



시큐어 코딩

- SQL 인젝션을 중심으로



목차

- 1 시큐어 코딩이란?
- 2 SQL 인젝션
- 3 이거 꼭 해야 되나..?



Part 1,

시큐어 코딩이란?





Secure Coding ?

소프트웨어(SW)를 개발함에 있어 개발자의 실수, 논리적 오류 등으로 인해 SW에 내포될 수 있는 보안취약점(vulnerability)을 배제하기 위한 코딩 기법을 뜻 한다.



Secure Coding ?

소프트웨어(SW)를 개발함에 있어 개발자의 실수, 논리적 오류 등으로 인해 SW에 내포될 수 있는 **보안취약점(vulnerability)**을 배제하기 위한 코딩 기법을 뜻 한다.



Security

소프트웨어(SW)를 개발함
으로 인해 SW에 내포될 수
하기 위한 코딩 기법을 뜻 함





470

게시물

812

팔로워

887

팔로잉

프로필 수정

투잡 휴일이나 퇴근후가볍게할수있는고수익부업알바
남성알바생채용합니다.투잡이나 주말근무가능!
급전필요하신남성분대환영알바시간 오후
4시이후2시간기본페이80만원+팁+교통비
전지역가능^^
여러분한분이라도벌어야사는세상입니다하루라도
달립시다

☎(주)매칭센터☎K상담카톡:





470

게시물

812

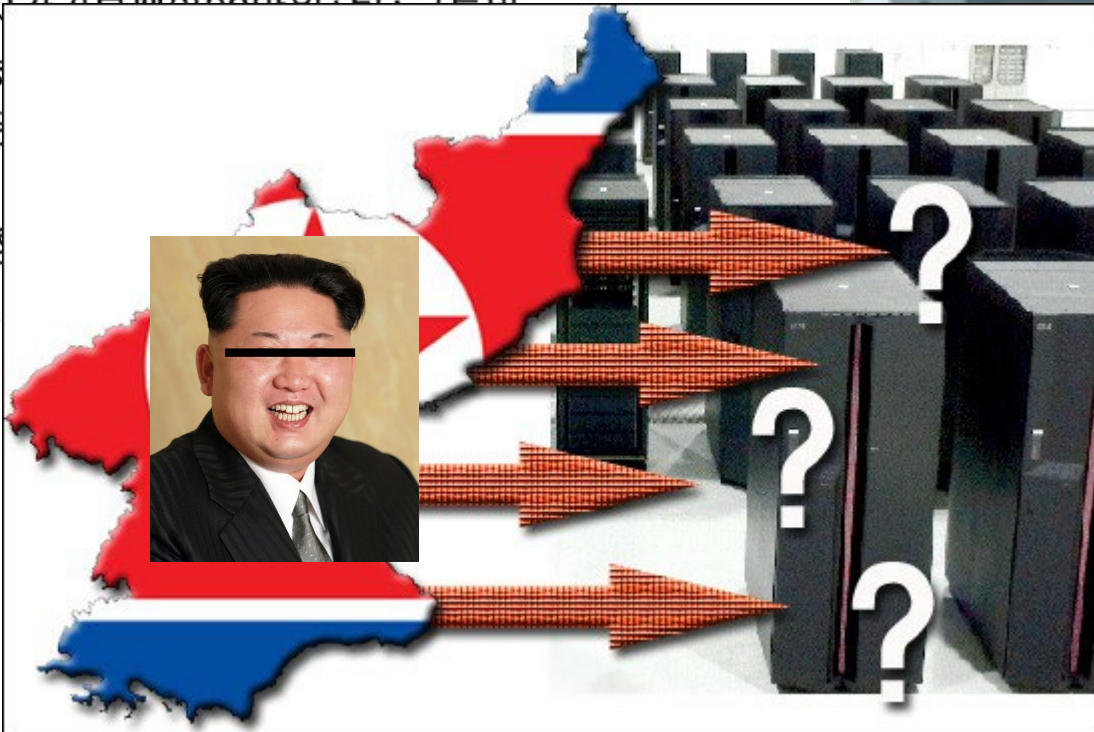
팔로워

887

팔로잉

프로필 수정

투잡 휴일이나 퇴근후가볍게할수있는고수익부업알바
남성알바생채용합니다.투잡이나 주말근무가능!
급전필요하신남성분대환영알바시간 오후
4시이후2시간까지
전지역가능
여러분한분
달립시다
(주)매칭





투잡 휴일이나 퇴근후
남성알바생채용합니다
급전필요하신남성분다
4시이후2시간까지
전지역가능
여러분한분
달립니다
(주)매





Secure Coding ?

소프트웨어(SW)를 개발함에 있어 개발자의 실수, 논리적 오류 등으로 인해 SW에 내포될 수 있는 **보안취약점(vulnerability)**을 배제하기 위한 코딩 기법을 뜻 한다.



2023 CWE Top 25 Most Dangerous Software Weaknesses

[Top 25 Home](#)[Share via:](#) [View in table format](#)[Key Insights](#)[Methodology](#)**1****Out-of-bounds Write**[CWE-787](#) | CVEs in KEV: 70 | Rank Last Year: 1**2****Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')**[CWE-79](#) | CVEs in KEV: 4 | Rank Last Year: 2**3****Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')**[CWE-89](#) | CVEs in KEV: 6 | Rank Last Year: 3**4****Use After Free**[CWE-416](#) | CVEs in KEV: 44 | Rank Last Year: 7 (up 3) ▲**5****Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')**[CWE-78](#) | CVEs in KEV: 23 | Rank Last Year: 6 (up 1) ▲**6****Improper Input Validation**[CWE-20](#) | CVEs in KEV: 35 | Rank Last Year: 4 (down 2) ▼**7****Out-of-bounds Read**[CWE-125](#) | CVEs in KEV: 2 | Rank Last Year: 5 (down 2) ▼**8****Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')**[CWE-22](#) | CVEs in KEV: 16 | Rank Last Year: 8**9****Cross-Site Request Forgery (CSRF)**[CWE-352](#) | CVEs in KEV: 0 | Rank Last Year: 9**10****Unrestricted Upload of File with Dangerous Type**[CWE-434](#) | CVEs in KEV: 5 | Rank Last Year: 10

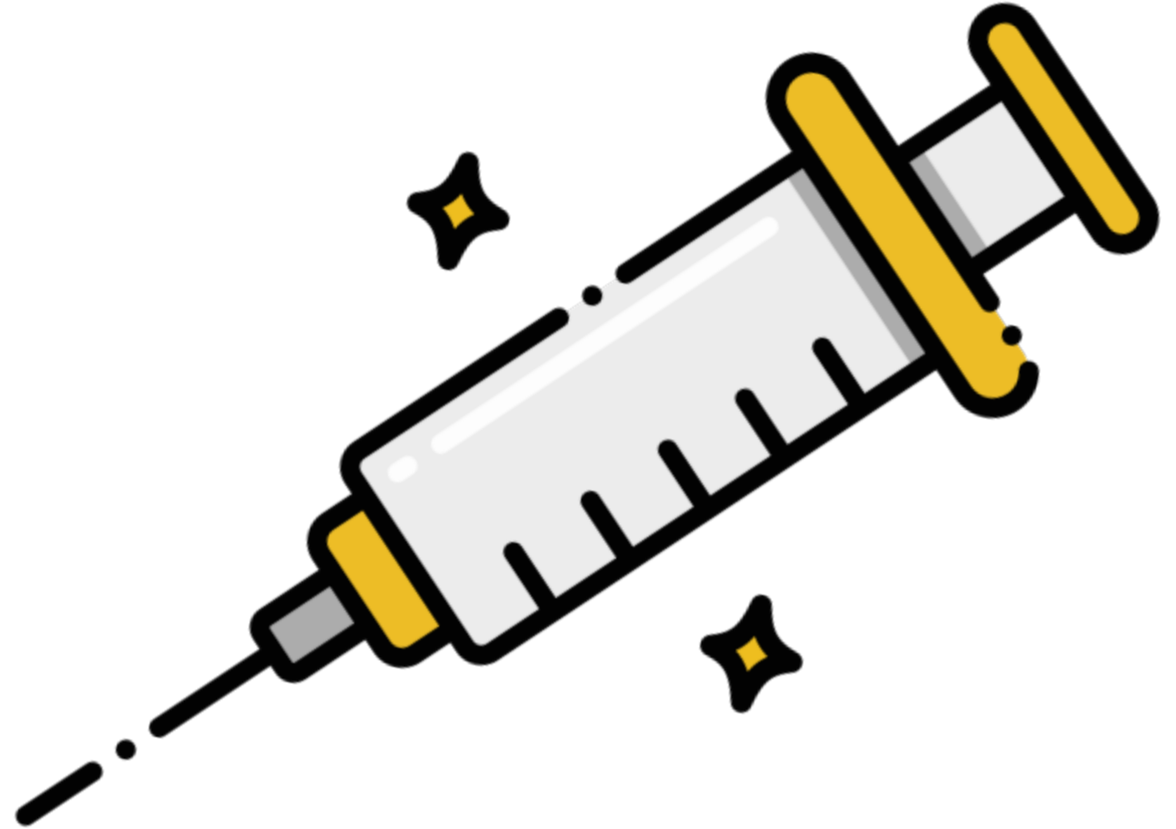


2023 CWE Top 25 Most Dangerous Software Weaknesses

[Top 25 Home](#)[Share via:](#) [View in table format](#)[Key Insights](#)[Methodology](#)**1****Out-of-bounds Write**[CWE-787](#) | CVEs in KEV: 70 | Rank Last Year: 1**2****Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')**[CWE-79](#) | CVEs in KEV: 4 | Rank Last Year: 2**3****Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')**[CWE-89](#) | CVEs in KEV: 6 | Rank Last Year: 3**4****Use After Free**[CWE-416](#) | CVEs in KEV: 44 | Rank Last Year: 7 (up 3) ▲**5****Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')**[CWE-78](#) | CVEs in KEV: 23 | Rank Last Year: 6 (up 1) ▲**6****Improper Input Validation**[CWE-20](#) | CVEs in KEV: 35 | Rank Last Year: 4 (down 2) ▼**7****Out-of-bounds Read**[CWE-125](#) | CVEs in KEV: 2 | Rank Last Year: 5 (down 2) ▼**8****Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')**[CWE-22](#) | CVEs in KEV: 16 | Rank Last Year: 8**9****Cross-Site Request Forgery (CSRF)**[CWE-352](#) | CVEs in KEV: 0 | Rank Last Year: 9**10****Unrestricted Upload of File with Dangerous Type**[CWE-434](#) | CVEs in KEV: 5 | Rank Last Year: 10



Part 2, **SQL 인젝션**





SQL + Injection



데이터베이스 관리 및 데이터 조작



```
SELECT * FROM Employees WHERE Department = 'Sales';
```

데이터베이스 관리 및 데이터 조작



```
SELECT * FROM Employees WHERE Department = 'Sales';
```

모든 데이터를 조회하자

Employees 테이블에서

Department 열(column)이 'Sales'인 행(row)을

작



Injection



gdsc.community.dev/dongguk-university/

로그인

Google 계정 사용

이메일 또는 휴대전화

Google



응답 1개



응답받기 ☒

요약

질문

개별 보기

이름을 입력해주세요

응답 1개

Test

연락받을 연락처를 입력해주세요 (ex:010-0000-0000)

응답 1개

01000000000



데이터베이스 관리 및 데이터 조작

```
SELECT * FROM Employees WHERE Department = 'Sales';
```

모든 데이터를 조회하자

Employees 테이블에서

Department 열(column)이 'Sales'인 행(row)을



Injection

 gdsc.community.dev/dongguk-university/



로그인

Google 계정 사용

이메일 또는 휴대전화

|



데이터베이스 관리 및 데이터 조작

```
SELECT * FROM Employees WHERE Department = 'Sales';
```

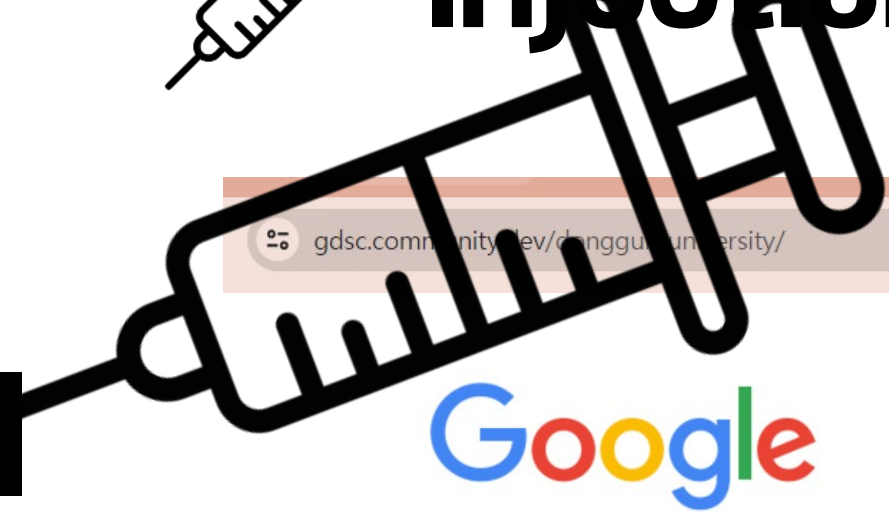
모든 데이터를 조회하자

Employees 테이블에서

Department 열(column)이 'Sales'인 행(rows)을



Injection



Google



로그인

Google 계정 사용

이메일 또는 휴대전화

|



https://demo.testfire.net/

IBM에서 웹 애플리케이션 보안 교육과 테스트를 목적으로 만들어진 가상의 은행 웹사이트



[Sign In](#) | [Contact Us](#) | [Feedback](#) | Search



ONLINE BANKING LOGIN	PERSONAL	SMALL BUSINESS	INSIDE ALTORO MUTUAL
<p>PERSONAL</p> <ul style="list-style-type: none">Deposit ProductCheckingLoan ProductsCardsInvestments & InsuranceOther Services <p>SMALL BUSINESS</p> <ul style="list-style-type: none">Deposit ProductsLending ServicesCardsInsuranceRetirementOther Services <p>INSIDE ALTORO MUTUAL</p> <ul style="list-style-type: none">About UsContact UsLocationsInvestor RelationsPress RoomCareersSubscribe	<p>Online Banking with FREE Online Bill Pay</p> <p>No stamps, envelopes, or checks to write give you more time to spend on the things you enjoy.</p>  <p>Real Estate Financing</p> <p>Fast. Simple. Professional. Whether you are preparing to buy, build, purchase land, or construct new space, let Altoro Mutual's premier real estate lenders help with financing. As a regional leader, we know the market, we understand the business, and we have the track record to prove it</p>	 <p>Business Credit Cards</p> <p>You're always looking for ways to improve your company's bottom line. You want to be informed, improve efficiency and control expenses. Now, you can do it all - with a business credit card account from Altoro Mutual.</p> <p>Retirement Solutions</p> <p>Retaining good employees is a tough task. See how Altoro Mutual can assist you in accomplishing this feat through effective Retirement Solutions.</p>	<p>Privacy and Security</p> <p>The 2000 employees of Altoro Mutual are dedicated to protecting your privacy and security. We pledge to provide you with the information and resources that you need to help secure your information and keep it confidential. This is our promise.</p>  <p>Win a Samsung Galaxy S10 smartphone</p> <p>Completing this short survey will enter you in a draw for 1 of 5 Samsung Galaxy S10 smartphones! We look forward to hearing your important feedback.</p>



[PERSONAL](#)

- [Deposit Product](#)
- [Checking](#)
- [Loan Products](#)
- [Cards](#)
- [Investments & Insurance](#)
- [Other Services](#)

[SMALL BUSINESS](#)

- [Deposit Products](#)
- [Lending Services](#)
- [Cards](#)
- [Insurance](#)
- [Retirement](#)
- [Other Services](#)

[INSIDE ALTORO MUTUAL](#)

- [About Us](#)
- [Contact Us](#)
- [Locations](#)
- [Investor Relations](#)
- [Press Room](#)
- [Careers](#)
- [Subscribe](#)

Online Banking Login

Username:

Password:

Login



Online Banking Login

Username:

Password:

Login



Online Banking Login

Login Failed: We're sorry, but this username or password was not found in our system. Please try again.

Username:

Password:

ID : gouyeonch

PW : asdfasd



Online Banking Login

Syntax error: Encountered "asdfasd" at line 1, column 71.

Username:

Password:

ID : gouyeonch'

PW : asdfasd

```
SELECT * FROM accounts
WHERE id='유저입력ID' AND pw='유저입력PW'
```

Online Banking Login

Syntax error: Encountered "asdfasd" at line 1, column 71.

Username:

Password:

ID : gouyeonch'

PW : asdfasd



```
SELECT * FROM accounts
WHERE id='유저입력ID' AND pw='유저입력PW'
```



Online Banking Login

Syntax error: Encountered "asdfasd" at line 1, column 71.

Username:

Password:

ID : gouyeonch'

PW : asdfasd

```
SELECT * FROM accounts
WHERE id='gouyeonch' ' AND pw='asdfasd' -> syntax error!!!
```




Online Banking L

**Syntax error: Encoun
column 71.**

Username:

Password:

Login



yeonch'

dfasd

```
SELECT * FROM accounts  
WHERE id='gouyeonch' ' AND pw='asdfasd' -> syntax error!!!
```

- 1.강제 로그인 가능
- 2.테이블 전체 출력 가능
- 3.DB 전체 삭제 가능

Online Banking Login

**Syntax error: Encountered "asdfasd" at line 1,
column 71.**

Username:

Password:

ID : gouyeonch'

PW : asdfasd

```
SELECT * FROM accounts  
WHERE id='gouyeonch' ' AND pw='asdfasd' -> syntax error!!!
```



Online Banking Login

Username:

Password:

Login

ID : tuser' --

PW : gsasgwartw

```
SELECT * FROM accounts  
WHERE id='tuser' -- ' AND pw='유저입력PW'
```



Online

[MY ACCOUNT](#)

[PERSONAL](#)

[SMALL BUSINESS](#)

[INSIDE ALTORO](#)

Username:

Password:

WANT TO ...

- [View Account Summary](#)
- [View Recent Transactions](#)
- [Transfer Funds](#)
- [Search News Articles](#)
- [Customize Site Language](#)

Hello Test User

Welcome to Altoro Mutual Online.

View Account Details:

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000!

Click [Here](#) to apply.

[Privacy Policy](#) | [Security Statement](#) | [Server Status Check](#) | [REST API](#) | © 2023 Altoro Mutual, Inc.

This web application is open source! [Get your copy from GitHub](#) and take advantage of advan

SELECT * FROM
WHERE id='tu

artw



Online Banking Login

Username:

Password:

ID : tuser' OR 1=1 --

PW : gsasgwartw

```
SELECT * FROM accounts
WHERE id='tuser' OR 1=1 -- ' AND pw='유저입력PW'
```



Online Banking Login

Username:

Password:

Id	pw
Admin	1234
Gouyeonch	asdfs12!@
Google	pwgoopw1156
dongguk	dongdong1230
....

ID : tuser' OR 1=1 --

PW : gsasgwartw

```
SELECT * FROM accounts  
WHERE id='tuser' OR 1=1 -- ' AND pw='유저입력PW'
```

Id

pw



Go



MY ACCOUNT

PERSONAL

SMALL BUSINESS

INSIDE ALTORO MUTUAL

Only

I WANT TO ...

- [View Account Summary](#)
- [View Recent Transactions](#)
- [Transfer Funds](#)
- [Search News Articles](#)
- [Customize Site Language](#)

ADMINISTRATION

- [Edit Users](#)

Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details:

800000 Corporate



GO

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000!

Click [Here](#) to apply.

SELECT * FROM
WHERE id



[Sign Off](#) | [Contact Us](#) | [Feedback](#) | Search

Go

AltoroMutual

DEMO
SITE
ONLY



[MY ACCOUNT](#)

[PERSONAL](#)

[SMALL BUSINESS](#)

[INSIDE ALTORO MUTUAL](#)

I WANT TO ...

- [View Account Summary](#)
- [View Recent Transactions](#)
- [Transfer Funds](#)
- [Search News Articles](#)
- [Customize Site Language](#)

ADMINISTRATION

- [Edit Users](#)

Transfer Funds

From Account:

800000 Corporate



To Account:

800000 Corporate



Amount to Transfer:

Transfer Money

[Privacy Policy](#) | [Security Statement](#) | [Server Status Check](#) | [REST API](#) | © 2023 Altoro Mutual, Inc.

***This web application is open source!** [Get your copy from GitHub](#) and take advantage of advanced features*

SELECT
WHERE



[MY ACCOUNT](#)

[PERSONAL](#)

[SMALL BUSINESS](#)

[INSIDE ALTORO MUTUAL](#)

WANT TO ...

- [View Account Summary](#)
- [View Recent Transactions](#)
- [Transfer Funds](#)
- [Search News Articles](#)
- [Customize Site Language](#)

ADMINISTRATION

- [Edit Users](#)

Transfer Funds

From Account:

800000 Corporate ▼

To Account:

800001 Checking ▼

Amount to Transfer:

1000000000

Transfer Money

SELECT
WHERE

[Sign Off](#) | [Contact Us](#) | [Feedback](#) | Search

Go

AltoroMutual



DEMO
SITE
ONLY



[MY ACCOUNT](#)

[PERSONAL](#)

[SMALL BUSINESS](#)

[INSIDE ALTORO MUTUAL](#)

I WANT TO ...

- [View Account Summary](#)
- [View Recent Transactions](#)
- [Transfer Funds](#)
- [Search News Articles](#)
- [Customize Site Language](#)

ADMINISTRATION

- [Edit Users](#)

Transfer Funds

From Account:

800000 Corporate

To Account:

800000 Corporate

Amount to Transfer:

Transfer Money

1.0E9 was successfully transferred from Account 800000 into Account 800001 at 11/29/23 4:36 AM.

[Privacy Policy](#) | [Security Statement](#) | [Server Status Check](#) | [REST API](#) | © 2023 Altoro Mutual, Inc.

This web application is open source! [Get your copy from GitHub](#) and take advantage of advanced features

SELEC
WHEP



MY ACCOUNT

PERSONAL

SMALL BUSINESS

INSIDE ALTORO MUTUAL

I WANT TO ...

- [View Account Summary](#)
- [View Recent Transactions](#)
- [Transfer Funds](#)
- [Search News Articles](#)
- [Customize Site Language](#)

ADMINISTRATION

- [Edit Users](#)

Recent Transactions

After

Before

Submit

yyyy-mm-dd

yyyy-mm-dd

Transaction ID	Transaction Time	Account ID	Action	Amount
5826	2023-11-29 04:36	800001	Deposit	\$1000000000.00
5825	2023-11-29 04:36	800000	Withdrawal	-\$1000000000.00
5824	2023-11-29 04:36	800001	Deposit	\$11.00
5823	2023-11-29 04:36	800000	Withdrawal	-\$11.00





Online Banking Login

**Syntax error: Encountered "asdfasd" at line 1,
column 71.**

Username:

Password:

ID : gouyeonch'

PW : asdfasd

```
SELECT * FROM accounts  
WHERE id='유저입력ID' AND pw='유저입력PW'
```

1. 입력값 검증

Online Banking Login

Syntax error: Encountered "asdfasd" at line 1,
column 71.

Username:

Password:

Login

???????

ID : gouyeonch'

PW : asdfasd

```
SELECT * FROM accounts
WHERE id='유저입력ID' AND pw='유저입력PW'
```

1. 입력값 검증

세미콜론(;), 단일 및 이중 따옴표(' , "), 백슬래시(\), 주석 기호(--, /*, */)

Online Banking Login

Syntax error: Encountered "asdfasd" at line 1, column 71.

Username:

Password:

Login

ID : gouyeonch'

PW : asdfasd

```
SELECT * FROM accounts
WHERE id='유저입력ID' AND pw='유저입력PW'
```




Online Banking Login

**Syntax error: Encountered "asdfasd" at line 1,
column 71.**

Username:

Password:

ID : gouyeonch'

PW : asdfasd

```
SELECT * FROM accounts
WHERE id='유저입력ID' AND pw='유저입력PW'
```

2. 에러 메시지 관리

???????

Online Banking Login

Syntax error: Encountered "asdfasd" at line 1,
column 71.

Username:

Password:

ID : gouyeonch'

PW : asdfasd

```
SELECT * FROM accounts
WHERE id='유저입력ID' AND pw='유저입력PW'
```



Online Banking Login

Syntax error: Encountered "asdfasd" at line 1, column 71.

Username:

Password:

ID : gouyeonch'

PW : asdfasd

```
SELECT * FROM accounts
WHERE id='유저입력ID' AND pw='유저입력PW'
```



Online Banking Login

Syntax error: Encountered "asdfasd" at line 1, column 71.

Username:

Password:

ID : gouyeonch'

PW : asdfasd



```
SELECT * FROM accounts
WHERE id='유저입력ID' AND pw='유저입력PW'
```

3. Prepared Statement (함수 안에 넣어서 필터 한 번 거치자)

Online Banking Login

Syntax error: Encountered "asdfasd" at line 1,
column 71.

Username:

Password:

ID : gouyeonch'

PW : asdfasd



```
SELECT * FROM accounts  
WHERE id='유저입력ID' AND pw='유저입력PW'
```




3. Prepared Statement (함수 안에 넣어서 필터 한 번 거치자)

```
String name = request.getParameter("inName");  
String sql = "SELECT * FROM board WHERE user = '" + name + "'";  
Connection con = db.getConnection();  
Statement stmt = con.createStatement();  
ResultSet rs = statement.executeQuery(sql);
```



3. Prepared Statement (함수 안에 넣어서 필터 한 번 거치자)

Input



```
String name = request.getParameter("inName");  
String sql = "SELECT * FROM board WHERE user = '" + name + "'";  
Connection con = db.getConnection();  
Statement stmt = con.createStatement();  
ResultSet rs = statement.executeQuery(sql);
```

3. Prepared Statement (함수 안에 넣어서 필터 한 번 거치자)

```
String name = request.getParameter("name");  
String sql = "SELECT * FROM board WHERE username = ?";  
Connection con = db.getConnection();  
PreparedStatement pstmt = con.prepareStatement(sql);  
pstmt.setString(1, name);  
ResultSet rs = pstmt.executeQuery();
```


3. Prepared Statement (함수 안에 넣어서 필터 한 번 거치자)

```
String name = request.getParameter("inName");  
String sql = "SELECT * FROM board WHERE user = '" + name + "'";  
Connection con = db.getConnection();  
Statement stmt = con.createStatement();  
ResultSet rs = statement.executeQuery(sql);
```



```
String name = request.getParameter("name");  
String sql = "SELECT * FROM board WHERE username = ?";  
Connection con = db.getConnection();  
PreparedStatement pstmt = con.prepareStatement(sql);  
pstmt.setString(1, name);  
ResultSet rs = pstmt.executeQuery();
```



3. Prepared Statement (함수 안에 넣어서 실행한 번 거치자)

```
String name = request.getParameter("inName");  
String sql = "SELECT * FROM board WHERE user = '" + name + "'";  
Connection con = db.getConnection();  
Statement stmt = con.createStatement();  
ResultSet rs = statement.executeQuery(sql);
```

```
String name = request.getParameter("name");  
String sql = "SELECT * FROM board WHERE username = ?";  
Connection con = db.getConnection();  
PreparedStatement pstmt = con.prepareStatement(sql);  
pstmt.setString(1, name);  
ResultSet rs = pstmt.executeQuery();
```



3. Prepared Statement (함수 안에 넣어서 실행할 때 한 번 거치자)

```
String name = request.getParameter("inName");  
String sql = "SELECT * FROM board WHERE user = '" + name + "'";  
Connection con = db.getConnection();  
Statement stmt = con.createStatement();  
ResultSet rs = statement.executeQuery(sql);
```

분리된 쿼리 구조와 데이터

자동적인 데이터 새니타이징

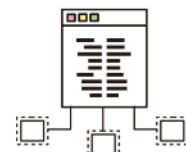
```
String name = request.getParameter("name");  
String sql = "SELECT * FROM board WHERE username = ?";  
Connection con = db.getConnection();  
PreparedStatement pstmt = con.prepareStatement(sql);  
pstmt.setString(1, name);  
ResultSet rs = pstmt.executeQuery();
```



Part 3, **이거 꼭 해야 되나..?**

저기, 미안한데
귀찮아.





JPA
Java Persistence API



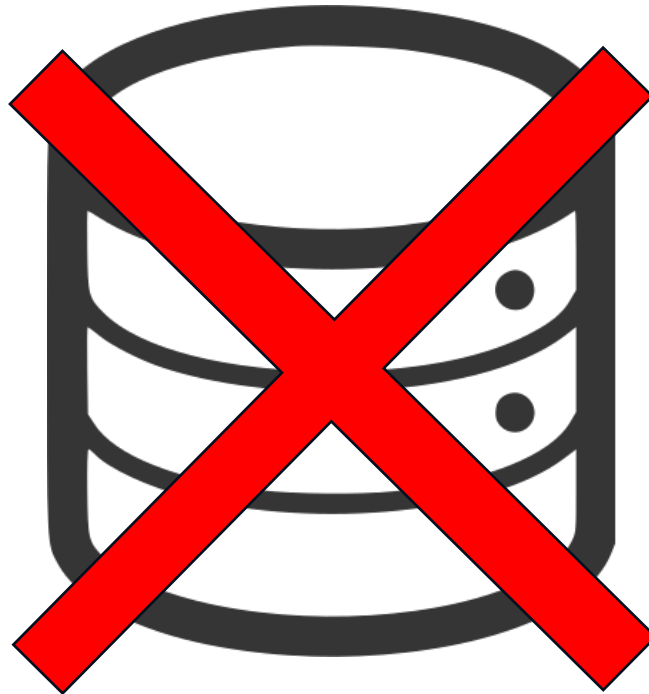
Query DSL
- JPA Module -



Django
ORM



DB 버려..?







감사합니다

Q/A