4-Week Learning Plan

Week 1: SQL Basics and Advanced Querying

- **Day 1-2**:
    - Introduction to relational databases
    - Basic queries (SELECT, INSERT, UPDATE, DELETE)
    - Data types and basic operators (AND, OR, IN, etc.)
- **Day 3-4**:
    - Joins (INNER, LEFT, RIGHT, FULL)
    - Subqueries and set operations (UNION, INTERSECT)
- **Day 5-7**:
    - Window functions (ROW_NUMBER(), RANK())
    - CTEs (Common Table Expressions)
    - Aggregation and Grouping (GROUP BY, HAVING, aggregate functions)

By the end of the first week, you should have a solid grasp of SQL fundamentals and advanced querying techniques, which form the backbone of your SQL knowledge.

Week 2: Data Manipulation, Database Design, and Performance

- **Day 8-9**:
    - Transactions (BEGIN, COMMIT, ROLLBACK) and ACID properties
    - Indexes and their impact on performance
- **Day 10-12**:
    - Constraints (PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK)
    - Views and Materialized Views
    - Normalization (1NF, 2NF, 3NF, BCNF)
- **Day 13-14**:
    - Query optimization techniques
    - Understanding and reading query execution plans

    - Index tuning

In week 2, you'll learn how to manipulate and manage data efficiently, design optimized database schemas, and start focusing on performance tuning.

Week 3: PostgreSQL-Specific Features and Advanced SQL

- **Day 15-16**:
    - PostgreSQL-specific features: JSON, Arrays, HSTORE
    - PL/pgSQL: Writing stored procedures and functions
- **Day 17-18**:
    - Full-text search in PostgreSQL
    - Extensions (e.g., PostGIS, TimescaleDB)
- **Day 19-21**:
    - Table partitioning and advanced indexing (GIN, GiST, BRIN)
    - PostgreSQL replication and high availability

By the end of week 3, you'll have a good understanding of PostgreSQL’s unique features, which can differentiate you from someone who only knows general SQL concepts.

Week 4: Security, Backup, ETL, and NoSQL Concepts

- **Day 22-23**:
    - Database security: Role-based access control, permissions
    - Preventing SQL Injection and encryption
- **Day 24-25**:
    - Backup and recovery strategies
    - Replication and high availability
- **Day 26-28**:
    - Overview of NoSQL: Differences, when to use SQL vs. NoSQL
    - Learn about NoSQL systems like MongoDB, Redis, and Cassandra

Week 4 will round off your understanding by covering security best practices, backup strategies, and basic exposure to NoSQL systems. This will give you a holistic view of the database landscape.

Daily Time Commitment

- If you can dedicate **2-4 hours per day**, this timeline can be feasible.
- **Consistency is key**: Ensure you're doing hands-on practice by writing SQL queries, experimenting with PostgreSQL, and reading execution plans.

Post-Month Review Plan

Since mastering SQL requires continuous practice, even if you cover all the topics in a month, it's important to:

- Revisit tough topics after the month ends.
- Solve SQL problems from sites like **LeetCode**, **HackerRank**, or **Mode Analytics**.
- Work on a personal project that involves database design and querying.

Additional Resources

- **Documentation**: Refer to the official PostgreSQL documentation for deeper insights.
- **Courses**: Platforms like Udemy, Coursera, and freeCodeCamp offer structured courses that complement your learning.

Feasibility

While **finishing the basics** and intermediate topics in one month is possible with intense focus, mastering advanced topics (like query optimization, performance tuning, and specific PostgreSQL features) might require more time. Practice will be essential for long-term mastery.