TOPS TECHNOLOGY

Module 4 – Introduction to DBMS

Presented By :

Nandni Vala

Introduction to PL/SQL

1. What is PL/SQL, and how does it extend SQL's capabilities?

- PL/SQL (Procedural Language/Structured Query Language) is Oracle's procedural extension to SQL that combines SQL with procedural programming features, enabling the creation of robust, scalable, and efficient database applications.
- **SQL's Capabilities**:
- **Procedural Features**: Adds constructs like loops, conditions, and exception handling.
- **Code Reusability**: Supports procedures, functions, and packages for reusable logic.
- **Error Handling**: Provides exception handling to manage runtime errors.
- **Performance**: Reduces network latency by bundling SQL statements into blocks.
- **Variables and Data Types**: Allows declaring variables and performing complex computations.
- **Triggers**: Automates tasks with event-driven programming.

- **Transaction Control**: Offers control over transactions with COMMIT, ROLLBACK, etc.
- **Dynamic SQL**: Executes dynamically constructed queries.
- **Example**:
- BEGIN
- UPDATE employees SET salary = salary * 1.10 WHERE department = 'Sales';
- COMMIT;
- END;

2. List and explain the benefits of using PL/SQL.

- **Benefits of Using PL/SQL**:
- **Improved Performance**:
  - PL/SQL reduces network traffic by executing multiple SQL statements in a single block, enhancing performance.

- **Code Reusability**:
  - PL/SQL allows the creation of reusable procedures, functions, and packages, reducing redundancy and simplifying maintenance.
- **Error Handling**:
  - Built-in exception handling ensures better management of runtime errors, allowing for custom error messages and actions.
- **Modularity**:
  - Code can be organized into logical units (procedures, functions, packages), improving readability and maintainability.
- **Security**:
  - PL/SQL provides control over user access by encapsulating logic within stored procedures and functions, limiting direct access to data.
- **Transaction Control**:
  - PL/SQL offers explicit transaction control with COMMIT, ROLLBACK, and SAVEPOINT, ensuring data consistency.

- **Flexibility**:
  - PL/SQL supports dynamic SQL, allowing the execution of SQL statements that are constructed at runtime, increasing flexibility.
- **Integration with SQL**:
  - Seamless integration with SQL enables the use of SQL queries alongside procedural constructs for more powerful database operations.