**********PostgreSQL architecture**********

Let's dive into the detailed explanation of the PostgreSQL architecture and its components:-

1. Client Applications:-

- Client applications interact with the PostgreSQL server to perform database operations such as querying data, inserting records, updating data, and executing administrative commands.

- Client applications can be developed in various programming languages and use appropriate libraries or drivers to establish a connection with the PostgreSQL server.

2. Postmaster:-

- The Postmaster process is the main control process of the PostgreSQL server. It starts and stops other processes, handles client connections, and manages the overall operation of the server.

- When a client application initiates a connection request, it is received by the Postmaster process, which then spawns a separate backend process to handle the client connection.

3. Backend Processes:-

- Backend processes are responsible for executing queries, managing transactions, and interacting with the storage subsystem.
- Each client connection is associated with a separate backend process that handles requests from the corresponding client application.
- Multiple backend processes can run concurrently to handle concurrent client connections and ensure scalability.
- Backend processes communicate with the Postmaster process for tasks such as connection management, query dispatching, and resource allocation.

4. Shared Buffer Cache:-

- The shared buffer cache is a portion of the server's memory used to cache frequently accessed data pages from the database.
- When a backend process needs to access a data page, it first checks if the page is already present in the shared buffer cache.
- If the page is found in the cache, it is read from memory, eliminating the need for disk I/O and improving performance.

- If the page is not in the cache, it is read from disk and then stored in the cache for future use.

5. Storage Subsystem::-

- PostgreSQL supports multiple storage engines and allows the use of different file systems.

- Data files are organized into tablespaces, which can span across multiple physical storage devices.

- Tables, indexes, system catalogs, and other database objects are stored as files on disk.
- The storage subsystem handles tasks such as reading and writing data pages, managing disk space, and ensuring data durability.

6. System Catalogs:-

- PostgreSQL maintains system catalogs, which are special tables that store metadata about the database objects.
- System catalogs contain information about tables, indexes, views, functions, and other database entities.
- Backend processes query system catalogs to access information about the database schema, object definitions, and other relevant metadata.

7. Write-Ahead Logging (WAL):-

- PostgreSQL uses a write-ahead log (WAL) mechanism to ensure durability and crash recovery.
- Before changes are written to data pages on disk, they are first written to the WAL, which serves

8. Autovacuum Launcher:-

- The Autovacuum Launcher process is responsible for automatic maintenance of the database by performing vacuuming and analyzing of tables.
- Vacuuming frees up space occupied by dead tuples, updates statistics, and optimizes the layout of tables to improve query performance.
- The Autovacuum Launcher monitors the workload and triggers vacuum and analyze operations based on predefined thresholds and configuration settings.

9. Background Writer (BGW):-

- The Background Writer process, also known as the bgwriter, is responsible for reducing disk I/O load by asynchronously writing dirty (modified) buffers from memory to disk.

- It writes data pages to disk in the background, reducing the frequency of synchronous disk writes during regular transactions and improving overall database performance.

10.Checkpointer:-

- The Checkpointer process performs a critical role in ensuring the durability and crash recovery of PostgreSQL.
- It writes the data from the shared buffers to disk periodically, known as a checkpoint, to ensure that changes made by committed transactions are durably stored on disk.
- The Checkpointer also helps in reducing the recovery time required during server startup after a crash.

11.WAL Writer:-

- The Write-Ahead Logging (WAL) Writer process writes the changes made to the database to the write-ahead log (WAL) files.
- The WAL Writer writes transaction logs to disk before the corresponding changes are applied to the data pages
- This mechanism ensures durability and crash recovery, as the transaction logs can be used to replay the changes and restore the database to a consistent state in the event of a crash.