## **Chapter -09**

**USING SET OPERATORS**

### **Set Operators**

* **UNION -** Rows from both queries after eliminating duplications
* **UNION ALL -** Rows from both queries, including all duplications
* **INTERSECT -** Rows that are common to both queries
* **MINUS -** Rows in the first query that are not present in the second query

### **Rules of Set Operators**

• The expressions in the SELECT lists must match in number.

• The data type of each column in the subsequent query must match the data type of its corresponding column in the first query.

• Parentheses can be used to alter the sequence of execution.

• ORDER BY clause can appear only at the very end of the statement.

### **Matching SELECT Statements**

You must match the data type (using the TO\_CHAR function or any other conversion functions) when columns do not exist in one or the other table.

### **Using the ORDER BY clause in set operations**

The ORDER BY clause can be used only once in a compound query. If used, the ORDER BY clause must be placed at the end of the query. The ORDER BY clause accepts the column name or an alias. By default, the output is sorted in ascending order in the first column of the first SELECT query.

## Chapter-10

Managing Tables Using DML Statements

### Data Manipulation Language

* + Transaction - A collection of DML statements

that form a logical unit of work.

### Inserting New Rows

* + This statement with the VALUES clause adds only one row at a time to a table.
  + Enclose character and date values within single quotation

Marks.

### Inserting Specific Date and Time Values

* + Supply the date value in the DD-MON-YYYY format. This is recommended.
  + Because it clearly specifies the century and does not depend on the internal RR format logic of specifying the correct century.
  + If a date must be entered in a format other than the default format (for example, with another century or a specific time), you must use the TO\_DATE function.

### Creating a Script

* + Use the & substitution in a SQL statement to prompt for values.
  + & is a placeholder for the variable value.

### Copying Rows from Another Table

* + In place of the VALUES clause, you use a subquery.
  + The *number of columns* and their *data types* in the column list of the INSERT clause must match the number of values and their data types in the subquery.

### UPDATE Statement Syntax

* + Update more than one row at a time (if required).
  + In general, use the primary key column in the WHERE clause to identify a single row for update.
  + If you omit the WHERE clause, values for all the rows in the table are modified.

### Updating Two Columns with a Subquery

* + Can update multiple columns in the SET clause of an UPDATE statement by writing multiple subqueries.

### Deleting Rows from a Table

* + Specific rows are deleted if you specify the WHERE clause.
  + All rows in the table are deleted if you omit the WHERE clause.
  + Use the subqueries in the DELETE statements to remove rows from a table based on values from another table.

### TRUNCATE Statement

* + Removes all rows from a table, leaving the table empty and the table structure intact
  + The TRUNCATE statement is a data definition language (DDL) statement and generates no rollback information.
  + If the table is the parent of a referential integrity constraint, you cannot truncate the table.

### Database Transactions

* + Transaction Types
    - Data manipulation language (DML)
      * Consists of any number of DML statements that the Oracle server treats as a single entity or a logical unit of work
    - Data definition language (DDL)
      * Consists of only one DDL statement
    - Data control language (DCL)
      * Consists of only one DCL statement
  + Transaction Begin when the first DML SQL statement is executed.
  + Transaction End with one of the following events:
    - A COMMIT or ROLLBACK statement is issued.
    - A DDL statement, such as CREATE, is issued.
    - A DCL statement is issued.
    - The user exits SQL Developer or SQL\*Plus.
    - A machine fails or the system crashes.
  + A DDL statement or a DCL statement is automatically committed and, therefore, implicitly ends a transaction.

### Explicit Transaction Control Statements

* + COMMIT ends the current transaction by making all pending data changes permanent.
  + SAVEPOINT name marks a savepoint within the current transaction.
  + ROLLBACK ends the current transaction by discarding all pending data changes.
  + ROLLBACK TO <savepoint> rolls back the current transaction to the specified savepoint, thereby discarding any changes and/or savepoints that were created after the savepoint to which you are rolling back.
  + Because savepoints are logical, there is no way to list the savepoints that you have created.
  + You cannot COMMIT to a SAVEPOINT. SAVEPOINT is not ANSI-standard SQL.

### Implicit Transaction Processing

* + Automatic commit: DDL statement or DCL statement issued

SQL Developer or SQL\*Plus exited normally, without explicitly issuing COMMIT or ROLLBACK commands

* + Automatic rollback: Abnormal termination of SQL Developer or SQL\*Plus or system failure

### State of Data

* + **Before COMMIT or ROLLBACK**
    - Every data change made during the transaction is temporary until the transaction is committed.
    - The current session can review the results of the data manipulation operations by querying the tables.
    - The affected rows are locked; other session cannot change the data in the affected rows.
  + **After COMMIT**
    - All sessions can view the results of the transaction.
    - The locks on the affected rows are released; the rows are now available for other sessions to perform new data changes.
    - All savepoints are erased.
  + **After ROLLBACK**
    - The previous state of the data is restored.
    - Locks on the affected rows are released.
  + Terminate your transactions explicitly by executing a COMMIT or ROLLBACK statement.

### Read Consistency

* + Read consistency ensures that, on the same data:
    - Readers do not wait for writers
    - Writers do not wait for readers
    - Writers wait for writers
  + Need read consistency so that the following occur:
    - The database reader and writer are ensured a consistent view of the data.
    - Readers do not view data that is in the process of being changed.
    - Writers are ensured that the changes to the database are done in a consistent manner.
    - Changes made by one writer do not disrupt or conflict with the changes being made by another writer.
  + FOR UPDATE clause in SELECT : Locks rows identified by the SELECT query.
  + Lock is released only when you issue a ROLLBACK or a COMMIT.
  + Oracle offers the FOR UPDATE clause of the SELECT statement to lock a set of records even before the change is done to them in the program.