NL2SQL (Text-to-SQL)

빅데이터 핀테크

Prof. Sang-Won Lee

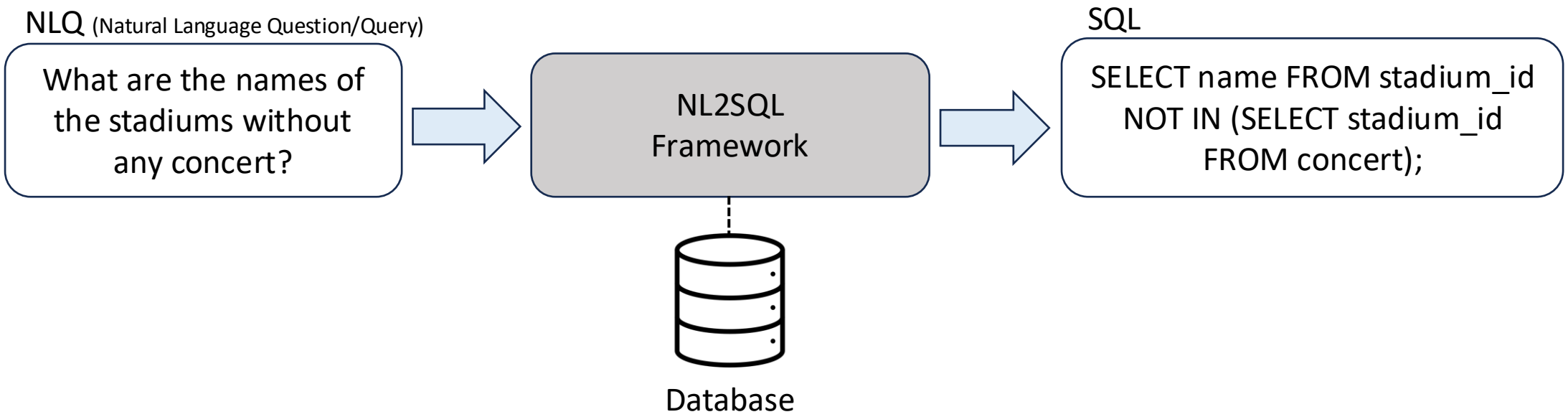
TA. KyongShik Lee

VLDB Lab.

VLDB Lab.

What is NL2SQL?

- **NL2SQL** (Natural Language to SQL) converts a user's natural-language question into an executable SQL query.
- It enables non-experts to query databases without writing SQL manually.
- The model must understand the question's intent and map it to the database schema (tables, columns, ..)



VLDB Lab.

Challenges of NL2SQL

- **Uncertain natural-language queries**
  - User questions are often ambiguous or incomplete, making it difficult for the system to infer the exact intended meaning.
- **Complex databases and dirty content**
  - Real-world databases have complex relationships across tables, and column names or values can be inconsistent or contain errors, which makes query generation harder.
- **From natural language to SQL**
  - Translating flexible natural language into the constrained, formal structure of SQL is inherently challenging.
- **Engineering challenges in NL2SQL systems**
  - Practical systems must consider factors such as cost-effective deployment and model efficiency (latency, resource usage, scalability)

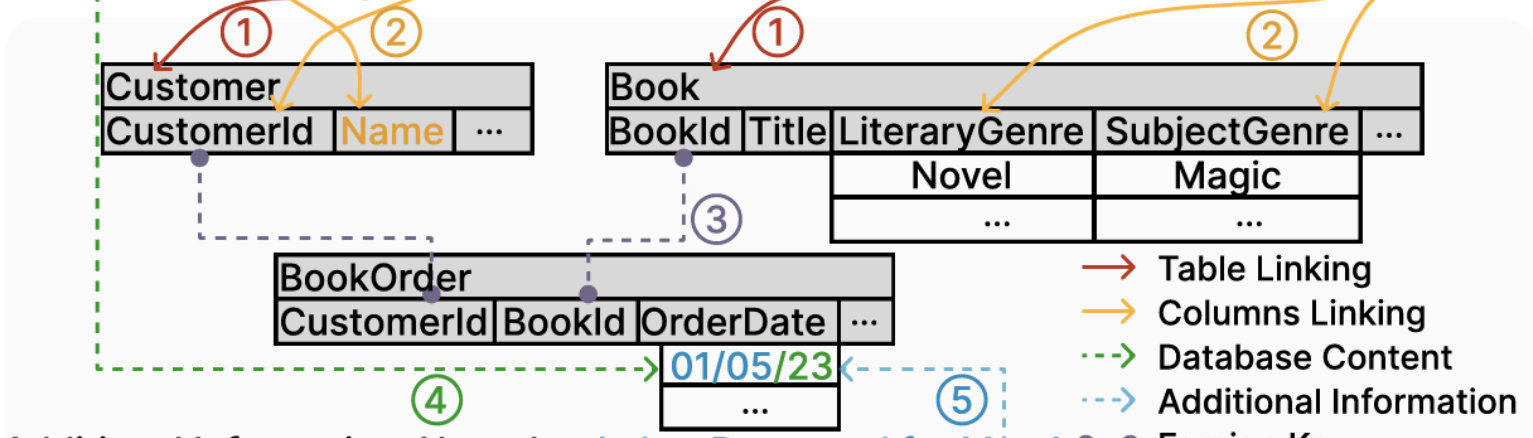
VLDB Lab.

Challenges of NL2SQL

NL Query:

Find the names of all customers who checked out books on exactly 3 different genres on Labor Day in 2023.

Database:



Additional Information: Note that Labor Day stand for May 1

SQL:

```
SELECT Name
FROM Customer
NATURAL JOIN BookOrder
NATURAL JOIN Book
WHERE OrderDate='01/05/23'
GROUP BY CustomerId, Name
HAVING COUNT(DISTINCT SubjectGenre)=3
```

LiteraryGenre

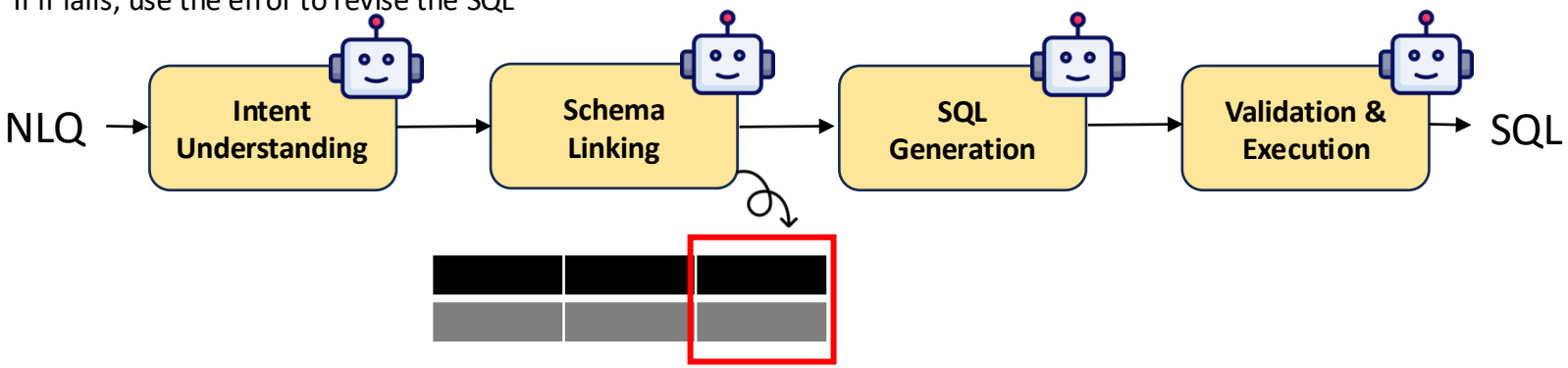
```
SELECT Name
FROM Customer
WHERE CustomerId = (SELECT CustomerId
FROM BookOrder NATURAL JOIN Book
WHERE OrderDate='01/05/23'
GROUP BY CustomerId
HAVING COUNT(DISTINCT SubjectGenre)=3)
```

LiteraryGenre

VLDB Lab.

Typical NL2SQL pipeline

1. **Intent Understanding**
   - What does the user really want? (filters, aggregation, time range, top-k, etc)
2. **Schema Linking**
   - Map words to tables/columns (and find the right join path)
3. **SQL Generation**
   - Produce a valid SQL query (SELECT, JOIN, GROUP BY, HAVING, …)
4. **Validation & Execution**
   - Check syntax/types, run the query, and show results
   - If if fails, use the error to revise the SQL



VLDB Lab.

Typical NL2SQL pipeline

Intent Understanding: What does the user really want? (filters, aggregation, time range, top-k, etc)

Example

You are an analyst that extracts the user's intent from a natural-language database question.
Do NOT write SQL.

Your job:
1) Restate the question in a clear and unambiguous way.
2) Identify the main task type (e.g., lookup, filter, aggregation, ranking, comparison, join).
3) Extract constraints (filters), metrics, grouping, sorting, and output columns.
4) If anything is ambiguous, list the ambiguity and propose 1–2 reasonable assumptions.
5) Produce the result in the JSON format below, with short strings.

Return ONLY valid JSON. No extra text.

Question: {NLQ}

VLDB Lab.

Typical NL2SQL pipeline

Schema Linking: Map words to tables/columns (and find the right join path)

Example 1.

```
CREATE TABLE DEPT (DEPTNO NUMBER(2) CONSTRAINT PK_DEPT PRIMARY
KEY, DNAME VARCHAR2(14) , LOC VARCHAR2(13) ) ;

CREATE TABLE EMP (EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
ENAME VARCHAR2(10), JOB VARCHAR2(9), MGR NUMBER(4), HIREDATE DATE,
SAL NUMBER(7,2), COMM NUMBER(7,2), DEPTNO NUMBER(2) CONSTRAINT
FK_DEPTNO REFERENCES DEPT);
```

Example 2.

```
DEPT (DEPTNO, DNAME, LOC)
EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
```

VLDB Lab.

Typical NL2SQL pipeline

Validation & Execution: Check syntax/types, run the query, and show results

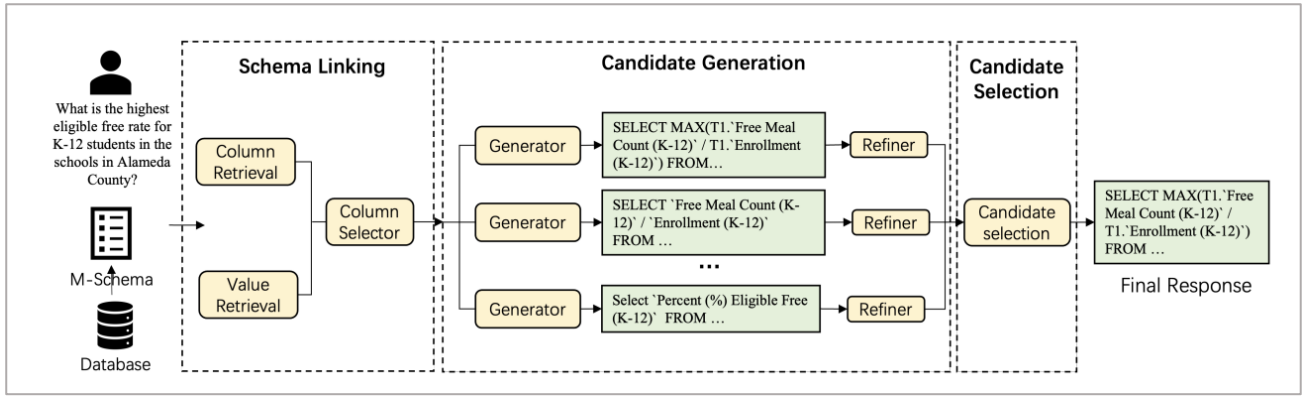
- SQL parse/compile check, execution-based validation, retry, explain/query plan feedback, debugger style traces

Example

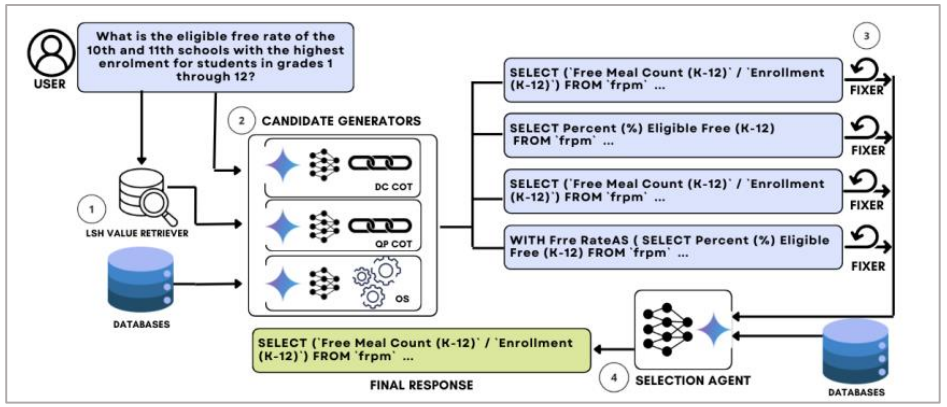
```
You generated a SQL query for SQLite, but it failed.

Rules:
- Return exactly ONE SQLite SELECT query.
- Use only the given schema.
- Keep it minimal (avoid unnecessary joins).
- Add LIMIT 50 if no limit is present.

User question:
{NLQ}

Schema:
{SCHEMA}

Your SQL:
{SQL}

SQLite error:
{ERROR}

Output:
1) Fixed SQL only
2) One-sentence reason for the fix
```

Current NL2SQL Frameworks

- Academic research has developed pipelines that integrate diverse techniques.



XiYan-SQL: A Multi-Generator Ensemble Framework for Text-to-SQL



CHASE-SQL: Multi-Path Reasoning and Preference Optimized Candidate Selection in Text-to-SQL

VLDB Lab.

NL2SQL Platform

- Recently, many platforms that support NL2SQL have emerged.
  - https://www.eversql.com/text-to-sql/
  - https://www.text2sql.ai
  - https://www.sqlai.ai/text-to-sql-ai

- Major DBMS vendors (e.g., Oracle) are also developing NL2SQL capabilities.
  - https://blogs.oracle.com/machinelearning/post/introducing-natural-language-to-sql-generation-on-autonomous-database

- Snowflake introduces a semantic layer that abstracts schema objects into business-friendly concepts that are easier for humans to read and interpret.
  - https://docs.snowflake.com/en/user-guide/views-semantic/overview

VLDB Lab.

Hands-On

- In this lab, we will practice **NL2SQL (Natural Language → SQL)** using a Jupyter/Colab notebook.
- We will use a **small sample database** (SCOTT-like tables) stored in **SQLite (in-memory)**.
- You will write a few SQL queries manually, then try NL2SQL where an LLM generates SQL from your question.

- **Prerequisite**
  - **Google Colab** (https://colab.google/)
  - No database installation needed: we use **SQLite in memory** inside the notebook
  - **OpenAI API key :** We will provide a key for this lab (do not share it publicly)