

Project Development Phase
Performance Test

| | |
|---------------|---|
| Date | 28 February 2026 |
| Team ID | LTVIP2026TMIDS46423 |
| Project Name | Intelligent SQL Querying with LLMs Using Gemini Pro |
| Maximum Marks | |

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

| S.No. | Parameter | Screenshot / Values |
|-------|-------------------------|---|
| 1. | Data Rendered | SQL-LLM dynamically renders relational database records in tabular format using Streamlit DataFrame. Data includes structured rows and columns fetched from SQLite/MySQL database. |
| 2. | Data Preprocessing | Natural language preprocessing, prompt structuring, schema extraction, SQL validation, restricted keyword filtering, and formatted result output. |
| 3. | Utilization of Filters | Filtering implemented via natural language conditions (e.g., "Show employees where salary > 50000"). Converted into SQL WHERE clause dynamically. |
| 4. | Calculation fields Used | SQL aggregate functions used: <ul style="list-style-type: none"> • COUNT() • SUM() • AVG() • MAX() • MIN() Example: <code>SELECT AVG(salary) FROM employees;</code> |
| 5. | Dashboard design | Streamlit-based interactive UI. No. of Visual Components: <ul style="list-style-type: none"> • Query Input Field – 1 • SQL Display Panel – 1 • Result Table – 1 • Error Message Panel – 1 Total Visual Components = 4 |
| 6 | Story Design | Logical user flow of interaction: Step 1 – Enter Query Step 2 – SQL Generation Step 3 – Validation Step 4 – Execution Step 5 – Display Results No. of Logical Interaction Steps = 5 |