Preparing Statements

Preparing Statements on menetelmä, joka mm. estää SQL-injektion mahdollisuuden. Menetelmää on suositeltavaa käyttää aina, kun SQL-lauseen osaksi luetaan dataa ulkoisesta lähteestä esim. käyttäjän syöttämänä.

Edellisessä luvussa esitetty SQL-injektio-esimerkki sql-inject.php voidaan kirjoittaa Preparing Statements -menetelmää käyttäen alla esitetyllä tavalla. Tässä materiaalissa esitellään ainoastaan nimettyjen place holdereiden käyttöä.

Esimerkki 2201: mysql05.php

- Rivillä 8 käytetään nimettyä place holderia :id, jota ei ympyröidä edes merkkijonojen tapauksessa heittomerkeillä
- Rivillä 10 metodilla \$db->prepare() SQL-kysely lähetetään ensin palvelimelle esikäännettäväksi
- Rivillä 11 esikäännöksen jälkeen place holderiin :id liitetään ukoisesta lähteestä tullut data muuttujasta \$_GET['id']. Tässä vaiheessa suoritettavan SQL-lauseen **merkitys** ei voi enää muuttua, jolloin SQL-injektion mahdollisuus estyy.
- Rivin 11 arvon liittäminen voi tapahtua merkkijonona esim. seuraavasti \$stmt->bindValue(':name', \$ GET['name'], PDO::PARAM STR);
- Rivillä 12 suoritetaan lopullinen SQL-kysely

```
<?php
1
    // mysq105.php
    require_once("/home/N1234/db-config/db-init.php");
3
4
    // Ota kommentti pois, jos et halua syöttää id:tä URLin avulla
5
    //$_GET['id'] = 1;
6
7
             = "SELECT * FROM customer WHERE id = :id";
    $sql
8
9
    $stmt = $db->prepare($sql);
10
    $stmt->bindValue(':id', $_GET['id'], PDO::PARAM_INT);
11
    $stmt->execute();
12
```

INSERT-, UPDATE- ja DELETE-lauseet ja Preparing Statements

Tällöin aiemmin esitetty *esimerkki 2012: mysql03.php* muuntuu seuraavaan muotoon.

Esimerkki 2202: mysql06.php

Huomio esimerkissä, että

- kyselyihin tuodaan "ulkoista dataa" esimerkin vuoksi riveillä 7-10, 24-25 ja 37-38
- ulkoinen data voidaan liittää place holdereihin myös hajautustaulukkona riveillä 15-17.
- kyselyjen suorittaminen tapahtuu execute () -metodilla riveillä 17, 30 ja 43

```
<?php
1
    // mysql06.php
2
    require once("/home/N1234/db-config/db-init.php");
 3
4
5
    // Lisätään yksi tietue
    // "Ulkoinen data":
7
    $id = 6;
    $nimi = 'Mieli Kaino';
    $pvm = '2011-01-01';
10
11
12
    $sql ="INSERT INTO customer VALUES(:id, :nimi, :pvm, (select )
13
    $stmt = $db->prepare($sql);
14
    $bind_array = array(':id' => $id, ':nimi' => $nimi, ':pvm' =>
15
```

```
16
    $affected rows = $stmt->execute($bind_array);
17
    echo "<br>>" . $affected_rows . " riviä lisättiin:<br>>";
18
19
    print customers($db);
20
21
22
    //-- -- -- --
23
    // Päivitetään yksi tietue
    // "Ulkoinen data":
24
    $pvm = '2019-01-09';
25
26
    $sql ="UPDATE customer SET birth date = :pvm WHERE name = 'Mie
27
    $stmt = $db->prepare($sql);
28
    $stmt->bindValue(':pvm', $pvm, PDO::PARAM STR);
29
    $affected rows = $stmt->execute();
30
    echo "<br>>" . $affected rows . " riviä muutettiin:<br>>";
31
32
33
    print customers($db);
34
35
36
    // Poistetaan yksi tietue
37
    // "Ulkoinen data":
38
    $nimi = 'Mieli Kaino';
39
    $sql ="DELETE FROM customer WHERE name = :nimi";
40
    $stmt = $db->prepare($sql);
41
42
    $stmt->bindValue(':nimi', $nimi, PDO::PARAM STR);
    $affected rows = $stmt->execute();
43
44
    echo "<br>" . $affected_rows . " riviä poistettiin:<br>";
45
46
47
    print customers($db);
48
             - -- -- -- -- -- -- -- -- -- -- -- --
49
    // Tulostetaan customer-taulu HTML-taulukkona
50
    function print_customers($db) {
51
52
     $result = $db->query('SELECT * FROM customer');
     $row count = $result->rowCount();
53
     echo "Näytetään " . $row_count. " riviä:<br>";
54
55
     echo "";
56
57
      while($row = $result->fetch(PDO::FETCH ASSOC)) {
        58
```

```
59 }
60 echo "";
61 }
62 ?>
```

Prepare Statements ja SQL-funktiot

SQL-funktiot (esim. now ()) on liitettävä seuraavalla tavalla

```
1  | $status = 'online';
2  | $stmt = $db->prepare("INSERT INTO table(`time`, `status`) VALU
3  | $stmt->bindValue(':status', $status, PDO::PARAM_STR);
4  | $stmt->execute();
```

Funktiolle voi antaa myös parametrit place holdereilla seuraavasti

Prepare Statements ja toistorakenteet

Kertaalleen esivalmistellun kyselyn voi suorittaa toistuvasti myös toistorakenteessa. Huomaa, että \$name-muuttuja pitää määritellä etukäteen (tyhjänä), jotta se voidan sitoa bindParam()-metodissa:

4

© #AriRantala