Prepared Statement 올바르게 사용해보기

```
SELECT name
FROM   cnu_dept_computer
JOIN   argos
ON     cnu_dept_computer.sid=argos.member_sid
AND    sid=201704150;          name='허강준'
```

Recommended_For =

DB를 이용한 프로그래밍에 입문하신 분들
어느정도 감을 잡고 좀 더 안전한 코딩을 원하시는 분들
조그만 실수 하나까지 바로잡고 싶으신 분들

// PHP와 SQL을 알고 있거나 막 배웠다면 더욱 잘 이해 가능

Contents =

```
Array (
    [0] => “SQL Injection”,
    [1] => “Prepared Statement”,
    [2] => “For Performance”,
    [3] => “About Missable Vulnerability”
);
```

SQL_Injection =

- SQL 쿼리에 공격을 위한 SQL을 Injection 하는 공격
- DB를 자유자재로 조작할 수 있어 파괴력이 강함
  (임의 실행, 데이터 탈취, 인증 우회 등…)

$sample --sql_injection

```
// vulnerable_signin.php
$id    = $_POST["user_id"];
$pw    = $_POST["user_pw"];
$query = mysqli_query($conn, "SELECT * FROM users WHERE id='$id' AND pw='$pw'");

if ($query->num_rows === 0) {
    // do something when signing fails...
    exit;
}

// do something to make status is signed in...
```

$sample --sql_injection

```
$id    = "admin';-- ";
$pw    = "1234";
$query = $conn->query("SELECT * FROM users WHERE id='$id' AND pw='$pw'");
```

```
SELECT * FROM users WHERE id='admin';-- ' AND pw='1234'
```

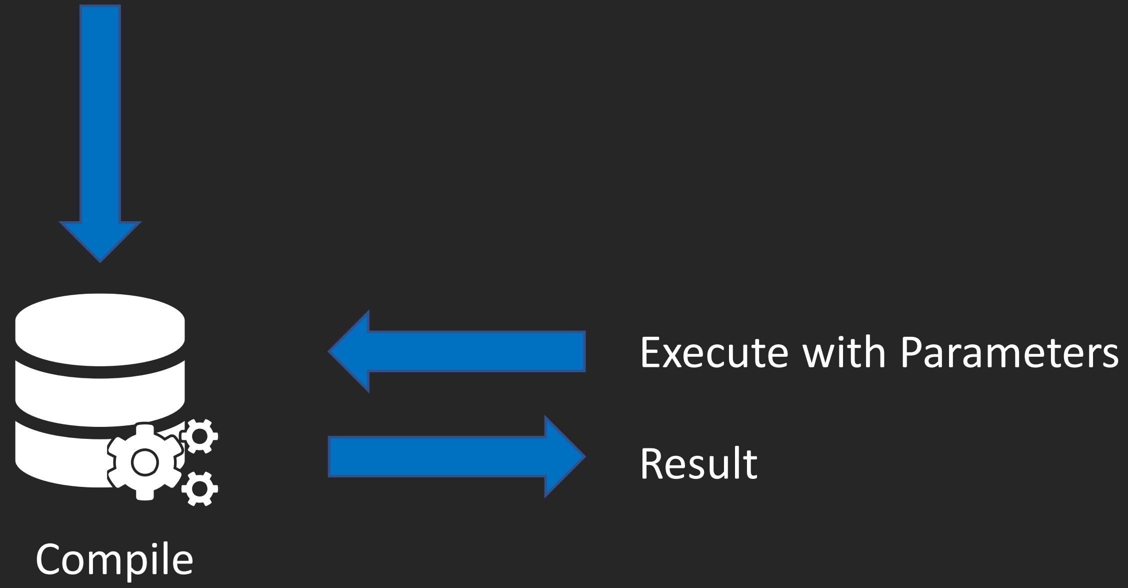
Protection =

- 신뢰할 수 없는 데이터를 쿼리에 포함하지 않기
- ~~입력 데이터를 이스케이핑 하기~~
- Prepared Statement 사용하기
- 기타등등…

Prepared_Statement =

Parameterized Query

```
SELECT *
FROM daechoong_con_tbl
WHERE attr_1=? AND attr_2 LIKE CONCAT('%', ?, '%');
```



Prepared_Statement =

```
// prepared_signin.php
$id   = $_POST["user_id"];
$pw   = $_POST["user_pw"];
$stmt = $conn->prepare("SELECT * FROM users WHERE id=? AND pw=?");
$stmt->bind_param("ss", $id, $pw);
$stmt->execute();

if ($query->num_rows === 0) {
    // do something when signing fails...
    exit;
}
// do something to make status is signed in...
```

Better way to use?

Useless_Prepared_Statement =

- Prepared Statement는 빠르다.
- 쿼리가 Compile & Evaluation 되는 과정이 한번만 일어남
- 이후 백엔드 시스템에서 Parameter만 Binding되어 동작
- 그러나 한번만 쓰고 버려지는 쿼리라면?

Useless_Prepared_Statement =

```
$date = date("Y");

// slow_select.php
$stmt = $conn->prepare("SELECT * FROM notes WHERE year=?");
$stmt->bind_param("s", $date);
$stmt->execute();

// better_select.php
$stmt = $conn->query("SELECT * FROM notes WHERE year=$date");
```

Useless_Prepared_Statement =

(miliseconds)

10000 times of iteration	1	2	3	4	5	AVG.
불필요한 Prepared Statement	945	1013	935	909	895	939.4
단순 Concat 쿼리	475	490	469	502	472	481.6

* 로컬 MariaDB, PHP 7.2.18에서 테스트되었음.

* https://b.patche.me/ups

Useless_Prepared_Statement =

(miliseconds)

10000 times of iteration	1	2	3	4	5	AVG.
PS with Loop	359	377	374	371	383	372.8

* 로컬 MariaDB, PHP 7.2.18에서 테스트되었음.

* https://b.patche.me/upsp

Useless_Prepared_Statement =

```
function auto_statement($conn, $query, ...$args) {
    // if (args !== null) -> return prepared statement object
    // if (args === null) -> return query result object
}

auto_statement($conn, $query, $param1, ...); // mysqli::stmt object
auto_statement($conn, $query);               // mysqli::result object
```

Using_For_Procedures =

```
-- vulnerable_procedure.sql
-- DELIMITER //
CREATE OR REPLACE PROCEDURE create_private_table(
  table_name TEXT
) BEGIN
    SET @SQL = CONCAT('CREATE TABLE ', table_name, '_ptable (...');
    PREPARE stmt FROM @SQL;
    EXECUTE stmt;
    DEALLOCATE PREPARE stmt;
  END; //
```

Using_For_Procedures =

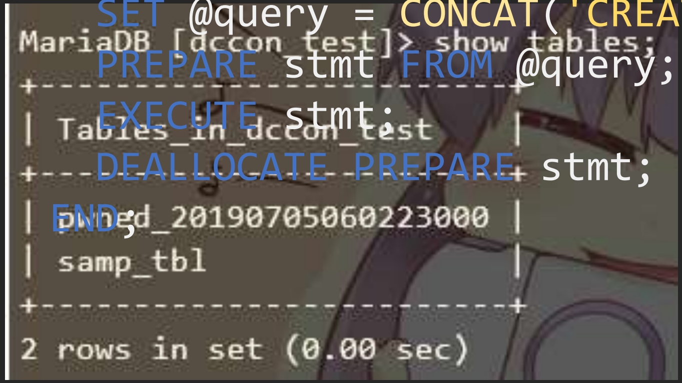
```
// freaking_great.php
$stmt = $conn->prepare("CALL create_private_table(?)");
$stmt->bind_param("s", $_POST['id']);
$stmt->bind_param("s", "pwned(id int)-- &#x20");

SET @SQL := CONCAT('CREATE TABLE', table_name, '(...');
SET @SQL := 'CREATE pwned(id int)-- &#x20table(...');
```

Using_For_Procedures =

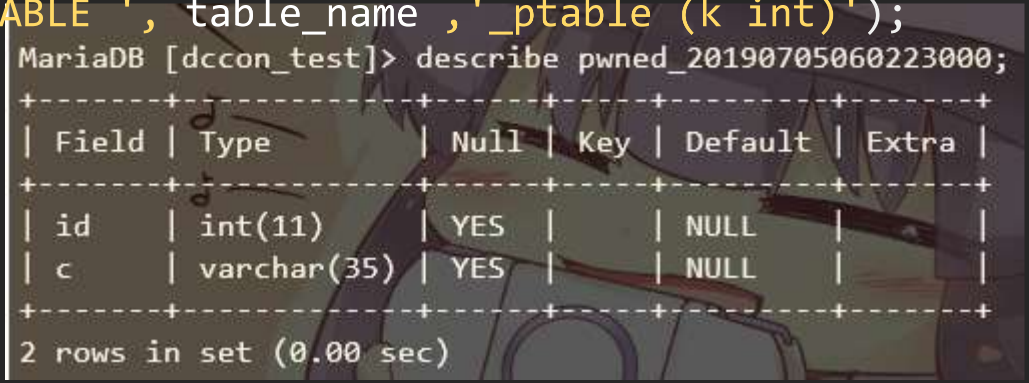
- https://b.patche.me/ufp

```
$datetime = date("Ymdhisv");
$query = "pwned_$datetime(id int, c varchar(35)); -- ";
$stmt      = mysqli_prepare($conn, "CALL create_private_table(?)");
```

```
CREATE PROCEDURE create_private_table (
 table_name TEXT
)
 BEGIN
   SET @query = CONCAT('CREATE TABLE ', table_name ,'_ptable (k int)');
   PREPARE stmt FROM @query;
   EXECUTE stmt;
   DEALLOCATE PREPARE stmt;
 END;
```



```
MariaDB [dccon_test]> show tables;
+-------------------------+
| Tables_in_dccon_test    |
+-------------------------+
| pwned_20190705060223000 |
| samp_tbl                |
+-------------------------+
2 rows in set (0.00 sec)
```



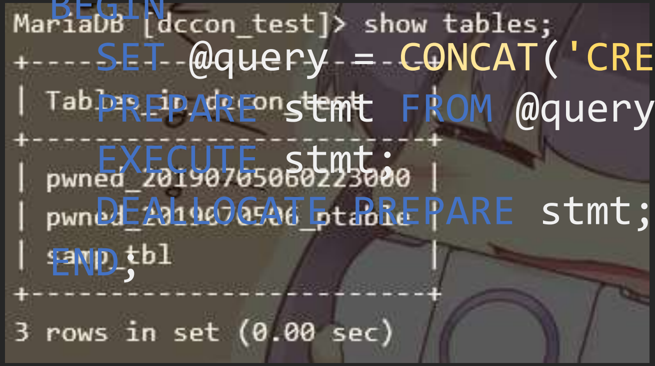
```
MariaDB [dccon_test]> describe pwned_20190705060223000;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | YES  |     | NULL    |       |
| c     | varchar(35) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Using_For_Procedures =

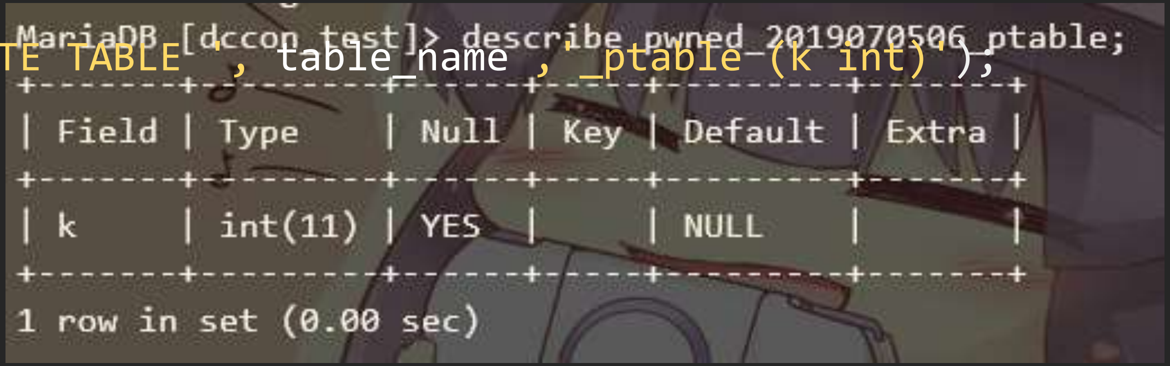
- https://b.patche.me/ufpp

```
$datetime = date("Ymdhis");
$query = "pwned_$datetime(id int, c varchar(35)); -- ";
```

```
CREATE PROCEDURE create_private_table_p (
 table_name VARCHAR(16)
)
  BEGIN
    SET @query = CONCAT('CREATE TABLE ', table_name , '_ptable (k int)');
    PREPARE stmt FROM @query;
    EXECUTE stmt;
    DEALLOCATE PREPARE stmt;
  END;
```



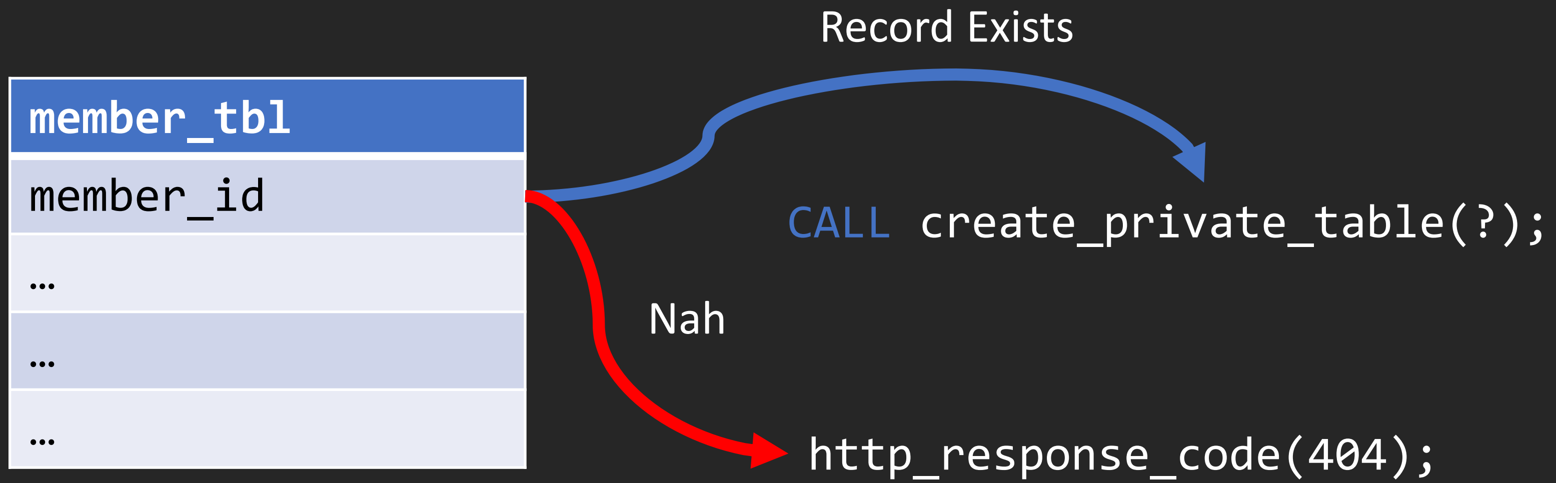
```
MariaDB [dccon_test]> show tables;
+-------------------------+
| Tables_in_dccon_test    |
+-------------------------+
| pwned_20190705060223000 |
| pwned_2019070506_ptable |
| samp_tbl                |
+-------------------------+
3 rows in set (0.00 sec)
```



```
MariaDB [dccon_test]> describe pwned_2019070506_ptable;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| k     | int(11) | YES  |     | NULL    |       |
+-------+---------+------+-----+---------+-------+
1 row in set (0.00 sec)
```

Using_For_Procedures =

```
SELECT member_id
FROM   member_tbl
WHERE  member_id=?
```



Conclusion =

```
Array (
    [0] => "성능을 위해 상황에 맞추어" .
           "Prepared Statement 사용하기",
    [1] => "놓치기 쉬운 취약점에 대비하기 위해" .
           "신뢰할 수 없는 데이터에 항상 주의하기"
);
```

QnA

Thank you!_Happy Hacking!