AI-Driven Optimization of Data Prompting and Extraction from Very Large Databases

Author: Steven Garner

Date: 29 Jan 2025

Introduction

As organizations deal with ever-growing databases, the challenge of efficiently retrieving relevant data becomes critical. Traditional query-writing processes are often time-consuming and require specialized knowledge. This white paper explores how AI can be leveraged to optimize data prompting and extraction from massive databases by automating SQL query generation, stored procedure execution, and intelligent decision-making regarding live querying, caching, or hybrid approaches.

AI-Driven Solution for Data Extraction

Step 1: Feeding AI with Data Schema and API Documentation

Understanding Database Schema

A database schema defines how data is structured within a database, including:

- **Tables:** Store actual data in structured formats
- **Indexes:** Speed up search queries and improve performance
- **Relationships:** Define how tables connect via foreign keys
- **Constraints:** Enforce data integrity (e.g., primary keys, unique constraints)

Extracting Schema as a PDF (MySQL Example)

To provide AI with an understanding of the database, we can export the schema as a PDF using MySQL:

```
None
mysqldump -u root -p --no-data database_name | pandoc -o
schema.pdf
```

This allows AI models to analyze and compare different versions for changes and optimizations. The schema can be formatted for multiple AI models to interpret, ensuring compatibility across various tools.

Creating Optimized Versions & Side Databases

If the existing schema is not normalized or contains inefficiencies, AI can:

- Identify redundant data and suggest normalization
- Create side databases for optimized queries
- Generate materialized views for frequent queries

For instance, AI can suggest denormalization for read-heavy applications while maintaining normalized structures for transactional consistency.

API Documentation Review & Optimization

APIs often serve as a bridge to access and write data. AI can:

- Analyze API endpoints for inefficiencies
- Review SQL and NoSQL queries used in API calls
- Suggest optimized endpoints and stored procedures

Example:

- If an API performs multiple redundant queries, AI can batch them into a single efficient call.
- AI can recommend caching strategies for frequently accessed endpoints.

Step 2: Providing Sample Data

Sample data enhances AI's ability to:

- Identify patterns and relationships within datasets
- Test and validate query outputs before execution
- Optimize data joins and aggregations for performance improvements

AI can generate synthetic data if real data is unavailable, ensuring privacy while maintaining data structure integrity.

Step 3: Supplying AI with Stored Procedures

Stored procedures encapsulate business logic, making them valuable for AI-driven query generation. By ingesting:

- **Existing stored procedures**
- **Business logic embedded within them**

AI can leverage pre-built optimizations and avoid redundant query generation.

AI can also review stored procedures for:

- Bottlenecks in execution
- Opportunities for parallel execution
- Suggestions for parameterized queries to improve performance

Step 4: AI-Driven Query and Stored Procedure Generation

AI can now:

- Generate optimized SQL queries tailored to specific data needs
- Suggest modifications or optimizations to existing stored procedures
- Automate repetitive query generation tasks to improve efficiency

AI-Optimized Query Execution Strategies

1. Live Query Execution

AI decides to execute queries directly against the live database when:

- **Data freshness is critical** (e.g., stock prices, real-time analytics)
- **Queries are infrequent and lightweight**
- **Security policies prevent local data storage**

Advantages: Always up-to-date results **Disadvantages:** Higher latency, potential database load issues

2. Cached Query Results

AI may choose to cache query results when:

- **Data is not frequently changing** (e.g., historical reports, reference data)
- **Performance is a priority** (reducing redundant queries)
- **Database access is limited due to rate limits or costs**

Advantages: Faster response times, reduced database load **Disadvantages:** Potential for outdated data if not refreshed appropriately

3. Hybrid Approach: Combining Live and Cached Data

A hybrid model balances speed and accuracy by:

- Storing **frequently used, less volatile data** in a cache
- Fetching **time-sensitive or dynamic data** from the live database
- Using AI to determine when to refresh caches based on query patterns

Advantages: Optimized for performance and accuracy **Disadvantages:** Complexity in implementation and maintenance

AI-Driven Performance Optimization

1. Query Optimization

AI can:

- Rewrite inefficient SQL queries for better execution plans
- Suggest index creation to speed up frequent queries
- Detect and eliminate redundant joins and subqueries

2. Stored Procedure Enhancements

AI can:

- Refactor stored procedures for efficiency
- Recommend parameterized queries to improve reusability
- Identify bottlenecks and propose fixes

3. Dynamic Query Execution Planning

AI can:

- Analyze past query performance and adjust execution strategies
- Choose between **live queries, cached results, or hybrid execution** based on workload patterns
- Prioritize query execution based on real-time user demand

4. Hardware & Architecture Considerations

To further enhance performance, AI can analyze:

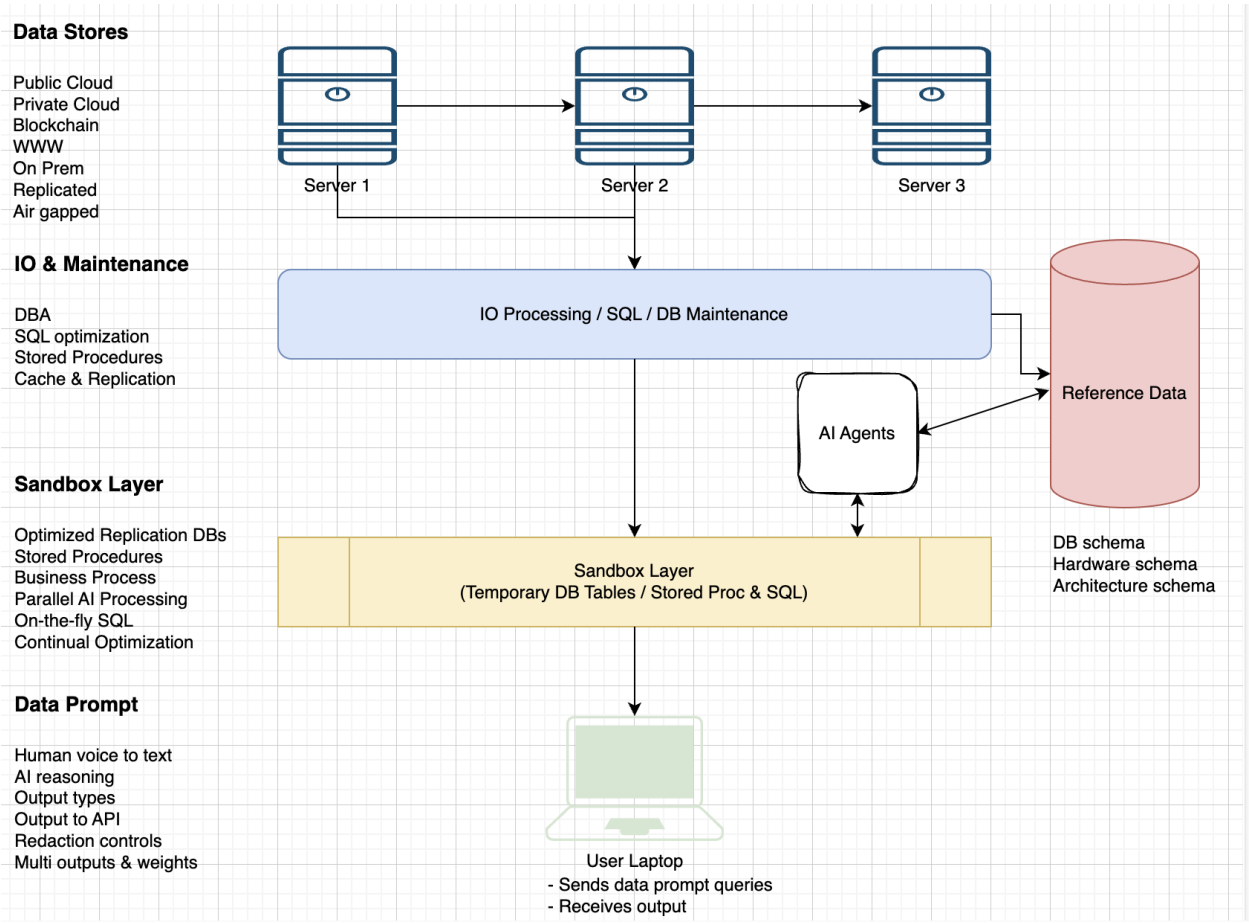
- **Database hardware bottlenecks** (CPU, memory, disk I/O)
- **Distributed database setups** for load balancing
- **Parallel query execution strategies**

On-the-Fly Query Optimization: AI can execute multiple query versions in parallel, learning which is fastest based on execution time. Over time, AI can preemptively select the best query plan for a given workload.

Example

Layers & Components

- **Top Layer: Connected Servers**
  - Multiple servers interconnected
- **Middle Layer: SQL Processing & Maintenance**
  - Handles input/output operations
  - Manages database maintenance
- **Side Layer: Reference Data**
  - Stores DB schema, hardware schemas, architecture diagrams
  - API Documentation
- **Sandbox Layer: Temporary DB Tables**
  - Creates temporary tables for processing
- **User Layer: Laptop for Data Prompts**
  - User sends queries
  - Output is received on the laptop
- **AI Agents: Optimization and Delivery**
  - DB Schema
  - Hardware Architecture Schema
  - External and Internal API Documentation
  - Create On The Fly SQL
  - Create On The Fly Non-Sql
  - Use Hadoop Clusters
  - AI Train of thought problem solving
- **Privacy: AI does not touch the underlying infrastructure**
  - Sandbox rules to determine redaction and prohibition



Conclusion

By integrating AI via agents into the data prompting and extraction process, organizations can significantly improve efficiency, reduce manual effort, and optimize database performance. The proposed approach—feeding AI with schema, API documentation, sample data, and stored procedures—enables AI to autonomously generate and execute optimized queries while intelligently balancing speed and privacy through live, cached, or hybrid execution models.

Future advancements in AI and database management will further enhance these capabilities, making AI-driven database interactions more autonomous, intelligent, and efficient.

References

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016).
   Deep Learning.
   MIT Press.
   https://www.deeplearningbook.org

2. LeCun, Y., Bengio, Y., & Hinton, G. (2015).

"Deep Learning."
Nature, 521(7553), 436–444.
https://doi.org/10.1038/nature14539

3. Silver, D., Huang, A., Maddison, C. J., et al. (2016).
"Mastering the game of Go with deep neural networks and tree search."
Nature, 529(7587), 484–489.
https://doi.org/10.1038/nature16961

4. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017).
"Attention is All You Need."
Advances in Neural Information Processing Systems (NeurIPS), 30, 5998–6008.
https://arxiv.org/abs/1706.03762

5. Sutton, R. S., & Barto, A. G. (2018).
Reinforcement Learning: An Introduction.
MIT Press.
http://incompleteideas.net/book/the-book-2nd.html