revisiting Substitution Variables–DEFINE Command– CASE Structure

DATA MANIPULATION LANGUAGE(DML)

- SQL’s Data Manipulation Language (DML) consists of three statements—INSERT, UPDATE, and DELETE.
- Data retrieval language, also known as a subset of DML, consists of the SELECT statement and its clauses
- A new row is added to a table with the INSERT statement.
- Data in existing rows are changed with the UPDATE statement.
- The DELETE statement removes rows from a table.
- The SELECT statement retrieves data from tables, but it does not affect the data in any way. In other words, the SELECT statement does not manipulate data; it only queries tables.

DATA MANIPULATION LANGUAGE(DML)

- The DML statements are not written perma- nently to the database unless they are committed.
- Many times, students do not exit properly from SQL* Plus, and they end up losing newly inserted rows or updated in- formation. Y
- You can enter a COMMIT statement anytime to write DML statements to the disk.
- You can use ROLLBACK to undo the last set of DML statements

ADDING A NEW ROW/RECORD

- The Data Manipulation Language (DML) statement INSERT is used to insert a new row/record into a table.
- A user can insert values for all columns or a selected list of columns in a record. The general syntax for the INSERT statement is

INSERT INTO tablename [(column1, column2, column3, . . .)] VALUES (value1, value2, value3, . . .);

Cont…

- For example, let us add a new record to the STUDENT table in the Indo–US (IU) College database:
- INSERT INTO student (StudentId, Last, First, Street, City, State, Zip, StartTerm, BirthDate, FacultyId, MajorId, Phone)VALUES (‘00100’, ‘Diaz’, ‘Jose’, ‘1 Ford Avenue #7’, ‘Hill’, ‘NJ’, ‘08863’, ‘WN03’, ‘12-FEB-80’,123, 100, ‘9735551111’);

ROUNDING BY INSERT

- If you insert value 543.876 in a NUMBER(6,2) column, the precision is 4, and the scale is 2. The resulting value will be 543.88, rounded to two decimal places, or a scale of 2.
- The rounded value will be entered into the column.

ENTERING NULL VALUES

- Null values are allowed in non–primary key columns that do not have a NOT NULL constraint

There are two methods for inserting a NULL value in a column:

- ***Implicit method:*** In the implicit method, the column’s name is omitted from the column list in an INSERT statement. For example,

INSERT INTO dept(DeptId, DeptName) VALUES(50, ‘Production’);

- ***Explicit method:*** In the explicit method, the null value is used as a value for a numeric column, and an empty string (‘’) is used for date or charac-ter columns. For example,

INSERT INTO dept(DeptId, DeptName, Location, EmployeeId) VALUES(60, ‘Personnel’, ‘Chicago’, NULL);

Entering Default Values

- With Oracle9i, the INSERT statement has added syntax that lets you insert default values with the key word DEFAULT in place of a value for a column.
- If a default value is assigned to the column during the table’s creation, that default value is inserted into the column.

SUBSTITUTION VARIABLES

- Inserting rows into a table is a very tedious task. The SQL language does have substitution variables, which enable you to create an interactive SQL script.
- The ampersand (&) character is used before the substitution variable in the query.
- The substitution variables for CHAR- and DATE-type columns are enclosed within a pair of single quotation marks

Cont..

```
SQL> INSERT INTO dept(DeptId, DeptName, Location, EmployeeId)
        2 VALUES(&dept_id, ’&dept_name’, ’&location’,
        &emp_id);
        Enter value for dept_id: 70
        Enter value for dept_name:
        Testing Enter value for location: Miami Enter value for emp_id:
        NULL
        old2: VALUES(&dept_id, ’&dept_name’, ’&location’,
       &emp_id)
      new 2: VALUES(70, ’Testing’, ’Miami’, NULL)
      1 row created.
 SQL>
```

CUSTOMIZED PROMPTS

- The substitution-variable prompts are standard. Oracle displays “Enter the value for” followed by the name of the substitution variable.
- The SQL* Plus command ACCEPT is used for customized prompts.
- The ACCEPT command does not use an ampersand in front of the variable name.
- If an ACCEPT statement is used for a variable, the value of that variable, once entered, is remembered during the session

Cont…

- The general syntax is

ACCEPT variablename PROMPT ‘prompt message’

```
SQL> ACCEPT dept_id PROMPT ’Please enter department number(10 to 99): ’
Please enter department number(10 to 99): 80
SQL> ACCEPT dept_name PROMPT ’Please enter department name(no nulls): ’
Please enter department name(no nulls): Accounting
SQL> ACCEPT location PROMPT ’Please enter location city: ’
Please enter location city: Monroe
SQL> ACCEPT manager PROMPT ’Please enter EmployeeId of Manager: ’
Please enter EmployeeId of Manager: NULL
SQL> INSERT INTO dept
2 VALUES(&dept_id, ’&dept_name’, ’&location’, &manager);
old 2: VALUES(&dept_id, ’&dept_name’, ’&location’, &manager)
new 2: VALUES(80, ’Accounting’, ’Monroe’, NULL)
1 row created.
SQL>
```

UPDATING EXISTING ROWS/RECORDS

- Once data are added to the tables for various entities, they may not stay the same forever.
- In SQL, the UPDATE statement is used for such modifications to data.
- Only one table can be updated at a time, but it is possible to change more than one column at a time.

The general syntax is

UPDATE tablename

SET column1 = newvalue [, column2 = newvalue]

[WHERE condition(s)];

RELATIONAL OPERATORS.

Relational Operator	Meaning
=	Equal to
<> or !=	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

Cont..

There are other operators for writing conditions like AND, OR, BETWEEN, AND, IN, and LIKE

Cont..

```
SQL> UPDATE student
SET MajorId = 700
WHERE StudentId = ’00103’;
UPDATE student
*
ERROR at line 1:
ORA-02291: integrity constraint (SYSTEM.STUDENT_MAJORID_FK) violated - parent key not found
SQL> 2
2* SET MajorId = 700
SQL> c/700/600
2* SET MajorId = 600
SQL> 1
UPDATE student
SET MajorId = 600
3* WHERE StudentId = ’00103’
SQL> /
1 row updated.
SQL> 2
2* SET MajorId = 600
SQL> c/600/500
2* SET MajorId = 500
SQL> /
1 row updated.
SQL>
```

DELETING EXISTING ROWS/RECORDS

- Deletion is another data maintenance operation
- In Oracle, the SQL statement DELETE is used for deleting un- wanted rows. Its general syntax is

DELETE [FROM] tablename

[WHERE condition(s)];

Cont..

- The DELETE statement without a condition will result in a table with no rows.
- A DELETE statement without a WHERE clause has same effect as a TRUNCATE statement

```
SQL> DELETE FROM dept
2 WHERE DeptId = 70;
1 row deleted.
SQL>
```

RETRIEVING DATA FROM A TABLE

- The main purpose of the SQL language is for querying the database
- The most important statement or query is the SELECT query.
- A user retrieves data from the underlying table or tables with a SELECT query.
- The output can be sorted and grouped, and information can be derived with the use of mathematical expressions and built-in functions.

Cont..

The general syntax is

SELECT columnlist

FROM tablename;

- The columns can be listed in any order.
- They do not have to be in the order given by the DESCRIBE command

SELECT (*)

- If you want to see all columns in a table, you do not have to list them all
- You can use an asterisk 1*2 in place of the column list, and all columns will be displayed in the same order as the underlying table structure

DISTINCT FUNCTION

- The DISTINCT function is used to suppress duplicate values.
- The word *DISTINCT* is used right after the keyword SELECT and before the column name

```
SQL> SELECT DISTINCT Building
2 FROM location;
BUILDING
- - - - - - -
Gandhi
Kennedy
Nehru
SQL>
```

Cont..

Column Alias

- When a SELECT query is executed, SQL* Plus uses the column’s name as the column heading.
- Normally, the user gives abbreviated names for columns, and they are not very descriptive
- Column aliases are useful, because they let you change the column’s heading

Cont..

- The column alias is written right after the column name with the optional keyword AS in between.
- The alias heading appears in uppercase by default. If an alias includes spaces or special characters or if you want to preserve its case, you must enclose it in double quotation marks (“ ”). The general syntax is

SELECT columnname [AS] alias . . .

Cont..

COLUMN Command

- SQL * Plus’ COLUMN command allows you to specify format columns for columns.
- The general syntax of the COLUMN command is

COLUMN columnname FORMAT format type

For example,

COLUMN State FORMAT A5

COLUMN Salary FORMAT $999,999

Cont…

Concatenation

- Concatenation means joining or linking
- In SQL, concatenation joins a column or a character string to another column.
- The result is a column that is a string or a sequence of characters.
- Two vertical bars or pipe symbols are used as the concatenation operator. The symbol appears on your keyboard with the back- slash (\) character.

ARITHMETIC OPERATIONS

- The arithmetic expressions are used to display mathematically calculated data.
- These expressions use columns, numeric values, and arithmetic operators
- When arithmetic operators are used with columns in the SELECT query, the under-lying data are not changed.
- The calculations are for output purposes only.

Cont..

Operator	Use
+	Addition
-	Subtraction
*	Multiplication
/	Division

Cont..

Order of Operation

The order of operation is as follows:

- Whatever is in parentheses is done first.
- Multiplication and division have higher precedence than addition and subtraction.
- If more than one operator of the same precedence is present, the operators are performed from left to right.

RESTRICTING DATA WITH A WHERE CLAUSE

- A WHERE clause is used with the SELECT query to restrict the rows that are picked.
- It is the implementation of a *selection* operation.
- The WHERE clause uses a simple condition or a compound condition.
- The rows that satisfy the supplied conditions are displayed in the output.
- The syntax of SELECT changes a little with an added WHERE clause.

Cont…

The general syntax of the WHERE clause is

SELECT columnlist

FROM tablename

[WHERE condition(s)];

Wild Cards

- There are times, however, when you do not know the exact string value.
- In these cases, you can select rows that match a pattern of characters. Such a search is known as a *wild-card search*. There are two wild cards for a pattern search.

Wild Card	Use
%	Represents zero or more characters
_(Underscore)	Represents any one character

SORTING

- The ORDER BY clause is used with the SELECT query to sort rows in a table.
- The rows can be sorted in ascending or descending order.
- The rows can also be sorted based on one or more columns.
- The expanded syntax of SELECT given here uses an ORDER BY clause, which is always used last in the statement.

The general syntax is

SELECT columnlist FROM tablename
[WHERE condition(s)]
[ORDER BY column|expression [ASC|DESC]];

In the syntax, ASC stands for ascending order

The default order is ascending, so there is no need to type ASC for ascending order.

The keyword DESC stands for descending or reverse order.

REVISITING SUBSTITUTION VARIABLES

- The substitution variables can be used in statements other than the INSERT statement.
- They can substitute for column names, table names, expressions, or text.
- Their use is to generalize queries by inserting them as follows:
- In the SELECT statement in place of a column name.
- In the FROM clause in place of a table name.
- In the WHERE clause as a column expression or text.
- As an entire SELECT statement.

DEFINE COMMAND

- A variable can be defined at the SQL> prompt. The variable is assigned a value that is held until the user exits from SQL* Plus or undefines it.

The general syntax is

DEFINE variable [= value]

For example,

DEFINE Last = Shaw

CASE STRUCTURE

- CASE structure is allowed anywhere expressions are allowed in SQL statements. SQL’s CASE structure is similar to the SELECT… CASE statement in Visual Basic language and the switch … case statement in C++ and Java.

The general syntax of CASE is

CASE WHEN condition1 THEN

Expression1

WHEN condition2 THEN

Expression2

. . .

[ELSE Expression]

END

WORKING WITH TABLES: FUNCTIONS AND GROUPING

- Built-in Functions
- Grouping Data
- Multiple Tables: Joins and Set operators

BUILT-IN FUNCTIONS

- The built-in functions provide a powerful tool for the enhancement of a basic query.
- A function takes zero or more arguments and returns a single value
- In Oracle’s SQL, there are two types of functions:
- ***Single-row functions,*** which work on columns from each row and return one result per row.
- ***Group functions*** or ***aggregate functions***, which manipulate data in a group of rows and return single result.

Cont…

Single-Row Functions

- The single-row functions take different types of arguments, work on a data item from each row, and return one value for each row.
- The arguments are in the form of a constant value, variable name, column, and/or expression.
- The value returned by a function may be of a different type than the argument(s) supplied.

Cont…

The general syntax is

Function(column **|** ***expression [, argument1, argument2,])***

where *function* is the name of the function, *column* is a column from a table, *expression* is a character string or a mathematical expression, and *argument* is any argument used by the function.

SINGLE-ROW FUNCTIONS

There are various types of single-row functions:

- **Character functions** take a character string or character-type column as an argument and return a character or numeric value.
- **Number functions** take a number or number-type column as an argument and return a numeric value.
- **Date functions** take a date value or date-type column as an argument and return date-type data. (Exception: The MONTHS_BETWEEN function re- turns a numeric value.)
- **Conversion functions** convert value from one data type to another.
- **General functions** perform different tasks.

Cont..

Group Functions

- The group functions perform an operation on a group of rows and return one result. Look at the EMPLOYEE and STUDENT tables:
- Who makes the lowest salary?
- Who got the maximum commission?
- What is the company’s total payroll?
- How many students started in the Winter 2003 semester?

GROUPING DATA

- The rows in a table can be divided into different groups to treat each group sepa-rately.
- The group functions can be applied to individual groups in the same fashion they are applied to all rows. The GROUP BY clause is used for grouping data

Cont..

The general syntax is

```
SELECT column, groupfunction(column)
FROM tablename [WHERE condition(s)]
[GROUP BY column | expression]
[ORDER BY column | expression [ASC |
    DESC]];
```

Cont..

- When you include a group function and the GROUP BY clause in your query, the individual column(s) appearing in SELECT must also appear in GROUP BY.
- The WHERE clause can still be used to restrict data before grouping.
- The WHERE clause cannot be used to restrict groups.
- A column alias cannot be used in a GROUP BY clause.
- The GROUP BY column does not have to appear in a SELECT query

- When a column is used in the GROUP BY clause, the result is sorted in ascending order by that column by default.
- In other words, GROUP BY has an implied ORDER BY. You can still use an ORDER BY clause explicitly to change the implied sort order.
- In Oracle9i, the order of the WHERE and GROUP BY clauses in the SELECT query does not matter, but traditionally, the WHERE clause is written before the GROUP BY clause.

Cont..

- **HAVING Clause**

  The HAVING clause can restrict groups. The WHERE clause restricts rows, the GROUP BY clause groups remaining rows, the Group function works on each group, and the HAVING clause keeps the groups that match the group condition.

NESTING GROUP FUNCTIONS

- The single-row functions can be nested to many levels, but the group functions can only be nested to two levels.

For example,

SELECT SUM(MaxCount) FROM crssection GROUP BY CourseId;

JOIN–SET OPERATORS

JOIN

When the required data are in more than one table, related tables are joined using a join condition. The join condition combines a row in one table with a row in another table based on the same values in the common columns.

Cont…

Cartesian Product

- A Cartesian product results from a multitable query that does not have a WHERE clause. The product operation joins each row in the first table with each row in the second table. The product normally results in an output with a large number of rows and is not very useful.

Cont..

There are four types of joins in Oracle:

- Equijoin.
- Nonequijoin.
- Outer join.
- Self-join.

Cont..

Equijoin

- The equijoin is a join with a join condition involving common columns from two tables.

The syntax is

SELECT *columnnames*

FROM *tablenames*

WHERE *join condition(s);*

Cont…

Nonequijoin

The join condition for these tables can be written using any operator other than the = operator. That is why it is called non-equijoin.

Cont..

Outer Join

Some of the faculty members are not any student’s advisor, so they did not get selected. Sup- pose you also want to see all those faculty advisor names. Then, you would change your query’s join condition and create a join known as an outer join.

Cont..

The general syntax is

SELECT tablename1.columnname, tablename2.columnname

FROM tablename1, tablename2

WHERE tablename1.columnname (+) = tablename2.columnname;

Cont…

Self-Join

- A self-join is joining a table to itself.
- A self-join is one join that is not so easy to understand.
- When a table is joined to itself, two copies of the same table are loaded or used. They are treated like any two different tables, and a join is produced from those two copies

SET OPERATORS

- The implementation of these operations is through the use of set operators.
- The union compatibility is achieved or the set operations are performed on results from two independent queries.
- The output from both queries must return the same number of columns, and respective columns must have a similar domain.

Cont..

The general syntax for any set operation is

SELECT-*Query1*

Set operator

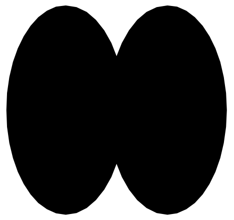
SELECT-*Query2;*

Cont..

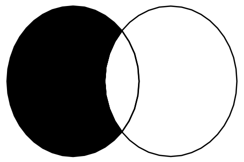
Set Operator	**Use**
Union	It returns all rows from both queries, but duplicate rows are not repeated
Union All	It returns all rows from both queries, and it displays all duplicate rows
Intersect	It returns all rows that appear in both queries' results.
Minus	It returns rows that are returned by the first query minus rows that are returned by the second query.

SET OPERATORS

- **Union**



- **Minus**



- **Intersect**