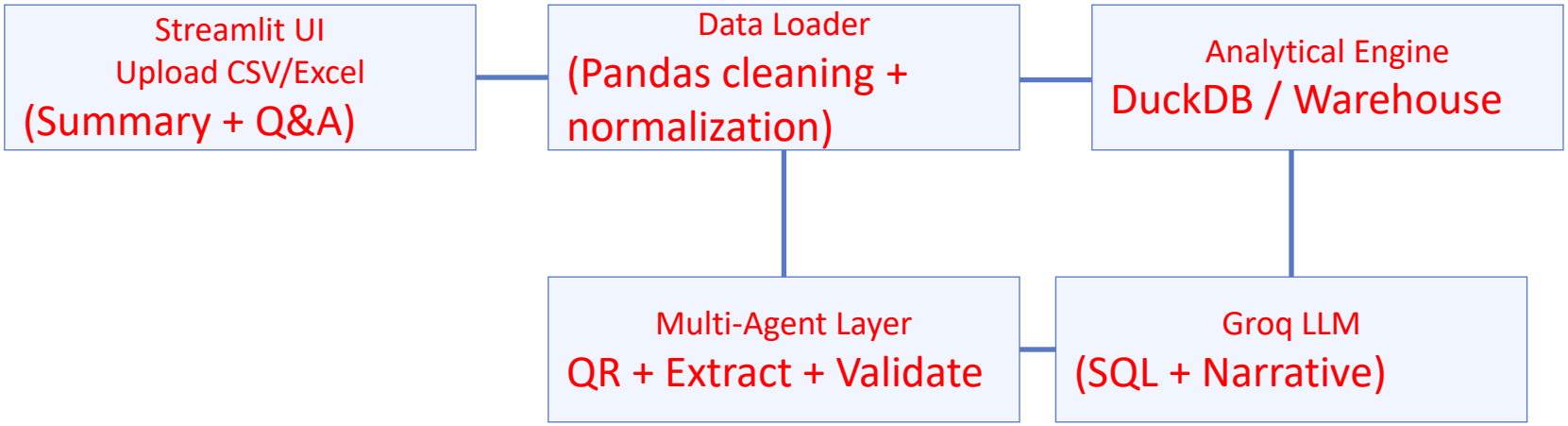
Retail Insights Assistant (GenAI + Multi-Agent System)

Blend360 GenAI Interview Assignment

Presentation by Vaishnavi Kukkala

10/01/2026

System Architecture & Data Flow



LLM Integration Strategy

- LLM is used for:
  - Intent understanding (user question → analytical task)
  - Schema-aware SQL generation
  - Executive narrative summaries / explanation of query output
- LLM is NOT used for:
  - Large scans, joins, aggregations, or metric computation
- Prompt grounding:
  - Inject DuckDB schema (PRAGMA table_info) to avoid hallucinations
  - Include recent conversation context (last N turns)
- Safety controls:
  - Allow SELECT-only queries
  - Block multi-statement SQL; validation gate; retry loop on errors

Multi-Agent Architecture

- Query Resolution Agent
  - Interprets user intent
  - Generates SQL grounded on schema
- Data Extraction Agent
  - Executes SQL on DuckDB (or warehouse at large scale)
  - Returns structured result tables
- Validation Agent
  - Checks empty / low-signal outputs
  - Guards against unsafe operations
- Self-correction loop
  - If SQL fails → pass error back to LLM → refine and retry

Summary

- Runs predefined SQL aggregates (Top categories, Top states, Order status split)
- Builds structured summary blocks from SQL outputs
- LLM converts metrics → executive narrative
- Outputs business recommendations grounded in retrieved data

Example Query → Response Pipeline

- User asks a question in natural language (Streamlit chat input)
- Query Resolution Agent generates schema-aware SQL
- Data Extraction Agent runs SQL in DuckDB
- Validation Agent checks result quality / emptiness
- LLM converts table output into a concise business insight

100GB+ Scale Design: Storage, Indexing & Retrieval

- Storage layers
  - Raw zone: S3 / GCS / Azure Data Lake
  - Curated zone: Partitioned Parquet / Delta Lake
- Compute & query engines
  - Spark/Databricks/dbt for batch ingestion
  - BigQuery/Snowflake/Athena/Trino for SQL query pushdown
- Indexing & optimization
  - Partition by date; cluster by state/category
  - Use pre-aggregated summary tables for common queries
- Optional semantic layer
  - Vector search for unstructured business docs / policies

Cost & Performance Considerations

- Latency controls
  - SQL pushdown + partition pruning
  - Pre-aggregations for frequent metrics
- Cost controls
  - Prompt templates; caching common requests
  - Send only needed schema/columns to LLM
- Monitoring
  - SQL success rate, retry count
  - Latency, token usage, cost

Thank You for your time!