

NL2SQL 멀티턴 설계

대화 맥락 기반 NL2SQL 후속 질의 이해 및 정확도 개선

LHG 2026-08-27

CONTENTS

왜 필요한가, 처리 흐름

원문과 완성 질문의 구분

형태소보다 중요한 의미

이전 대화를 구조로 기억하기

슬롯필링, 원칙, 기대 효과

이 글에서 쓰는 용어

처음 보는 사람도 이해되게 먼저 풀어둔다

- **완성 질문(normalizedIntent)**: 이전 대화 없이 읽어도 뜻을 알 수 있는 자연어 질문
- **슬롯필링**: 조회에 필요한 빈칸(계좌, 기간 등)을 채우고 부족한 것만 되묻는 기능
- **의미 역할(role)**: 단어가 조회에서 맡는 역할. 명사인지 여부보다 중요하다
- **후속 질문**: "그중 서울만"처럼 이전 대화를 이어받는 질문
- **Coordinator**: 정규화와 슬롯필링을 담당하는 처리 단계

사용자는 매번 완전한 질문을 하지 않는다

LHG

첫 질문: 급여계좌의 최근 한 달 거래내역 보여줘

다음 질문: 출금만 보여줘

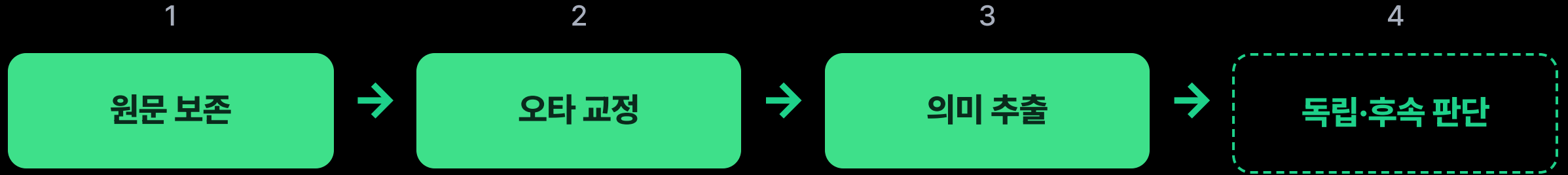
두 번째 문장만 보면 계좌도 기간도 알 수 없다. 사람은 앞선 대화를 보고 이렇게 이해한다.

급여계좌의 최근 한 달 출금 거래내역을 보여줘

이 설계의 목적은 불완전한 질문을
혼자서도 실행할 수 있는 정확한 질문으로 완성하는 것이다.

1. 처리 흐름

보안 검사를 통과한 뒤, Planner가 계획을 세우기 전에 끝낸다



- 이전 대화에서 필요한 값만 재사용하고 부족하거나 애매하면 사용자에게 확인한다
- 완성된 질문만 NL2SQL에 전달한다

Coordinator 상세 프로세스, 준비 단계

1. 원문에서 숫자, 날짜, 계좌번호, 식별자를 보호
2. 오타 교정 + 이전 대화 요약 조회 + 슬롯 정의 조회를 동시에
3. 보호한 값이 교정 중에 바뀌지 않았는지 확인
4. 교정문에서 날짜, 금액, 비교 조건을 코드로 파싱

Coordinator 상세 프로세스, 분석과 확정 단계

LHG

1. 문장 의미와 후속 질문 여부 분석
2. 현재 질문의 조회 종류 분류
3. 후속 질문이면 이전 값 중 맞는 것만 병합
4. 계좌·상품처럼 조회가 필요한 값은 후보를 찾아 해소
5. 기본값 적용, 값 검증, 부족한 필수값 계산
6. 부족하면 되묻고 채워지면 완성 질문 확정

오타 교정, 문장 분석, 이전 대화 참고를 **한 작업으로 뭉개지 않는다.**
각 단계의 입력과 결과를 남겨야 어디서 의미가 달라졌는지 확인할 수 있다.

2. 원문과 완성 질문은 따로 보관한다

사용자가 입력한 원문을 직접 덮어쓰지 않는다

세 가지 값, 세 가지 용도

LHG

원문

- 감사, 재현, 오교정 복구용

교정문

- 오타와 띄어쓰기를 정리한 분석용 문장

완성 질문

- 이전 문맥까지 반영해 NL2SQL에 전달할 문장

예시, 오타와 숫자를 함께 보호한다

LHG

원문: 그중 지난달 천마년 이상 출금만

교정문: 그중 지난달 천만원 이상 출금만

완성 질문: 직전 조회 계좌에서 지난달 출금 거래 중
천만원 이상인 내역을 조회해줘

숫자, 날짜, 계좌번호, 회사명은 교정 전에 보호하고 교정 후에도 같은지 확인한다.



3. 형태소보다 중요한 것은 의미다

명사인지 아닌지보다, 조회에서 어떤 역할인지가 중요하다

단어의 역할을 함께 뽑아야 한다

"매출"이 명사라는 사실만으론 부족하다. 조회할 지표(METRIC)인지
가 중요하다

- "올해"는 기간, "고객별"은 분류 기준, "매출"은 지표, "5개"는 개수 제한
- "매출이 가장 많이 증가한 상품"처럼 증가액인지 증가율인지 모르면 임의로 고르지 않는다

분석 결과 예시

```
{
  "originalQuery": "올해 고객별 매출 상위 5개 보여줘",
  "semanticSlots": {
    "metrics": ["매출"], "dimensions": ["고객"],
    "timeRange": {"kind": "THIS_YEAR"},
    "aggregation": "SUM",
    "sort": {"key": "매출", "direction": "DESC"},
    "limit": 5
  },
  "normalizedIntent": "올해 고객별 매출 합계를 내림차순으로 정렬해 상위 5개를 조회해줘"
}
```

문법 종류와 의미 역할은 다르다

- **pos**: 명사, 동사, 형용사, 숫자 같은 문법적 종류
- **role**: **METRIC** (조회할 수치), **DIMENSION** (나누는 기준), **TIME_RANGE** (기간), **SORT**, **LIMIT** 같은 실제 역할
- 형태소 정보는 내부 분석 자료로만 관리하고 NL2SQL에는 완성된 자연어 질문만 전달한다

4. 이전 대화는 구조로 기억한다

답변 문장을 다시 요약하지 않는다

왜 문장이 아니라 구조로 남기나

AI가 답변을 다시 요약하면 이전 오류가 사실처럼 굳어질 수 있다

- 지표, 분류 기준, 기간, 정렬, 결과 건수, 순위 참조 가능 여부를 코드가 직접 구조화된 결과에서 뽑는다
- 이 정보가 있어야 "그중 서울만", "두 번째 회사는?" 같은 질문을 해석할 수 있다
- 이전 결과에 순위나 지표가 없으면 임의로 만들지 않고 사용자에게 묻는다

프롬프트에 넣는 짧은 요약

LHG

```
<n12sql_turn complete="true">
```

질의: 올해 고객별 매출 합계 상위 5개

지표: 매출 합계 | 분류: 고객 | 기간: 2026년

정렬: 매출 합계 내림차순 | 결과: 5건 | 순위참조: 1~5

```
</n12sql_turn>
```

생성된 SQL문과 전체 결과 데이터는 일반 대화 기록에 넣지 않는다.

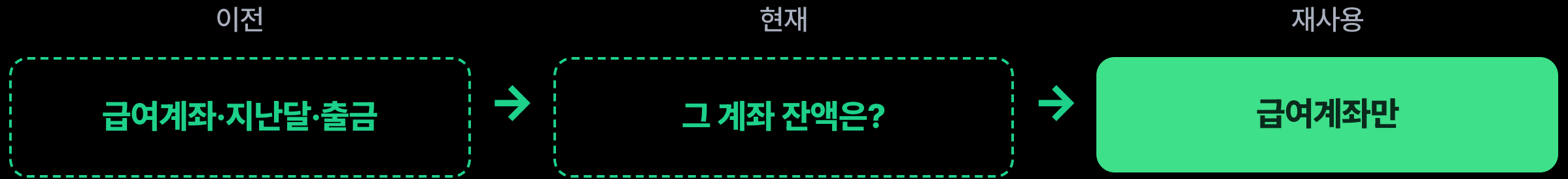


5. 이전 값은 전부 가져오지 않는다

현재 질문과 맞는 값만 재사용한다

계좌는 이어받고 기간·조건은 버린다

LHG



- 현재 질문에서 새로 말한 값이 이전 값보다 우선한다
- 완전히 새로운 질문이면 이전 값을 가져오지 않는다
- 값마다 어디서 왔는지(이전 턴, 현재 발화) 출처를 함께 남긴다

6. 부족한 값은 슬롯필링으로 채운다

계좌, 기간처럼 조회에 필요한 빈칸을 채운다

없는 값만 골라서 묻는다

LHG

사용자: 거래내역 보여줘

필요한 값: 계좌, 기간 / 현재 값: 없음

시스템: 어느 계좌인가요? [급여계좌] [운영계좌]

사용자: 급여계좌

시스템: 기간을 선택해 주세요. [1주일] [1개월] [3개월]

선택지는 AI가 임의로 만들지 않고 관리자가 등록한 값이나 실제 조회 후보에서 만든다.

문장 분석은 후보를, 슬롯필링은 확정을

LHG



- 필요한 값이 모두 채워졌을 때만 다음 단계로 넘어간다

7. 반드시 지켜야 할 원칙

일곱 가지로 정리한다

- 원문을 보존한다

교정문과 완성 질문은 별도 값으로 관리한다

- 없는 내용을 만들지 않는다

확인할 수 없으면 사용자에게 묻는다

- 현재 질문을 우선한다

이전 값은 맞는 것만 가져온다

- 결과를 구조로 기억한다

AI가 만든 답변을 다시 사실 근거로 쓰지 않는다

- **민감정보를 보호한다**

계좌번호와 개인정보는 마스킹 후에만 남긴다

- **완성 질문을 실제로 사용한다**

NL2SQL과 게이트웨이는 원문보다 완성 질문을 우선한다

- **실패하면 추측하지 않는다**

원문·교정문으로 안전하게 돌아가거나 사용자에게 확인한다

8. 기대 효과

무엇이 좋아지나

- 오타와 띄어쓰기 때문에 조회 조건이 빠지는 문제가 줄어든다
- "그중", "그 계좌", "지난번처럼" 같은 꼬리물기 질문을 처리한다
- 이전 조건을 잘못 이어붙이는 오류가 줄어든다
- 근거 없는 조건이나 결과를 만들어내는 문제가 줄어든다

9. 한 줄 정리

질의 정규화 및 분석은 단순한 오타 교정이 아니다.

원문·교정문·완성 질문을 따로 관리하고 이전 대화는 구조로 기억해
후속 질문에서 맞는 값만 가져다 쓴다.

확인할 수 없는 내용은 만들지 않고 꼭 필요한 것만 되묻는다는

원칙 하나가 꼬리물기 질문을 안정적으로 처리하는 핵심이다.