**Insert and Select Query Flow**

Detailed explanation of how an **INSERT statement flows through the PostgreSQL architecture**, from client to disk.

**PostgreSQL Architecture: Key Components Involved**

1. **Client** – Sends SQL queries (like INSERT) to the server.
2. **Postgres Process** – Receives and parses the query.
3. **Parser** – Converts SQL into internal query structure.
4. **Planner/Optimizer** – Generates the best execution plan.
5. **Executor** – Runs the plan step-by-step.
6. **Buffer Manager** – Manages data pages in memory (shared buffers).
7. **WAL Writer** – Writes changes to the Write-Ahead Log (WAL).
8. **Checkpointer** – Periodically flushes dirty pages to disk.
9. **Data Files** – Where the final data is stored on disk.

**Step-by-Step Flow of an INSERT Statement**

**Example SQL**

INSERT INTO employees (name, department, salary)

VALUES ('John Doe', 'HR', 50000);

**1. Client Sends Query**

* Application or user sends the INSERT via libpq (PostgreSQL's protocol library).

**2. PostgreSQL Server Parses the Query**

* The parser checks syntax and converts the SQL to a parse tree.

**3. Planner/Optimizer Evaluates**

* The planner determines the **cheapest plan** for executing the INSERT, even though it’s straightforward.
* It checks constraints (like indexes, foreign keys).

**4. Executor Executes Plan**

* The executor:
  + Allocates a tuple slot.
  + Builds a new row in memory.
  + Applies any rules or triggers.
  + Inserts the row into the **shared buffers** (PostgreSQL's memory cache).

**5. Buffer Manager Writes to Shared Buffers**

* The row is inserted into a memory page in the **shared buffer pool**.
* The page is marked as **dirty** (modified but not yet flushed to disk).

**6. Write-Ahead Log (WAL) Written**

* Before confirming the insert to the client, PostgreSQL logs the insert into the **WAL**.
* WAL ensures durability: if a crash happens, changes can be replayed.
* WAL is written **sequentially**, making it very fast.

**7. Acknowledgment Sent to Client**

* Once WAL is flushed to disk, **PostgreSQL returns success** to the client, even though the data page may still be in memory.

**8. Checkpointer Flushes to Disk (Eventually)**

* Later, the **checkpointer** or **bgwriter** writes the dirty page from shared buffers to disk.
* The actual disk file (e.g., base/12345/67890) gets updated.

A diagram of a computer program

AI-generated content may be incorrect.

**Summary Table**

| **Component** | **Role in INSERT** |
| --- | --- |
| Client | Sends SQL query |
| Parser | Parses SQL into a tree |
| Planner | Builds an optimal plan |
| Executor | Executes insert logic |
| Shared Buffers | Holds in-memory pages |
| WAL | Logs changes for durability |
| Checkpointer | Writes dirty pages to disk later |
| Data Files | Final destination of inserted rows |