

NL2SQL 슬롯필링 설계

슬롯필링과 되묻기 단계

LHG 2026-08-27

CONTENTS

슬롯필링이란, 다른 서비스 사례

두 가지 슬롯 구분

전체 처리 프로세스와 예시

AI와 코드의 역할, 멀티턴 판정

되묻기 정책과 소유 경계

이 글에서 쓰는 용어

신청서에 빈칸을 채우는 과정이라고 생각하면 쉽다

- **슬롯**: 채워야 하는 한 칸. 계좌, 기간, 금액 같은 값
- **레시피**: 어떤 조회에 어떤 칸이 필요한지 적은 규칙표
- **missing**: 필요한데 아직 안 채워진 칸(required – filled)
- **resolver**: "급여계좌" 같은 별칭을 실제 값으로 찾아주는 기능
- **clarify**: 부족하거나 애매한 값을 사용자에게 묻는 것



1. 슬롯필링이란

빈칸이 있는 조회 신청서를 채우는 과정이다

"거래내역 보여줘"에는 빈칸이 있다

의도: 거래내역 조회

필요한 슬롯: 계좌, 기간

현재 채워진 슬롯: 없음

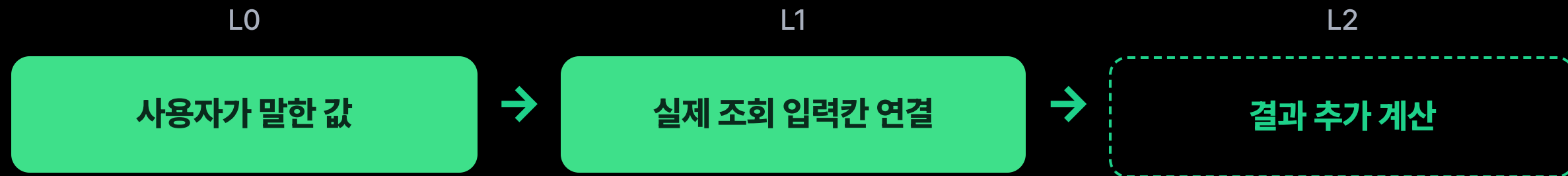
부족한 슬롯: 계좌, 기간

현재 발화 추출 → 이전 대화 재사용 → 후보 자동 조회 → 기본값 적용 → 그래도 부족하면 되묻기

L0, L1, L2로 나눠 생각한다

LHG

사용자 말이 실제 조회와 답변으로 바뀌는 세 단계다



- 슬롯필링의 범위는 L0를 모으고 부족한 값을 채워 어떤 조회를 할지 확정하는 데까지다

2. 다른 서비스들은 몇 번 물어보고 언제 확인 하나

Lex, Dialogflow, Rasa, Copilot Studio를 찾아봤다

재질문 상한은 대체로 2~3번

Amazon Lex

- 0~5회 사이로 설정, 실무는 보통 2~3회
- 초과 시 다른 처리로 넘어간다

Google Dialogflow

- 문구를 바꿔가며 최대 6번까지 재질문
- 그래도 안 되면 기본 처리로

Rasa

- 기본 횟수 제한 없음
- 못 알아들으면 바로 다른 방식으로

Copilot Studio

- 한 발화에서 여러 슬롯을 동시 추출
- 질문 횟수 자체를 줄인다

확인 방식은 신뢰도로 나눈다

LHG

낮은 신뢰도, 따로 확인

- "계좌번호가 1234인가요?"

높은 신뢰도, 슬쩍 확인

- "네, 1234로 확인했습니다. 기간은요?"

- 이 설계의 "전체 되묻기 예산 3턴"도 같은 방향이다



3. 두 가지 슬롯을 구분한다

사용자가 말한 값과 SQL로 해석하는 값은 다르다

L0 업무 슬롯, 사용자가 말한 그대로

- 계좌, 상품군, 기간, 거래방향, 금액조건, 거래내용, 이체조건, 정렬·건수, 회차
- 어려운 내부 필드명이 아니라 대화에서 바로 찾을 수 있는 값으로 관리한다
- 정확한 전체 목록과 검증 규칙은 별도 슬롯 사전이 정본이다

NL2SQL 의미 슬롯, SQL로 해석한 값

- `metrics` (매출) · `dimensions` (고객) · `filters` (서울)
- `timeRange` (올해) · `aggregation` (SUM) · `comparison` (전년 대비)
- `sort` (DESC) · `limit` (5) · `negation` (제외)
- L0 "기간=최근 1개월"은 `timeRange` 로, "거래방향=출금"은 `filters` 로 연결된다

무엇을 조회하나, 어떻게 보여주나

타입, 필요한 슬롯을 정한다

- 계좌 목록 → 계좌 불필요
- 잔액 조회 → 계좌만 필요
- 거래내역 → 계좌와 기간 필요

질문유형, 답변 방식을 정한다

- 단순, 분석, 비교, 추이
- 계좌·기간을 정하는 기준이 아니다
- 비교 기준은 별도 구조로 보존



4. 전체 처리 프로세스

문장을 바로 SQL로 바꾸지 않는다

앞쪽 절반, 발화에서 슬롯까지

LHG

1. 발화 수신
2. 문장 정규화(오타자·동의어·상대 날짜)
3. 새 질문·후속 질문 판정
4. 조회 타입 분류
5. 타입별 레시피 선택
6. 현재 발화에서 슬롯 추출

뒤쪽 절반, 병합에서 실행까지

1. 이전 대화의 호환 슬롯 병합
2. 값 검증과 자동 해소
3. 부족한 슬롯 계산 ($\text{missing} = \text{required} - \text{filled}$)
4. 부족하면 질문, 다 채워지면 완성 질의 생성
5. NL2SQL이 의미 구조로 변환
6. SQL 생성, 안전 검증, 실행



5. 실제로 어떻게 동작하나

예시 네 가지로 살펴본다

필요한 값이 다 들어온 경우

LHG

"급여계좌의 최근 한 달 출금 내역을 금액 큰 순서로 10건 "

```
{  
  "slots": {"account": "급여계좌", "period": "최근 한 달",  
            "direction": "출금", "limit": 10},  
  "missing": [], "status": "COMPLETE"  
}
```

되묻기 없이 바로 완성한다.

필수값이 부족한 경우

LHG

사용자: 거래내역 보여줘

시스템: 어느 계좌의 거래내역을 조회할까요? [급여계좌] [운영계좌]

사용자: 급여계좌

시스템: 조회 기간을 선택해 주세요. [1주일] [1개월] [3개월]

한 번에 여러 값을 묻지 않고 우선순위 높은 것부터 확인한다.

조회 종류가 바뀌면 일부만 이어받는다

"급여계좌 지난달 출금 내역" 다음에 "그 계좌 잔액은?"

이어받음

- 계좌 = 급여계좌

이어받지 않음

- 기간 = 지난달, 거래방향 = 출금

- 전부 이어받으면 잔액 조회에 불필요한 조건이 SQL에 섞여 들어간다

후보가 여러 개인 경우

LHG

"운영계좌 잔액 알려줘" → 후보: 연구운영계좌, 사업운영계좌

```
{
  "kind": "clarify",
  "text": "어느 운영계좌를 조회할까요?",
  "choices": [{ "id": "ACC_01", "label": "연구운영계좌" },
               { "id": "ACC_02", "label": "사업운영계좌" } ]
}
```

AI가 후보 중 하나를 임의로 고르면 안 된다.



6. 슬롯필링과 NL2SQL의 책임 경계

각자 맡는 일이 다르다

슬롯필링

- 업무 조건 수집
- 이전 조건의 선택적 승계
- 부족·모호한 값 확인
- 완성 질의 생성

NL2SQL

- 완성 질의를 SQL 구조로 변환
- 스키마·컬럼·조인 결정
- SQL 생성과 안전 검증
- 조회 결과 생성



7. AI가 판단할 일, 코드가 확정할 일

의미 판단과 구조 판정을 분리한다

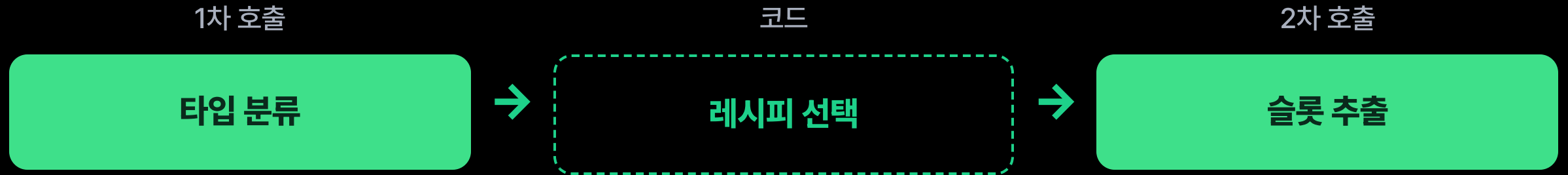
필요한 값 {계좌, 기간} - 채워진 값 {기간, 거래방향} = 부족한
값 {계좌}

- AI: 타입·질문유형 분류, 슬롯값 추출, 후보 충돌 해소, 되묻기 문장 표현
- 코드: 레시피 선택, 값 검증, 병합, 후보 조회, 부족한 값 계산
- AI는 "무엇이 부족한가"를 결정하지 않는다. 코드가 집합 차이로 계산한다

판단을 두 번에 나눠 한다

LHG

작은 모델도 한 번에 하나의 판단에 집중하게 한다



- 전체 규칙과 예시를 매번 다 보내지 않고 목적에 맞는 부분만 전달한다



8. 멀티턴, 이어지는 질문인지 먼저 판단한다

새 질문인지 후속 질문인지부터 정한다

연속성 값 네 가지

- **NEW**: 이전 요청과 독립된 새 질문
- **CONTINUE**: 이전 요청의 일부 조건을 이어가는 질문
- **UNCERTAIN**: 독립·후속 여부가 불명확해서 확인이 필요
- **CLARIFICATION_REPLY**: 직전에 시스템이 물은 것에 대한 답변

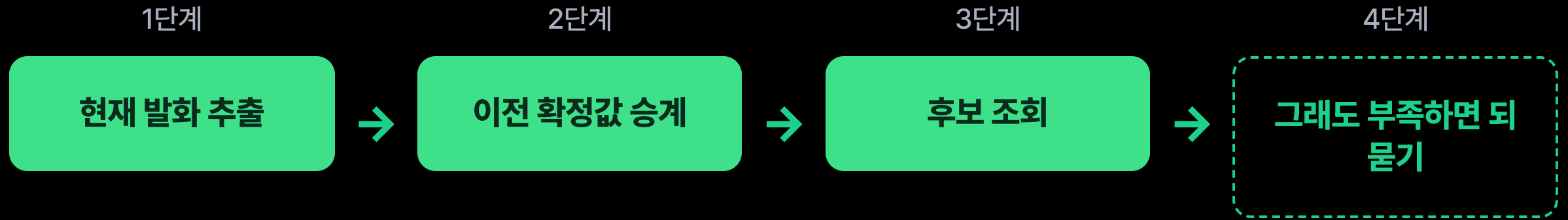
CONTINUE라고 해서 채워진 슬롯을 전부 이어받으면 위험하다.
새 타입이 허용하는 슬롯만 선택적으로 이어받는다.

9. 먼저 자동으로 찾고 정말 필요할 때만 묻는다

되묻기는 최후 수단이다

되묻기 전에 거치는 단계

LHG



- 후보 1건은 자동 확정, 소수는 선택지, 다수는 검색형, 없으면 재입력 요청
- 초기 정책은 전부 표시 3건, 선택지 5건, 되묻기 예산 3턴

값 확인과 실행 승인은 다르다

- 되묻기는 부족한 값을 채우거나 후보 중 하나를 고르기 위한 절차다
- 실행을 승인받는 절차가 아니다
- 선택지 값은 SI가 새로 만들지 않고 후보 조회 결과나 등록된 값에서 만든다



10. Admin, Agent, Front의 소유 경계

판단이 여러 곳에 있으면 정본이 갈라진다

누가 무엇을 갖고 있나

LHG

- **관리 화면**: 슬롯·타입·레시파·되묻기 정책 등록과 편집
- **관리 API**: 저장, 검증, 버전 발행, 설정 묶음 제공
- **Agent(코드)**: 정규화, 분류, 추출, 병합, 검증, 부족한 값 계산
- **Front**: 발화 전달, 선택지 화면 결과 렌더링. 필수값 검사는 하지 않는다

정리

- 슬롯필링은 빈칸이 있는 조회 신청서를 채우는 과정이다
- 사용자가 말한 값(L0)과 SQL로 해석하는 값은 서로 다른 구조로 연결만 한다
- 되묻기는 최후 수단이다. 먼저 자동으로 찾고 정말 부족할 때만 묻는다
- 다른 서비스들의 관행(재질문 2~3회, 신뢰도별 확인)을 코드 규칙으로 명시한다