

Text-to-SQL Project Documentation

Text-to-SQL Conversion System Project Documentation

Overview

This project is a **Text-to-SQL conversion system** that translates natural language questions into SQL queries using a fine-tuned LLM (OpenAI GPT-3.5 Turbo). It operates on the **Northwind** database and supports both **CLI** and **Streamlit frontend** interaction.

It includes:

- Schema-aware prompt generation
- SQL query optimization & execution
- Evaluation system using precision, recall, F1, and execution match
- Modular architecture

Project Structure

File / Folder	Purpose
----- -----	
`app/prompt_generator.py`	Sends NL + prompt to fine-tuned LLM; returns SQL
`app/sql_executor.py`	Executes SQL queries safely using SQLite; ensures only SELECT
`app/sql_optimizer.py`	Optimizes model output (e.g., replaces `SELECT *`, adds LIMIT)
`setup/initialize_db.py`	Loads schema and index into SQLite DB from SQL files
`setup/create_indexes.sql`	Defines indexes on the Northwind database for performance
`data/northwind.sql`	Northwind schema and data (SQLite format)
`tests/test_cases.json`	Evaluation test cases: NL question + expected SQL
`tests/evaluate.py`	Evaluates SQL output using metrics + execution match

Text-to-SQL Project Documentation

`prompts/schema_prompt.txt`	Prompt template with schema + instructions	
`main.py`	CLI interface for the system	
`app.py`	Streamlit-based interactive frontend	
`.env`	Holds your OpenAI API key	
`requirements.txt`	Python dependencies	

Setup Instructions

1. Install Dependencies

```
```bash

pip install -r requirements.txt

```
```

2. Add your OpenAI API key

```
```env

OPENAI_API_KEY=your-openai-key

```
```

3. Initialize the database

```
```bash

python setup/initialize_db.py

```
```

How to Run the Application

CLI (Terminal-Based)

Text-to-SQL Project Documentation

```
```bash
```

```
python main.py
```

```
```
```

Web UI (Streamlit)

```
```bash
```

```
streamlit run app.py
```

```
```
```

Evaluation System

Located in `tests/evaluate.py`. Supports:

- Exact Match
- Precision / Recall / F1
- SQL Similarity
- Execution Match

```
```bash
```

```
python -m tests.evaluate
```

```
```
```

Sample Result

```
```
```

Exact SQL Match : 80%

Execution Match : 95%

Avg Precision : 0.945

# Text-to-SQL Project Documentation

Avg Recall : 0.964

Avg F1 Score : 0.953

Avg SQL Similarity : 0.98

...

## Architecture Overview

![[Architecture]](your\_architecture\_image\_link)

## Notes & Extras

- Evaluation is optional and triggered for known test cases
- Streamlit has clean tabbed UI
- Safe by design: only SELECT queries allowed

## Future Improvements

- Better JOIN & nested query support
- Confidence scoring
- Deploy with Streamlit Cloud