

Maven & JDBC

SC363204

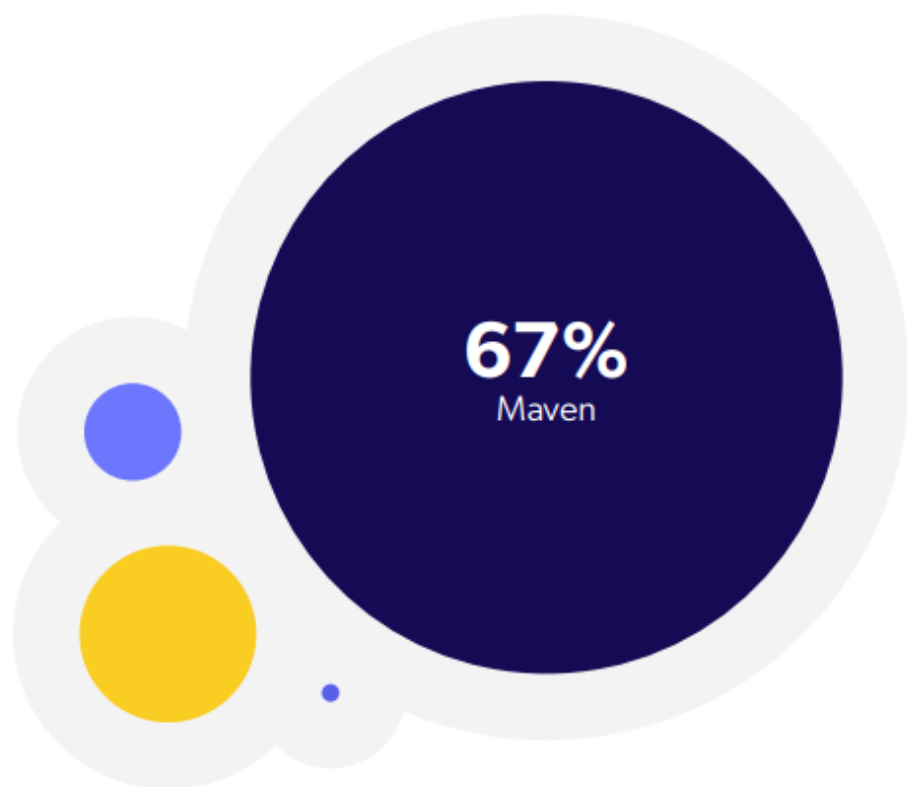
Java Web Application Development

การพัฒนาโปรแกรมประยุกต์บนเว็บด้วยภาษาจาวา





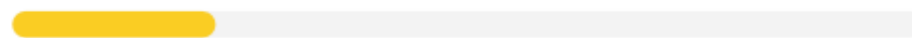
What build tool do you use in your main application?



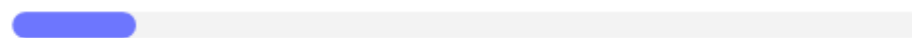
Maven **67%**



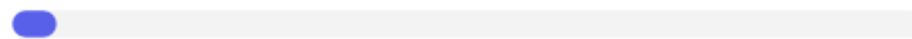
Gradle **20%**



Ant **11%**



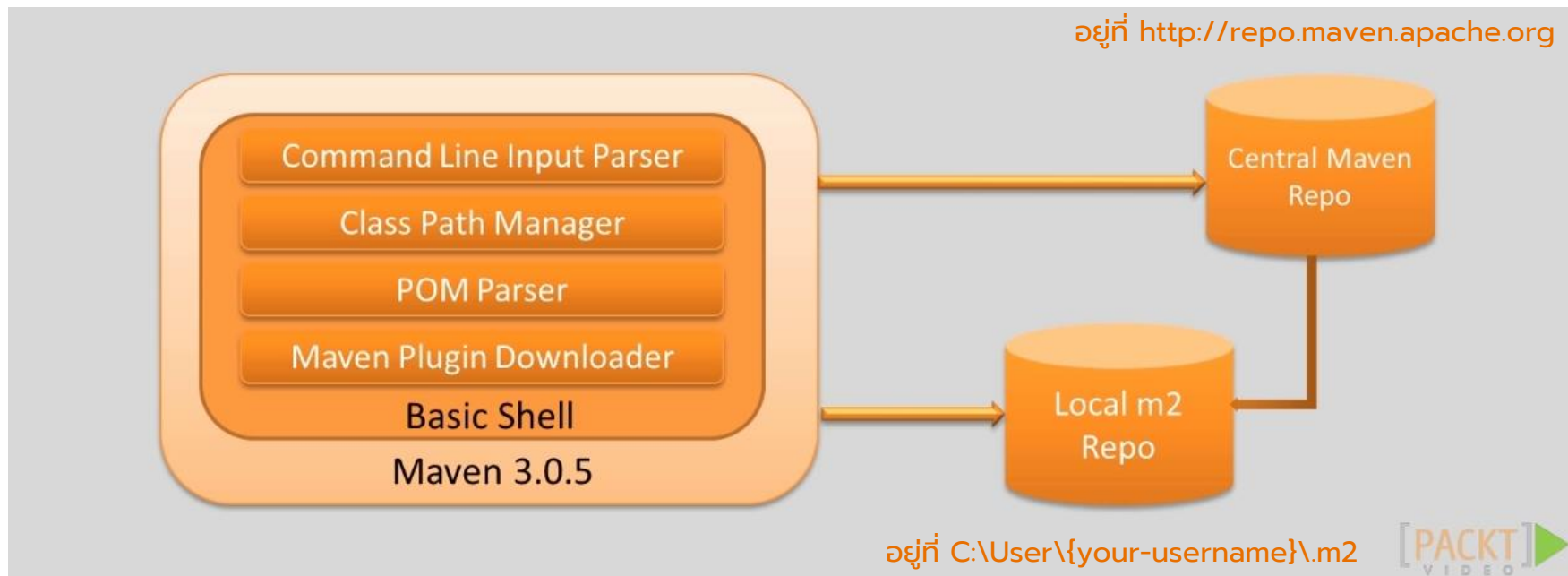
Other **2%**



- **Maven** คือ เครื่องมือสำหรับการบริหารจัดการ Project โดยมีเป้าหมายคือ ช่วยในการ build project แบบอัตโนมัติ
- Maven ใช้ **Project Object Model (POM)** ซึ่งเป็นไฟล์ข้อมูลเกี่ยวกับการสร้าง Project ในรูปแบบ XML นักพัฒนาสามารถกำหนดข้อมูล เช่น
 - รายการไลบรารีที่อ้างอิง (**Dependency list**) ช่วยให้ไม่ต้องไปโหลดไฟล์ Library เอง
 - ส่วนการทำ Unit Test
 - การกำหนด environment ต่าง ๆ
- Maven มีแม่แบบ (archetype) ในการช่วยสร้างแอปพลิเคชันชนิดต่างๆ
- ใน **Eclipse** มี Maven ติดมาแล้ว สามารถสร้าง Project ในรูปแบบ Maven ได้เลย

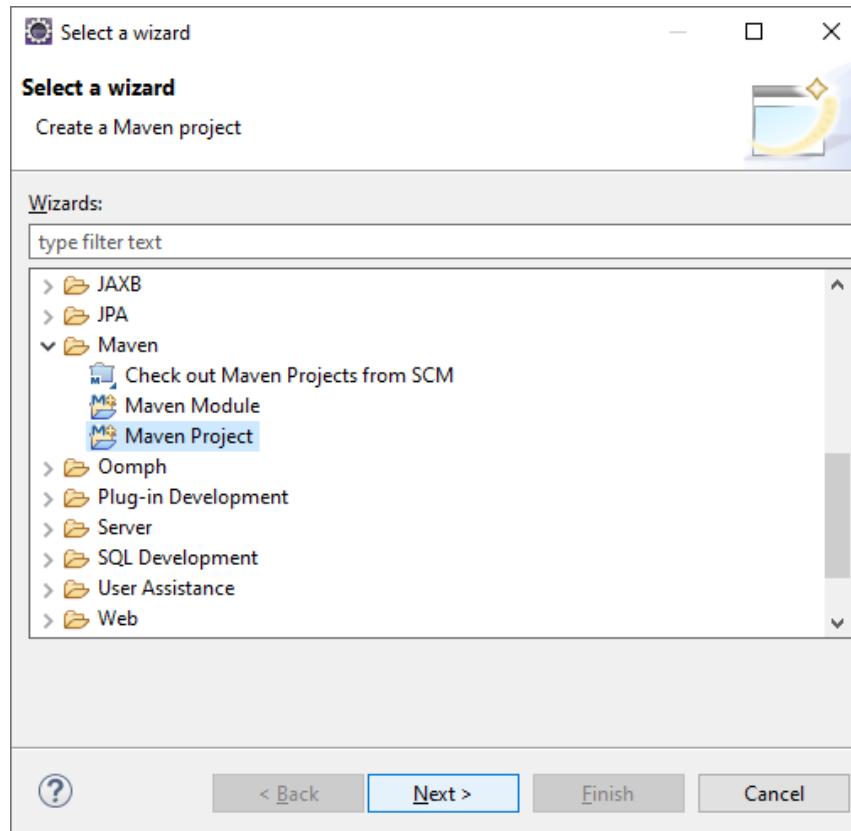
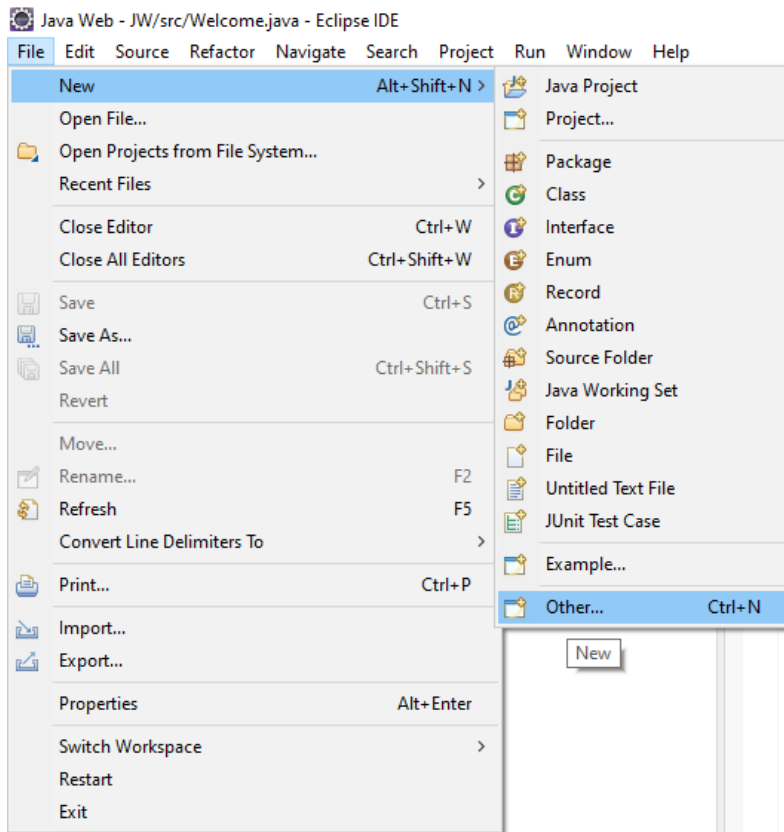
Maven Architecture

- **Central Maven Repository** คือ Server ที่ใช้เก็บ Library ทั้งหมด
- Library ที่ถูกอ้างอิงมาใช้แล้ว จะอยู่บน **Local m2 Repository** เพื่อใช้กับ Project อื่นๆไม่ต้องโหลดใหม่จาก Central Maven Repository ทุกครั้ง



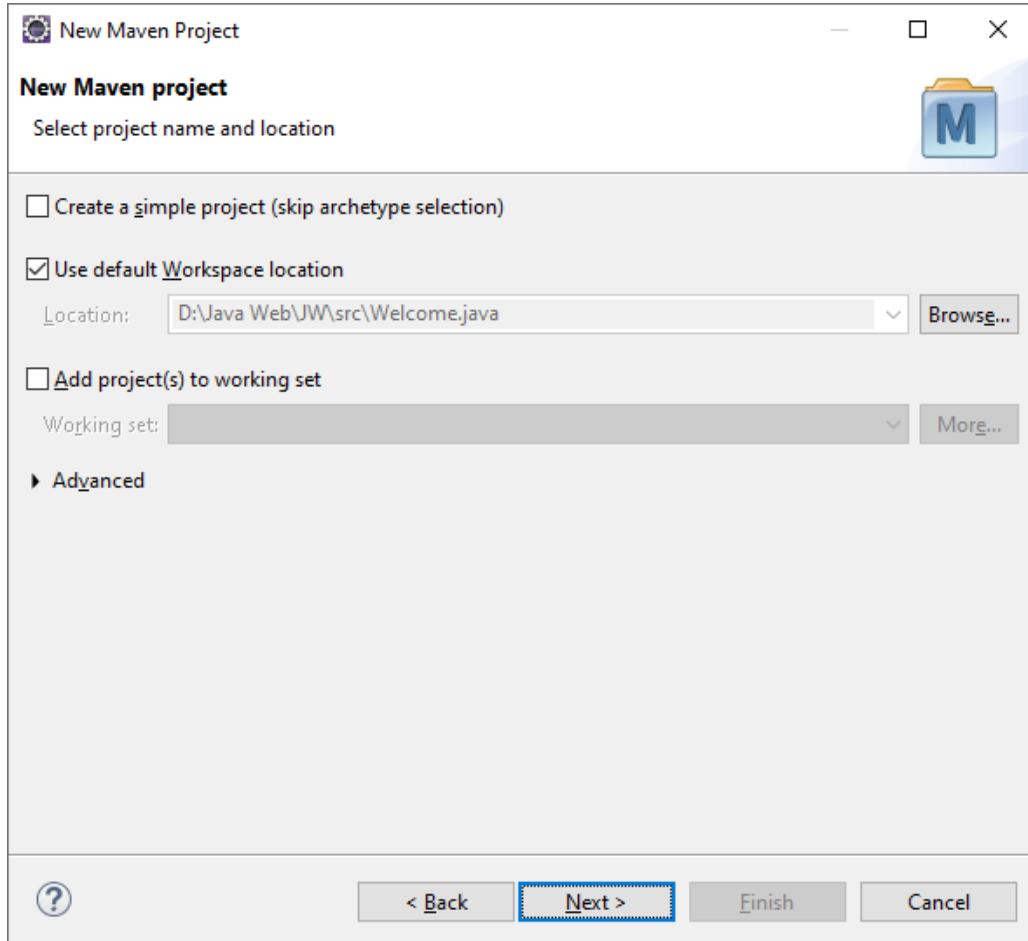
Create Maven

- เลือกเมนู **File** ▶ **New** ▶ **Maven Project**



Create Maven

- **na Next**



New Maven Project

New Maven project
Select project name and location

☐ Create a simple project (skip archetype selection)

☒ Use default Workspace location

Location: D:\Java Web\JW\src\Welcome.java Browse...

☐ Add project(s) to working set

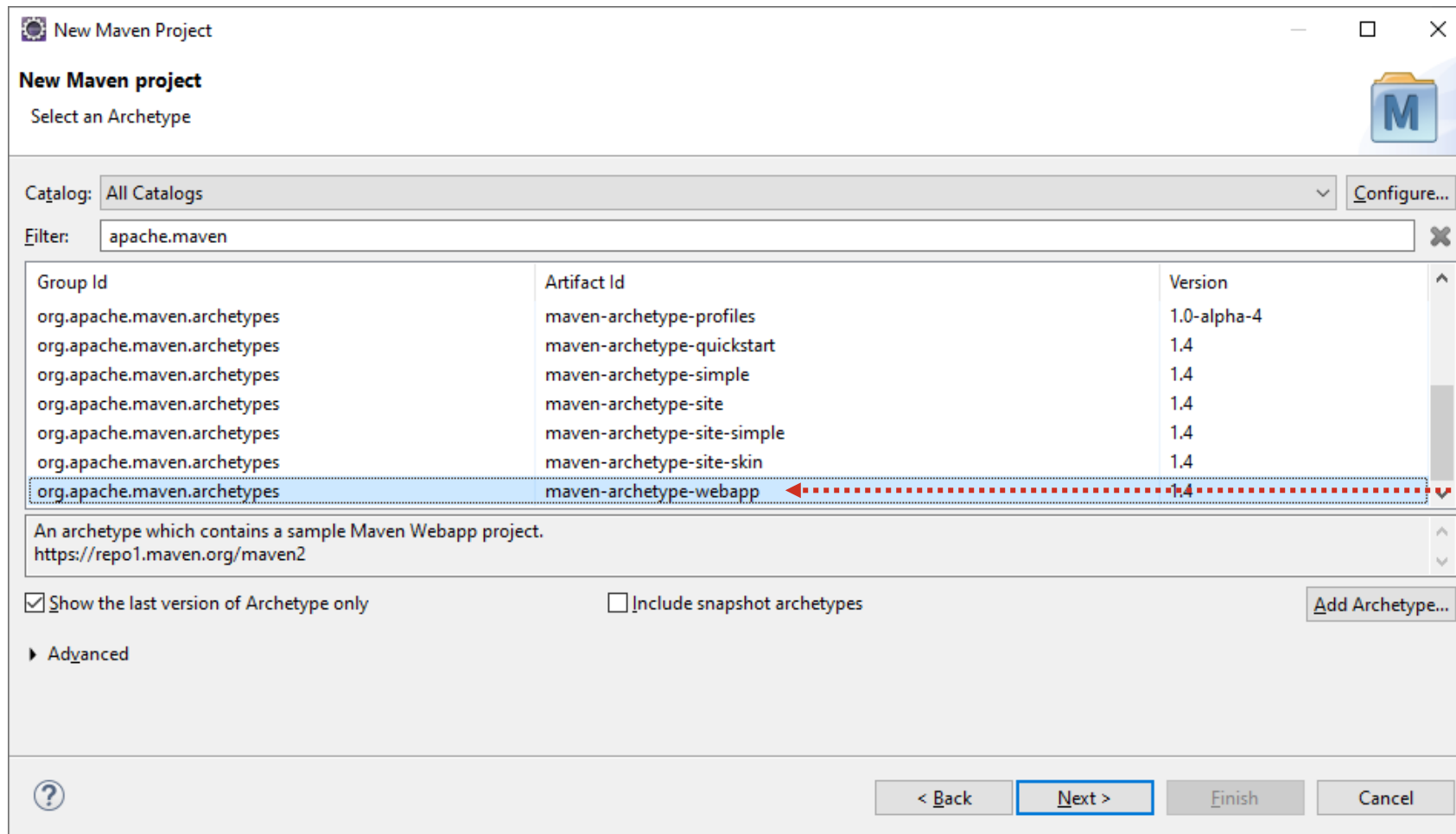
Working set: More...

► Advanced

? < Back Next > Finish Cancel

Create Maven

- ข้อ Filter กรอง: apache.maven



เลือกใช้แม่แบบเว็บแอปพลิเคชัน
(Web application archetype)

Create Maven

New Maven Project

New Maven project

Specify Archetype parameters

Group Id: com.javawebapp

Artifact Id: firstmaven

Version: 0.0.1-SNAPSHOT

Package: com.javawebapp.firstmaven

☒ run archetype generation interactively

Properties available from archetype:

Name	Value

Advanced

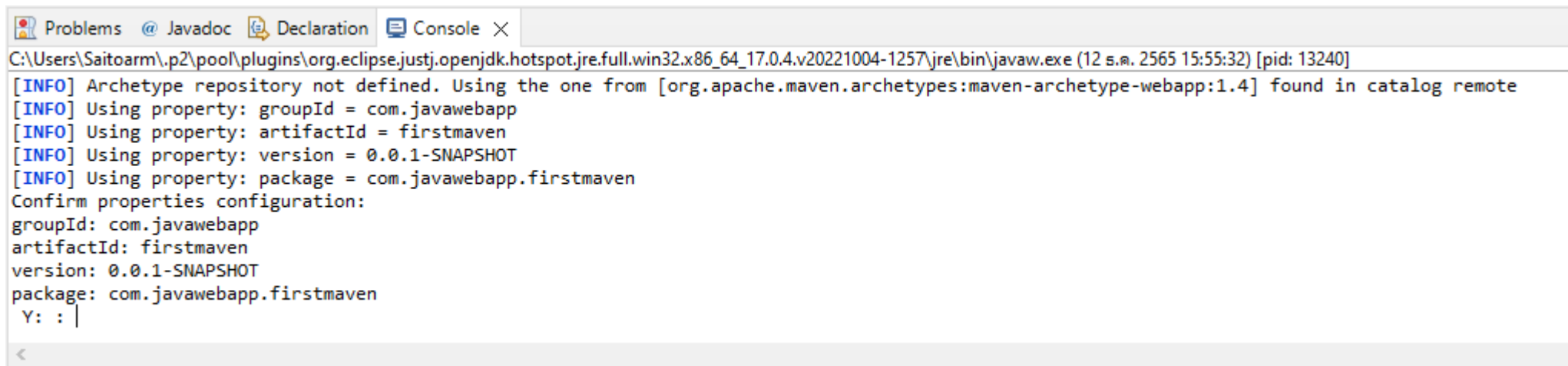
< Back Next > Finish Cancel

ชื่อกลุ่ม Project - อาจใช้ชื่อ domain ขององค์กรเจ้าของ project นำมาเขียนจากหลังไปหน้า

ชื่อ Project ต้องกำหนดเป็นค่า unique

Create Maven

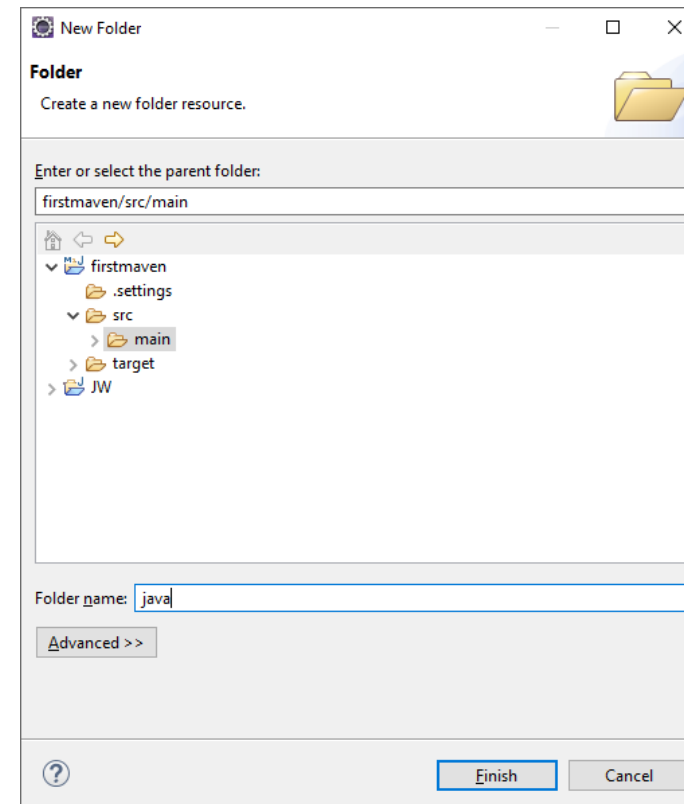
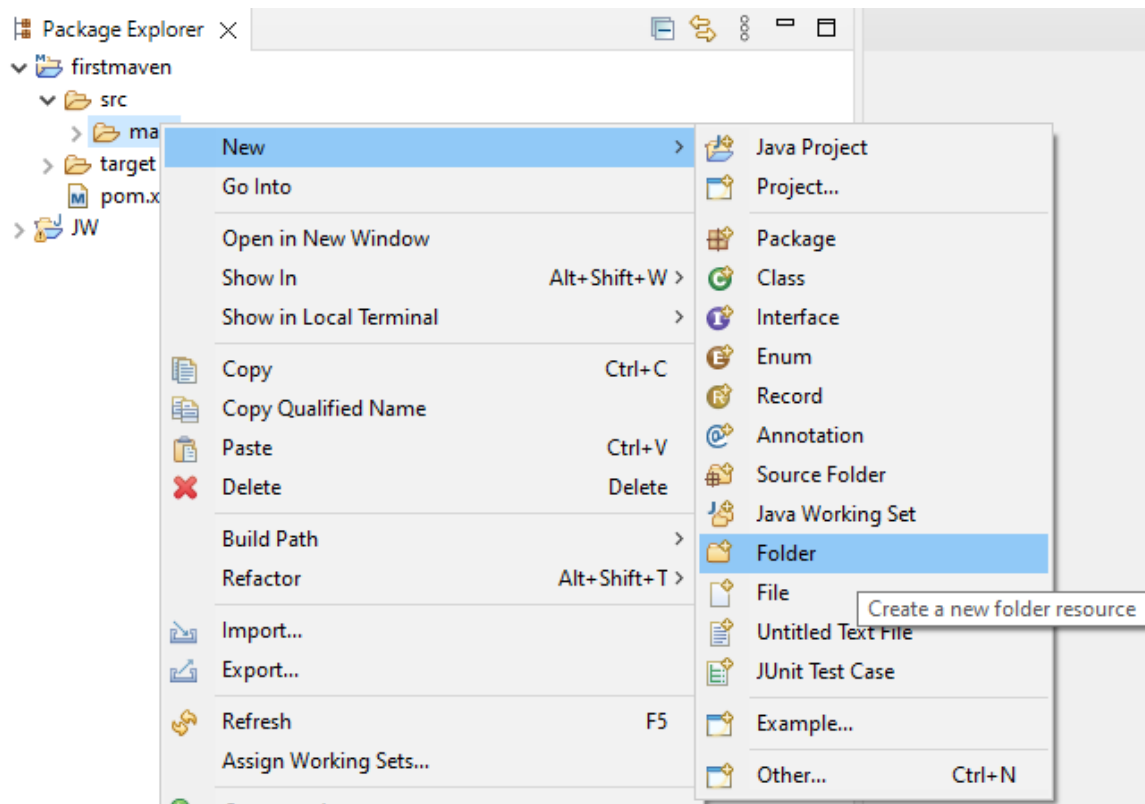
- กด y เพื่อยืนยันการสร้าง maven project



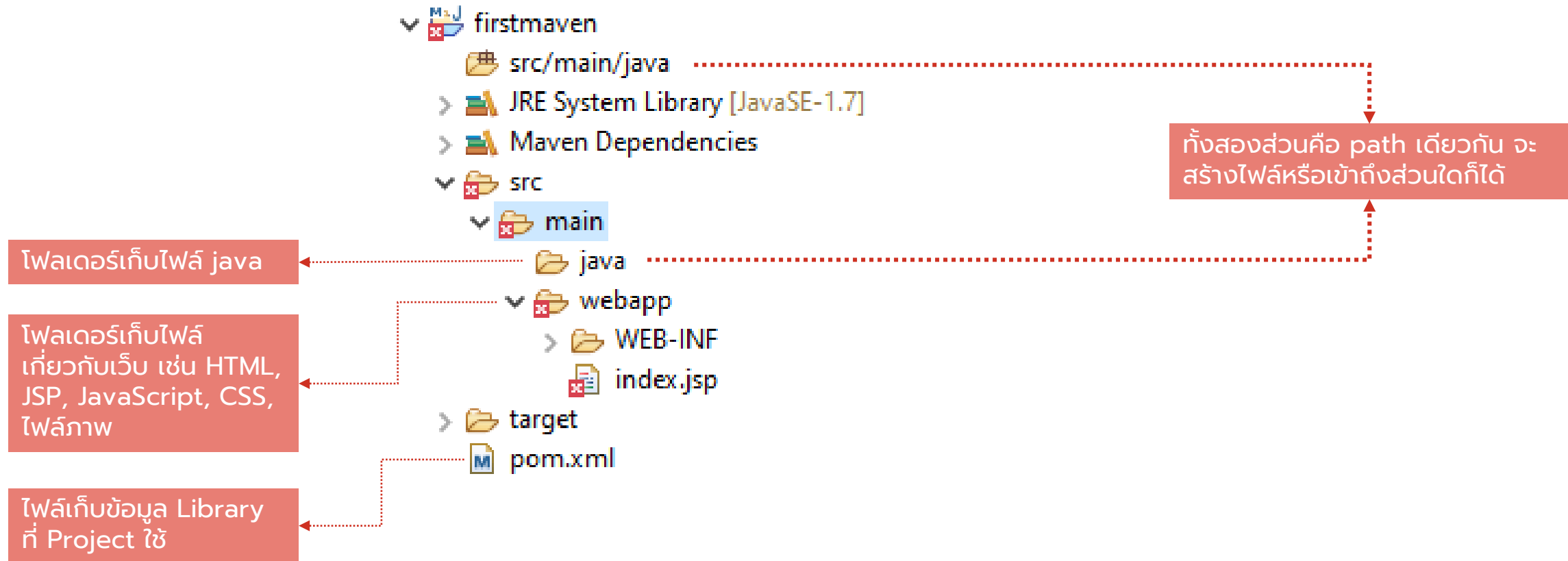
```
Problems @ Javadoc Declaration Console X
C:\Users\Saitoarm\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20221004-1257\jre\bin\javaw.exe (12 ส.ค. 2565 15:55:32) [pid: 13240]
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.4] found in catalog remote
[INFO] Using property: groupId = com.javawebapp
[INFO] Using property: artifactId = firstmaven
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = com.javawebapp.firstmaven
Confirm properties configuration:
groupId: com.javawebapp
artifactId: firstmaven
version: 0.0.1-SNAPSHOT
package: com.javawebapp.firstmaven
Y: : |
<
```

Create Maven

- แม่แบบเว็บแอปพลิเคชันจะไม่สร้างโฟลเดอร์สำหรับเก็บไฟล์ .java มาให้ ให้สร้างโฟลเดอร์ใหม่เอง
ชื่อ java ภายในโฟลเดอร์ main
- คลิกขวาที่โฟลเดอร์ main เลือก New > Folder > ใส่ชื่อโฟลเดอร์ "java"



Maven Project



Maven Project

- การเพิ่มไลบรารีสำหรับ Maven Project สามารถนำไลบรารีมาใช้ โดยหาได้จาก 2 แหล่ง
 - เว็บไซต์ผู้สร้าง Library โดยตรง
 - Central Maven Repository
- การเพิ่ม Library คือ การนำชื่อ และรุ่นของ Library มาเพิ่มข้อมูลลงในไฟล์ pom.xml โดยวางภายในแท็ก `<dependencies></dependencies>`
 - หลังจากบันทึกไฟล์ pom.xml แล้ว Maven จะติดต่อไปยัง Maven Repository เพื่อโหลดไฟล์ที่เกี่ยวข้องมาเก็บไว้ใน Project อัตโนมัติ ดังนั้นหากไม่เคยใช้ Library นั้นในเครื่องที่พัฒนาใช้ จะต้องต่อ internet ด้วย
- การโหลด Library จาก Maven Repository จะเฉพาะครั้งแรกและครั้งเดียวเท่านั้น หากเป็น Library ที่เคยเรียกใช้แล้วจะนำไฟล์จาก cache มาเพิ่มให้โดยไม่มีการดาวน์โหลดใหม่

Maven Project

- การค้นหาชื่อ **Library** จาก Maven Repository สามารถค้นหาชื่อ Library เพื่อนำมากำหนดใช้ใน Project ได้จาก <https://mvnrepository.com/>

The screenshot shows the Maven Repository search results for the query 'jdbc:mysql'. The page displays 4897 results. The top result is 'MySQL Connector Java' (mysql » mysql-connector-java) with 6,673 usages. The description states: 'MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implementation of the MySQL protocol and does not rely on the MySQL client libraries. This driver supports auto-registration with the Driver Manager, standardized validity checks, categorized SQLExceptions, support for large update counts, support for local and offset date-time variants from the java.time package, support for JDBC-4.x XML processing, support for per connection client information and support for the NCHAR, NVARCHAR ...'. The last release is on Oct 18, 2022. The second result is 'AnyLine JDBC MySQL' (org.anyline » anyline-jdbc-mysql) with 23 usages.

Repository

- Central 3.8k
- Sonatype 914
- Spring Plugins 515
- Spring Lib M 476
- Clojars 164
- JCenter 146
- ICM 105
- Spring Lib Release 94

Group

- com.github 455
- org.apache 279
- io.github 171

Found 4897 results

Sort: **relevance** | popular | newest

1. **MySQL Connector Java** 6,673 usages
mysql » mysql-connector-java

MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implementation of the MySQL protocol and does not rely on the MySQL client libraries. This driver supports auto-registration with the Driver Manager, standardized validity checks, categorized SQLExceptions, support for large update counts, support for local and offset date-time variants from the java.time package, support for JDBC-4.x XML processing, support for per connection client information and support for the NCHAR, NVARCHAR ...

Last Release on Oct 18, 2022

2. **AnyLine JDBC MySQL** 23 usages
org.anyline » anyline-jdbc-mysql

The screenshot shows the Maven Repository artifact page for 'mysql:mysql-connector-java'. The page displays a table of versions and their associated vulnerabilities, repository, usages, and dates.

Central (90) | Jahia (1) | Redhat GA (7) | Redhat EA (2) | Imcode (1)

ICM (9)

Version	Vulnerabilities	Repository	Usages	Date
8.0.31		Central	64	Oct 18, 2022
8.0.30		Central	242	Jul 25, 2022
8.0.29		Central	272	Apr 25, 2022
8.0.28		Central	376	Jan 17, 2022
8.0.27	1 vulnerability	Central	281	Oct 18, 2021
8.0.26	2 vulnerabilities	Central	222	Jul 19, 2021
8.0.25	2 vulnerabilities	Central	232	May 10, 2021
8.0.24	2 vulnerabilities	Central	60	Apr 19, 2021
8.0.23	2 vulnerabilities	Central	226	Jan 17, 2021
8.0.22	2 vulnerabilities	Central	304	Oct 17, 2020
8.0.21	2 vulnerabilities	Central	308	Jul 12, 2020

Maven Project



MySQL Connector Java » 8.0.31

MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implementation of with the Driver Manager, standardized validity checks, categorized SQLExceptions, support for JDBC-4.x XML processing, support for per connection client information and support for the NC

Categories	JDBC Drivers
Tags	database sql jdbc driver connector mysql
Date	Oct 18, 2022
Files	pom (2 KB) jar View All
Repositories	Central
Ranking	#68 in MvnRepository (See Top Artifacts) #1 in JDBC Drivers
Used By	6,673 artifacts

Note: This artifact was moved to:

[com.mysql » mysql-connector-j](#)

MySQL Connector/J artifacts moved to reverse-DNS compliant Maven 2+ coordinates.

Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.31</version>
</dependency>
```

☒ Include comment with link to declaration

POM File

Tag ใช้ควบคุม Library
ทั้งหมดใน Project

```
<groupId>com.javawebapp</groupId>  
<artifactId>firstmaven</artifactId>  
<version>0.0.1-SNAPSHOT</version>  
<packaging>war</packaging>
```

ข้อมูล Project

```
<name>firstmaven Maven Webapp</name>
```

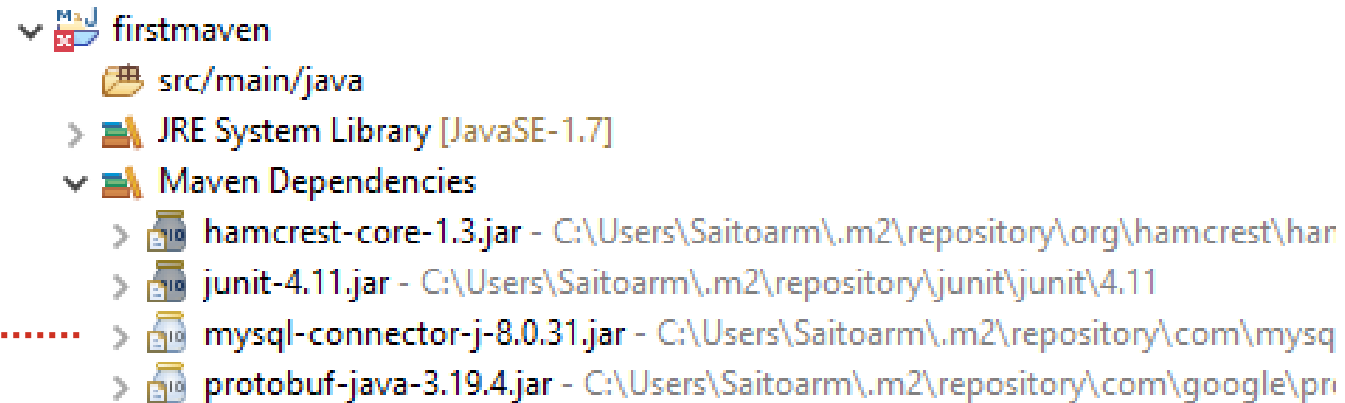
```
<dependencies>
```

```
<dependency>  
  <groupId>junit</groupId>  
  <artifactId>junit</artifactId>  
  <version>4.11</version>  
  <scope>test</scope>  
</dependency>
```

Tag ที่ใช้ครอบคลุม 1 Library

```
</dependencies>
```


POM File



Library ที่ Maven
โหลดมาให้อัตโนมัติ

- หากมีการลบแท็ก <dependency> ของไลบรารีใดออก ไฟล์ .jar ก็จะถูกดึงออกจาก Project แบบอัตโนมัติ
- แต่ไฟล์ .jar นั้นยังเก็บพัก (cache) ไว้ในโปรแกรม Eclipse อยู่จะถูกนำกลับมาใช้ได้เมื่อมีการเพิ่มไลบรารีอีกครั้งใน Project ใด ๆ ก็ได้
- เมื่อได้ Library ที่ต้องการแล้ว ไฟล์ต่างๆจะอยู่ใน Class Path อัตโนมัติ
- สามารถ import ในโค้ดโปรแกรม อ้างอิงคลาสจาก Library ที่ Maven โหลดมาให้ได้ทันที

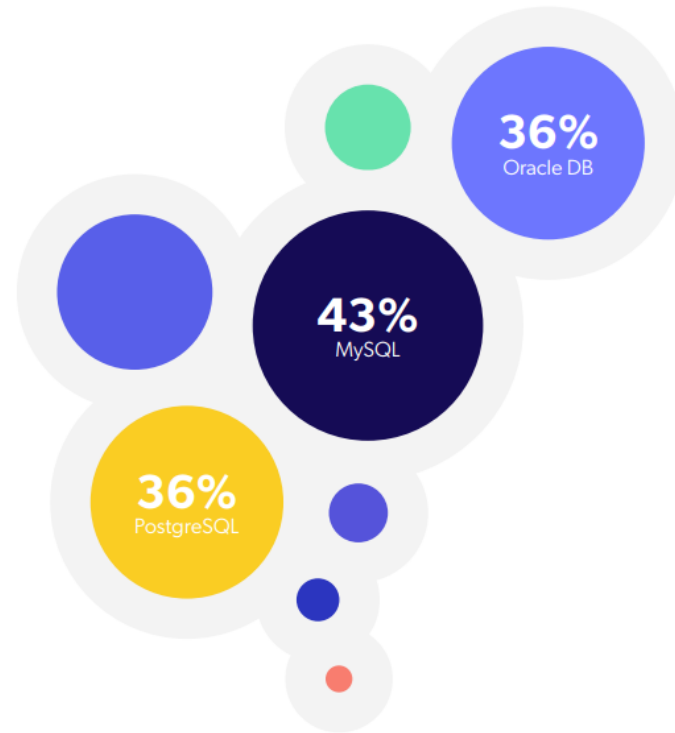
Java Database Connectivity (JDBC)

SC363204

Java Web Application Development
การพัฒนาโปรแกรมประยุกต์บนเว็บด้วยภาษาจาวา



What databases are you using?



MySQL **43%**

Oracle DB **36%**

PostgreSQL **36%**

MongoDB **29%**

Other SQL **16%**

Other NoSQL **11%**

Cassandra **8%**

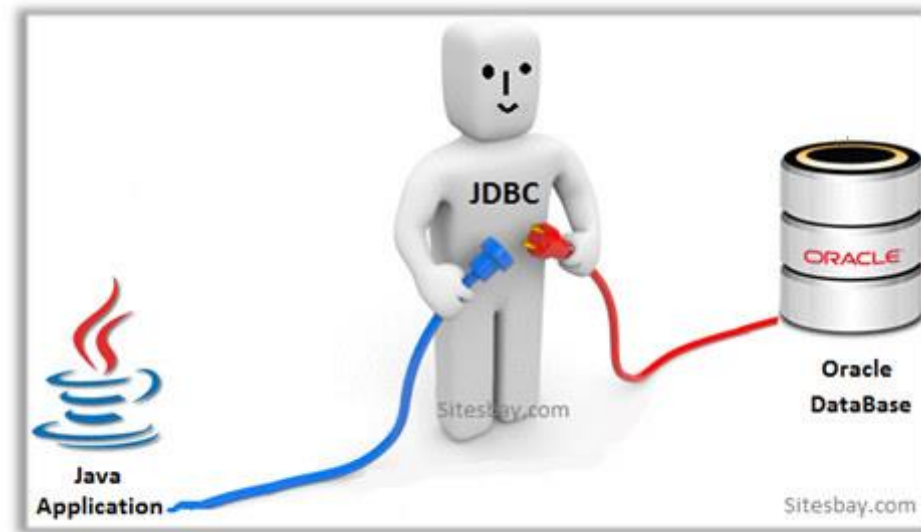
Neo4j **5%**

JDBC

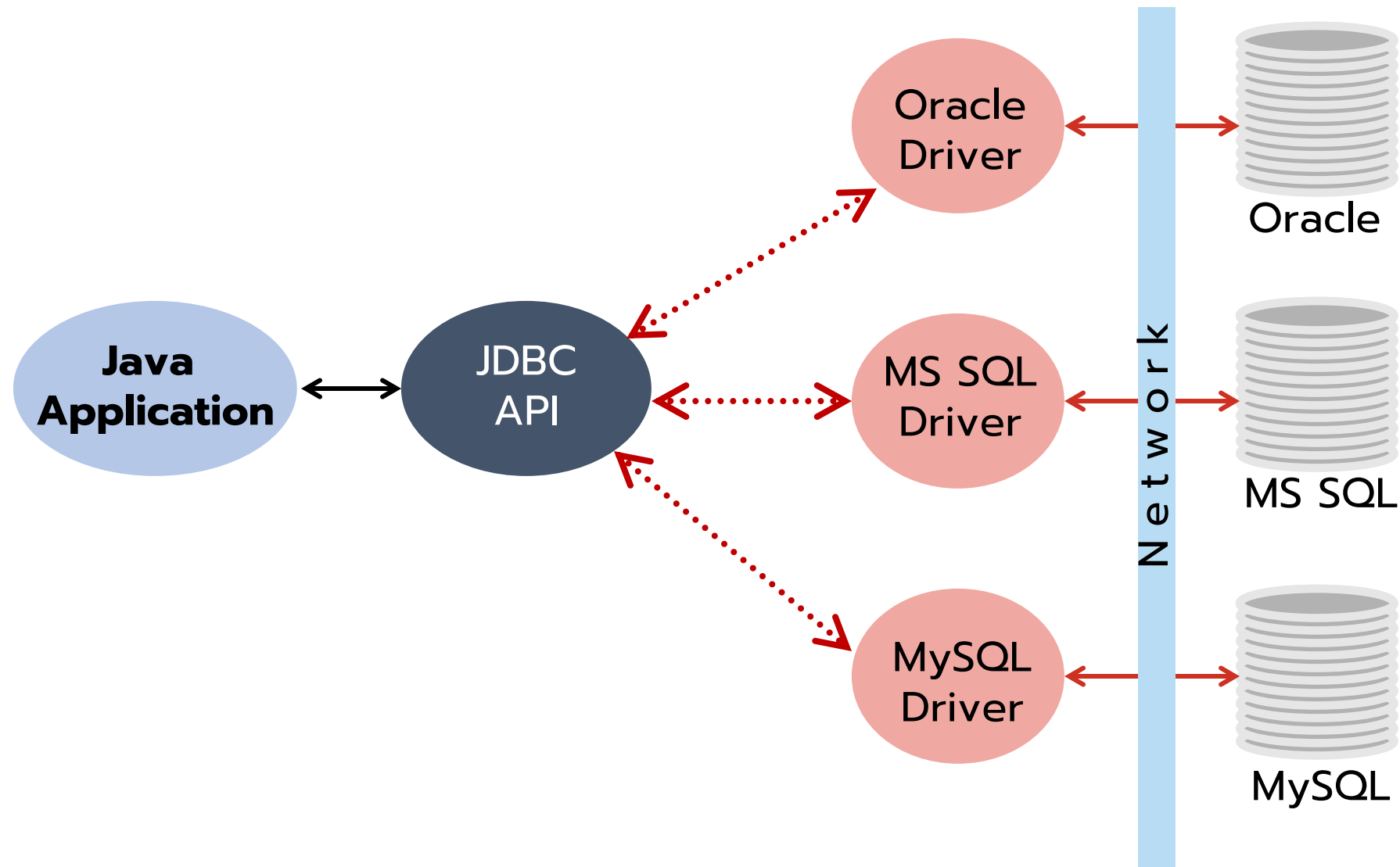
- JDBC คือ ชุดคำสั่งมาตรฐาน (Library หรือ API) สำหรับทำงานกับฐานข้อมูล
- JDBC กำหนดให้มีคำสั่งสำหรับการจัดการฐานข้อมูล
 - สร้างฐานข้อมูล
 - ดึงข้อมูล
 - เพิ่ม ลบ และแก้ไข
- JDBC อยู่ใน package ชื่อ `java.sql` (อยู่ใน Java Standard Edition)

JDBC Driver

- JDBC Driver คือ ชุดคำสั่งที่ผู้ผลิตซอฟต์แวร์ฐานข้อมูล implement ตามข้อกำหนดในของ JDBC เพื่อสนับสนุนการเชื่อมต่อกับฐานข้อมูลของตนเอง
- JDBC Driver จะอยู่ในรูปแบบของไฟล์ที่บีบอัดในนามสกุล .jar เพื่อให้นักพัฒนานำไปเก็บไว้ในแอปพลิเคชัน หรืออ้างอิงถึง



JDBC Architecture



JDBC Driver

- MySQL

<https://dev.mysql.com/downloads/connector/j/>

- Oracle

<http://www.oracle.com/technetwork/database/features/jdbc/>

- Microsoft SQL Server

<https://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx>

Tag ใช้ควบคุม Library
ทั้งหมดใน Project

```
<groupId>com.javawebapp</groupId>  
<artifactId>firstmaven</artifactId>  
<version>0.0.1-SNAPSHOT</version>  
<packaging>war</packaging>
```

ข้อมูล Project

```
<name>firstmaven Maven Webapp</name>
```

```
<dependencies>
```

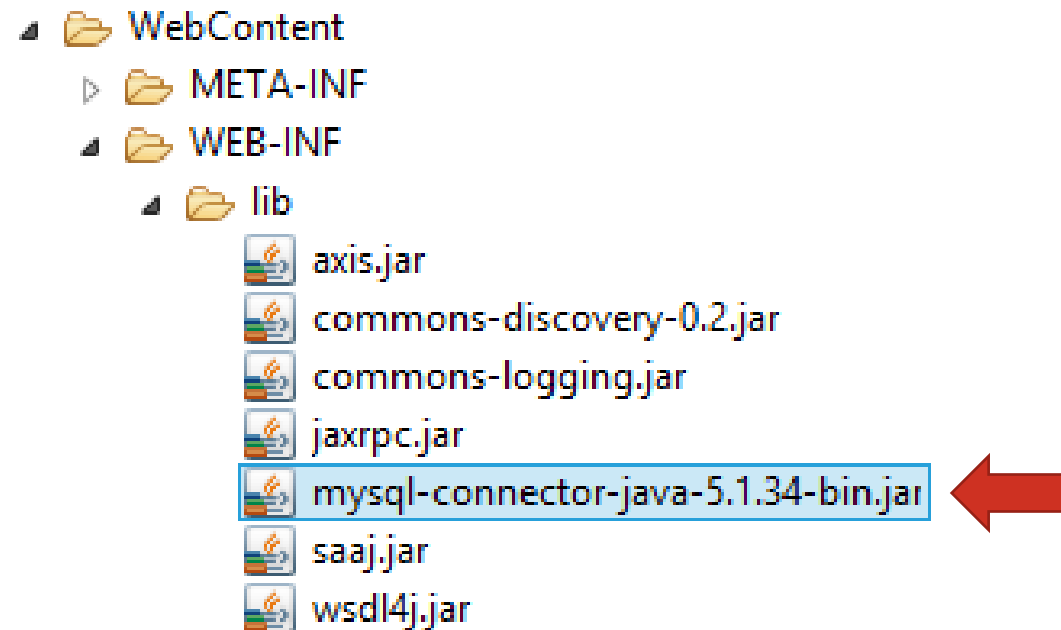
```
<dependency>  
  <groupId>junit</groupId>  
  <artifactId>junit</artifactId>  
  <version>4.11</version>  
  <scope>test</scope>  
</dependency>
```

Tag ที่ใช้ครอบคลุม 1 Library

```
</dependencies>
```


JDBC Driver

- การเพิ่ม JDBC Driver ใน Project
- โฟลเดอร์ที่ใช้เก็บ Library (ไฟล์ .jar) สำหรับโปรเจก จะอยู่ใน **WebContent/WEB-INF/lib**
- เมื่อโหลด JDBC Driver ของ MySql แล้ว สามารถลากวาง .jar ในโฟลเดอร์นี้ได้เลย เพื่อให้โปรเจกรู้จัก



JDBC Process

1. โหลด JDBC Driver
2. กำหนด URL สำหรับติดต่อกับฐานข้อมูล
3. เชื่อมต่อกับฐานข้อมูล
4. เตรียมคำสั่ง SQL ที่จะประมวลผล
5. ส่งคำสั่ง SQL ไปยังฐานข้อมูล
6. อ่านผลลัพธ์ที่ฐานข้อมูลส่งกลับ
7. ปิดการเชื่อมต่อ

1. โหลด JDBC Driver

- คลาสหลักสำหรับ JDBC Driver จะอยู่ใน package ที่แตกต่างกันไปขึ้นอยู่กับผู้ผลิตฐานข้อมูล โดยใช้เมธอด `Class.forName()` ดังนี้

`Class.forName("ชื่อ package และคลาส JDBC Driver");`

- การโหลดจะต้องดัก Exception ชื่อ `ClassNotFoundException` เสมอ

```
try {  
    // 1. โหลด JDBC Driver  
    Class.forName("com.mysql.cj.jdbc.Driver");  
    ...  
} catch (ClassNotFoundException e) {  
    System.err.println("Error loading driver: " + e);  
}
```

1. โหลด JDBC Driver

- ชื่อคลาส JDBC Driver ของฐานข้อมูลต่างๆ

- MySQL

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

- Oracle

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

- Microsoft SQL Server

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
```

2. กำหนด URL สำหรับติดต่อกับฐานข้อมูล

- URL สำหรับติดต่อกับฐานข้อมูล เป็น **String** ซึ่งมีรูปแบบดังนี้

String **dbURL** = "jdbc:mysql://localhost/blueshop?characterEncoding=utf-8";

The diagram illustrates the components of the JDBC URL "jdbc:mysql://localhost/blueshop?characterEncoding=utf-8" using brackets and labels:

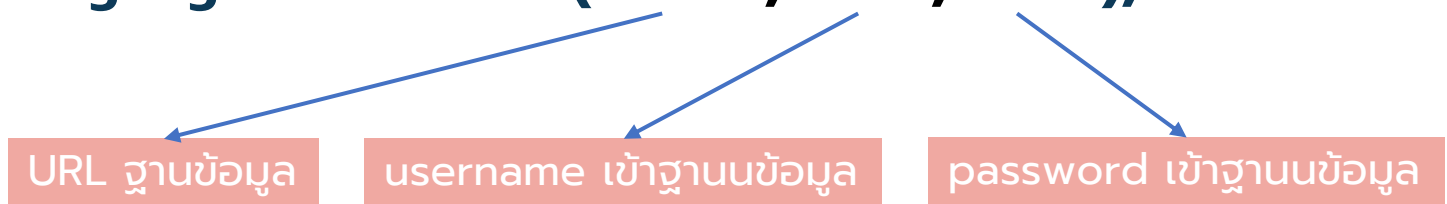
- ชื่อ Protocol**: jdbc:mysql
- ชื่อ domain หรือ ip ของ database server**: localhost
- ชื่อฐานข้อมูล**: blueshop
- ชื่อชุดอักขระ**: characterEncoding=utf-8

3. เชื่อมต่อกับฐานข้อมูล

- ประกาศ object ของคลาส Connection สำหรับการเชื่อมต่อ
- เรียกเมธอด getConnection() จากคลาส DriverManager ซึ่งมีอาร์กิวเมนต์ดังนี้

```
String dbURL = "jdbc:mysql://localhost/blueshop?characterEncoding=utf-8";
```

```
Connection con = DriverManager.getConnection(dbURL, "root", "1234");
```



4. เตรียมคำสั่ง SQL ที่จะประมวลผล

- ประกาศตัวแปร **PreparedStatement** และเรียกเมธอด **prepareStatement()** ของ **Connection Object** โดยส่งรูปแบบคำสั่ง SQL ไปด้วย
- หลังจากนั้นกำหนดค่าให้กับ **?** ในรูปแบบ SQL ด้วยเมธอด **setXxx()**

```
Connection con = DriverManager.getConnection(dbURL,"root","1234");
```

```
PreparedStatement pStatement = con.prepareStatement("SELECT *  
FROM product WHERE pid = ? or pname LIKE ?");
```

```
pStatement.setInt(1, 3);
```

```
pStatement.setString(2, "g%");
```

argument แรก แทน
ลำดับของ ? ในประโยค
SQL

argument ที่สอง แทนค่าที่
ต้องการนำไปแทนในประโยค
SQL

การใช้เมธอดกำหนดค่าให้กับ **?**

- ใช้ **setInt()** เมื่อฟิลด์ในฐานข้อมูลเป็นตัวเลข
- ใช้ **setString()** เมื่อฟิลด์ในฐานข้อมูลเป็น
ชุดอักขระ

5. ส่งคำสั่ง SQL ไปยังฐานข้อมูล

- เรียกเมธอด `executeQuery()` จาก `PreparedStatement` Object
- ประกาศตัวแปรของคลาส `ResultSet` เพื่อรับค่าผลลัพธ์

```
ResultSet resultSet = pStatement.executeQuery();
```



Object สำหรับเก็บ
ผลลัพธ์ที่ฐานข้อมูลส่งกลับ

6. อ่านผลลัพธ์ที่ฐานข้อมูลส่งกลับ

- ผลลัพธ์ที่ได้จากฐานข้อมูลจะถูกเก็บลง **ResultSet** Object
- สามารถโหลดข้อมูลที่ละแถวด้วยเมธอด **next()**
- เข้าถึงข้อมูลในแต่ละคอลัมน์ด้วยเมธอด **getXxx()**

```
while (resultSet.next()) {  
    int pid = resultSet.getInt("pid");  
    String pname = resultSet.getString("pname");  
    String pdetail = resultSet.getString("pdetail");  
    int price = resultSet.getInt("price");  
    System.out.println(pid + "," + pname + "," + pdetail + "," + price);  
}
```

ดึงข้อมูลออกมาเป็น **Integer**

ดึงข้อมูลออกมาเป็น **String**

ระบุชื่อคอลัมน์

6. ปิดการเชื่อมต่อ

- เรียกเมธอด `close()` จาก `Connection` Object เพื่อหยุดการเชื่อมต่อกับฐานข้อมูล
- การปิด `Connection` ควรทำทุกครั้ง เพื่อเพิ่มประสิทธิภาพการทำงานกับฐานข้อมูล

```
// 1. โหลดคลาส JDBC Driver
Class.forName("com.mysql.cj.jdbc.Driver");

// 2. กำหนด URL สำหรับติดต่อกับฐานข้อมูล
String dbURL = "jdbc:mysql://localhost/blueshop?characterEncoding=utf-8";

// 3. สร้าง Connection
Connection con = DriverManager.getConnection(dbURL, "root", "1234");

.....

// 7. ปิดการเชื่อมต่อ
con.close();
```

```

import java.sql.*;
public class JDBCStep {
    private static Connection con; // ประกาศตัวแปรสำหรับใช้ติดต่อฐานข้อมูล
    public static void main(String[] args) {
        try {
            // 1. โหลด JDBC Driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // 2. กำหนด URL สำหรับติดต่อกับฐานข้อมูล
            String dbURL = "jdbc:mysql://localhost/blueshop?characterEncoding=utf-8";

            // 3. เชื่อมต่อกับฐานข้อมูล
            con = DriverManager.getConnection(dbURL, "root", "1234");

            // 4. เตรียมคำสั่ง SQL ที่จะประมวลผล
            PreparedStatement pStatement = con.prepareStatement("SELECT * FROM product WHERE pid = ? or pname LIKE ?");
            pStatement.setInt(1, 3);
            pStatement.setString(2, "g%");

            // 5. ส่งคำสั่ง SQL ไปยังฐานข้อมูล
            ResultSet resultSet = pStatement.executeQuery();

            // 6. อ่านผลลัพธ์ที่ฐานข้อมูลส่งกลับ
            while (resultSet.next()) {
                int pid = resultSet.getInt("pid");
                String pname = resultSet.getString("pname");
                String pdetail = resultSet.getString("pdetail");
                int price = resultSet.getInt("price");
                System.out.println(pid + "," + pname + "," + pdetail + "," + price);
            }

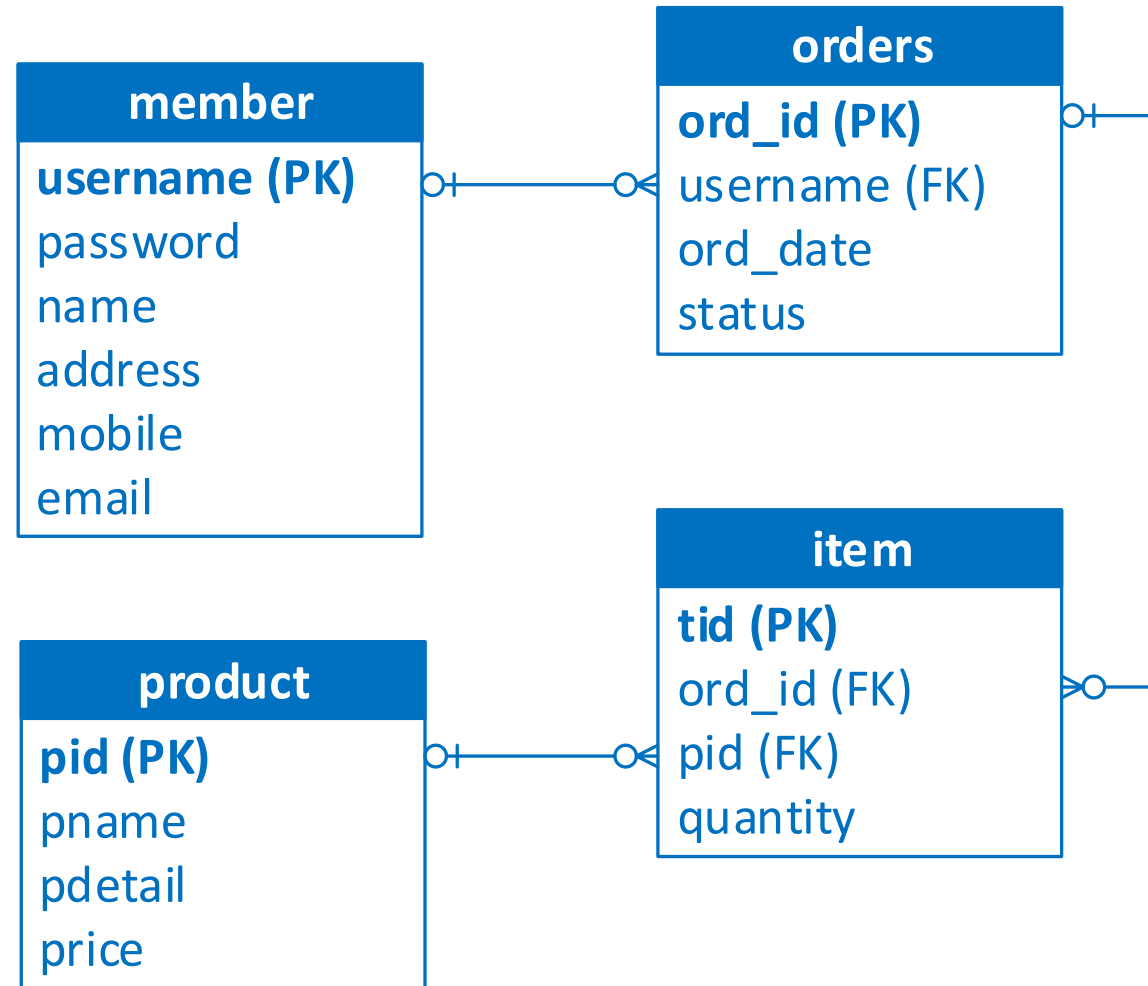
            // 7. ปิดการเชื่อมต่อ
            con.close();

        } catch (ClassNotFoundException e) {
            System.err.println("Error loading driver: " + e);
        } catch (SQLException e) {
            System.err.println("Error database connection: " + e);
        }
    }
}

```



blueshop DB



blueshop DB

ตาราง member

username	password	name	address	mobile	email
somsak	1899	สมศักดิ์ สุรเสถียร	174 ถ.มิตรภาพ จ.ขอนแก่น		somsak@gmail.com
baramee	aaaff1	บารมี บุญหลาย	123 ถ.วิภาวดีรังสิต กรุงเทพฯ	08-9446-9955	baramee@gmail.com
metasit	m345	เมธาสิทธิ์ สอนสั่ง	98/9 ถ.ศรีจันทร์ จ.ขอนแก่น	08-4456-9877	metasit@outlook.com

ตาราง product

pid	pname	pdetail	price
1	Centrum	วิตามินรวมจาก A ถึง Zinc	350
2	Caltrate	บำรุงกระดูก เสริมวิตามินดี	760
3	Ester-C	วิตามินซี 500 mg ไม่กัดกระเพาะ	500
4	Glucosamine	บำรุงข้อต่อ ป้องกันข้อเสื่อม	1200

ตาราง item

tid	ord_id	pid	quantity
1	1	2	2
2	1	3	5
3	1	4	1
4	2	1	2
5	2	3	4
6	2	4	3
7	3	2	3
8	3	4	5
9	4	1	5
10	4	3	1

ตาราง orders

ord_id	username	ord_date	status
1	baramee	16/7/2013 23:25	wait
2	metasit	12/2/2013 23:25	pay
3	baramee	27/12/2013 23:26	send
4	metasit	11/12/2013 23:27	pay

Insert/Update/Delete

- การใช้คำสั่ง SQL ประเภทดึงข้อมูล จะใช้เมธอด `executeQuery()` แต่การใช้คำสั่ง SQL ที่เป็น

การ **Insert/Update/Delete** จะใช้เมธอด `executeUpdate()`

การใช้คำสั่งดึงข้อมูล

Object สำหรับเก็บ
ผลลัพธ์ที่ฐานข้อมูล
ส่งกลับ

```
PreparedStatement pStatement = con.prepareStatement("SELECT * FROM product WHERE pid = ?");  
pStatement.setInt(1, 3);  
ResultSet resultSet = pStatement.executeQuery( );
```

String ที่อยู่ในรูป
คำสั่ง SQL ประเภท
ดึงข้อมูล

การใช้คำสั่งเพิ่ม/ลบ/แก้ไขข้อมูล

```
PreparedStatement pStatement = con.prepareStatement("UPDATE product SET price=? WHERE pid=?");  
pStatement.setInt(1, 3000);  
pStatement.setInt(2, 3);  
int row = pStatement.executeUpdate( );
```

เก็บจำนวนแถวที่มี
ผลกระทบ

String ที่อยู่ในรูปคำสั่ง
SQL ประเภท
Insert/Update/Delete