Proposal: Internal AI Co-Pilot for Enterprise Data Access

Overview: What We're Building

We aim to build a secure, on-premise AI Co-Pilot that allows internal users to ask natural-language questions over company systems (Oracle DB, Excel, reports) and receive answers, insights, and summaries instantly. This will:

- Simplify data access (e.g., "What was Vendor A's total spend in Q3?")
- Generate reports (e.g., Excel sheets of high-variance SKUs)
- Validate data (e.g., flag GST mismatches)
- Summarize trends (e.g., procurement highlights)

All data and processing will remain securely inside our infrastructure.

Technology Stack

- **Database:** Oracle DB with Python access via `cx_Oracle`
- **Vector Store:** Chroma, Weaviate, FAISS
- **Embedding Models:** SentenceTransformers (MiniLM, BGE, etc.)
- **LLMs:** Mistral, LLaMA 2, Falcon (quantized, open-source)
- **Frameworks:** LangChain, LlamaIndex

Capabilities

- Query structured data from Oracle via SQL (manual or generated by LLM)
- Perform semantic search over documents and past records
- Generate Excel reports and summaries via Python
- Answer contextual questions using Retrieval-Augmented Generation (RAG)
- Keep everything local for maximum privacy and control

RAG Workflow Explained

What We Actually Do:

1. **Prepare Internal Data**
    - Pull data from Oracle using Python
    - Convert tables/reports to readable text chunks
2. **Embed & Index**
    - Use embedding models to convert text chunks into vectors

    - Store vectors in Chroma/Weaviate
3. **User Query → Semantic Search**
    - User asks a question ("top 10 SKUs with variance")
    - System retrieves relevant text chunks
4. **Prompt the LLM with Retrieved Chunks**
    - Pass both user question + retrieved context to LLM
    - Get accurate, grounded answer

Why This Works

- No fine-tuning needed
- Minimal hallucination (grounded answers)
- Secure and efficient

Real-Time Oracle DB Integration

Can We Pull Data Live from Oracle?

Using Python libraries:

- `cx_Oracle` for direct SQL
- `SQLAlchemy` for ORM
- `Select AI` (if available) for natural language to SQL

Security Best Practices

- Use service accounts with read-only access
- Encrypt connections (SSL/TLS)
- Store credentials securely
- Audit logs & IP firewalls

Hybrid Approach

- Static embeddings for documents, reports
- Live SQL for real-time metrics

This ensures speed + freshness.

Sample Query Flow

User: "What were total sales in Region A last week?"

1. LLM generates:

```
SELECT SUM(sales_amount) FROM sales_data WHERE region='A' AND sale_date
BETWEEN SYSDATE-7 AND SYSDATE;
```

2. Python executes, fetches result

3. Result passed back to LLM:

"Total sales in Region A last week were ₹1.25 million."