

JAVA PROGRAMMING BASICS

Module 3: Java Standard Edition

Training program

1. Java I/O Streams
2. Java Serialization
3. **Java Database Connectivity**
4. Java GUI Programming
5. The basics of Java class loaders
6. Reflections
7. Annotations
8. The proxy classes
9. Java Software Development
10. Garbage Collection

Module contents

Java Data Base Connectivity

- An Introduction to Relation Data Bases
- An Introduction to SQL
- The CRUD Operations
- The JDBC
- The DriverManager and Connection
- The Statement and ResultSet
- PreparedStatement and CallableStatement
- The Database Transactions
- The Database MetaData

Module contents

Java Data Base Connectivity

- An Introduction to Relation Data Bases
- An Introduction to SQL
- The CRUD Operations
- The JDBC
- The DriverManager and Connection
- The Statement and ResultSet
- PreparedStatement and CallableStatement
- The Database Transactions
- The Database MetaData

Persistence concept

- **Persistence** means the ability of an object to exist longer than the process that created it.
- Persistence is achieved by storing object status data in files and databases.



Data Persistence

Module contents

Java Data Base Connectivity

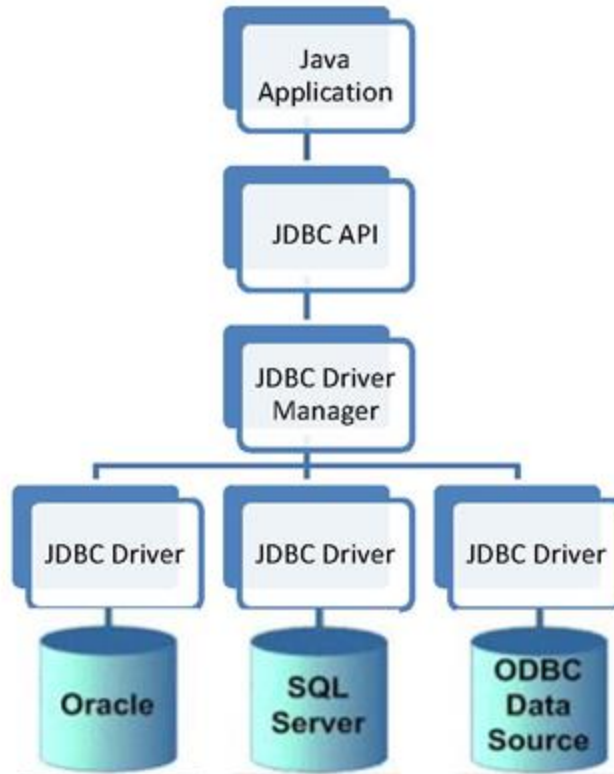
- An Introduction to Relation Data Bases
- An Introduction to SQL
- The CRUD Operations
- **The JDBC**
- The DriverManager and Connection
- The Statement and ResultSet
- PreparedStatement and CallableStatement
- The Database Transactions
- The Database MetaData

The JDBC

- **The Java Database Connectivity (JDBC)**
- The Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases [SQL databases and other tabular data sources, such as spreadsheets or flat files
- The JDBC API provides a call-level API for SQL-based database access

The JDBC

- A JDBC driver is a software component enabling a Java application to interact with a database



JDBC API Components

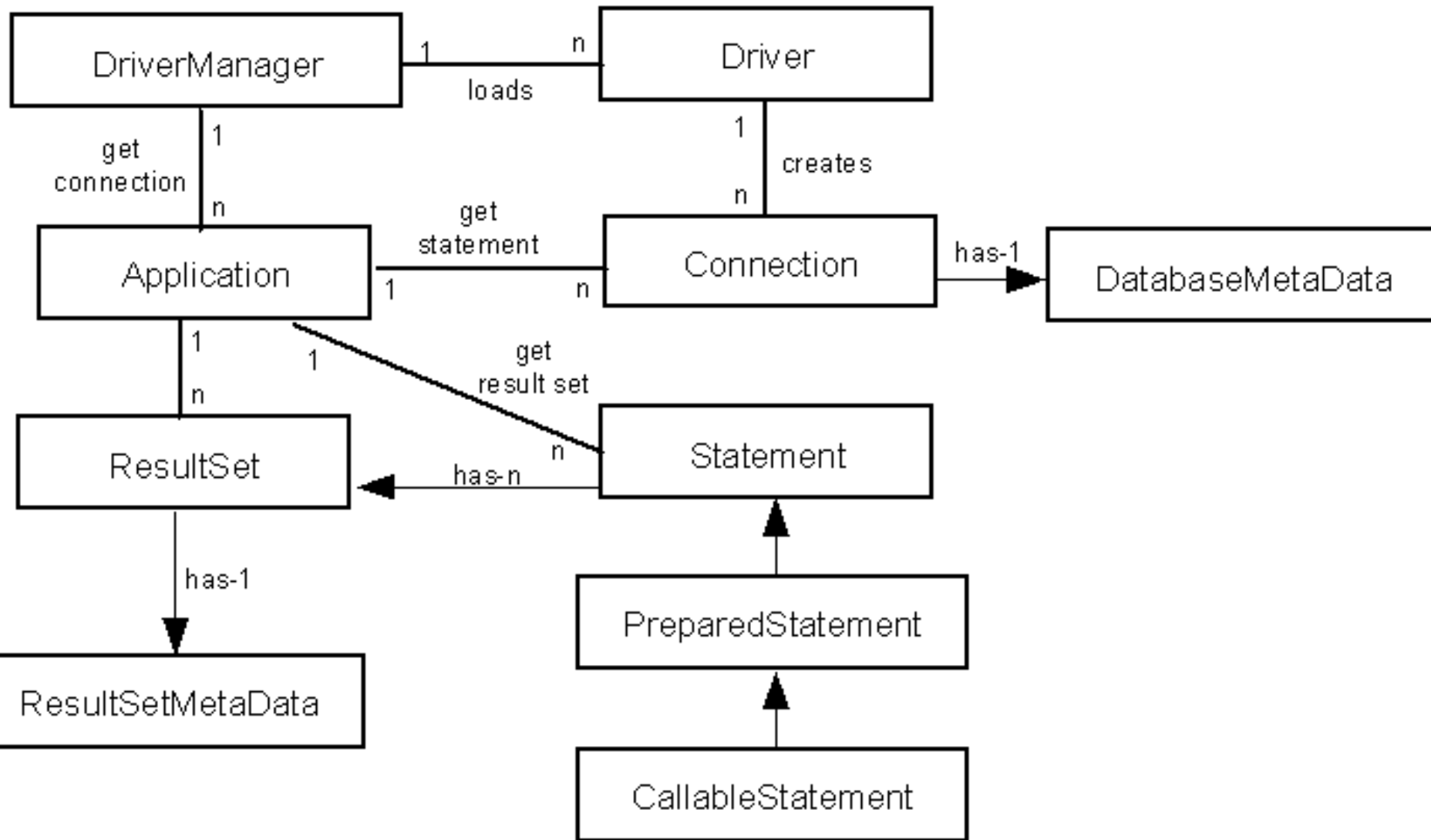
- Class `java.sql.DriverManager` manages the list of database drivers registered in the Java application and creates a connection to the database using the appropriate database driver data.
 - Interface `java.sql.Driver` is implemented by all database drivers, responsible for communication with the database. Interface `java.sql.Connection` provides methods for working with the database.
 - Interface `java.sql.Statement` - implementations of this interface are used to execute static SQL queries and get the results of these queries.
- ...

JDBC API Components

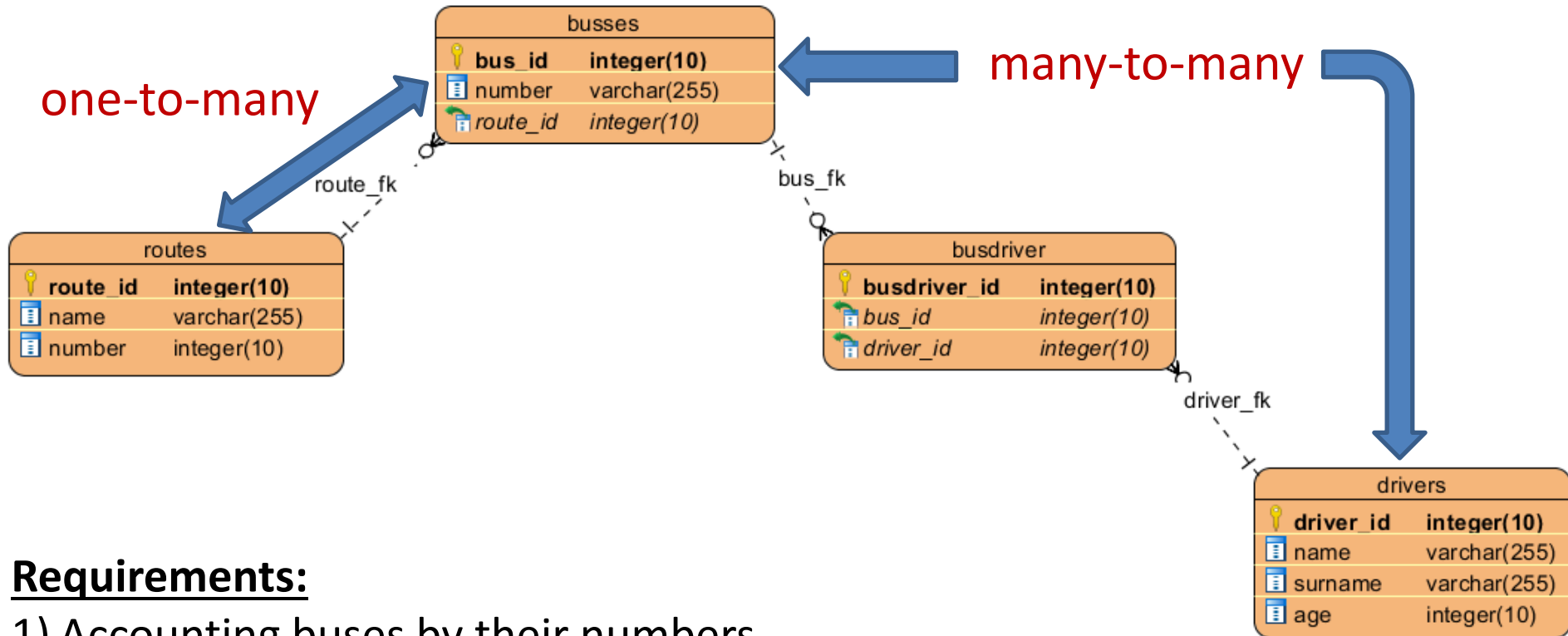
...

- Interface `java.sql.PreparedStatement` - implementations of this interface are used to execute SQL-queries that contain basic parameters (denoted by the symbol `?` in the body of the query), and get the results of these queries.
- Interface `java.sql.CallableStatement` - implementations of this interface are used to call stored procedures.
- Interface `java.sql.ResultSet` - implementations of this interface contain data that was obtained as a result of executing an SQL query.
- Class `java.sql.SQLException` handles errors that can occur when using the JDBC API.

java.sql package



jdbcstudy ERD



Requirements:

- 1) Accounting buses by their numbers.
- 2) Accounting routes by their names: starting point - end point and route number.
- 3) Accounting bus drivers by name, surname and age.
- 4) Several buses can travel on the same route at once.
- 5) One driver can drive different buses.
- 6) One bus can be driven by different drivers.

jdbcstudy database creation

The screenshot displays the MySQL Workbench interface for creating a foreign key. The 'SCHEMAS' tree on the left shows the 'jdbcstudy' database selected, with the 'buses' table and its columns ('bus_id', 'number', 'route_id') visible. The 'Foreign Key' tab is active, showing a table with the following data:

Foreign Key Name	Referenced Table
buses_ibfk_1	'jdbcstudy`.`routes`

The 'Foreign Key Options' section is highlighted with a red box and contains the following settings:

- On Update: RESTRICT
- On Delete: RESTRICT
- Skip in SQL generation

The 'Foreign Key Comment' field is empty. At the bottom right, the 'Apply' button is circled in red, and a red arrow points to it.

jdbcstudy database

SCHEMAS

Filter objects

- autopark
- hibernatestudy
- jdbcstudy
 - Tables
 - busdriver
 - buses
 - Columns
 - bus_id
 - number
 - route_id
 - Indexes
 - PRIMARY
 - route_id
 - Foreign Keys
 - buses_ibfk_1
 - Triggers

SCHEMAS

Filter objects

- jdbcstudy
 - Tables
 - busdriver
 - buses
 - drivers

SCHEMAS

Filter objects

- jdbcstudy
 - Tables
 - busdriver
 - Columns
 - busdriver_id
 - bus_id
 - driver_id
 - Indexes
 - PRIMARY
 - bus_id
 - driver_id
 - Foreign Keys
 - busdriver_ibfk_1
 - busdriver_ibfk_2
 - Triggers

SCHEMAS

Filter objects

- jdbcstudy
 - Tables
 - busdriver
 - buses
 - drivers
 - routes
 - Columns
 - route_id
 - name
 - number
 - Indexes
 - PRIMARY
 - Foreign Keys
 - Triggers

jdbcstudy database dump

```
C:\>mysqldump -u root -p jdbcstudy > jdbcstudy.sql
```

```
CREATE TABLE `drivers` (  
  `driver_id` int NOT NULL PRIMARY KEY,  
  `name` varchar(255) NOT NULL,  
  `surname` varchar(255) NOT NULL,  
  `age` int NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
  COLLATE = utf8mb4_unicode_ci;
```

```
LOCK TABLES `drivers` WRITE;  
INSERT INTO `drivers` VALUES  
(1,'Микола','Іванченко',45),(2,'Ігор','Дорошенко',35),  
(3,'Артур','Пиріжков',29),(4,'Петро','Гасанов',23),(5,'Андрій','Пе  
тренко',57),  
(6,'Олег','Головін',41);  
UNLOCK TABLES;
```

jdbcstudy database dump

```
...  
CREATE TABLE `routes` (  
  `route_id` int NOT NULL PRIMARY KEY,  
  `name` varchar(255) NOT NULL,  
  `number` int NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE = utf8mb4_unicode_ci;
```

```
LOCK TABLES `routes` WRITE;  
INSERT INTO `routes` VALUES (1,'вул. Милославська-ст. м.  
"Героїв Дніпра"',39),  
(2,'Кінотеатр "Братислава"-вул.Светлицького',157),  
(3,'Контрактова площа-Залізничний вокзал "Дарниця"',74),  
(4,'ст. м. "Чернігівська"-ст. м. "Політехнічний інститут"',388),  
(5,'ст. м. "Васильківська"-вул. Смелянська',114),(6,'Рибальський  
півострів-вул. Північна',11),  
(7,'ст. м. "Виставковий центр"-с. Жуляни',7);  
UNLOCK TABLES;
```


jdbcstudy database dump

...

```
CREATE TABLE `buses` (  
  `bus_id` int NOT NULL PRIMARY KEY,  
  `number` varchar(255) NOT NULL,  
  `route_id` int,  
  FOREIGN KEY (`route_id`) REFERENCES routes(`route_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
  COLLATE = utf8mb4_unicode_ci;
```

```
LOCK TABLES `buses` WRITE;  
INSERT INTO `buses` VALUES  
(1,'к254тр',1),(2,'т436ку',7),(3,'п398тм',2),(4,'п387тс',4),  
(5,'о567рп',2),(6,'а576рн',1),(7,'ч265ек',2),(8,'т764дл',5),(9,'п765  
ра',5),(10,'г459пи',3),  
(11,'ц255ка',7),(12,'р536ка', 6);  
UNLOCK TABLES;
```

jdbcstudy database dump & restore




...

```
CREATE TABLE `busdriver` (  
  `busdriver_id` int NOT NULL PRIMARY KEY,  
  `bus_id` int NOT NULL,  
  `driver_id` int NOT NULL,  
  FOREIGN KEY (`bus_id`) REFERENCES buses(`bus_id`),  
  FOREIGN KEY (`driver_id`) REFERENCES drivers(`driver_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
  COLLATE = utf8mb4_unicode_ci;
```

```
LOCK TABLES `busdriver` WRITE;  
INSERT INTO `busdriver` VALUES  
(1,1,1),(2,1,4),(3,2,1),(4,2,2),(5,3,2),(6,3,3),  
(7,4,3),(8,4,4),(9,5,3),(10,6,5),(11,7,5),(12,8,5),(13,9,1),(14,9,4),(15  
,10,2),(16,11,3);  
UNLOCK TABLES;
```


```
C:\>mysql -u root -p jdbcstudy < jdbcstudy.sql
```

Add JDBC driver to project

→    dev.mysql.com/downloads/connector/j/

MySQL Community Downloads

← Connector/J

General Availability (GA) Releases Archives 

Connector/J 8.3.0

Select Operating System:

Platform Independent

Select Operating System...

Ubuntu Linux

Debian Linux

SUSE Linux Enterprise Server

Red Hat Enterprise Linux / Oracle Linux

Fedora

Platform Independent

Source Code

Platform Independent (Architecture Independent),

4.1M	Download
f3920e62be749cde1f61e3e1e Signature	
4.8M	Download
(mysql-connector-j-8.3.0.zip)	MD5: 8b4e005c4371adb851bf070c4365fa30 Signature

ZIP Archive

We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

Add ConnectorJ driver to plain project of IDE

Create & test database connection with JDBC

```
package connection;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
```

```
public class DriverManagerDemo {
```

```
    public static void main(String[] args) {
```

```
        final String DB_CONNECTION = "jdbc:mysql://localhost:3306/jdbcstudy";
```

```
        final String DB_USER = "root";
```

```
        final String DB_PASSWORD = "root";
```

```
        try (Connection connection = DriverManager.getConnection(DB_CONNECTION,  
                                                                DB_USER, DB_PASSWORD)) {
```

```
            System.out.println("The " + connection.getCatalog() + " database is connected");
```

```
        } catch (SQLException ex) {
```

```
            ex.printStackTrace();
```

```
        }
```

```
    }
```

```
}
```

Java Build Tools, Maven Essentials

- Java ecosystem is dominated with three build tools:
- Apache Ant with Ivy
- Maven
- Gradle



ANT - build.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="AntStudyApp" default="archive" >
  <target name="init">
    <mkdir dir="build/classes" />
    <mkdir dir="dist" />
  </target>
  <target name="compile" depends="init" >
    <javac includeantruntime="false"
      srcdir="src" destdir="build/classes"/>
  </target>
  <target name="archive" depends="compile" >
    <jar destfile="dist/project.jar"
      basedir="build/classes"/>
  </target>
  <target name="clean" depends="init">
    <delete dir="build" />
    <delete dir="dist" />
  </target>
</project>
```

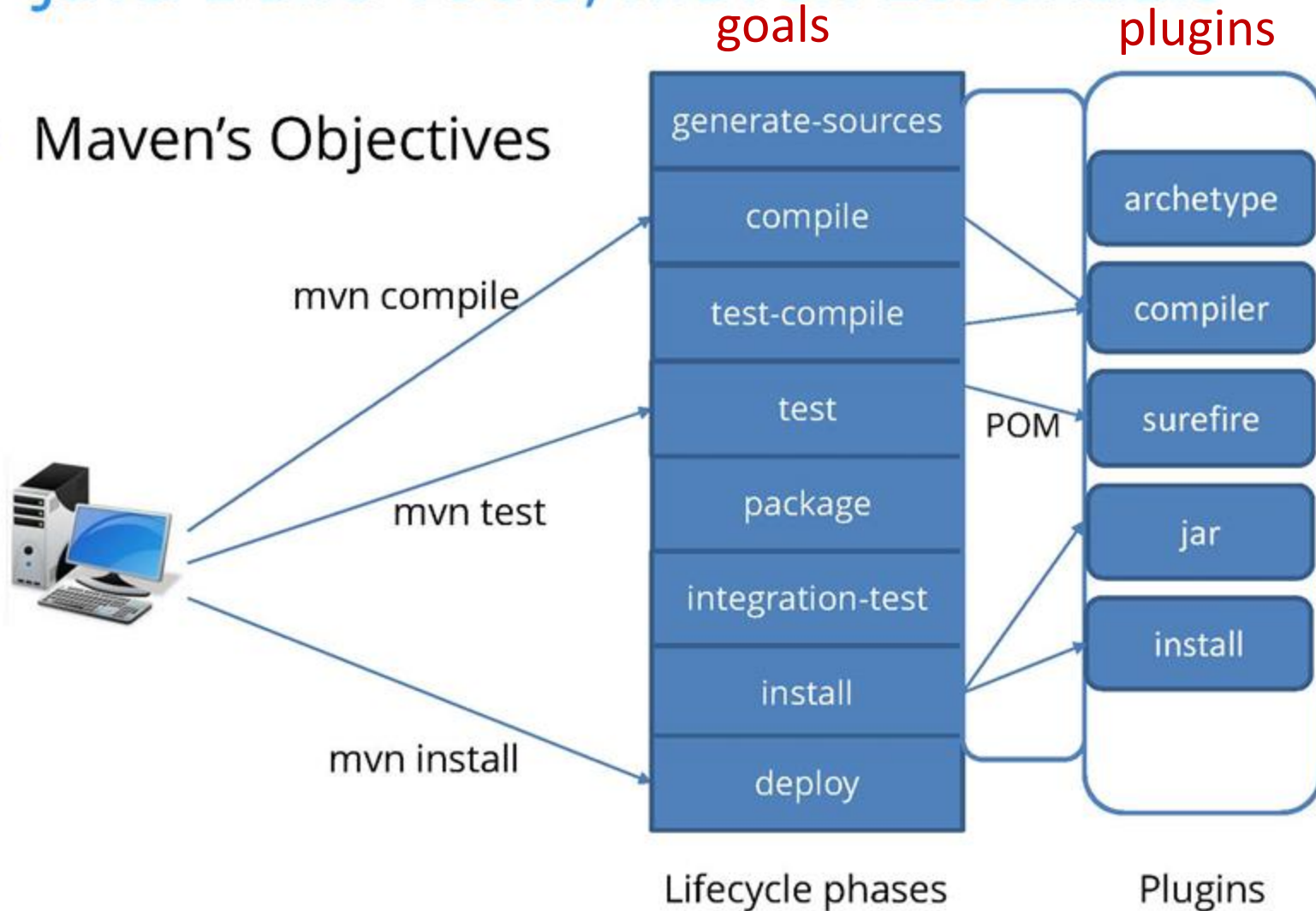
Java Build Tools, Maven Essentials



- Maven's Objectives
- Maven's primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. In order to attain this goal there are several areas of concern that Maven attempts to deal with:
 - Making the build process easy
 - Providing a uniform build system
 - Providing quality project information
 - Providing guidelines for best practices development
 - Allowing transparent migration to new features

Java Build Tools, Maven Essentials

- Maven's Objectives



`mvn help:describe -Dcmd=PHASENAME`

Maven using

```
c:\Users\kgp\Dropbox\EDU\KT\JDBCStudyApp>mvn clean
```

```
[INFO] Scanning for projects...
```

```
[INFO]
```

```
[INFO] -----< ua.edu.znu:jdbcstudy >-----
```

```
[INFO] Building jdbcstudy 1.0-SNAPSHOT
```

```
[INFO]   from pom.xml
```

```
[INFO] -----[ jar ]-----
```

```
[INFO]
```

```
[INFO] --- clean:3.2.0:clean (default-clean) @ jdbcstudy ---
```

```
[INFO]
```

```
[INFO] BUILD SUCCESS
```

```
[INFO]
```

```
[INFO] Total time: 0.509 s
```

```
[INFO] Finished at: 2024-01-24T14:51:04+02:00
```

```
[INFO]
```

```
[INFO]
```

```
c:\Users\kgp\Dropbox\EDU\KT\JDBCStudyApp>mvn compile
```

```
[INFO] Scanning for projects...
```

```
[INFO]
```

```
[INFO] -----< ua.edu.znu:jdbcstudy >-----
```

```
[INFO] Building jdbcstudy 1.0-SNAPSHOT
```

```
[INFO]   from pom.xml
```

```
[INFO] -----[ jar ]-----
```

```
[INFO]
```

```
[INFO] --- resources:3.3.0:resources (default-resources) @ jdbcstudy ---
```

```
[INFO] Copying 4 resources
```

```
[INFO]
```

```
[INFO] --- compiler:3.10.1:compile (default-compile) @ jdbcstudy ---
```

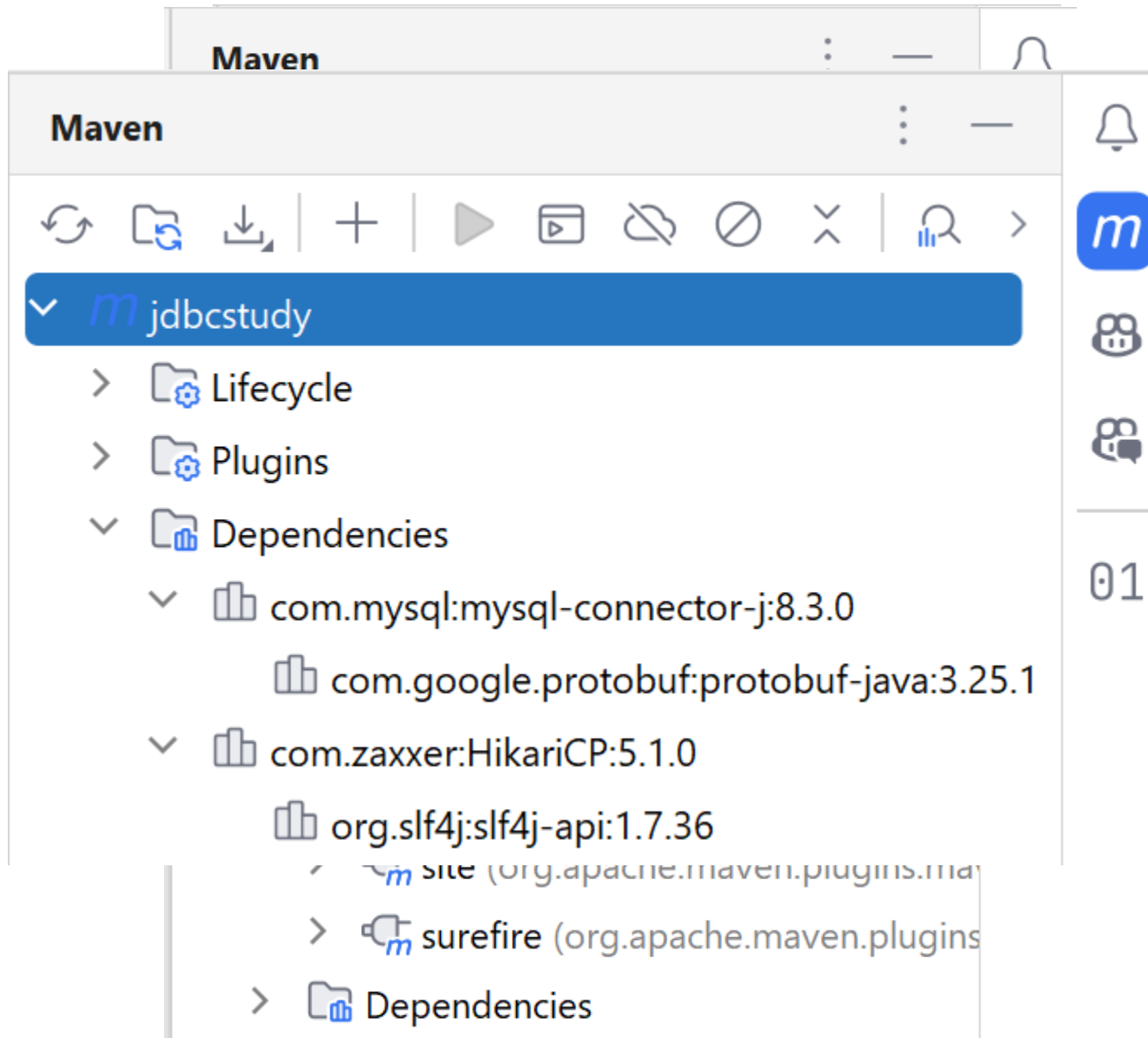
```
[INFO] Changes detected - recompiling the module!
```

```
[INFO] Compiling 9 source files to c:\Users\kgp\Dropbox\EDU\KT\JDBCStudyApp\target\classes
```

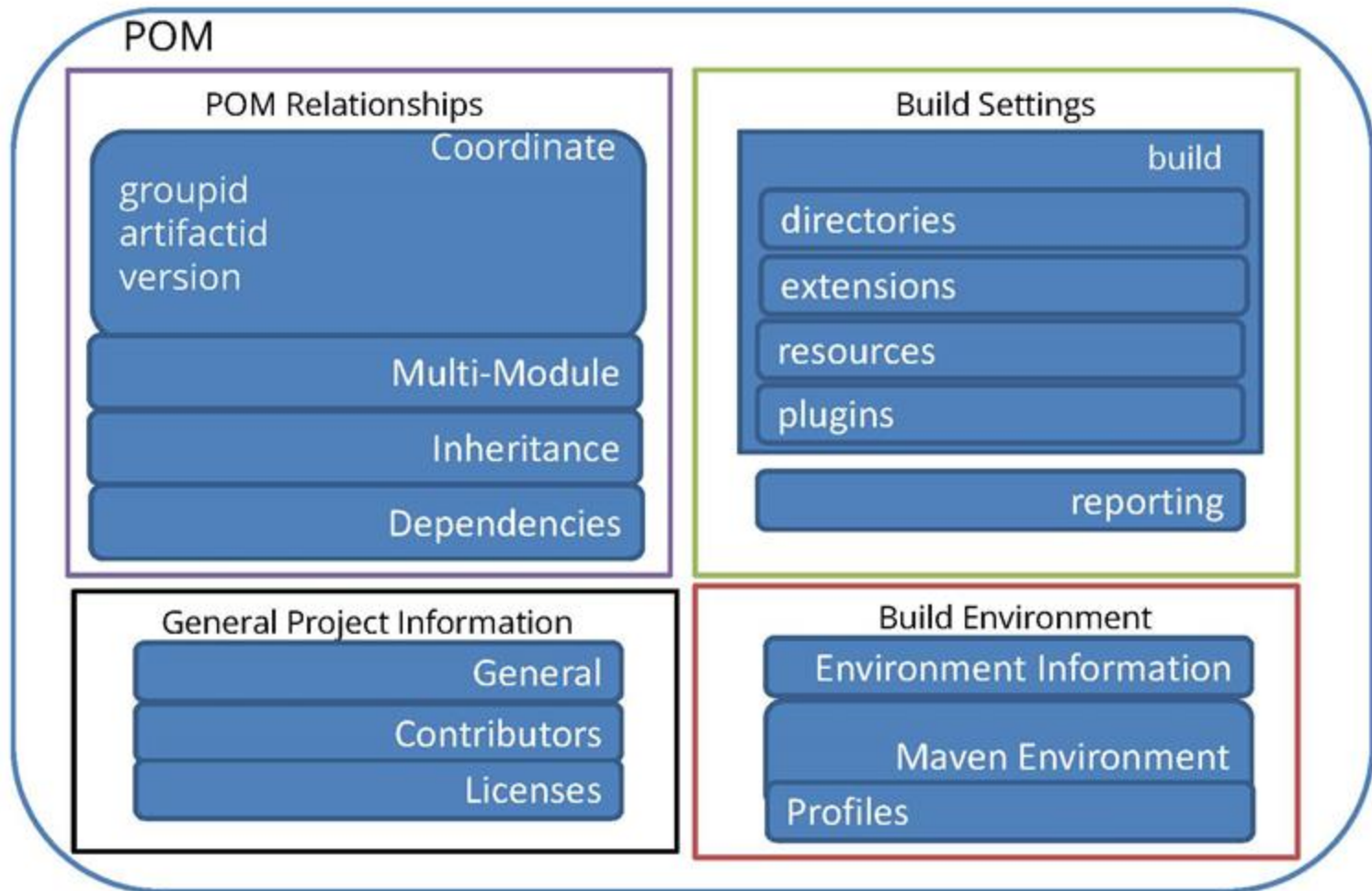
```
[INFO]
```

```
[INFO] BUILD SUCCESS
```

IDE Maven tools



Java Build Tools, Maven Essentials



Maven Project Object Model (POM) File

```
<project xmlns=http://maven.apache.org/POM/4.0.0
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation=
"http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany</groupId>
  <artifactId>MavenStudyApp</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>MavenStudyApp</name>
  <description>
    Project for Maven studying
  </description>
```

...

Maven Project Object Model (POM) File

...

```
<dependencies>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-engine
    </artifactId>
    <version>5.7.0</version>
    <scope>test</scope>
  </dependency>
</dependencies>
<build>
  <plugins>
```













...

Maven Project Object Model (POM) File

...














```
    <plugin>
      <groupId>org.apache.maven.plugins
      </groupId>
      <artifactId>maven-surefire-plugin
      </artifactId>
      <version>3.0.0-M5</version>
    </plugin>
  </plugins>
</build>
</project>
```

maven-archetype-quickstart

- ▼  MavenStudyApp
 - >  .settings
 - ▼  src
 - ▼  main
 - ▼  java
 -  Main.java
 -  resources
 - ▼  test
 - ▼  java
 -  MainTest.java
 -  resources
 - ▼  target

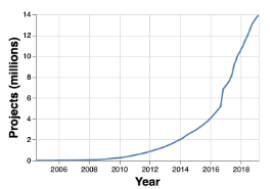
...

maven-archetype-quickstart

- ...
- ▼  target
 - ▼  classes
 - >  META-INF
 -  Main.class
 - >  maven-archiver
 - >  maven-status
 - >  surefire-reports
 - ▼  test-classes
 -  MainTest.class
 -  MavenStudyApp-0.0.1-SNAPSHOT.jar
 -  .classpath
 -  .project
 -  pom.xml

Maven repository

Indexed Artifacts (37.4M)



Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- JVM Languages
- Language Runtime
- Core Utilities
- Mocking
- Web Assets
- Annotation Libraries

Home » Repositories » Central



Central Repository
https://repo1.maven.org/maven2/

global repo

URL	https://repo1.maven.org/maven2/
Storage	39038.4 GBs
Packages	12,910,217 indexed packages

Published Packages by Year

Year	Count
2024	146,540
2023	2,433,196
2022	2,251,027
2021	2,045,263
2020	1,435,839
2019	1,223,821
2018	929,167
2017	712,956

commons-digester logkit
commons-fileupload mysql
commons-httpclient nekohtml

local repo


Indexed Repositories (1984)

- Central
- Atlassian
- Sonatype
- Hortonworks
- Spring Plugins
- Spring Lib M
- JCenter
- JBossEA
- Atlassian Public
- KtorEAP

Popular Tags

- aar android apache api application
- arm assets aws build build-system bundle client clojure
- cloud commons config cran data
- database eclipse example extension

Maven-project creation





 New Project ✕

🔍


New Project

Empty Project

Generators

-  Maven Archetype
-  Spring 
-  Compose for Desktop

Name:


Location: 

Project will be created in: ~\OneDrive ...g Academy\Java-HOC\JDBCStudy

Create Git repository

Language: Java Kotlin Groovy HTML


Build system: IntelliJ Maven Gradle


JDK: 


Add sample code

Generate code with onboarding tips

▼ Advanced Settings

GroupId: 

ArtifactId: 



Maven-project pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>ua.edu.znu</groupId>
  <artifactId>jdbcstudy</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>21</maven.compiler.source>
    <maven.compiler.target>21</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <dependency>
      <groupId>com.mysql</groupId>
      <artifactId>mysql-connector-j</artifactId>
      <version>8.3.0</version>
    </dependency>
  </dependencies>
</project>
```

Module contents

Java Data Base Connectivity

- An Introduction to Relation Data Bases
- An Introduction to SQL
- The CRUD Operations
- The JDBC
- **The DriverManager and Connection**
- The Statement and ResultSet
- PreparedStatement and CallableStatement
- The Database Transactions
- The Database MetaData

The JDBC

- Processing SQL Statements with JDBC

0. Load database driver



1. getConnection()



2. createStatement()



3a. SELECT: executeQuery()



5. close()



3b. INSERT/UPDATE/DELETE:
executeUpdate()

4b. int



5. close()

4a(i) next()

4a(ii) getXxx()

Data

5. close()



See connection package

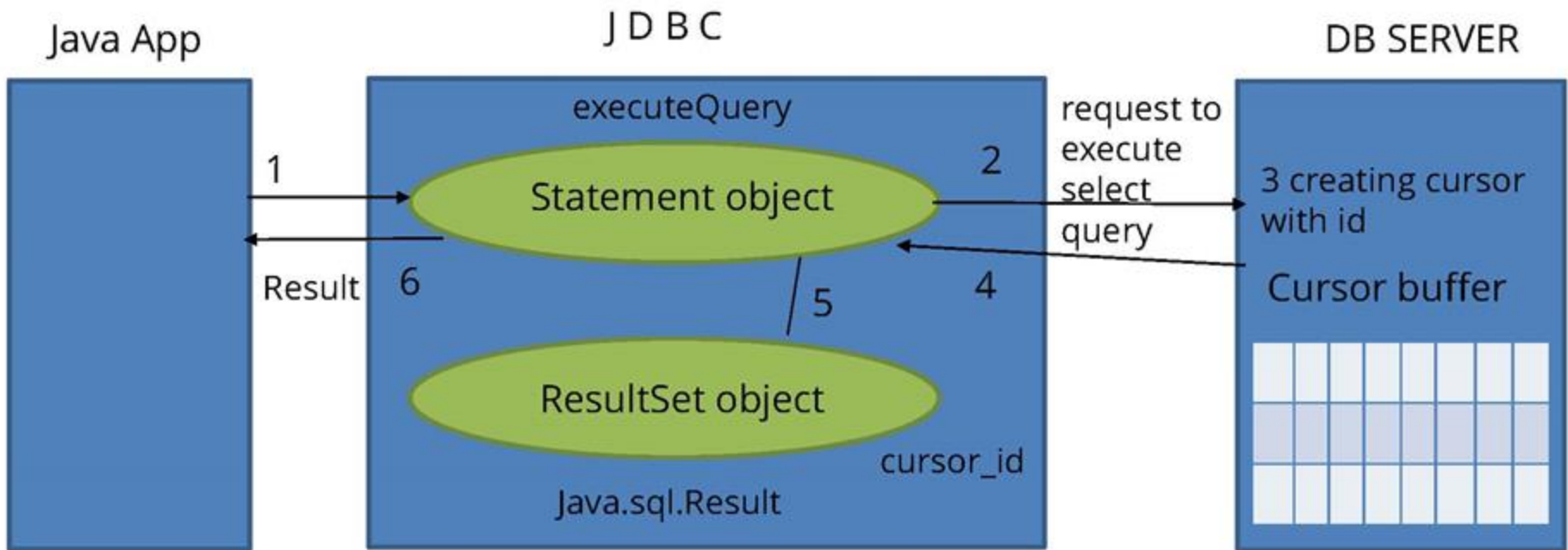
Module contents

Java Data Base Connectivity

- An Introduction to Relation Data Bases
- An Introduction to SQL
- The CRUD Operations
- The JDBC
- The DriverManager and Connection
- **The Statement and ResultSet**
- PreparedStatement and CallableStatement
- The Database Transactions
- The Database MetaData

The Statement and ResultSet

- The **Statement** used for executing a static SQL statement and returning the results it produces



See [statementresultset/StatementResultSetDemo](#)

Statement execute query methods



executeQuery()

This method is used to execute the SQL statements which retrieve some data from the database.

This method returns a ResultSet object which contains the results returned by the query.

This method is used to execute only select queries.

Ex : SELECT

executeUpdate()

This method is used to execute the SQL statements which update or modify the database.

This method returns an int value which represents the number of rows affected by the query. This value will be the 0 for the statements which return nothing.

This method is used to execute only non-select queries.

Ex :
DML → INSERT, UPDATE and DELETE
DDL → CREATE, ALTER

execute()

This method can be used for any kind of SQL statements.

This method returns a boolean value. TRUE indicates that query returned a ResultSet object and FALSE indicates that query returned an int value or returned nothing.

This method can be used for both select and non-select queries.

This method can be used for any type of SQL statements.

The ResultSet

Statement createStatement(**int resultSetType**,
int resultSetConcurrency) throws SQLException

ResultSet.TYPE_FORWARD_ONLY (default value) - the cursor can only move forward;

ResultSet.TYPE_SCROLL_INSENSITIVE - the cursor can be moved forward and backward, as well as set to any position. The ResultSet object does not reflect data changes in the database after its creation;

ResultSet.TYPE_SCROLL_SENSITIVE - the cursor can be moved forward and backward, as well as set to any position. The ResultSet object reflects changes to the database after it is created.

ResultSet.CONCUR_READ_ONLY - the created ResultSet object cannot be modified by its updateTYPE* methods;

ResultSet.CONCUR_UPDATABLE - The created ResultSet object can be modified by its updateTYPE methods.

* **TYPE updateTYPE (String columnLabel)** throws SQLException or **TYPE updateTYPE (int columnIndex)** throws SQLException, where TYPE, including numeric Object, Array, Date and others of this interface

The ResultSet

next	Moves the Cursor to the next row in your table. If there are no more rows in the table, a value of False will be returned.
previous	Moves the Cursor back one row in your table. If there are no more rows in the table? A value of False will be returned.
first	Moves the Cursor to the first row in your table.
last	Moves the Cursor to the last row in your table.
absolute	Moves the Cursor to a particular row in the table. So absolute(5) will move the Cursor to row number 5 in the table

See [statementresultset\ResultSetCursorDemo](#)

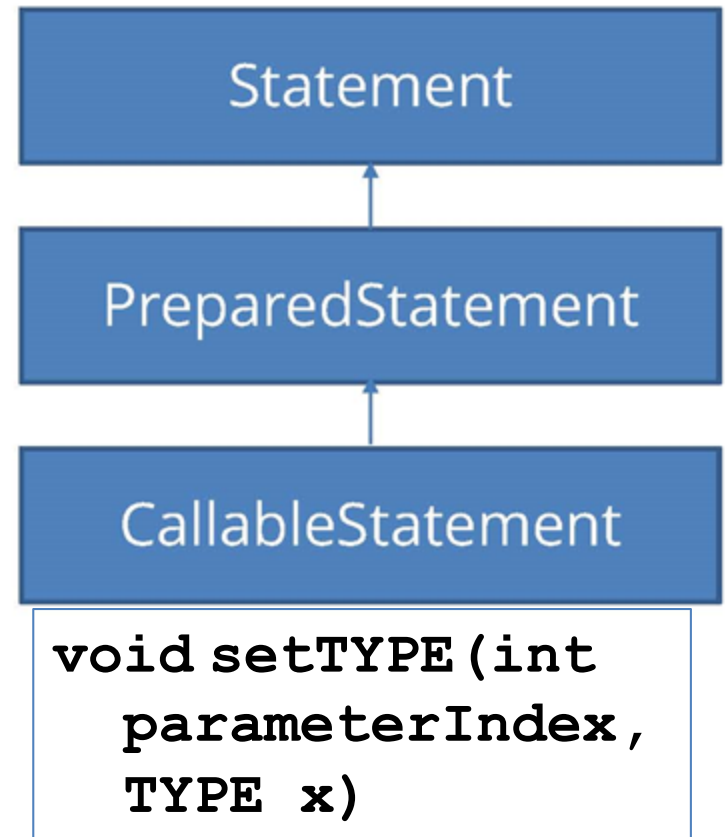
Module contents

Java Data Base Connectivity

- An Introduction to Relation Data Bases
- An Introduction to SQL
- The CRUD Operations
- The JDBC
- The DriverManager and Connection
- The Statement and ResultSet
- **PreparedStatement and CallableStatement**
- The Database Transactions
- The Database MetaData

PreparedStatement

- A SQL statement is precompiled and stored in a PreparedStatement object.
- This object can then be used to efficiently execute this statement multiple times.



See [preparedstatement\PreparedStatementDemo](#)

CallableStatement

```
USE `jdbcstudy`;  
DROP PROCEDURE IF EXISTS getDriverName;  
  
CREATE PROCEDURE getDriverName  
(IN DRIVER_ID INT, OUT DRIVER_SURNAME VARCHAR(50), OUT  
DRIVER_NAME VARCHAR(50))  
SELECT surname, `name` INTO DRIVER_SURNAME, DRIVER_NAME  
FROM drivers  
WHERE driver_id = DRIVER_ID  
LIMIT 1;
```

IN - parameter, the value of which is known at the time when the request is created;
OUT - parameter whose value is returned by SQL - query;
INOUT - parameter that uses input and output values.

CallableStatement

Connection: jdbc:mysql://localhost:3306/jdbcstudy?zeroDateTimeBehavior=CONVERT_TO_NU..

```
1 SHOW PROCEDURE STATUS;
```

SHOW PROCEDURE STATUS ×

Max. rows: 100 | Fetched Rows: 27 | Matching Rows:

#	Db	Name	Type	Definer	Modif
1	jdbcstudy	getDriverName	PROCEDURE	root@localhost	2020-1
2	sys	create_synonym_db	PROCEDURE	mysql.sys@localhost	2020-1
3	sys	diagnostics	PROCEDURE	mysql.sys@localhost	2020-1

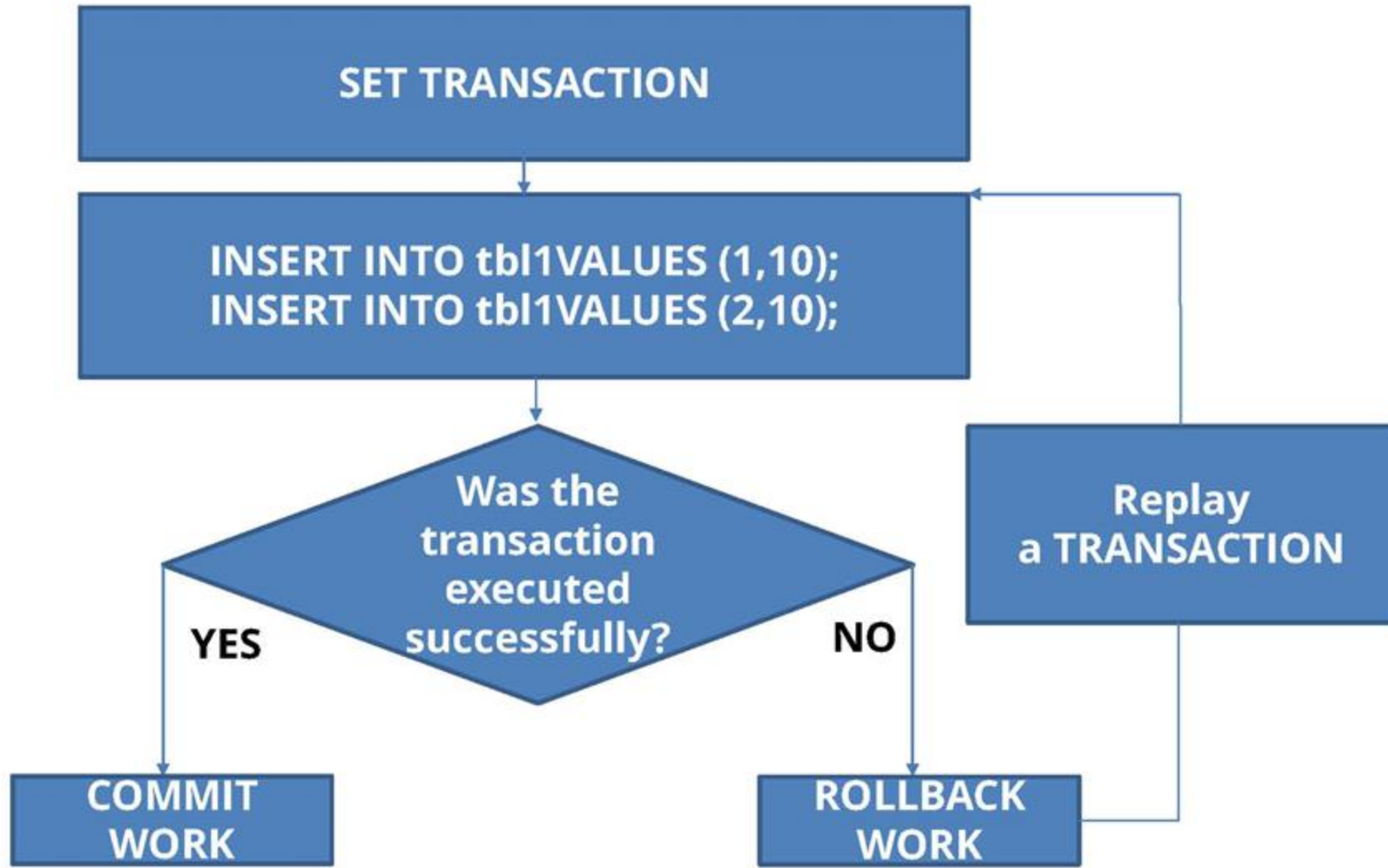
See [callablestatement\CallableStatementDemo](#)

Module contents

Java Data Base Connectivity

- An Introduction to Relation Data Bases
- An Introduction to SQL
- The CRUD Operations
- The JDBC
- The DriverManager and Connection
- The Statement and ResultSet
- PreparedStatement and CallableStatement
- **The Database Transactions**
- The Database MetaData

The Database Transactions



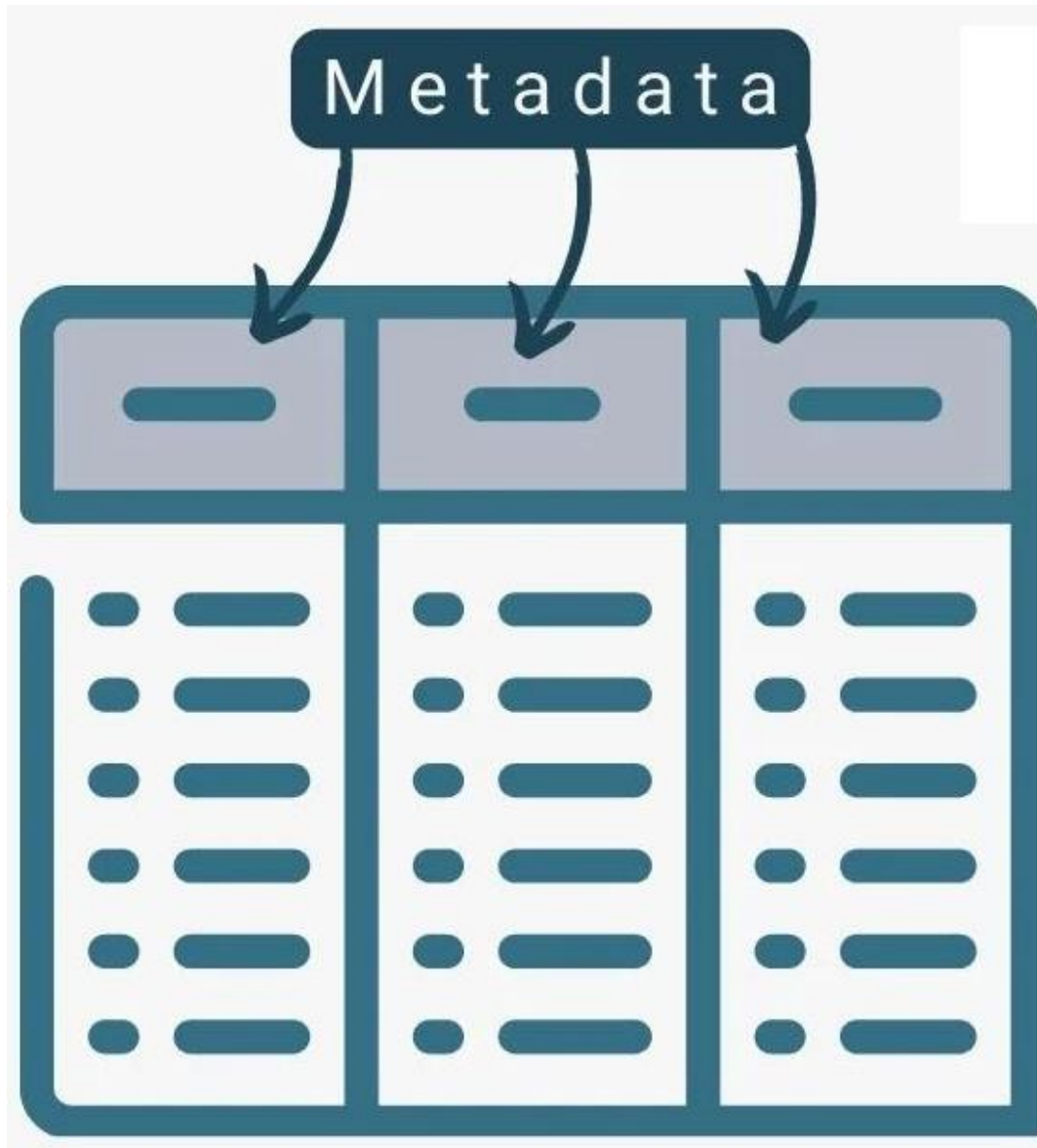
See transaction\TransactionDemo

Module contents

Java Data Base Connectivity

- An Introduction to Relation Data Bases
- An Introduction to SQL
- The CRUD Operations
- The JDBC
- The DriverManager and Connection
- The Statement and ResultSet
- PreparedStatement and CallableStatement
- The Database Transactions
- The Database MetaData

The Database MetaData



See `metsdata\MetaDataDemo`