**N.B.** The UNION ALL operator does not eliminate duplicate rows. UNION returns all distinct rows selected by either query. UNION ALL returns all rows selected by either query, including all duplicates.

Example: *SELECT employee\_id, job\_id,department\_id FROM employees UNION SELECT employee\_id, job\_id,department\_id FROM job\_history ORDER BY employee\_id;*

**Intersect Operator:** The INTERSECT operator returns rows that are common to both queries. [*the INTERSECT operator to return all rows that are common to multiple queries.*]

Example: SELECT employee\_id, job\_id FROM employees INTERSECT SELECT employee\_id, job\_id FROM job\_history;

{*Display the employee IDs and job IDs of those employees who currently have a job title that is the same as a previous job title.}*

**Guidelines**

• The number of columns and the data types of the columns being selected by the SELECT

statements in the queries must be identical in all the SELECT statements used in the query.

The names of the columns need not be identical.

• Reversing the order of the intersected tables does not alter the result.

• INTERSECT does not ignore NULL values.

**Minus Operator:** The MINUS operator returns rows in the first query

that are not present in the second query.

For Example: SELECT employee\_id FROM employees MINUS SELECT employee\_id FROM job\_history;

**Set Operator Guidelines**

• The expressions in the SELECT lists must match in number and data type.

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

• The ORDER BY clause:

– Can appear only at the very end of the statement

– Will accept the column name, aliases from the first SELECT statement, or the positional notation.

**Oracle Server and Set Operators**

• Duplicate rows are automatically eliminated except in UNION ALL.

• Column names from the first query appear in the result.

• The output is sorted in ascending order by default except in UNION ALL.

**Chapter-8**

**What is DML statement? When does it execute?**

Data manipulation language (DML) is a core part of SQL. When we want to add, update, or delete data in the database, we execute a DML statement. A collection of DML statements that form a logical unit of work is called a *transaction*.

**How to add a new row in a table?**

We can add new rows to a table by using the INSERT statement. *INSERT INTO table [(column [, column...])] VALUES (value [, value...]);*

With this syntax, only one row is inserted at a time.

Or

• Insert a new row containing values for each column.

• List values in the default order of the columns in the table.

• Optionally, list the columns in the INSERT clause.

• Enclose character and date values in single quotation marks.

Example: *INSERT INTO departments(department\_id, department\_name, manager\_id, location\_id) VALUES (70, 'Public Relations', 100, 1700);*

**Method for inserting Null values:**

**Implicit** Omit the column from the column list.

**Explicit** Specify the NULL keyword in the VALUES list; specify the empty string ('') in the VALUES list for character strings and dates.

**What are the common errors that occur during user input?**

Common errors that can occur during user input:

• Mandatory value missing for a NOT NULL column

• Duplicate value violates uniqueness constraint

• Foreign key constraint violated

• CHECK constraint violated

• Data type mismatch

• Value too wide to fit in column

**How to update a table?**

* Modify existing rows with the UPDATE statement.
* Update more than one row at a time (if required).
* Specific row or rows are modified if you specify the WHERE clause.
* All rows in the table are modified if you omit the WHERE clause.

Syntax: UPDATE *table* SET *column* = *value* [, *column* = *value, ...*][WHERE *condition*];

Example: UPDATE employees SET department\_id = 70 WHERE employee\_id = 113;

**How to delete rows from a table?**

We can remove existing rows from a table by using the DELETE statement. Specific rows are deleted if we specify the WHERE clause. All rows in the table are deleted if you omit the WHERE clause.

For Example: DELETE FROM departments WHERE department\_name = 'Finance';

**What is Truncate Statement?**

The **TRUNCATE** TABLE statement is used to remove all records from a table in **Oracle**. It performs the same function as a DELETE statement without a WHERE clause.

**N.B. T**he **TRUNCATE** TABLE statement cannot be rolled back.

**What is the characteristic of Truncate Statement?**

• The TRUNCATE statement is a data definition language (DDL) statement and generates no rollback information.

• Truncating a table does not fire the delete triggers of the table.

• If the table is the parent of a referential integrity constraint, you cannot truncate the table. You need to disable the constraint before issuing the TRUNCATE statement. Disabling constraints is covered in a subsequent lesson.

**What is Database Transactions?**

A **transaction** is a logical unit of work that contains one or more SQL statements. A **transaction** is an atomic unit. The effects of all the SQL statements in a **transaction** can be either all committed (applied to the **database**) or all rolled back (undone from the **database**).

**A database transaction consists of one of the following:**

• DML statements that constitute one consistent change to the data

• One DDL statement

• One data control language (DCL) statement

**When Does a Transaction Start and End?**

A transaction begins when the first DML statement is encountered and ends when one of the following occurs:

• 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.

**What is Rollback?**

In **Oracle**, the **ROLLBACK** statement is used to undo the work performed by the current transaction or a transaction that is in doubt.

**What are the advantages of COMMIT and ROLLBACK Statements?**

Advantages of COMMIT and ROLLBACK Statements

With COMMIT and ROLLBACK statements, you can:

• Ensure data consistency

• Preview data changes before making changes permanent

• Group logically related operations

**How to control Transaction?**

We can control the logic of transactions by using the COMMIT, SAVEPOINT, and ROLLBACK

Statements.

**Note:** In SQL\*Plus, the AUTOCOMMIT command can be toggled ON or OFF. If set to ON, each individual DML statement is committed as soon as it is executed. You cannot roll back the changes. If set to OFF, the COMMIT statement can still be issued explicitly.

**How many ways a user can access the database?**

Database users access the database in two ways:

• Read operations (SELECT statement)

• Write operations (INSERT, UPDATE, DELETE statements)

**Chapter-9**

**What are the Objects of Database?**

**Database Objects are:**

• **Table:** Stores data

• **View:** Subset of data from one or more tables

• **Sequence:** Generates numeric values

• **Index:** Improves the performance of some queries

• **Synonym:** Gives alternative names to objects

**What are the rules of naming Database Table?**

Table names and column names:

• Must begin with a letter

• Must be 1–30 characters long

• Must contain only A–Z, a–z, 0–9, \_, $, and #

• Must not duplicate the name of another object owned by

the same user

• Must not be an Oracle server–reserved word

**What is Schema?**

A schema is a collection of objects such as tables, views, and sequences. The schema is owned by a database user and has the same name as that user. Schema objects include tables, views, synonyms, sequences, stored procedures, indexes, clusters, and database links.

**What are the guidelines for creating a table?**

**Guidelines**

• A LONG column is not copied when a table is created using a subquery.

• A LONG column cannot be included in a GROUP BY or an ORDER BY clause.

• Only one LONG column can be used per table.

• No constraints can be defined on a LONG column.

• You might want to use a CLOB column rather than a LONG column.

**What is Timestamp Data Type?**

The TIMESTAMP data type is an extension of the DATE data type. It stores the year, month, and day of the DATE data type plus hour, minute, and second values. This data type is used for

storing precise time values.

**What is Constraint?**

A CONSTRAINT clause is an optional part of a [CREATE TABLE statement](https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqlj24513.html#rrefsqlj24513) or [ALTER TABLE statement](https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqlj81859.html#rrefsqlj81859). A constraint is a rule to which data must conform. Constraint names are optional.

**What are the Constraints Types?**

The following constraint types are valid:

– NOT NULL

– UNIQUE

– PRIMARY KEY

– FOREIGN KEY

– CHECK