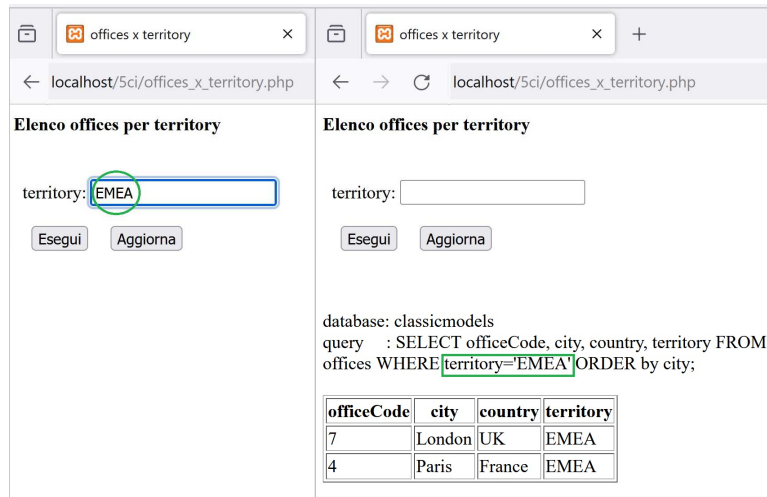


PHP – SQL Injection e Prepared Statement

L'input mediante caselle di testo per i parametri da utilizzare in comandi SQL può essere soggetto ad **SQL Injection** (inserimento doloso di istruzioni SQL nell'input - *malicious SQL*).

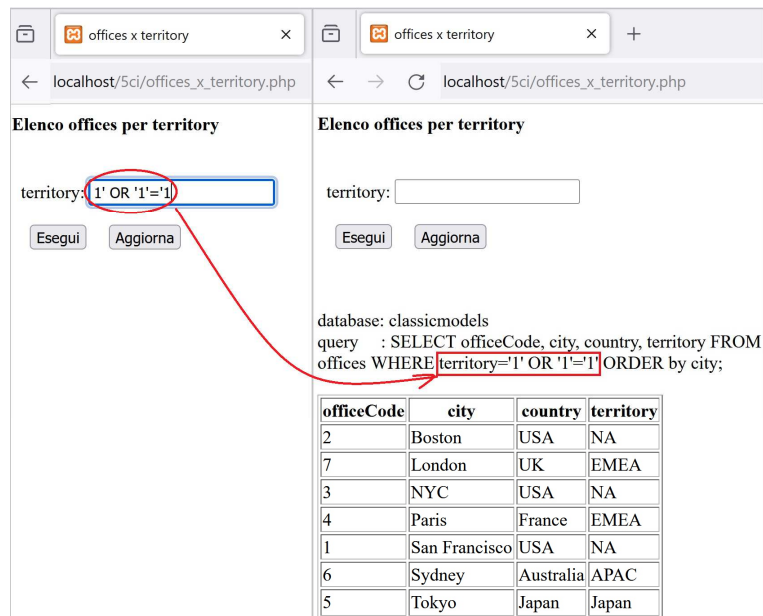
Un esempio: lo script di selezione degli offices in base ad un territory:



The screenshot shows a web browser with two tabs for 'offices x territory'. The left tab shows the 'territory' input field containing 'EMEA'. The right tab shows the result of the query: a table with 2 rows and 4 columns (officeCode, city, country, territory).

officeCode	city	country	territory
7	London	UK	EMEA
4	Paris	France	EMEA

Questo il risultato se nella casella di testo si digita **1' OR '1'='1**



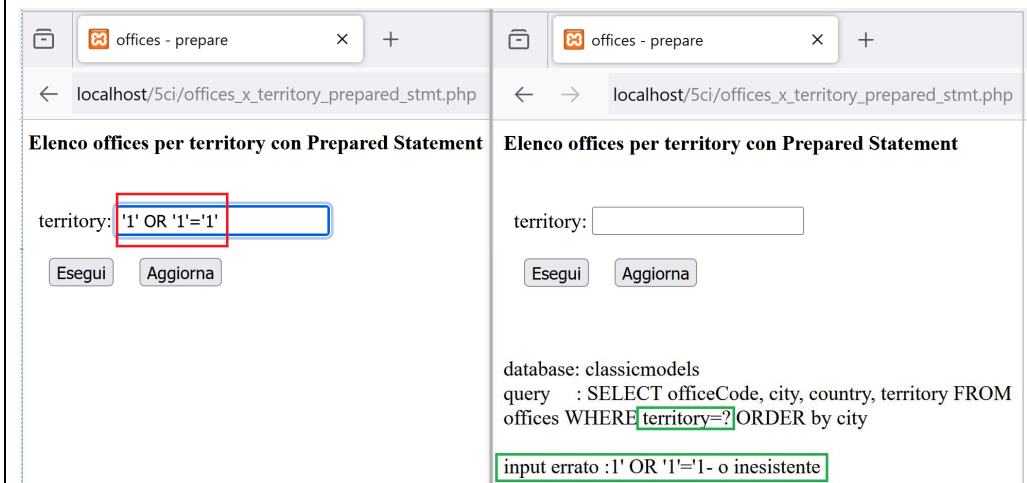
The screenshot shows the same web application. The 'territory' input field now contains '1' OR '1'='1'. The result table shows 7 rows of data from different territories, indicating that the condition is always true.

officeCode	city	country	territory
2	Boston	USA	NA
7	London	UK	EMEA
3	NYC	USA	NA
4	Paris	France	EMEA
1	San Francisco	USA	NA
6	Sydney	Australia	APAC
5	Tokyo	Japan	Japan

Nell'esempio, nonostante la query sia condizionata (mediante la **WHERE territory = '\$terr'**) e non esistano offices con territory='1', vengono estratte **tutte le righe della tabella** perchè la stringa **\$sql** viene espansa con una **condizione in OR ('1'='1')** che è **sempre vera**.

Si evita questo inconveniente utilizzando una **Prepared Statement** che **separa la struttura dell'SQL dai dati immessi dall'utente**; questo permette di avere una protezione rispetto alle *SQL Injection*: la query viene preparata (e analizzata da MySQL per verificarne la correttezza sintattica) con la funzione **mysqli_prepare(\$conn, \$sql)** utilizzando nella stringa **\$sql** dei segnaposto (**?**, *placeholder*) che vengono successivamente abbinati alle variabili PHP mediante la funzione **mysqli_stmt_bind_param(\$stmt, "s", \$terr)**, in cui va specificato il tipo atteso da MySQL ("**i**" per variabili da associare a dati di tipo **intero**, "**d**" per i **decimal**, "**s**" per le **stringhe**, "**b**" per i BLOB - Binary Large Object).

Nello script *offices_x_territory_prepared_stmt.php* (riportato nella pagina successiva a destra) il dato **\$terr**, che viene digitato dall'utente, può solo dare valore alla condizione sulla colonna territory (**WHERE territory=?**) ma **non può aggiungere comandi SQL** come **1' OR**.



The screenshot shows a web browser with two tabs for 'offices - prepare'. The left tab shows the 'territory' input field containing '1' OR '1'='1'. The right tab shows the result of the query: a table with 0 rows, indicating that the prepared statement correctly filters out the malicious input.

input errato :1' OR '1'='1- o inesistente

offices_x_territory.php	offices_x_territory_prepared_stmt.php
<pre> <html><head><title>offices x territory</title></head> <body><h4>Elenco offices per territory </h4> <form action = "<?php echo \$_SERVER['PHP_SELF']; ?>" method="post" >
&nbsp;&nbsp;&nbsp;territory: <input type="text" name="T" >

 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="submit" name="Esegui" value="Esegui"> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="submit" name="Aggiorna" value="Aggiorna">

</form> <?php if(isset(\$_POST["Aggiorna"])) header("Location: " . \$_SERVER['PHP_SELF']); if(isset(\$_POST["Esegui"])) { \$db_nome = "classicmodels"; require 'connect.php'; require 'TAG_cost.php'; \$terr = \$_POST ["T"]; //--valorizz.ne \$terr necessaria costruendo \$sql \$sql = "SELECT officeCode, city, country, territory"; \$sql .= " FROM offices WHERE territory='\$terr' ORDER by city;"; echo "
database: \$db_nome
\n"; echo "query &nbsp;&nbsp;&nbsp;: \$sql

\n"; \$result = mysqli_query (\$conn , \$sql); if (mysqli_num_rows (\$result) === 0) exit ("input errato :\$terr-
\n"); echo \$tagTa; echo \$tagRa; while (\$field = mysqli_fetch_field (\$result)) echo \$tagHa . \$field->name . \$tagHch; echo \$tagRch; while (\$row = mysqli_fetch_assoc (\$result)) { echo \$tagRa; foreach (\$row as \$key => \$val) echo \$tagDa . \$val . \$tagDch; echo \$tagRch; } echo \$tagTch . "</body></html>"; mysqli_free_result (\$result); mysqli_close (\$conn); } //----- fine isset su Esegui ?> </pre>	<pre> <html><head> <title>offices - prepare </title></head> <body><h4>Elenco offices per territory con Prepared Statement</h4> <form action = "<?php echo \$_SERVER['PHP_SELF']; ?>" method="post" >
&nbsp;&nbsp;&nbsp;territory: <input type="text" name="T" >

 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="submit" name="Esegui" value="Esegui"> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="submit" name="Aggiorna" value="Aggiorna">

</form> <?php if(isset(\$_POST["Aggiorna"])) header("Location: " . \$_SERVER['PHP_SELF']); if(isset(\$_POST["Esegui"])) { \$db_nome = "classicmodels"; require 'connect.php'; require 'TAG_cost.php'; \$sql = "SELECT officeCode, city, country, territory"; \$sql .= " FROM offices WHERE territory=? ORDER by city"; echo "
database: \$db_nome
\n"; echo "query &nbsp;&nbsp;&nbsp;: \$sql

\n"; \$stmt = mysqli_prepare(\$conn, \$sql); mysqli_stmt_bind_param(\$stmt, "s", \$terr); \$terr = \$_POST ["T"]; //--valorizz.ne \$terr necessaria prima di execute mysqli_stmt_execute(\$stmt); \$result = mysqli_stmt_get_result(\$stmt); if (mysqli_num_rows (\$result) === 0) exit ("input errato :\$terr- o inesistente
\n"); echo \$tagTa; echo \$tagRa; while (\$field = mysqli_fetch_field (\$result)) echo \$tagHa . \$field->name . \$tagHch ; echo \$tagRch; while (\$row = mysqli_fetch_row(\$result)){ echo \$tagRa; foreach (\$row as \$key => \$r) echo \$tagDa . \$r. \$tagDch; echo \$tagRch; } echo \$tagTch . "</body></html>"; mysqli_free_result (\$result); mysqli_stmt_close(\$stmt); mysqli_close(\$conn); } //----- fine isset su Esegui ?> </pre>

