Problem Statement

You are tasked with building a **production-grade application** that:

1. Ingests **multiple Excel/CSV files**, each representing **data for a month/quarter/year**.
2. Extracts **numeric and categorical columns** from each table.
3. Performs **LLM-generated operations** (average, sum, growth, trend analysis, etc.) both:
    - **Individually (per table)** — e.g., “average revenue in Nov 2024.”
    - **Across tables (temporal comparison)** — e.g., “compare Q3 vs Q4 revenue for Widget-A.”
4. Stores results in a **relational DB** with session/user context.
5. Provides a **“Talk to Your Data” conversational interface**, which allows reasoning-based questions such as:
    - *“Has online revenue grown consistently across the last three months?”*
    - *“Which region had the sharpest revenue decline in Q4 2024 compared to Q3 2024?”*
    - *“If discounts continue at current levels, what’s the projected impact on Q1 2025 revenue?”*
6. Uses **LangGraph** to orchestrate the reasoning flow, with nodes for parsing, planning, code generation, execution, explanation, and response.

Example Dataset

File 1: `sales_nov_2024.csv`

Order ID	Customer ID	Order Date	Region	Product	Units	Unit Price	Discount	Revenue	Delivery Date	Meta
1001	CUST-01	15/11/2024	APAC	Widget-A	32	10.5	0.05	319.2	20/11/2024	{"channel":"online", "priority":"high"}
1002	CUST-02	16/11/2024	EU	Widget-B	100	20.0	0.10	1800.0	25/11/2024	{"channel":"retail"}
1003	CUST-03	17/11/2024	NA	Widget-A	15	10.5	0.00	157.5	19/11/2024	{"channel":"online"}

File 2: `sales_dec_2024.csv`

Order ID	Customer ID	Order Date	Region	Product	Units	Unit Price	Discount	Revenue	Delivery Date	Meta
2001	CUST-04	05/12/2024	APAC	Widget-C	50	50.0	0.20	2000.0	15/12/2024	{"channel":"online"}
2002	CUST-01	07/12/2024	EU	Widget-B	80	20.0	0.05	1520.0	18/12/2024	{"channel":"retail"}
2003	CUST-05	10/12/2024	NA	Widget-A	40	10.5	0.00	420.0	15/12/2024	{"channel":"partner"}

File 3: `sales_q1_2025.csv` (aggregated quarterly file)

Quarter	Region	Product	Total_Units	Avg_Unit Price	Avg_Discount	Total_Revenue
Q1-2025	APAC	Widget-A	120	10.5	0.10	1134.0
Q1-2025	EU	Widget-B	300	20.0	0.08	5520.0
Q1-2025	NA	Widget-C	80	50.0	0.15	3400.0

Expectations

Backend

- Handle **multiple files** (monthly/quarterly/yearly).

- Support **cross-table reasoning**:
    - Compute aggregates per file.
    - Compare across time windows (growth, trends, YoY, MoM, QoQ).

- LLM-generated **analysis plans** must distinguish:
    - *Single-table operation*: “average revenue in December 2024.”
    - *Cross-table operation*: “growth of Widget-B from November 2024 to Q1 2025.”

Conversational Interface

- Must handle **temporal reasoning**: understand “last month,” “quarterly trend,” “compared to last year.”

- Must generate **actionable insights**, not just numbers:

  “Revenue for Widget-B grew by 15% in the EU between Nov and Dec 2024, but average discount also increased, suggesting margin pressure.”

- Should allow **multi-turn follow-ups**:
    - Q: “How did Widget-A perform in APAC?”
    - Q2: “Was the growth consistent across Nov and Dec?”

LangGraph

Nodes should include:

1. `parse_files` — ingest multiple files, extract schemas, align columns.
2. `plan_operations` *(LLM)* — decide if single-table or cross-table.
3. `align_timeseries` *(LLM helper)* — align tables by month/quarter/year before codegen.
4. `generate_code` → `validate_code` → `execute_code`.
5. `trend_analysis` *(optional LLM node)* — detect patterns (growth, seasonality, anomalies).
6. `explain_result` — narrative + recommended actions.

- Graph transitions should follow a **clean DAG**.
- Log and trace each node's input/output.
- Use LangGraph's StateGraph to define your graph structure.
- Add failure handling and retry if any node fails.

Example Queries & Expected LangGraph Flow

Q1. "Show **average Revenue** by Region for Nov 2024 and Dec 2024, and the **MoM growth** per region."

- **Intent:** cross-table comparison, group-by Region, MoM growth
- **Flow:** `ingest_query → retrieve_context → analyze_intent → plan_analysis (files=[nov, dec], ops=[groupby mean, growth]) → generate_code → validate_code → execute_code → explain_result → return_chat`
- **Plan (LLM):**
    - Load `sales_nov_2024`, `sales_dec_2024`; group by `Region`, compute `mean(Revenue)` per month.
    - Join Nov vs Dec on Region; compute `growth_pct = (Dec-Nov)/Nov`.
- **Outputs:** Table `[Region, AvgRev_Nov, AvgRev_Dec, MoM_Growth%]` + narrative (call out biggest mover) + 1 action.

Q2. "For **Widget-B in the EU**, did **discount increases** correlate with **revenue growth** from **Nov→Dec 2024**? Show correlation and a 1-line takeaway."

Q3. "Detect **anomalies** in **Units** for **APAC** across **Nov, Dec 2024 and Q1-2025**, and suggest 2 actions."

Q4. "What % of **online** channel revenue (from `Meta.channel`) comes from **Widget-A** in **APAC** over **Nov & Dec 2024**? Also show **YoY** if prior year files exist."

Deliverables

- `frontend/` (upload + config + chat), `backend/` (FastAPI), `langgraph/` (graph + nodes), `workers/`

- `db/` (migrations), `tests/`, `README.md` (setup, .env, sample curls), architecture diagram, guardrails note (optional)

- **Sample data**: 3–6 files

Note:

- Ensure that the system is designed for **scale** (can support **> 10,000** concurrent users/requests), and is **robust and fault-tolerant**.

- You are free to use any **open-source language models** on **HuggingFace/Ollama** (you can also use **OpenAI/Anthropic** models if you prefer, but **DO NOT SHARE THE KEYS** in the response codebase).

- You can **make assumptions** about the data & overall system architecture, but please **state your assumptions clearly in the code**.