

## **PostgreSQL Course Content**

### **1. Introduction to PostgreSQL**

Overview of PostgreSQL and its architecture.

Key features of PostgreSQL (e.g., open-source, extensibility, compliance with SQL standards).

Installation and setup of PostgreSQL on various platforms.

PostgreSQL ecosystem and tools (e.g., psql, pgAdmin, DBeaver).

### **2. PostgreSQL Data Types**

Supported data types:

Numeric types (INTEGER, BIGINT, NUMERIC, SERIAL).

Character types (CHAR, VARCHAR, TEXT).

Date and time types (DATE, TIMESTAMP, INTERVAL).

Boolean type (BOOLEAN).

JSON and JSONB types.

Array and composite types.

Choosing the appropriate data type for your application.

### **3. Database and Schema Management**

Creating and managing databases.

Understanding schemas and their role in PostgreSQL.

Managing objects within schemas (tables, views, functions).

Schema search paths and case sensitivity.

### **4. SQL Queries in PostgreSQL**

Writing basic and advanced SQL queries.

Using LIMIT, OFFSET, and window functions.

Common Table Expressions (CTEs) and recursive queries.

Working with joins, subqueries, and set operations.

## 5. Indexing in PostgreSQL

Types of indexes:

B-tree (default index type).

GIN (Generalized Inverted Index) for full-text search.

GiST (Generalized Search Tree) for spatial data.

BRIN (Block Range Index) for large datasets.

Creating and managing indexes.

Understanding index-only scans and query optimization.

## 6. Transactions and Concurrency

PostgreSQL's implementation of MVCC (Multi-Version Concurrency Control).

Transaction management:

BEGIN, COMMIT, and ROLLBACK.

Savepoints for partial rollbacks.

Isolation levels (READ COMMITTED, REPEATABLE READ, SERIALIZABLE).

Deadlocks and how to avoid them.

## 7. Functions and Stored Procedures

Writing functions in PostgreSQL using PL/pgSQL.

Differences between functions and stored procedures.

Working with input and output parameters.

Error handling and exception management in PL/pgSQL.

## 8. Advanced Query Features

- Full-text search using PostgreSQL.
- Working with JSON and JSONB data types.
- Querying JSON data with operators (->, ->>, @>).
- Window functions for advanced analytics.
- Using PostgreSQL's WITH clause for query modularity.

## 9. Performance Tuning and Optimization

Understanding the query planner and execution plans (EXPLAIN, EXPLAIN ANALYZE).

- Optimizing queries using indexes and materialized views.
- Vacuuming and autovacuum for table maintenance.
- Partitioning strategies (declarative partitioning, inheritance-based partitioning).

## 10. Backup and Recovery

- Logical backups using pg\_dump and pg\_dumpall.
- Physical backups using pg\_basebackup.
- Restoring databases from backups.
- Point-in-time recovery (PITR) using Write-Ahead Logs (WAL).

## 11. Security and Access Control

- Role-based access control (roles, privileges, and permissions).
- Managing authentication with pg\_hba.conf.
- Encryption options (SSL/TLS for connections).
- Row-level security policies.

## 12. PostgreSQL Configuration

Key configuration files:

`postgresql.conf` (server settings).

`pg_hba.conf` (authentication settings).

Tuning memory, connections, and performance parameters.

Logging and monitoring settings.

## 13. Extensions and Advanced Features

Popular PostgreSQL extensions:

PostGIS for spatial data.

`pg_stat_statements` for query monitoring.

`pg_trgm` for text similarity searches.

Installing and managing extensions.

Using Foreign Data Wrappers (FDWs) for external data sources.

## 14. High Availability and Replication

Setting up streaming replication.

Logical replication for selective data replication.

Understanding failover and high availability solutions (e.g., Patroni, PgBouncer).

Backup and disaster recovery strategies.

## 15. Monitoring and Maintenance

Monitoring database performance with system views (`pg_stat_activity`, `pg_stat_user_tables`).

Understanding PostgreSQL logs for debugging.

Regular maintenance tasks:

Vacuuming and analyzing tables.

Reindexing and managing bloat.

## 16. Migration from Oracle to PostgreSQL (Newly Added Topic)

Key Differences Between Oracle and PostgreSQL:

Data types mapping (e.g., NUMBER to NUMERIC, VARCHAR2 to VARCHAR).

Handling sequences and auto-increment columns (SERIAL/BIGSERIAL in PostgreSQL).

PL/SQL vs PL/pgSQL syntax and procedural differences.

Managing case sensitivity and object naming conventions.

Migration Tools:

ora2pg (open-source tool for schema and data migration).

pgloader (data migration tool).

Commercial tools like DBConvert.

Steps for Migration:

Schema migration (converting Oracle schemas to PostgreSQL).

Data migration (exporting and importing data).

Migrating stored procedures, functions, and triggers.

Application code changes for PostgreSQL compatibility.

Testing and Validation:

Verifying data integrity after migration.

Performance testing and query optimization.

Common Challenges:

Handling Oracle-specific features (e.g., CONNECT BY, START WITH).

Migrating large datasets efficiently.

Addressing differences in transaction handling.

## 17. PostgreSQL Ecosystem and Tools

Familiarity with PostgreSQL client tools:

- psql (command-line client).

- pgAdmin (GUI-based client).

- Third-party tools like DBeaver, DataGrip.

Integration with application frameworks and ORMs (e.g., Django ORM, SQLAlchemy, Hibernate).

## 18. PostgreSQL Best Practices

- Writing efficient SQL queries.

- Designing normalized database schemas.

- Using transactions effectively.

- Monitoring and optimizing database performance.