

# SQL Injection

## Query

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

## Syntax & Grammatik

String Terminator : `'`

Statement Terminator: `;`

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. <b>Note:</b> In some versions of SQL this operator may be written as !=
BETWEEN	Between a certain range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

NOT invertieren

## LIKE Clause

- In Anführungsstrichen: `SELECT a FROM b WHERE name LIKE ''`
- Zwei Wildcards: `%` und `_`
  - `%`: Wie ein \* Wildcard in Shells. Matching von 0 oder mehr Characters
  - `_`: Wie ein . in regulären Ausdrücken: Ein beliebiges Zeichen

# Exploit

Oft ist es genug, die WHERE Bedingung wahr werden zu lassen mit einem OR 1=1.

```
$name = "steve";
$password = "x' or '1='1";
$query = "SELECT * FROM users WHERE name = 'steve'
          AND password = 'x' or '1='1'";
```

Condition is always true  
(password irrelevant)

Es funktioniert weil:

- **The developer's view**

```
$sql = "SELECT * FROM users WHERE passwd = '' + $pass + """;
```

Code

Data

- **The database's view**

```
$sql = "SELECT * FROM users WHERE passwd = '' + $pass + """;
```

Code

Data

?

- **An attack mixing code and data**

```
$sql = "SELECT * FROM users WHERE passwd = 'x' or '1' = '1'";
```

Code

Data

Attack