SQL Handouts

Name: E.R Harish

1. Introduction to Databases

- SQL/NoSQL: SQL databases use structured schemas; NoSQL databases are flexible with unstructured data.
- Advantages & Disadvantages: SQL offers reliability and ACID compliance, while NoSQL excels with scalability and flexibility.
- Managing Databases: Understanding SQL Server's system databases is key for managing configurations and metadata.

2. Normalization

- 1NF: Ensures atomicity; no repeating groups.
- 2NF: Meets 1NF and removes partial dependencies.
- 3NF: Removes transitive dependencies from 2NF.
- BCNF: A stricter form of 3NF, handling anomalies in composite keys.

3. Managing Databases

- System Databases in SQL Server: Databases like `master`, `model`, `msdb`, `tempdb` are integral to SQL Server.
- User-Defined Databases: Creation, renaming, and deletion processes for custom databases.

4. Managing Tables

- Data Types: SQL supports various types like INT, VARCHAR, DATE.
- Creating/Modifying Tables: Defining or changing table structures.
- Renaming/Dropping Tables: Renaming or permanently removing tables as needed.

5. Manipulating Data with DML Statements

- Insert, Update, Delete: Basic operations for adding, changing, or removing data.
- Data Retrieval: Selecting specific columns and rows, filtering with `WHERE`, `IN`, `DISTINCT`, `LIKE`, etc.
- Demo: Practical applications of inserting, updating, and retrieving data.

6. Data Integrity and Functions in DML

- Data Integrity: Enforcing rules for valid data (constraints, keys).
- Functions: Customize outputs using string, date, math, and system functions.
- Grouping: Summarize data with aggregate functions and grouping operations.

7. Querying Data with Joins

- Inner Join: Retrieves common records between tables.
- Left/Right Join: Retrieves all records from the left/right table, and matched ones from the other.
- Full Outer Join: Retrieves all records when there is a match in either table.
- Cross Join: Creates a Cartesian product.
- Demo: Demonstrates join usage in data analysis.

8. Joins and Subqueries

- Equi Join: Uses equality operator to match columns.
- Self Join: Joins a table to itself.
- Advanced Grouping: Grouping with sets, `HAVING` clause.
- Demo: Practical use of joins and grouping for complex queries.

9. Subqueries and Set Operations

- Subqueries: Nested queries providing intermediate results.
- EXISTS, ANY, ALL: Test subquery results for specific conditions.
- UNION, INTERSECT, EXCEPT, MERGE: Combine results from multiple queries.