#### 쿼리 디자이너

쿼리 디자이너는 사용자가 로그프레소 쿼리에 대해 깊은 지식이 없어도 특정한 작업을 수행할 수 있도록 해주는 쿼리 작성 도구입니다. 쿼리문을 크게 데이터 조회, 데이터 가공, 최종적으로 데이터를 텍스트 파일, 데이터베이스, 로그프레소 테이블에 기록하는 형태로 구분할 수 있습니다. 쿼리 디자이너를 통해 작성할 수 있는 스니펫 커맨드를 유형에 따라 구분해보면 다음과 같습니다.

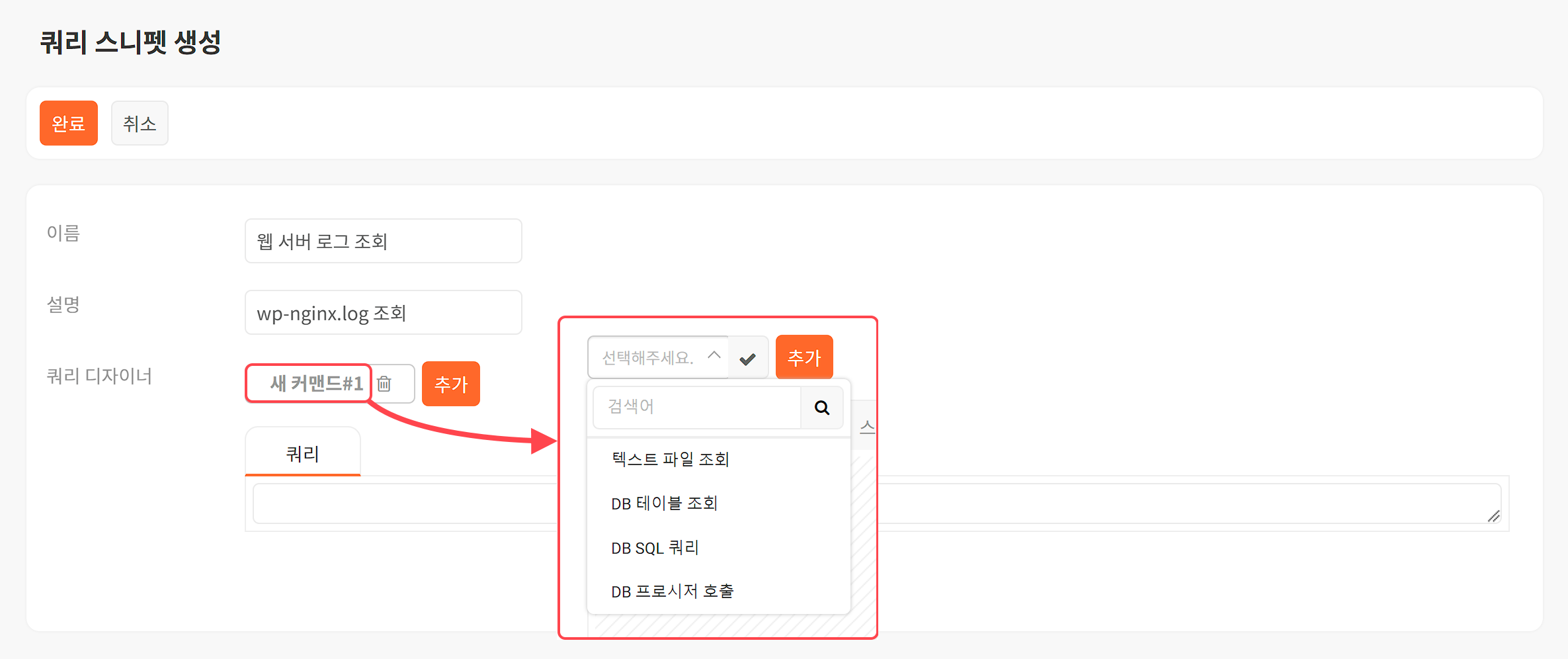
|  |  |  |
| --- | --- | --- |
| 작업 유형 | 스니펫 커맨드 | 관련 쿼리 명령 |
| 데이터 조회 | [텍스트 파일 조회](https://docs.logpresso.comnull) | [textfile](https://docs.logpresso.comnull) |
| 데이터 조회 | [DB 테이블 조회](https://docs.logpresso.comnull) | [dbquery](https://docs.logpresso.comnull) |
| 데이터 조회 | [DB SQL 쿼리](https://docs.logpresso.comnull) | [dbquery](https://docs.logpresso.comnull) |
| 데이터 조회 | [DB 프로시저 호출](https://docs.logpresso.comnull) | [dbcall](https://docs.logpresso.comnull) |
| 데이터 가공 | [조건 검색](https://docs.logpresso.comnull) | [search](https://docs.logpresso.comnull) |
| 데이터 가공 | [필드 할당](https://docs.logpresso.comnull) | [eval](https://docs.logpresso.comnull) |
| 데이터 가공 | [쿼리 매개변수 할당](https://docs.logpresso.comnull) | [evalc](https://docs.logpresso.comnull) |
| 데이터 적재 | [텍스트 파일 적재](https://docs.logpresso.comnull) | [outputtxt](https://docs.logpresso.comnull) |
| 데이터 적재 | [DB 테이블 적재](https://docs.logpresso.comnull) | [dboutput](https://docs.logpresso.comnull) |
| 데이터 적재 | [테이블 적재](https://docs.logpresso.comnull) | [import](https://docs.logpresso.comnull) |

사용법

쿼리 디자이너는 쿼리 [스니펫을 추가](https://docs.logpresso.comnull), [수정](https://docs.logpresso.comnull)하는 작업에서 이용할 수 있습니다.

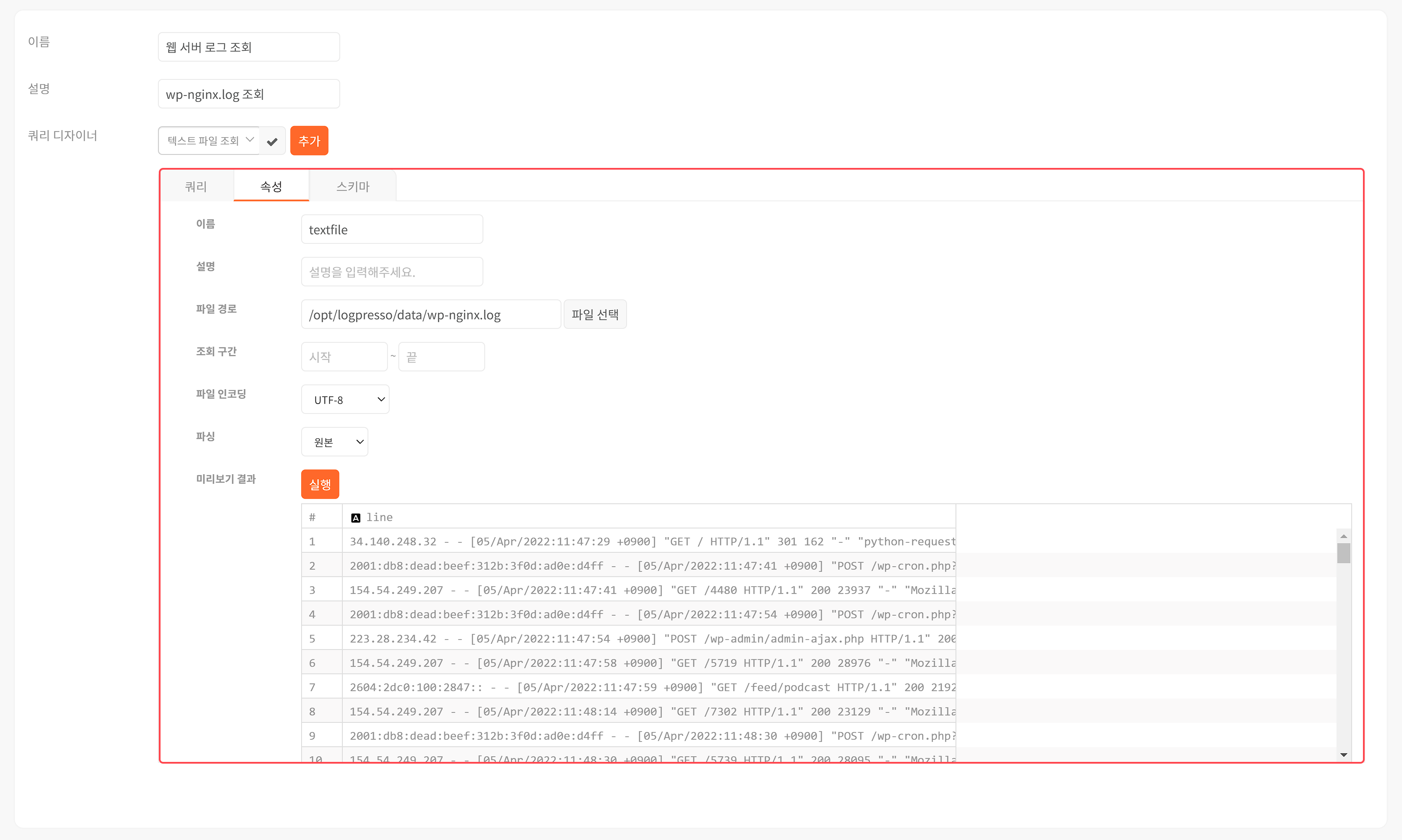
첫 번째 쿼리 커맨드

**쿼리 디자이너**에서 **새 커맨드 #1**을 눌러 [쿼리 스니펫 유형](https://docs.logpresso.comnull)을 선택합니다. 첫 커맨드는 데이터 조회 유형의 커맨드 중에서 선택합니다.



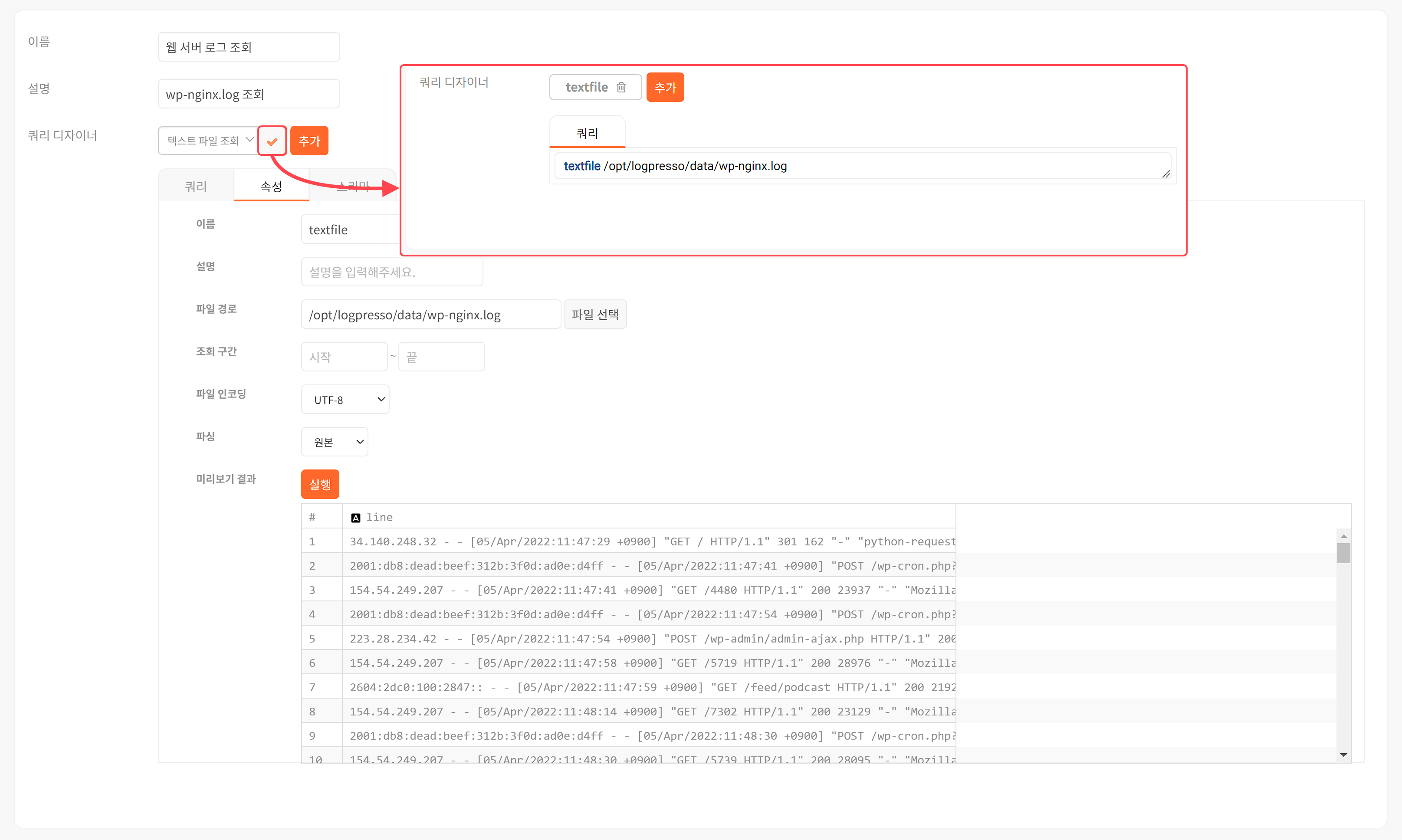
* **텍스트 파일 조회**: 로그프레소 서버에 있는 텍스트 파일을 조회
* **DB 테이블 조회**: DBMS에 저장된 테이블 데이터를 조회(접속 프로파일 필요)
* **DB SQL 쿼리**: DBMS에 저장된 데이터를 SQL 쿼리문으로 조회(접속 프로파일 필요)
* **DB 프로시저 호출**: DBMS에 저장된 프로시저를 호출해 결과값을 조회(접속 프로파일 필요)

커맨드 유형을 선택하면 각 커맨드별로 설정 항목들이 표시됩니다. 필요한대로 설정을 입력합니다. 설정 항목은 [스니펫 커맨드 유형](https://docs.logpresso.comnull)을 참조하십시오.



* **쿼리** 탭을 누르면 속성에 입력된 값을 이용해 작성된 쿼리문을 보여줍니다.
* **속성** 탭 패널에서 쿼리문 작성에 필요한 속성을 입력합니다.
* **스키마** 탭은 입력 데이터와 출력 데이터 필드 정보를 보여줍니다.

작업 중인 커맨드 항목의 오른쪽에 있는 체크 표시를 누르면 첫번째 쿼리 수식의 편집이 종료되고, 작성된 쿼리 스니펫을 볼 수 있습니다.

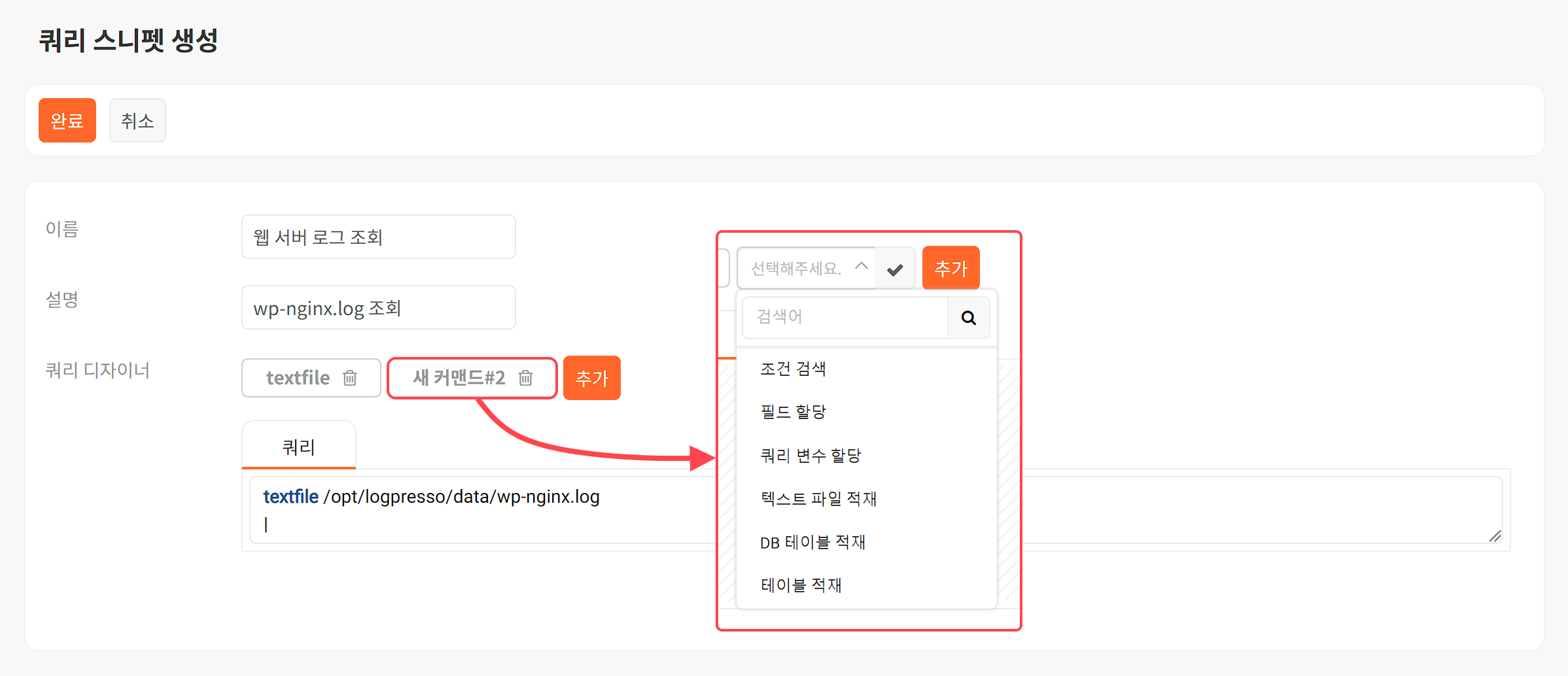


첫 번째 이후의 쿼리 커맨드

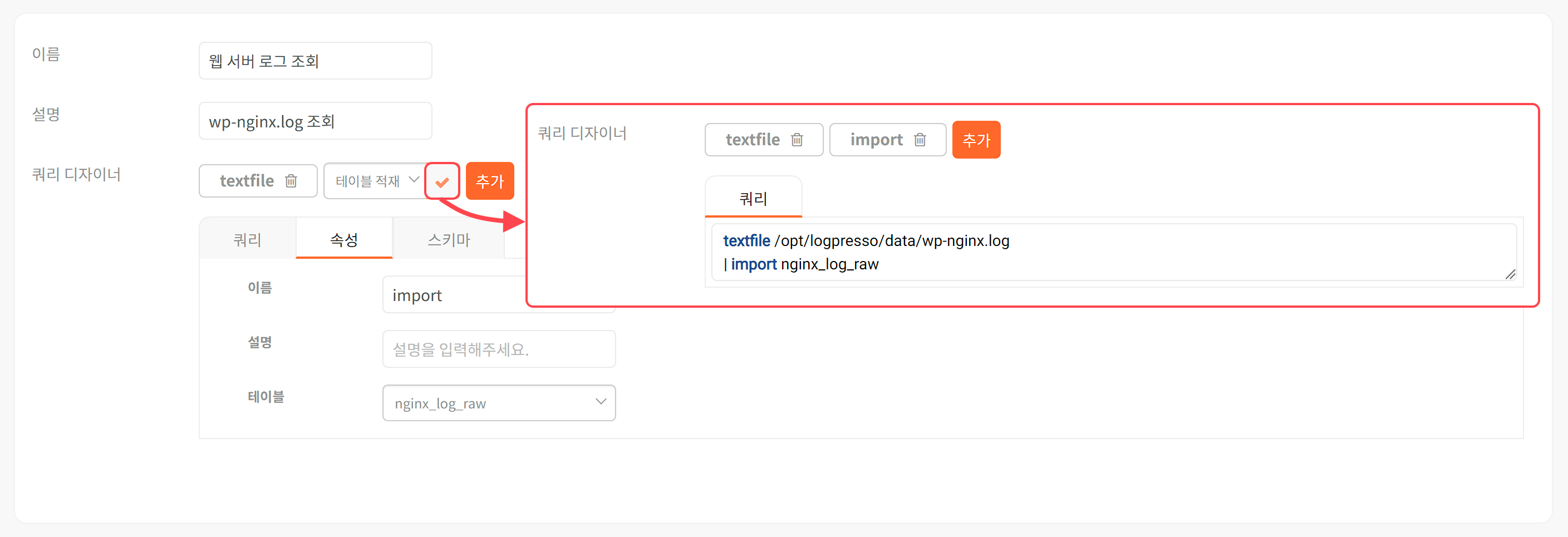
데이터 조회 커맨드에 이어 데이터를 가공하는 두 번째 쿼리 수식을 작성할 수 있습니다.

**쿼리 디자이너**에서 **추가**를 누릅니다.

**새 커맨드 #2**를 눌러 커맨드 유형을 선택합니다.

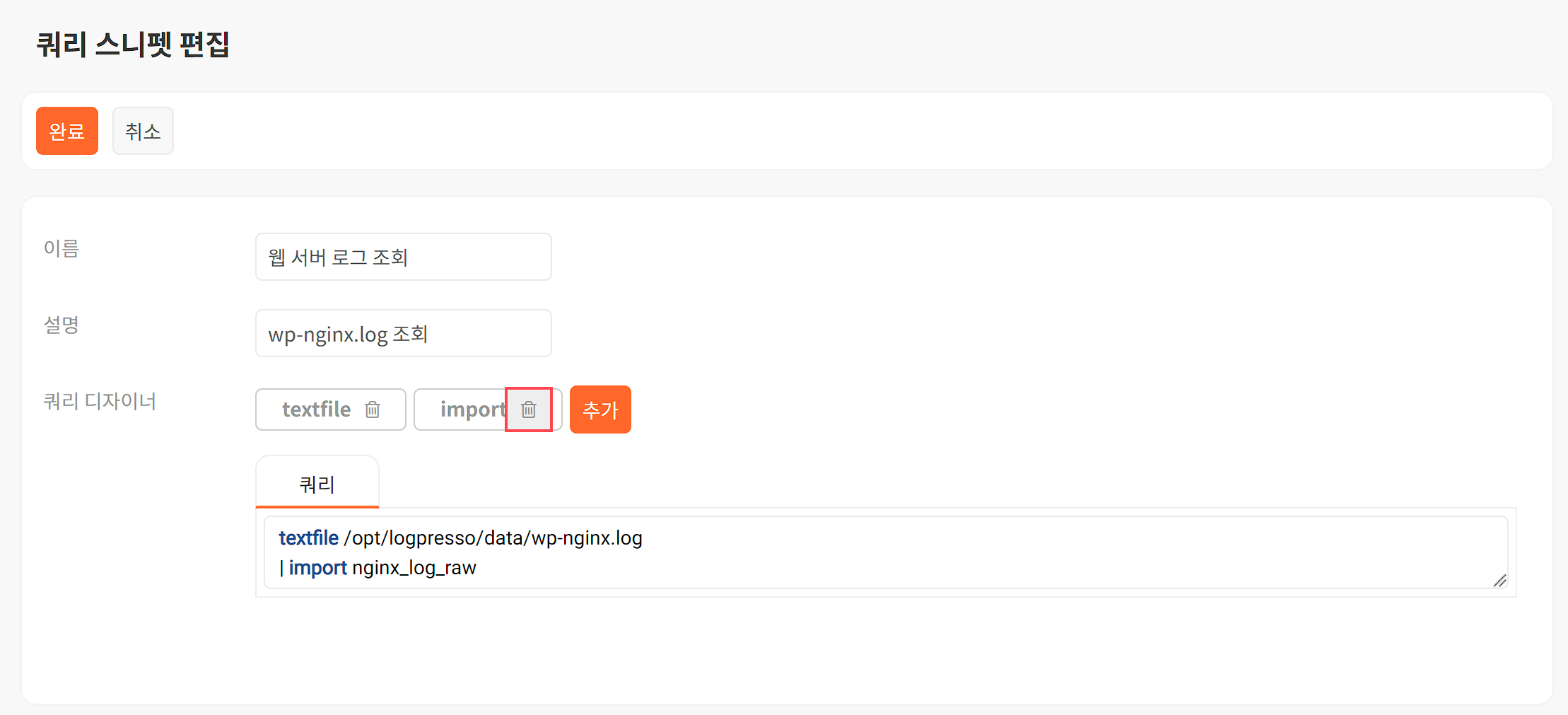


* 데이터 가공 스니펫
* **조건 검색**: 특정 조건에 맞는 데이터를 추출
* **필드 할당**: 값을 필드에 할당
* **쿼리 변수 할당**: 값을 쿼리 변수에 할당
* 데이터 적재 스니펫
* **텍스트 파일 적재**: 쿼리 결과를 파일에 기록
* **DB 테이블 적재**: 쿼리 결과를 DBMS에 기록
* **테이블 적재**: 쿼리 결과를 테이블에 기록

작업 중인 커맨드 항목의 오른쪽에 있는 체크 표시를 누르면 쿼리 수식의 편집이 종료되고, 작성된 쿼리 스니펫을 볼 수 있습니다.

쿼리 커맨드 삭제

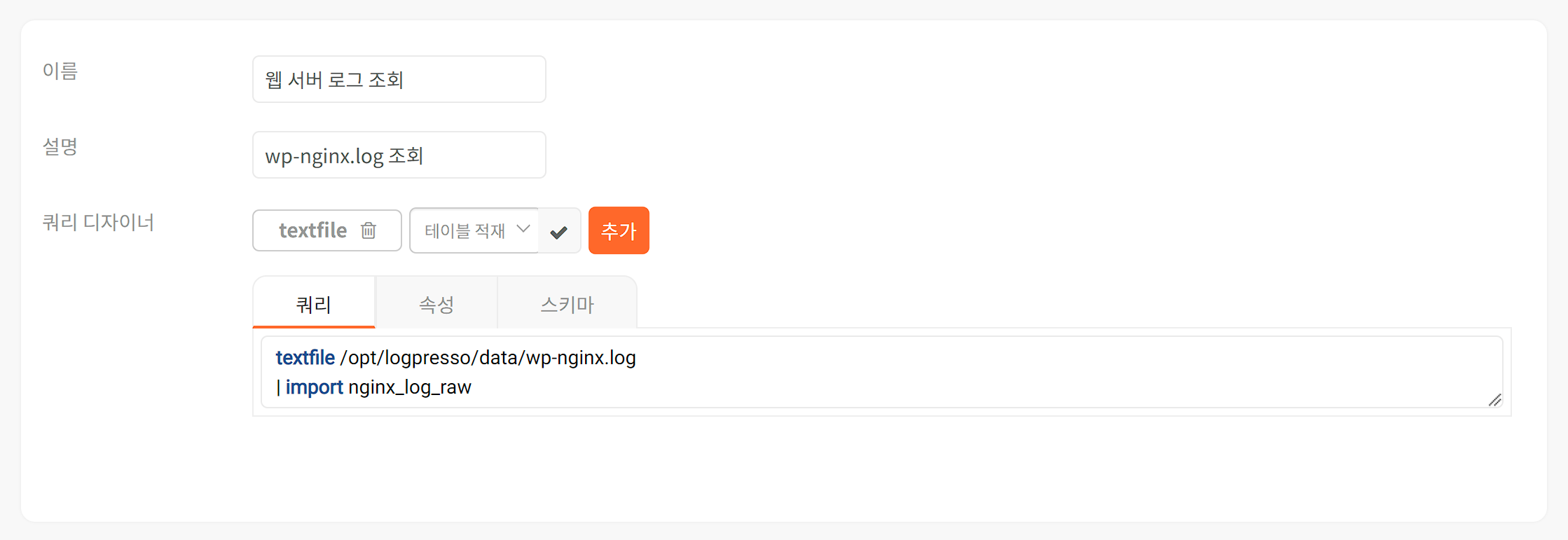
쿼리 디자이너를 이용해 추가한 스니펫 커맨드를 삭제하려면 추가한 커맨드 항목에 표시된 휴지통 버튼을 누르십시오.



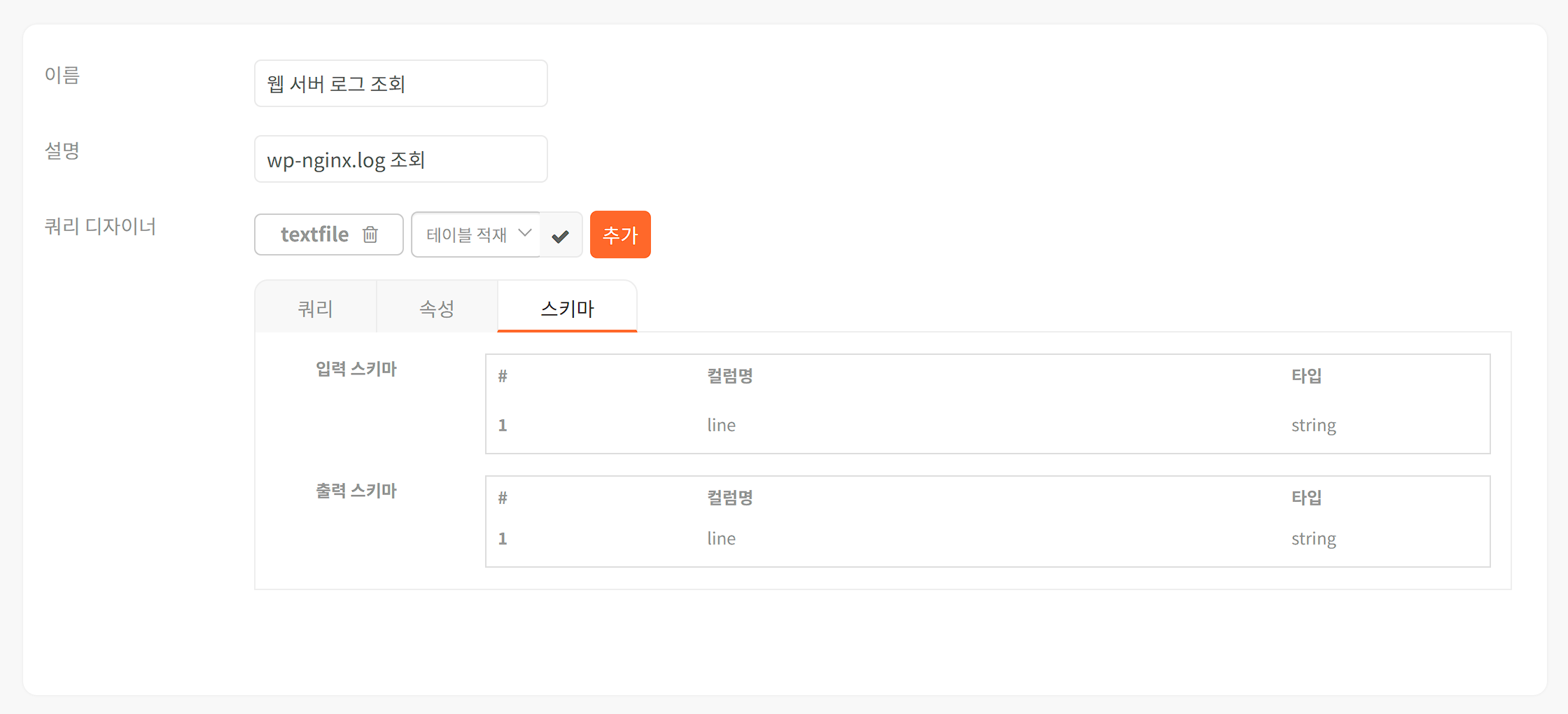
쿼리문과 스키마 확인

각 쿼리 수식의 스키마를 확인하고 전체 쿼리문을 조회해서 원하는대로 쿼리문이 작성되고 있는지 확인하십시오. 쿼리 커맨드 선택 후 스키마 탭을 클릭하면 해당 커맨드 실행 전후 필드 이름 및 타입 목록이 표시됩니다.

**쿼리** 탭을 누르면 지금까지 쿼리 디자이너로 작성한 쿼리문이 표시됩니다.



**스키마** 탭을 누르면 해당 커맨드 실행 전후 필드 이름과 타입 목록이 표시됩니다.

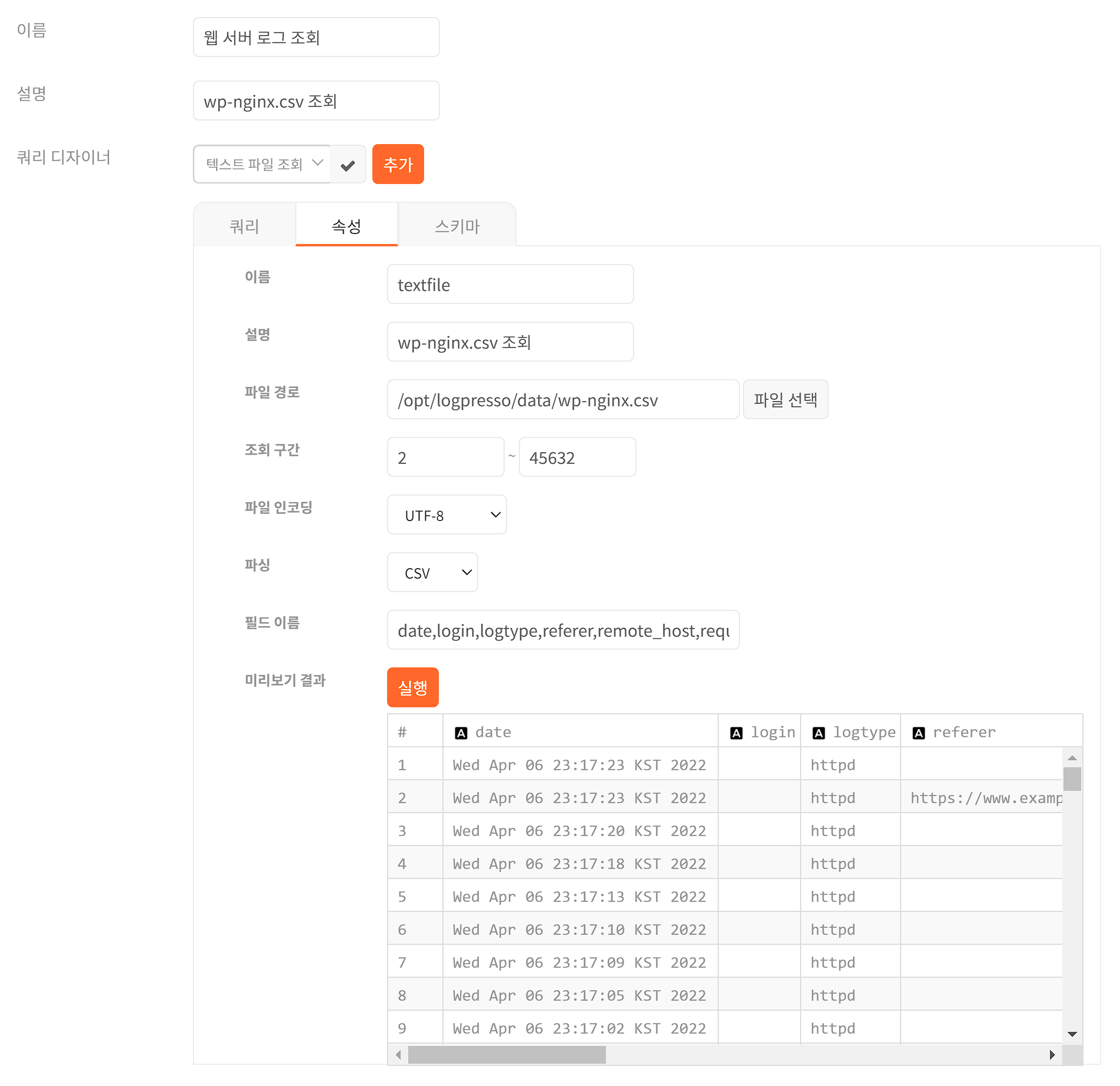


스니펫 커맨드 유형

쿼리 디자이너로 구성할 수 있는 스니펫 커맨드들은 다음과 같습니다.

텍스트 파일 조회

텍스트 파일 조회 스니펫 커맨드는 텍스트 기반 파일에서 내용을 읽어와 파싱합니다.



텍스트 파일 조회 스니펫 커맨드의 속성은 다음과 같습니다.

**이름**

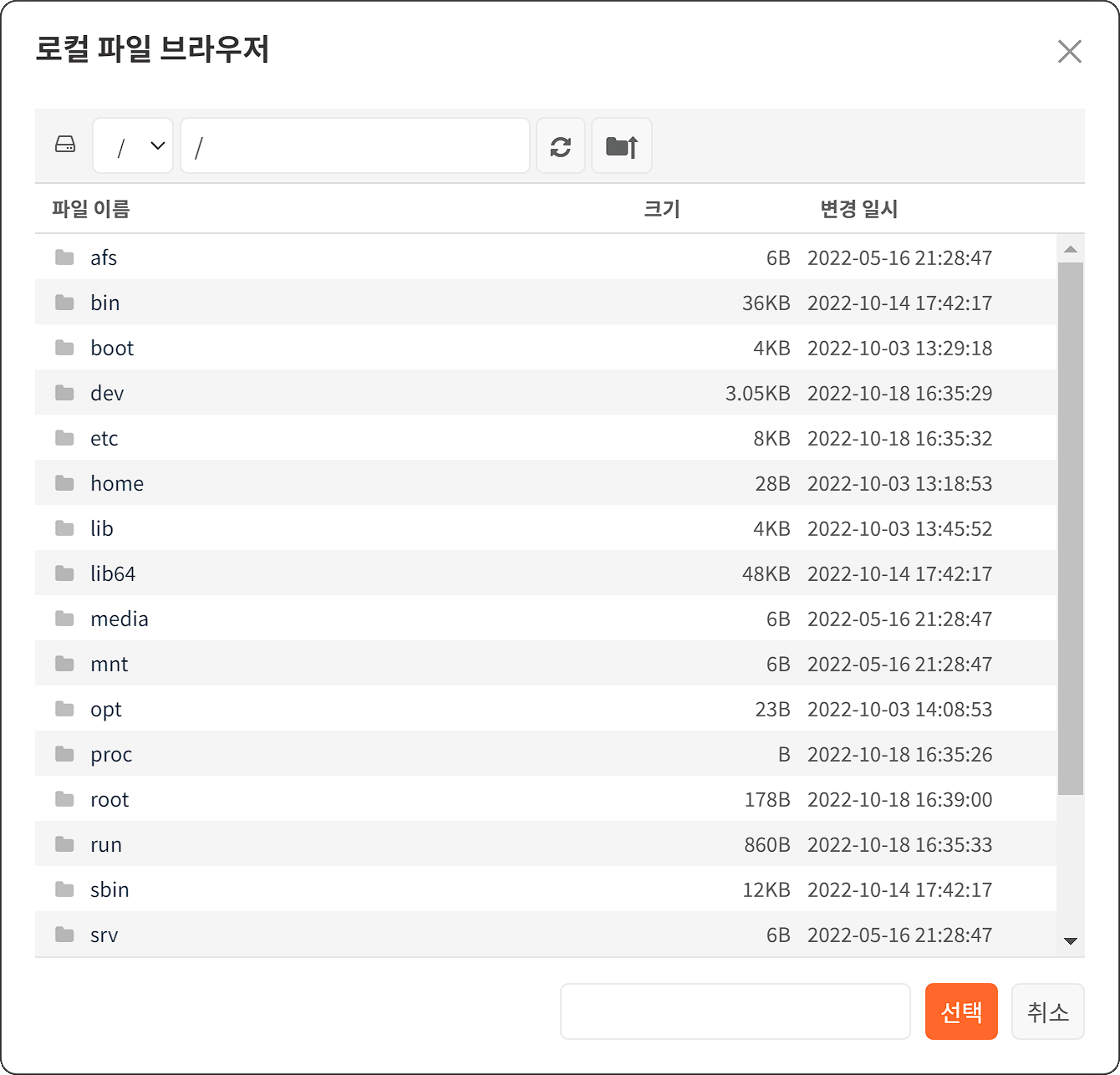
쿼리 커맨드를 식별할 이름

**설명**

쿼리 커맨드에 대한 설명

**파일 경로**

읽어올 파일 경로. **파일 선택**을 누른 후 **로컬 파일 브라우저**에서 원하는 파일을 찾아서 입력할 수 있습니다.



**조회 구간**

읽어올 구간의 줄 번호를 입력합니다.

**파일 인코딩**

파일 인코딩을 **UTF-8**, **확장완성형** 중에서 선택합니다(기본값: UTF-8).

**파싱**

파서를 **원본**, **구분자**, **CSV**, **TSV**, **JSON** 중에서 선택합니다(기본값: 원본).

**원본**

텍스트 원본 1행을 하나의 레코드로 간주해 파싱해 line 필드에 기록합니다.

**구분자**

사용자가 지정한 구분자를 기준으로 필드를 구분하여 레코드를 파싱합니다. 구분자로 사용할 문자와 파싱된 결과를 기록할 필드 이름을 지정해야 합니다.

**CSV**

텍스트 파일을 CSV(Comma-separated values) 형식으로 인식해 레코드를 파싱합니다. 파싱된 결과를 기록할 필드 이름을 지정해야 합니다.

**TSV**

텍스트 파일을 TSV(Tab-separated values) 형식으로 인식해 레코드를 파싱합니다. 파싱된 결과를 기록할 필드 이름을 지정해야 합니다.

**JSON**

텍스트 파일을 JSON 형식으로 인식해 레코드를 파싱합니다.

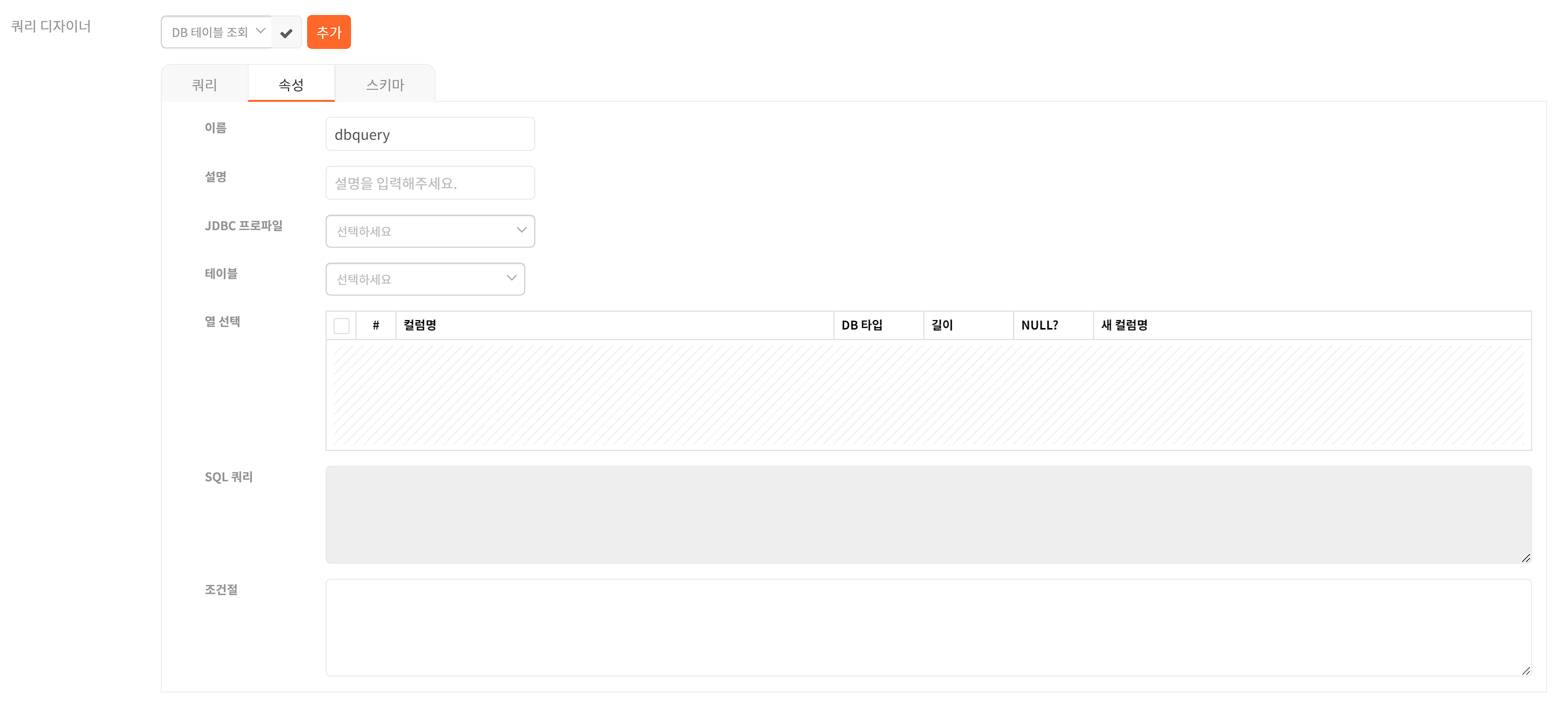
**미리보기 결과**

파일을 조회하고 파싱한 결과를 미리 확인할 수 있습니다.

CSV 파일(https://raw.githubusercontent.com/logpresso/dataset/main/wp-nginx.csv)을 이용해 실습해보십시오.

DB 테이블 조회

DB 테이블 조회 스니펫 커맨드는 DBMS 테이블에 저장된 내용을 읽어와 파싱합니다.



DB 테이블 조회 스니펫 커맨드의 속성은 다음과 같습니다.

**이름**

쿼리 커맨드를 식별할 이름

**설명**

쿼리 커맨드에 대한 설명

**JDBC 프로파일**

DBMS에 접속할 때 사용할 접속 프로파일을 입력합니다.

**테이블**

JDBC 프로파일을 지정하면 자동으로 테이블 목록을 가져와 보여줍니다. 조회할 테이블을 선택합니다.

**열 선택**

DBC 프로파일을 지정하면 자동으로 테이블의 열 목록을 가져와 보여줍니다. 조회할 열 목록을 선택합니다.

**SQL 쿼리**

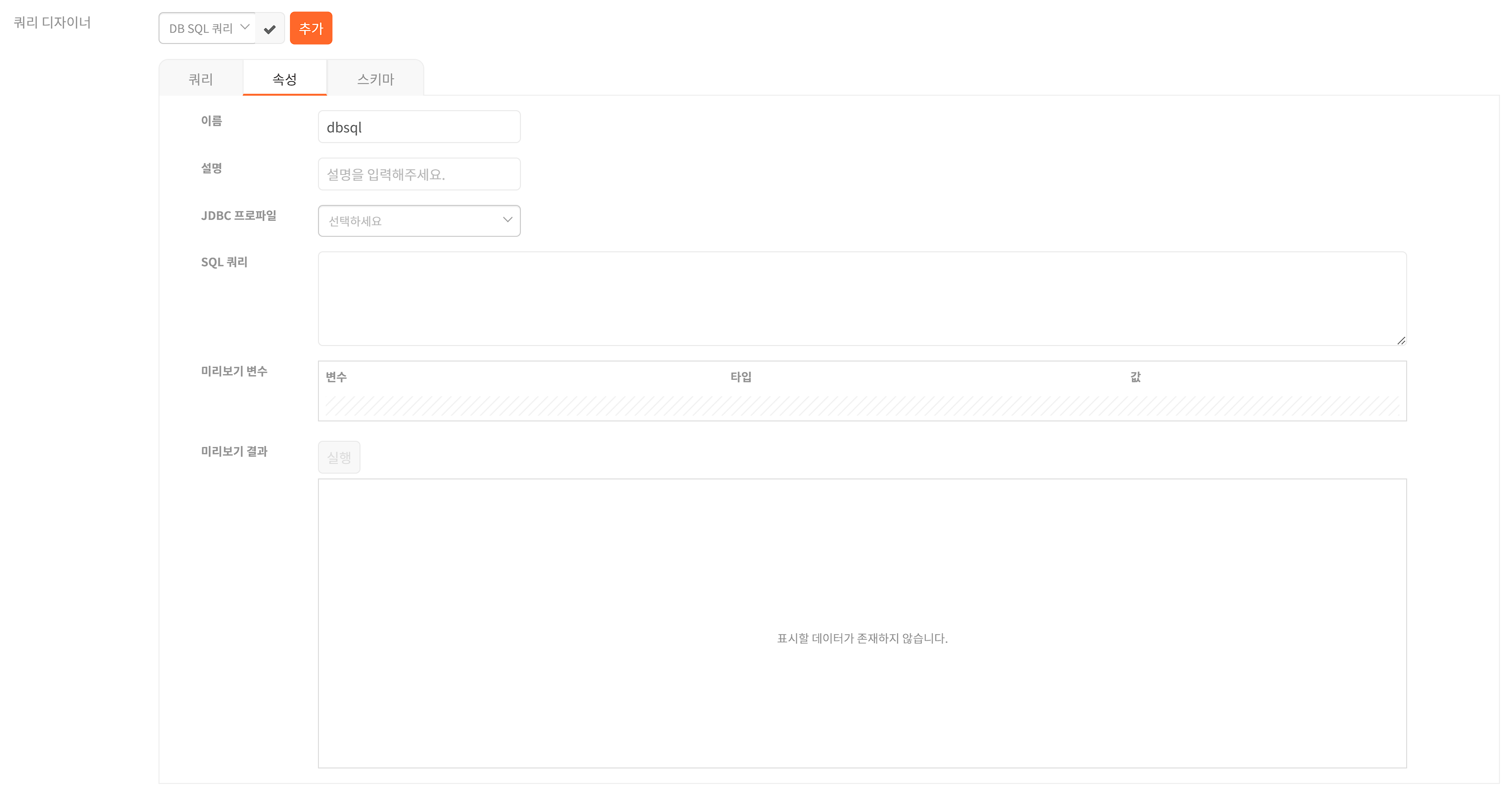
테이블 이름, 열 목록을 기반으로 생성된 SELECT SQL 쿼리문을 보여줍니다.

**조건절**

추가 조회 조건을 SQL 쿼리 형식으로 작성합니다.

DB SQL 쿼리

DB SQL 쿼리 스니펫 커맨드는 사용자가 직접 입력한 SQL 쿼리문을 실행하여 DBMS에 저장된 데이터를 읽어올 수 있는 커맨드입니다. 입력할 수 있는 쿼리는 SELECT 문만 가능합니다.



DB SQL 쿼리 스니펫 커맨드의 속성은 다음과 같습니다.

**이름**

쿼리 커맨드를 식별할 이름

**설명**

쿼리 커맨드에 대한 설명

**JDBC 프로파일**

DBMS에 접속할 때 사용할 접속 프로파일을 입력합니다.

**SQL 쿼리**

데이터를 조회할 SQL 쿼리문을 작성합니다. SQL 쿼리문에 콜론(:)으로 시작하는 입력 매개변수와 출력 매개변수를 정의할 수 있습니다.

* 입력 매개변수는 :name 형식으로 작성합니다. [set](https://docs.logpresso.comnull) 명령으로 정의한 쿼리 매개변수가 삽입됩니다.
* 출력 매개변수는 :name(type) 형식으로 작성합니다. 사용할 수 있는 출력 매개변수 타입은 문자열(varchar), 정수(int), 날짜(datetime)가 있습니다.

**미리보기 변수**

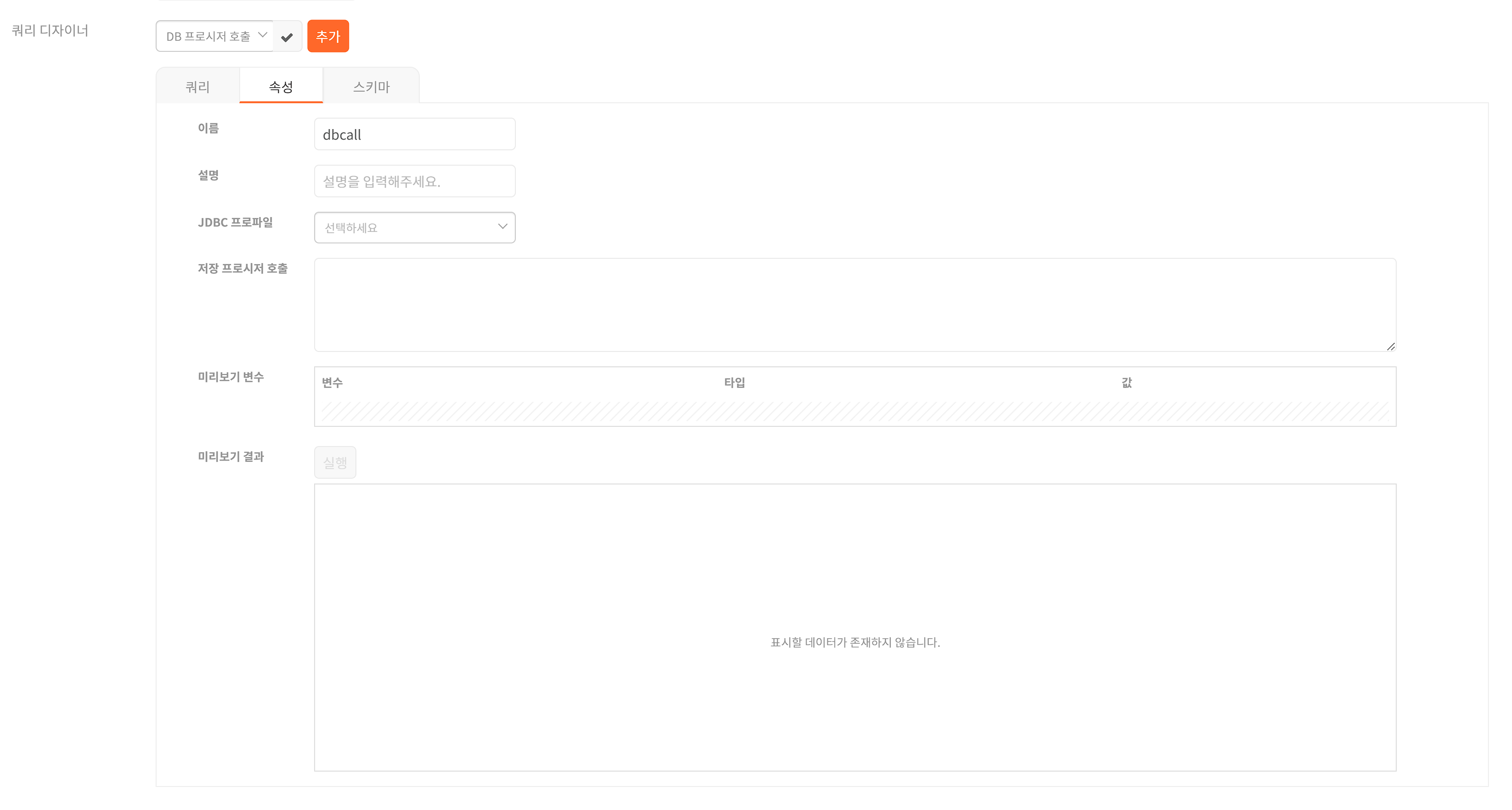
SQL 쿼리문에 있는 매개변수를 인식해서 표시합니다. 입력 매개변수는 :name 형식으로 작성합니다. 매개변수 값을 입력할 수 있습니다.

**미리보기 결과**

SQL 쿼리를 실행하여 결과를 미리 확인할 수 있습니다.

DB 프로시저 호출

DB 프로시저 호출 스니펫 커맨드는 DBMS에 저장된 프로시저를 호출하여 결과를 읽어옵니다.



DB 프로시저 호출 스니펫 커맨드의 속성은 다음과 같습니다.

**이름**

쿼리 커맨드를 식별할 이름

**설명**

쿼리 커맨드에 대한 설명

**JDBC 프로파일**

DBMS에 접속할 때 사용할 접속 프로파일을 입력합니다.

**저장 프로시저 호출**

프로시저를 호출하는 SQL 쿼리문을 작성합니다. SQL 쿼리문에 콜론(:)으로 시작하는 입력 매개변수와 출력 매개변수를 정의할 수 있습니다.

* 입력 매개변수는 :name 형식으로 작성합니다. [set](https://docs.logpresso.comnull) 명령으로 정의한 쿼리 매개변수가 삽입됩니다.
* 출력 매개변수는 :name(type) 형식으로 작성합니다. 사용할 수 있는 출력 매개변수 타입은 문자열(varchar), 정수(int), 날짜(datetime)가 있습니다.

**미리보기 변수**

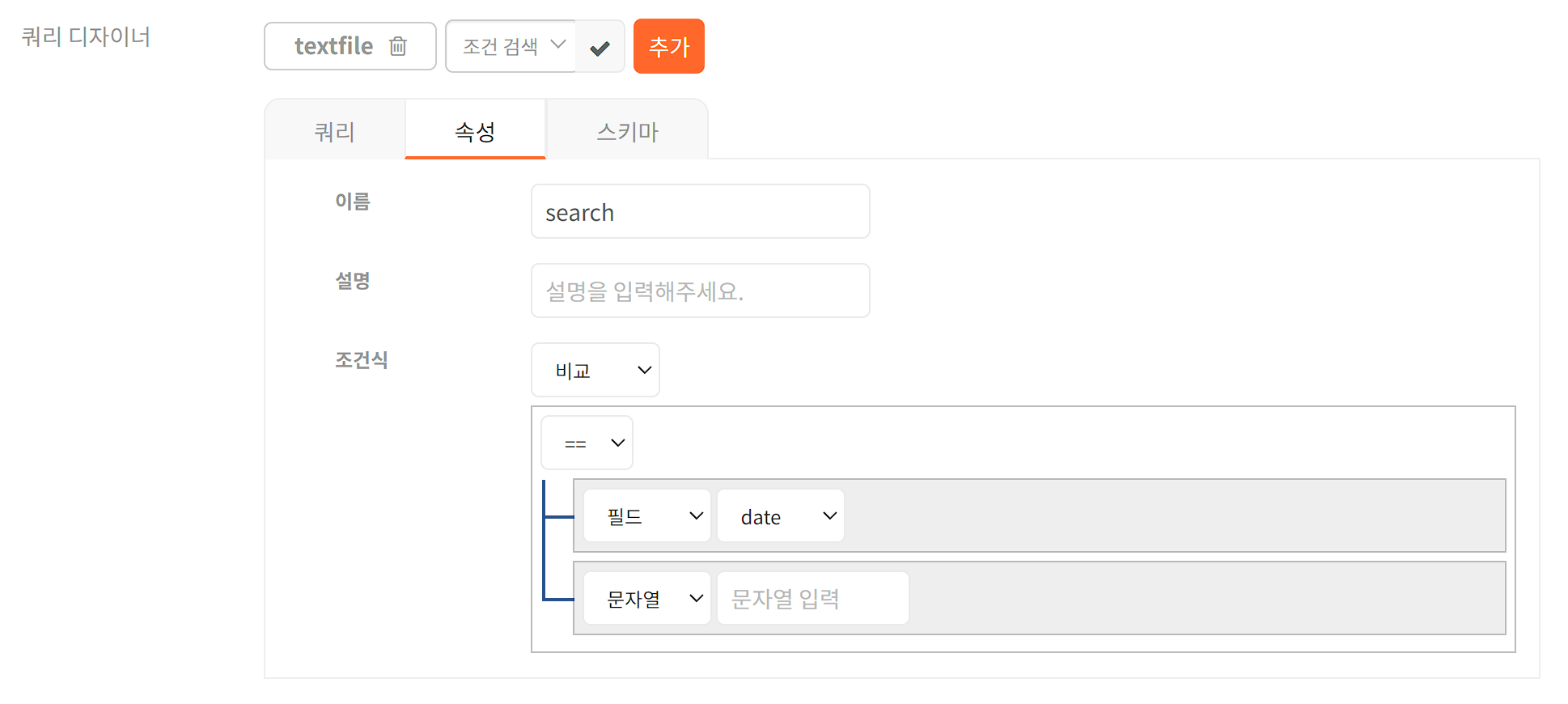
프로시저를 호출하는 SQL 쿼리문에 있는 매개변수를 인식해서 표시합니다. 입력 매개변수는 :name 형식으로 작성합니다. 매개변수 값을 입력할 수 있습니다.

**미리보기 결과**

프로시저를 호출하는 SQL 쿼리문을 실행하여 결과를 미리 확인할 수 있습니다.

조건 검색

조건 검색 스니펫 커맨드는 특정한 조건과 일치하는 레코드만 검색해서 결과를 반환합니다.



조건 검색 스니펫 커맨드의 속성은 다음과 같습니다.

**이름**

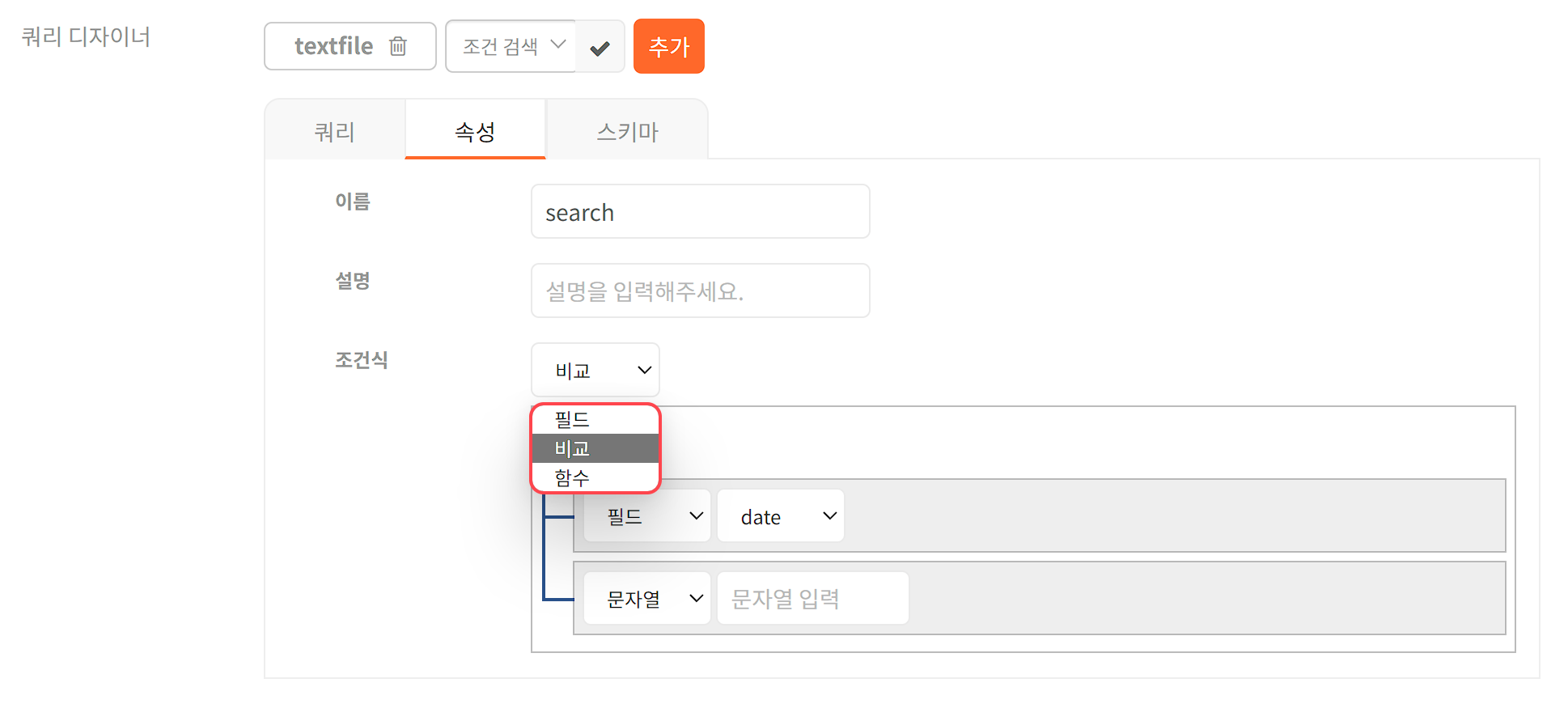
쿼리 커맨드를 식별할 이름

**설명**

쿼리 커맨드에 대한 설명

**조건식**

특정 필드 값이 존재하는 레코드를 필터링합니다. 해당 필드 값이 null인 경우 출력되지 않습니다. 조건식으로는 **필드**, **비교**, **함수**가 있습니다(기본값: 비교).



**필드**

특정 필드 값이 존재하는 레코드를 검색합니다. 해당 필드 값이 null인 경우 출력되지 않습니다.

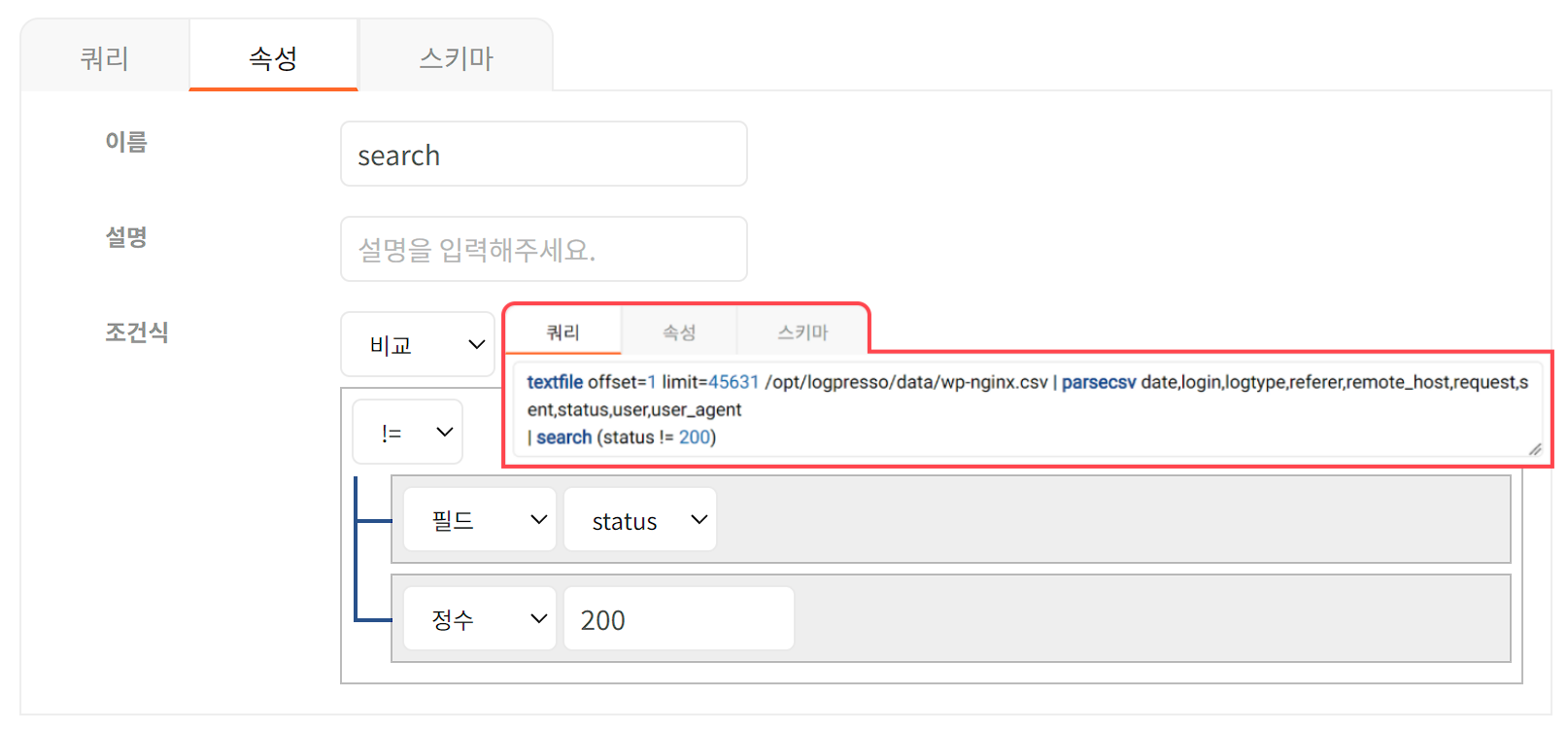
**함수**

함수의 반환값이 포함된 레코드를 검색합니다. 사용할 수 있는 함수는 다음 표를 참조하십시오.

|  |  |
| --- | --- |
| 대상 표현식 | 관련 함수 |
| 32비트 정수 변환 | [int()](https://docs.logpresso.comnull) |
| 64비트 정수 변환 | [long()](https://docs.logpresso.comnull) |
| 64비트 실수 변환 | [double()](https://docs.logpresso.comnull) |
| IP 타입 변환 | [ip()](https://docs.logpresso.comnull) |
| 날짜 타입 변환 | [date()](https://docs.logpresso.comnull) |
| 문자열 포함 여부 | [contains()](https://docs.logpresso.comnull) |
| 문자열 여부 | [isstr()](https://docs.logpresso.comnull) |
| 숫자형 여부 | [isnum()](https://docs.logpresso.comnull) |
| NULL 여부 | [isnull()](https://docs.logpresso.comnull) |
| NOT NULL 여부 | [isnotnull()](https://docs.logpresso.comnull) |
| NULL 기본값 | [nvl()](https://docs.logpresso.comnull) |
| GUID 생성 | [guid()](https://docs.logpresso.comnull) |
| NULL 여부 | [contains()](https://docs.logpresso.comnull) |
| 문자열 길이 | [len()](https://docs.logpresso.comnull) |
| 공백 제거 | [trim()](https://docs.logpresso.comnull) |
| 소문자 변환 | [lower()](https://docs.logpresso.comnull) |
| 대문자 변환 | [upper()](https://docs.logpresso.comnull) |
| 문자열 연결 | [concat()](https://docs.logpresso.comnull) |
| 문자열 변경 | [replace()](https://docs.logpresso.comnull) |
| 왼쪽 부분 문자열 | [left()](https://docs.logpresso.comnull) |
| 오른쪽 부분 문자열 | [right()](https://docs.logpresso.comnull) |
| 부분 문자열 | [substr()](https://docs.logpresso.comnull) |
| 현재 시각 | [now()](https://docs.logpresso.comnull) |
| 이전 시각 | [ago()](https://docs.logpresso.comnull) |
| 날짜 덧셈 | [dateadd()](https://docs.logpresso.comnull) |
| 날짜 차이 | [datediff()](https://docs.logpresso.comnull) |
| 조건 분기 | [if()](https://docs.logpresso.comnull) |

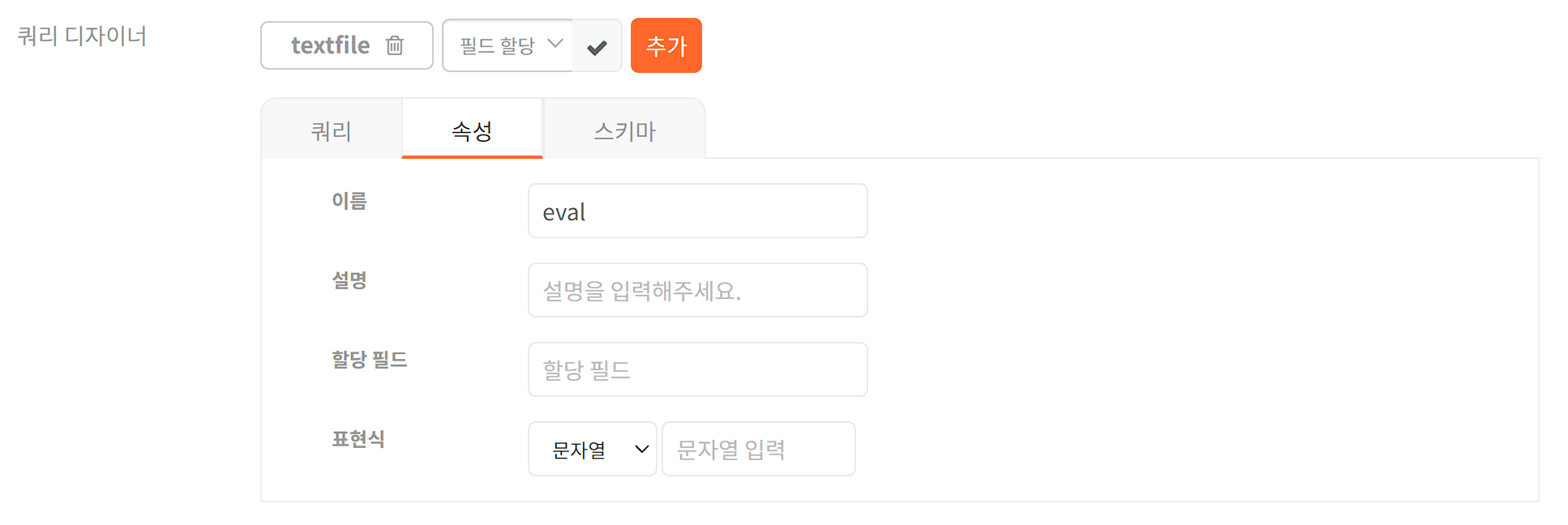
**비교**

두 값 또는 표현식의 비교 결과가 true인 레코드를 검색합니다. 사용할 수 있는 비교 연산자로 ==(같음), !=(다름), <(작음), <=(작거나 같음), >(큼), >=(크거나 같음)이 있습니다. 비교 연산자는 항상 2개의 비교 대상이 필요합니다. 비교할 표현식으로 **필드**, **문자열**, **정수**, **날짜**, **비교**, **함수** 표현식을 사용할 수 있습니다.

* **필드**: 필드의 값을 다른 표현식과 비교합니다.
* **문자열**: 문자열 값을 다른 표현식과 비교합니다. 문자열에 와일드카드 문자(\*)를 사용할 수 있습니다.
* **정수**: 정수 값을 다른 표현식과 비교합니다.
* **날짜**: 날짜 값을 다른 표현식과 비교합니다.
* **비교**: 두 표현식을 비교한 결과를 다른 표현식과 비교합니다. 비교 결과는 true, false 혹은 null입니다.
* **함수**: 함수가 반환하는 값을 다른 표현식과 비교합니다.
* status 필드 값이 200이 아닌 다른 값일 때 참인 비교 조건식은 다음과 같습니다. 빨간 선 안에 있는 쿼리식과 비교해보십시오.
* 

필드 할당

필드 할당 스니펫 커맨드는 새로운 필드에 값을 할당하거나 이미 존재하는 필드에 새 값을 할당합니다.



필드 할당 스니펫 커맨드의 속성은 다음과 같습니다.

**이름**

쿼리 커맨드를 식별할 이름

**설명**

쿼리 커맨드에 대한 설명

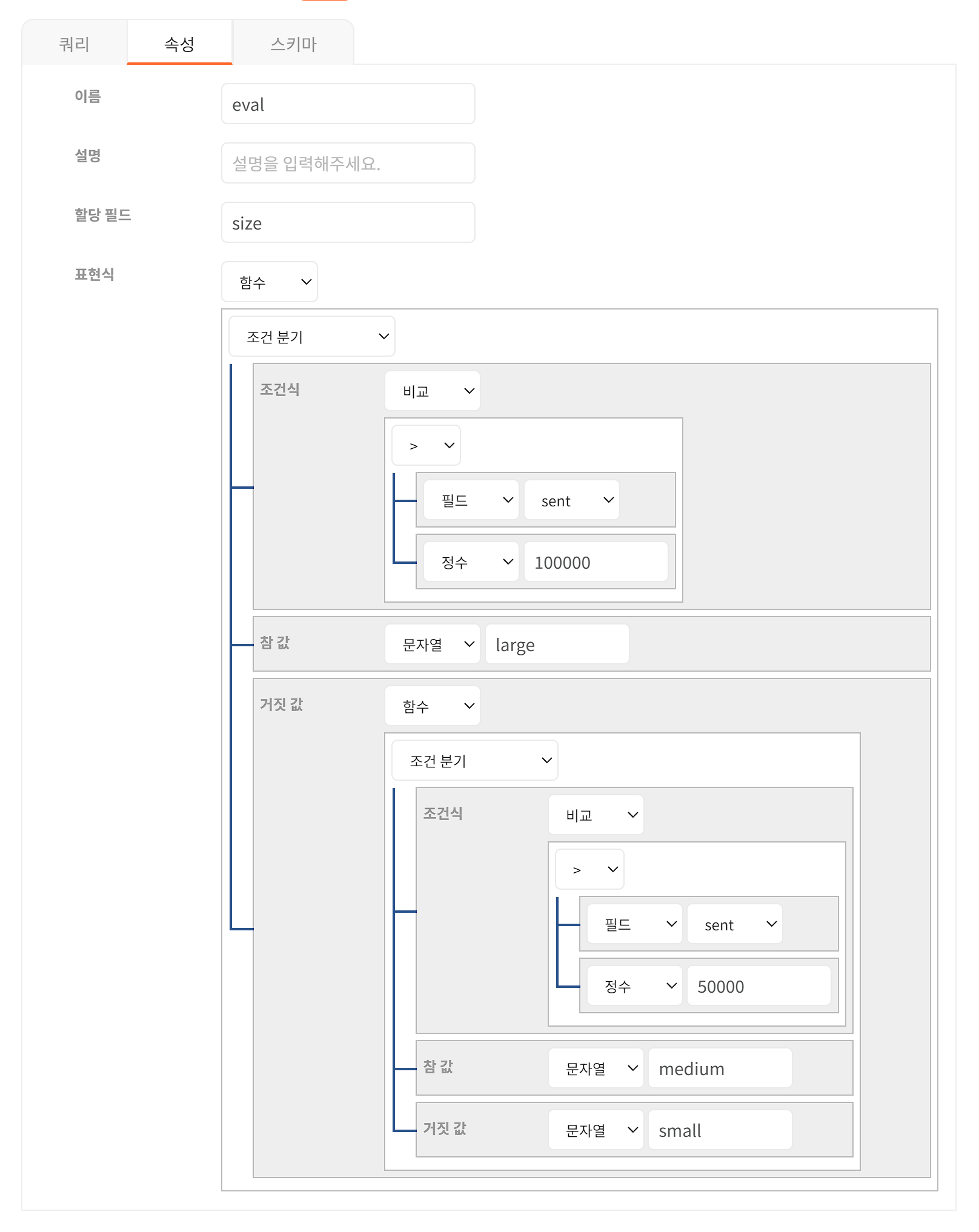
**할당 필드**

값을 할당할 필드 이름

**표현식**

필드에 할당할 값을 반환하는 표현식. 조건식으로는 **필드**, **문자열**, **정수**, **날짜**, **비교**, **함수**가 있습니다(기본값: 문자열). [조건 검색](https://docs.logpresso.comnull) 스니펫 커맨드의 비교를 참조하십시오.

sent 필드 값에 따라 size 필드에 large(100,000보다 클 때), medium(50,000보다 클 때), small(50,0000 보다 작을 때) 문자열을 할당하는 필드 할당 표현식은 다음과 같습니다.



쿼리 변수 할당

쿼리 변수 할당 스니펫 커맨드는 쿼리 매개변수에 값을 할당합니다. 쿼리 매개변수는 쿼리문에서 $("변수이름") 형태로 사용할 수 있습니다.



쿼리 변수 할당 스니펫 커맨드의 속성은 다음과 같습니다.

**이름**

쿼리 커맨드를 식별할 이름

**설명**

쿼리 커맨드에 대한 설명

**쿼리 매개변수**

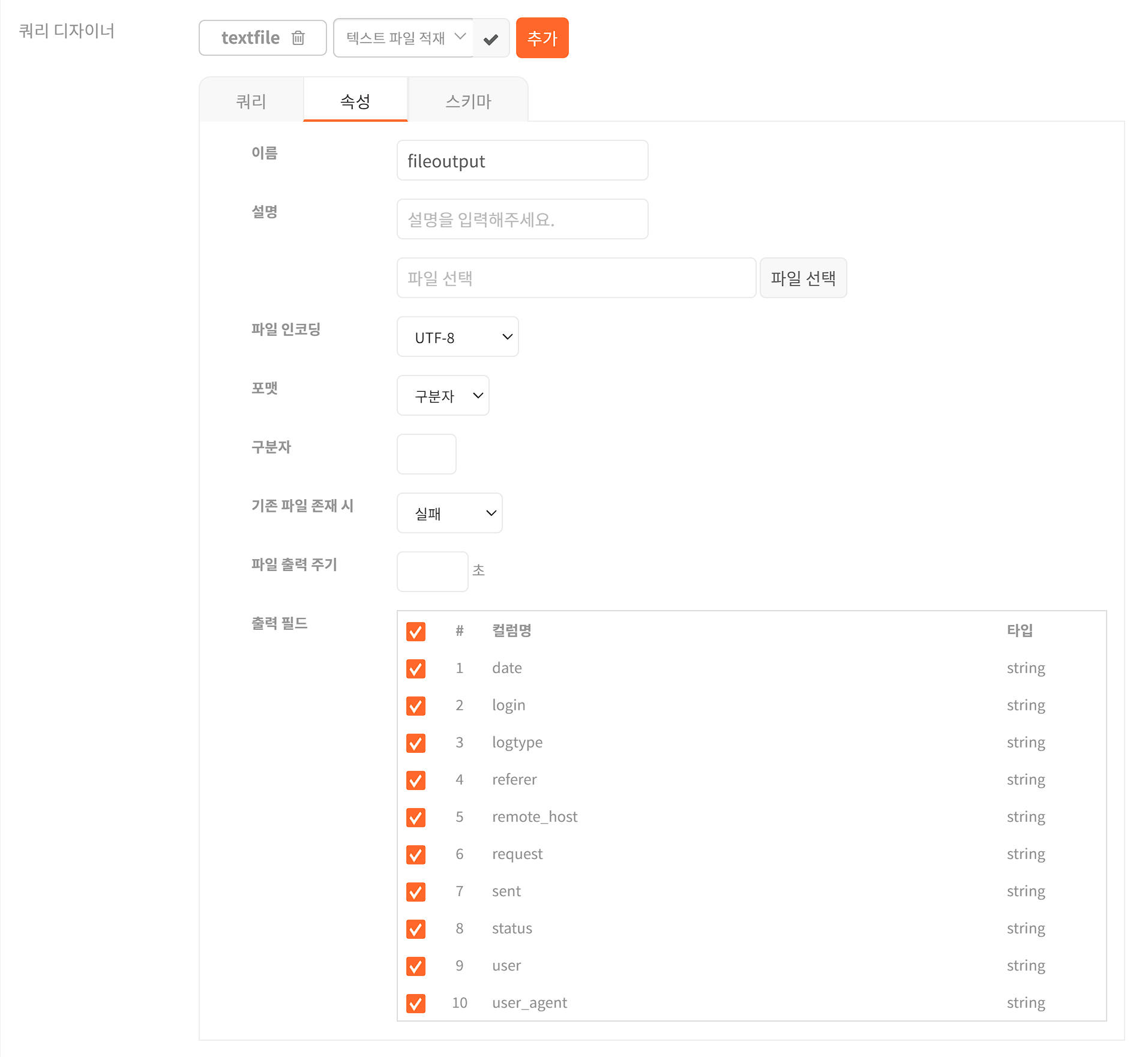
값을 할당할 쿼리 매개변수

**표현식**

쿼리 매개변수에 할당할 값을 반환하는 표현식. 조건식으로는 **필드**, **문자열**, **정수**, **날짜**, **비교**, **함수**가 있습니다(기본값: 문자열). [조건 검색](https://docs.logpresso.comnull) 스니펫 커맨드의 **비교**를 참조하십시오.

텍스트 파일 적재

텍스트 파일 적재 스니펫 커맨드는 쿼리 결과를 텍스트 파일에 저장합니다.



텍스트 파일 적재 스니펫 커맨드의 속성은 다음과 같습니다.

**이름**

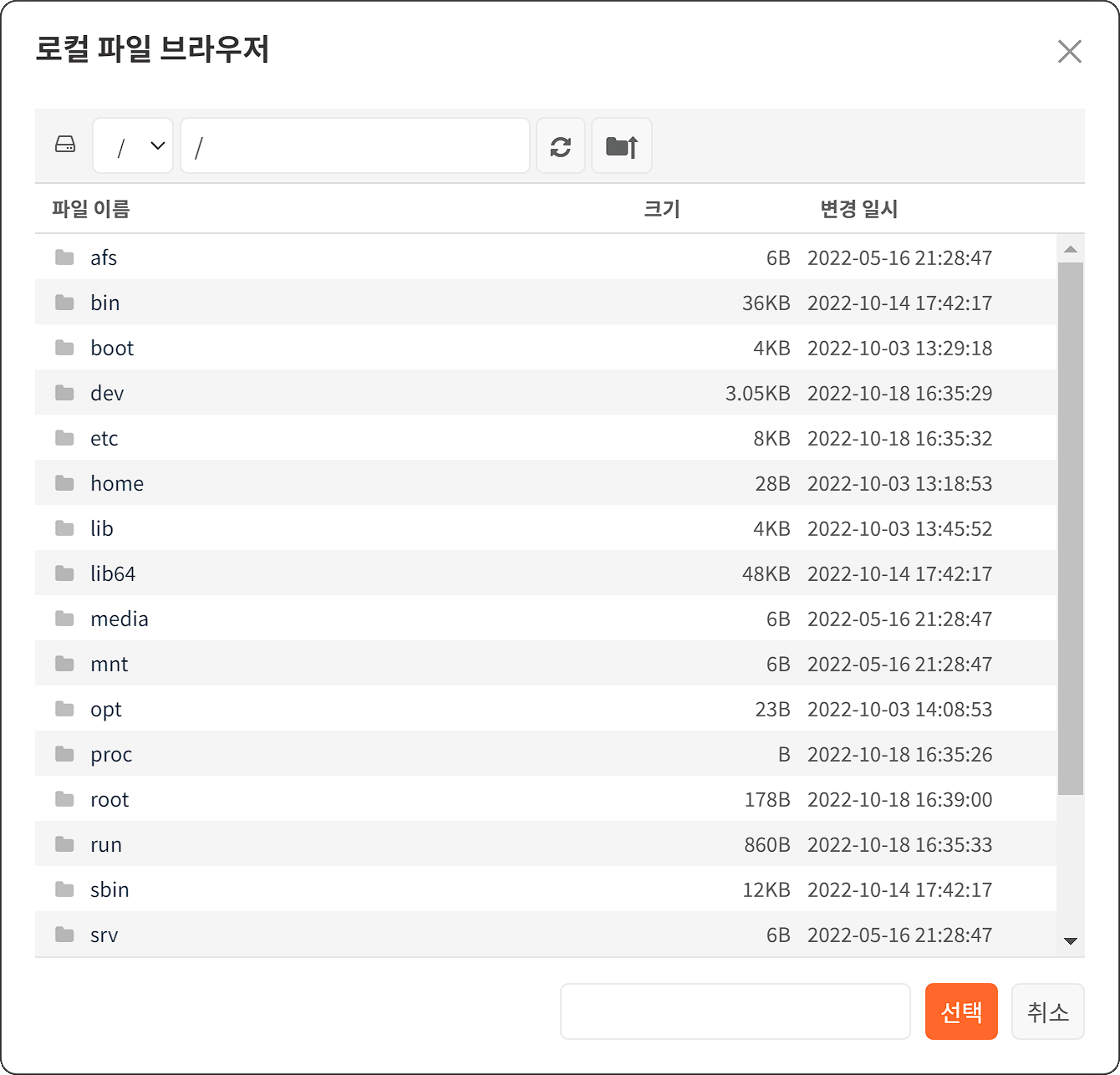
쿼리 커맨드를 식별할 이름

**설명**

쿼리 커맨드에 대한 설명

**파일 경로**

파일을 저장할 경로. **파일 선택**을 누른 후 **로컬 파일 브라우저**에서 이미 존재하는 파일 경로를 선택하거나, 입력창에 경로를 직접 입력할 수 있습니다.



**파일 인코딩**

파일 인코딩을 **UTF-8**, **확장완성형** 중에서 선택합니다(기본값: UTF-8).

**포맷**

파일 형식을 **구분자**, **CSV**, **TSV**, **JSON** 중에서 선택합니다(기본값: 구분자).

**구분자**

사용자가 지정한 구분자를 기준으로 필드를 구분하여 쿼리 결과를 저장합니다. 구분자로 사용할 문자를 지정해야 합니다.

**CSV**

텍스트 파일을 CSV(Comma-separated values) 형식으로 쿼리 결과를 저장합니다.

**TSV**

텍스트 파일을 TSV(Tab-separated values) 형식으로 쿼리 결과를 저장합니다.

**JSON**

텍스트 파일을 JSON 형식으로 인식해 레코드를 파싱합니다.

**기존 파일 존재 시 동작**

파일 경로에 같은 이름의 파일이 존재할 경우 동작을 선택합니다. **실패**, **덮어쓰기**, **이어쓰기** 중에서 선택할 수 있습니다(기본값: 실패).

**파일 출력 주기**

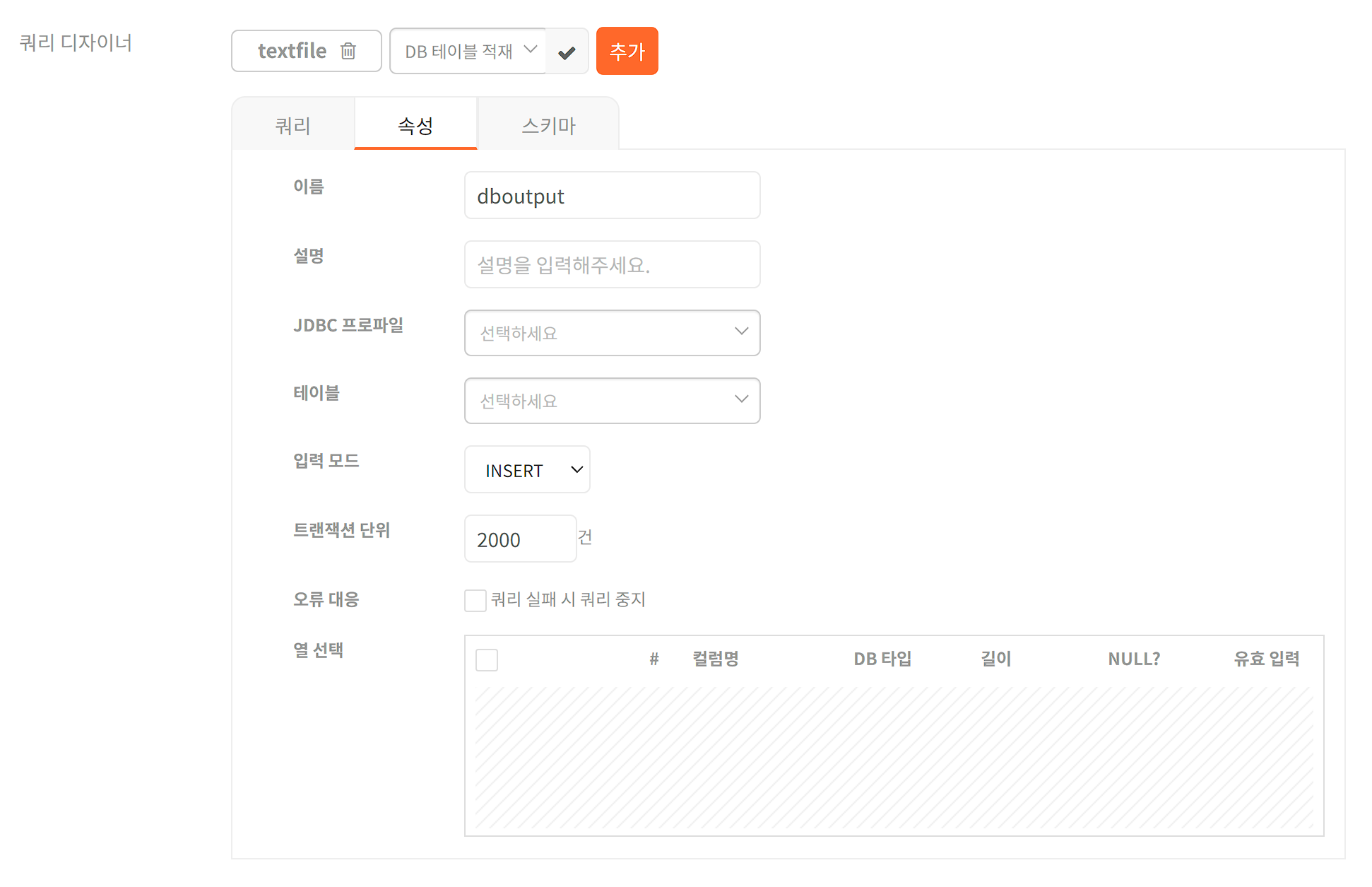
파일 출력 주기를 입력합니다.

**출력 필드**

파일에 출력할 필드를 목록에서 선택합니다.

DB 테이블 적재

DB 테이블 적재 스니펫 커맨드는 쿼리 결과를 DBMS 테이블에 저장합니다.



DB 테이블 적재 스니펫 커맨드의 속성은 다음과 같습니다.

**JDBC 프로파일**

DBMS에 접속할 때 사용할 JDBC 프로파일을 선택합니다.

**테이블**

데이터를 입력할 테이블.

**입력 모드**

입력 모드를 **INSERT**, **UPDATE** 중에서 선택합니다(기본값: INSERT). INSERT는 테이블 입력을 시도합니다. UPDATE는 기준 키로 해당 행이 이미 존재하는지 확인해서, 존재하면 UPDATE, 존재하지 않으면 INSERT를 수행합니다.

**트랜잭션위**

트랜잭션 배치 처리 단위 갯수. 단위가 크면 한 번에 많이 커밋되므로 효율적이지만, 트랜잭션이 실패했을 때 롤백되는 양도 늘어나므로 적당한 수를 지정해야 합니다. 배치 처리 단위를 지정하지 않으면 1건씩 자동 커밋되므로 매우 느릴 수 있습니다.

**오류 대응**

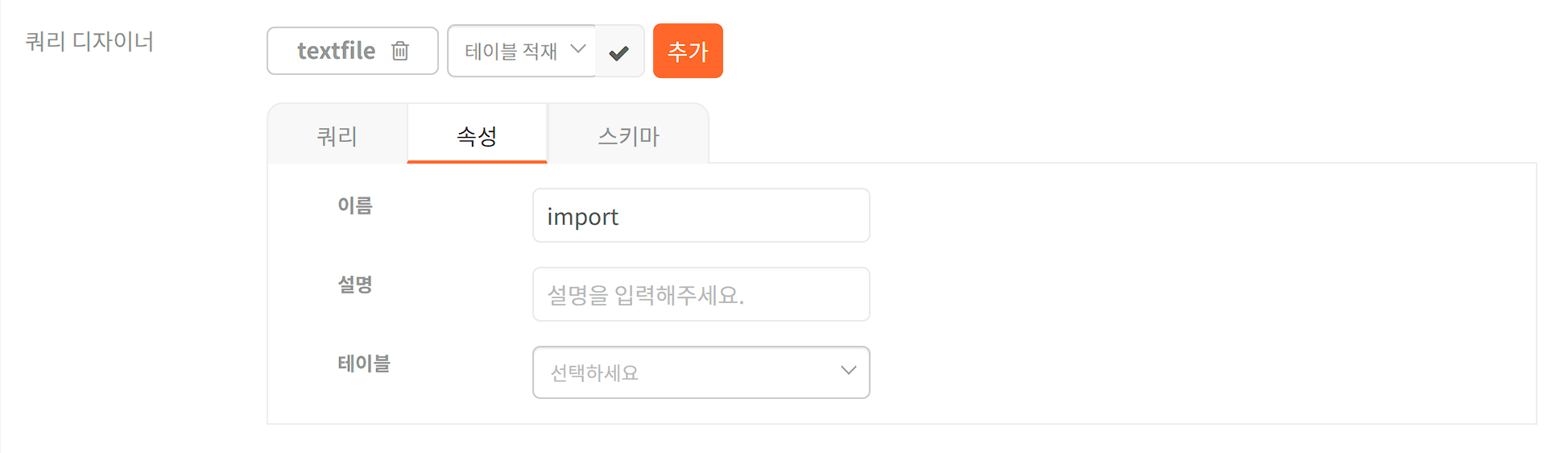
커밋 실패 시 동작을 지정합니다. 기본값은 커밋 실패 시 해당 트랜잭션 범위만 실패하고 다음 트랜잭션 범위로 넘어가도록 되어 있습니다. 선택 해제 시 해당 트랜잭션 범위가 실패하면 SQL 쿼리를 취소합니다.

**열 선택**

DBMS 테이블에 입력할 열을 선택합니다. 열 이름과 순서는 테이블 컬럼의 이름과 순서와 일치해야 합니다. **입력 모드**가 **UPDATE**일 경우 기준 키로 사용할 열 목록도 선택해야 합니다.

테이블 적재

테이블 적재 스니팻 커맨드는 쿼리 결과를 로그프레소 테이블에 저장합니다. 테이블 목록에서 데이터를 저장할 테이블을 선택합니다.



테이블 적재 스니팻 커맨드의 속성은 다음과 같습니다.

**이름**

쿼리 커맨드를 식별할 이름

**설명**

쿼리 커맨드에 대한 설명

**테이블**

데이터를 입력할 로그프레소 테이블을 목록에서 선택합니다.