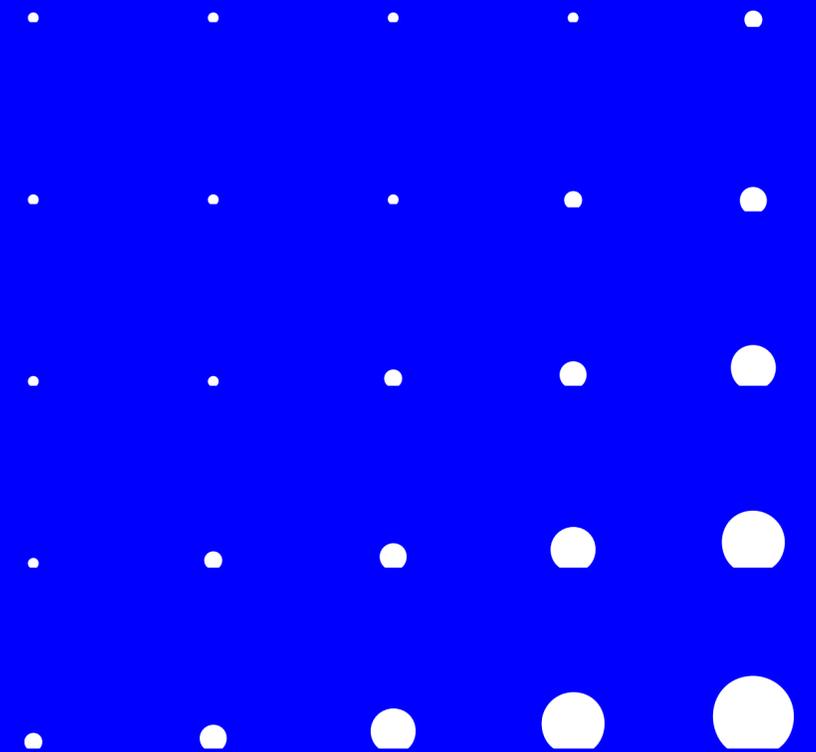


Java Database Connectivity
JDBC



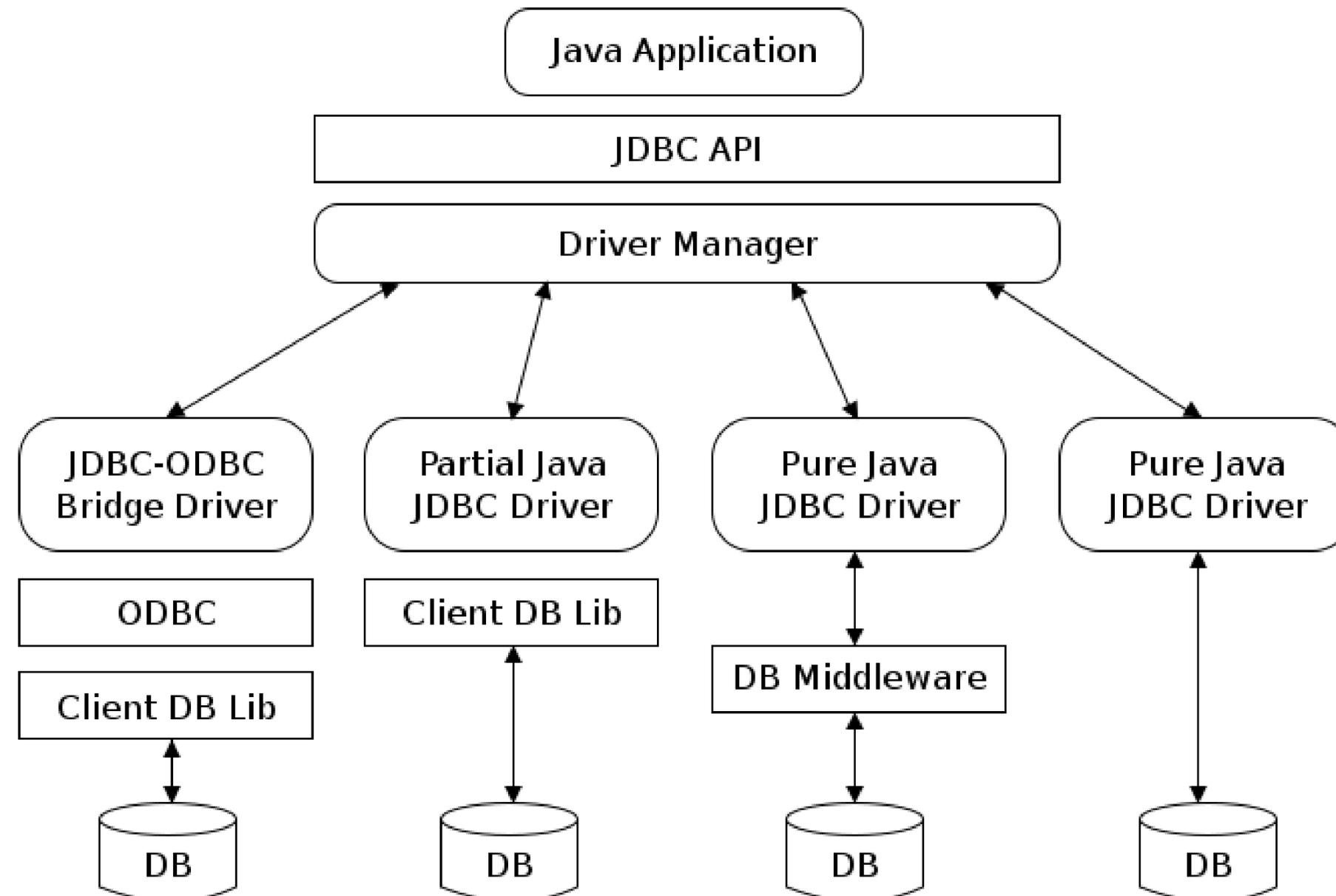
Was ist JDBC?

- Die Java Database Connectivity (JDBC) ist eine Datenbankschnittstelle der Java-Plattform, die eine einheitliche Schnittstelle zu Datenbanken verschiedener Hersteller bietet und speziell auf relationale Datenbanken ausgerichtet ist.
- JDBC ist in seiner Funktion als universelle Datenbankschnittstelle vergleichbar mit z. B.
 - ODBC unter Windows oder
 - DBI unter Perl.
- Für jede spezifische Datenbank sind eigene Treiber erforderlich, welche die JDBC-Spezifikation implementieren.
- Diese Treiber werden meist vom Hersteller des DBS geliefert.

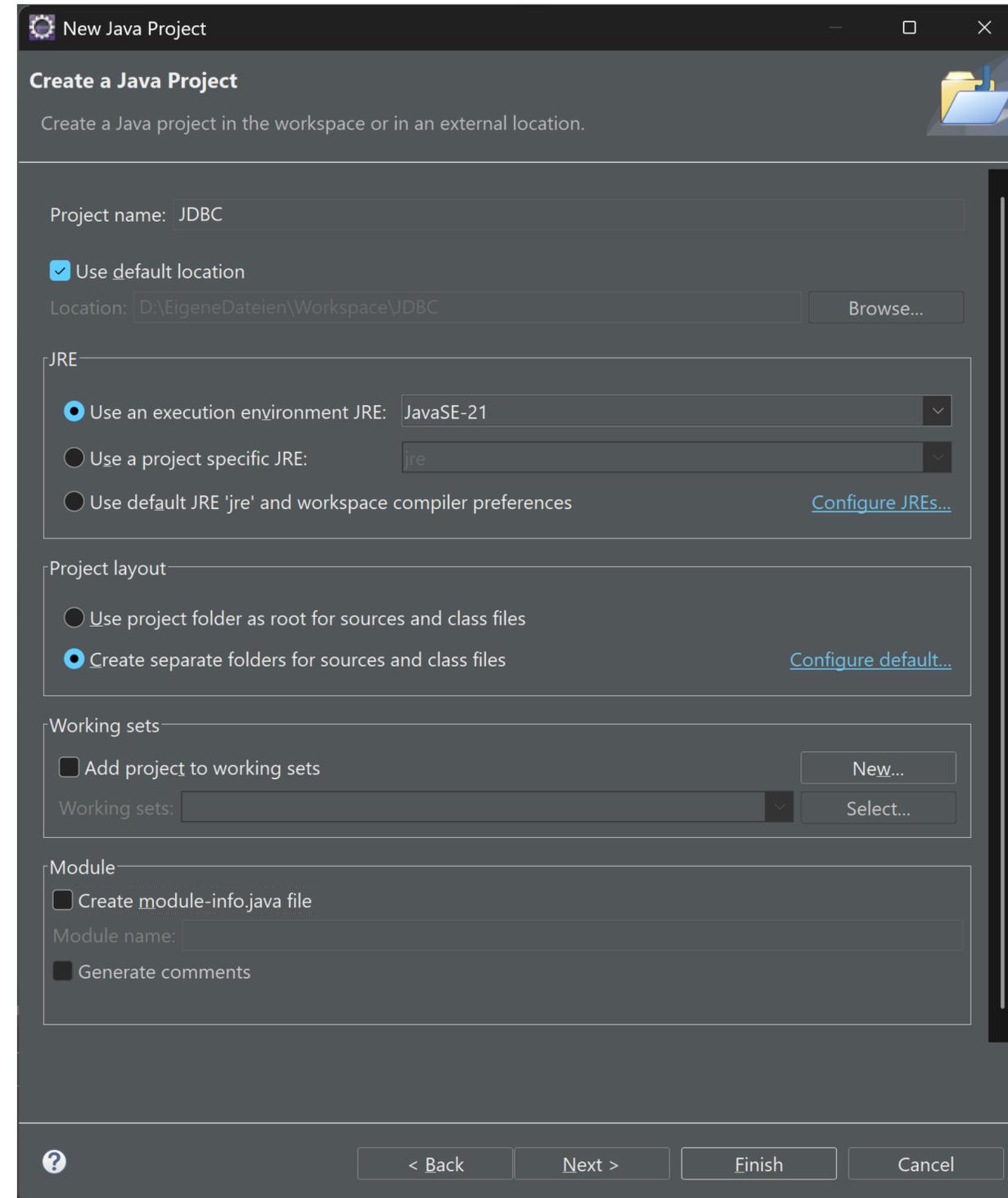
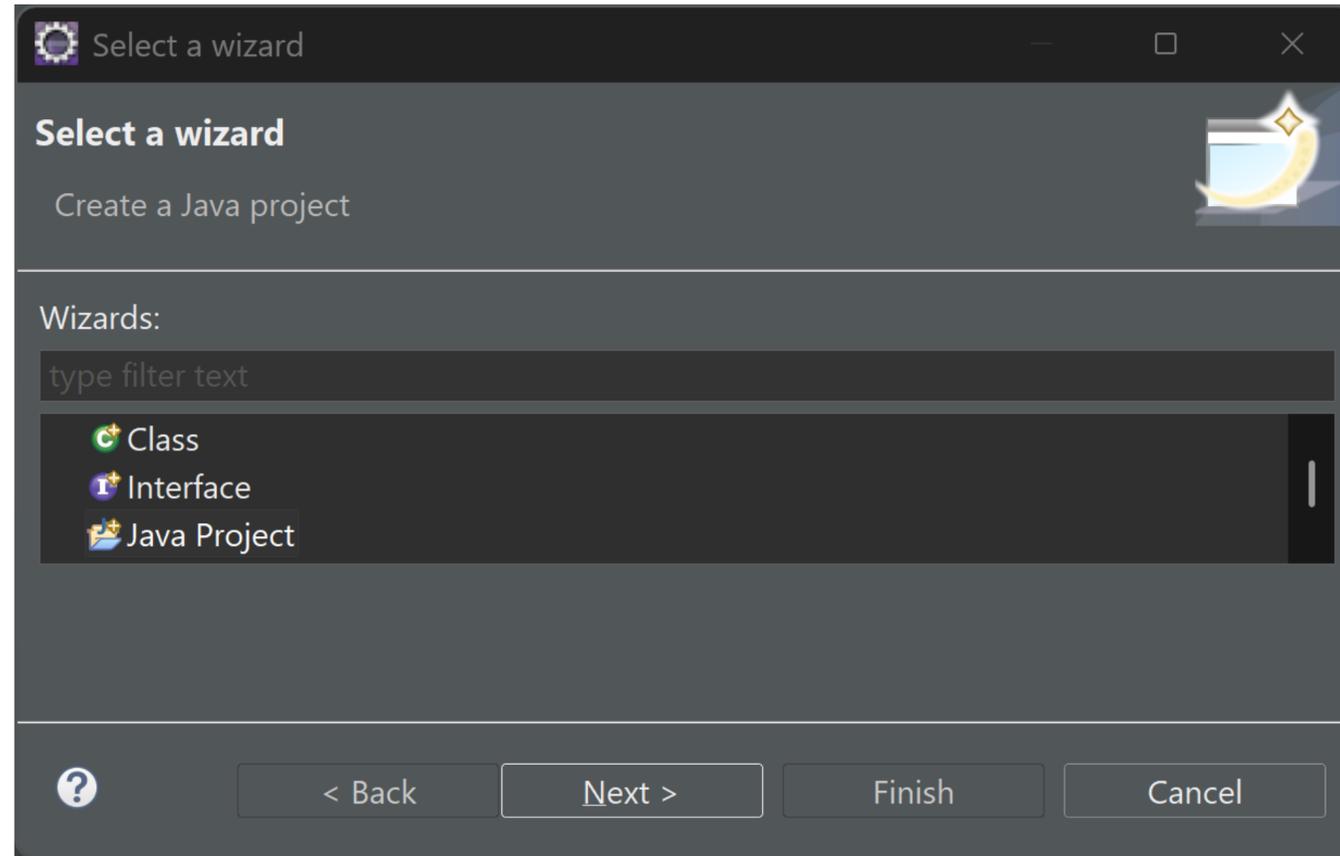
Aufgaben von JDBC

- Zu den Aufgaben von JDBC gehört es,
 - Datenbankverbindungen aufzubauen und
 - sie zu verwalten,
 - SQL-Anfragen an die Datenbank weiterzuleiten und
 - die Ergebnisse in eine für Java nutzbare Form umzuwandelnund dem Programm zur Verfügung zu stellen.

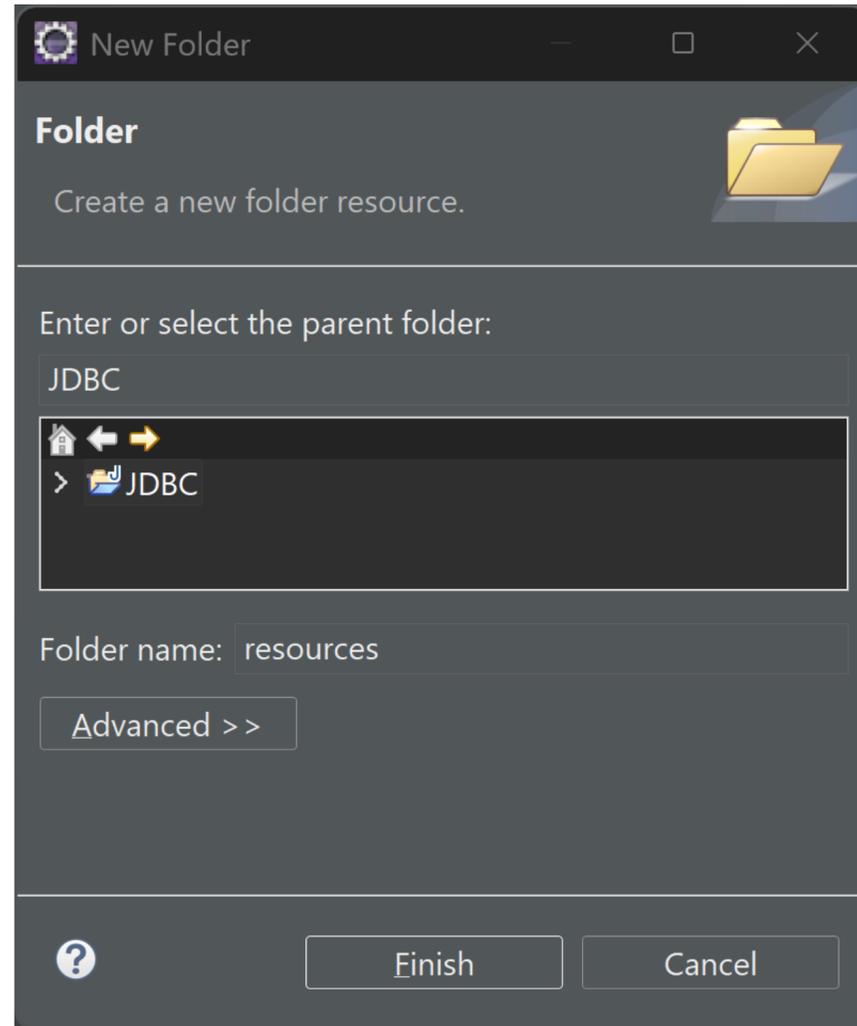
Mögliche JDBC-Anbindungen



Neues Java-Projekt anlegen...



Einbinden des JDBC-Connectors



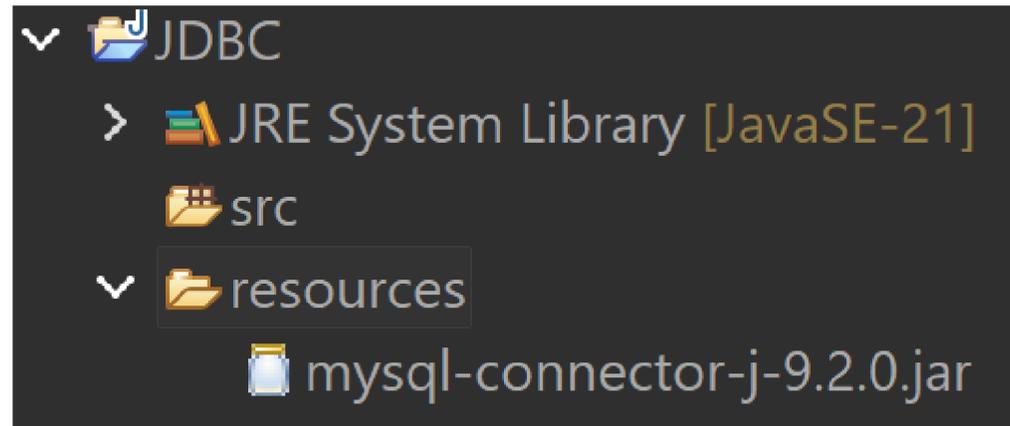
- Dann die JAR-Datei (JAva Archiv) aus der ZIP holen...

▼ \YouTube\InformatikGrundlagen\Datenmanagement\02 - Anwendungen\mysql-connector-j-9.2.0.zip\mysql-connector-j-9.2.0*.*

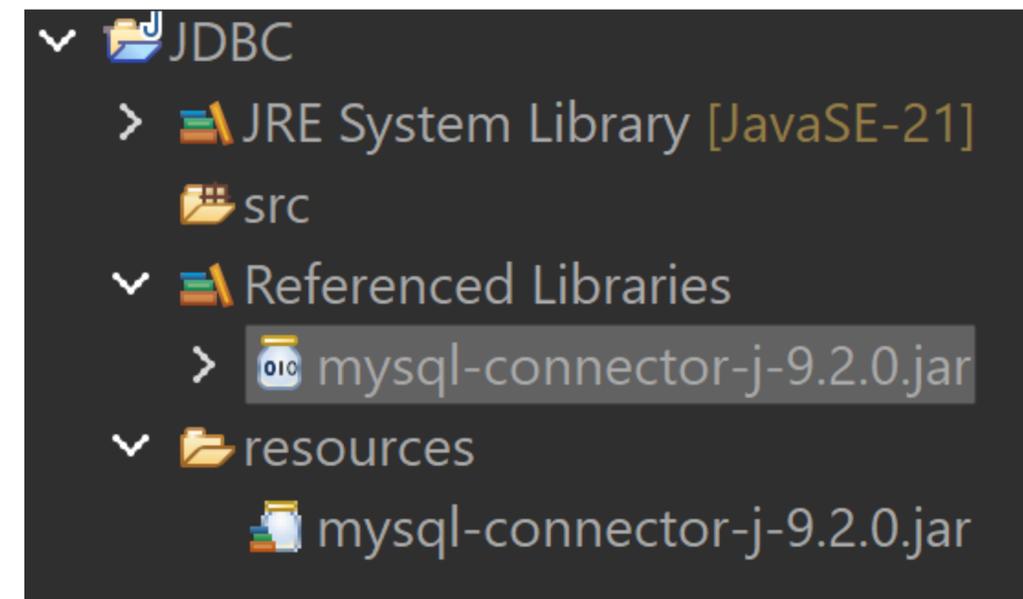
Name	Erw.	Größe	Datum	Attr.
[.]		<DIR>	12.02.2025 14:35-	
[src]		<DIR>	16.12.2024 12:18-	
README		2.179	16.12.2024 12:18-	
mysql-connector-j-9.2.0	jar	2.609.756	16.12.2024 12:18-	
LICENSE		82.899	16.12.2024 12:18-	
INFO_SRC		134	16.12.2024 12:18-	
INFO_BIN		186	16.12.2024 12:18-	
CHANGES		286.736	16.12.2024 12:18-	
build	xml	91.633	16.12.2024 12:18-	

Einbinden des JDBC-Connectors

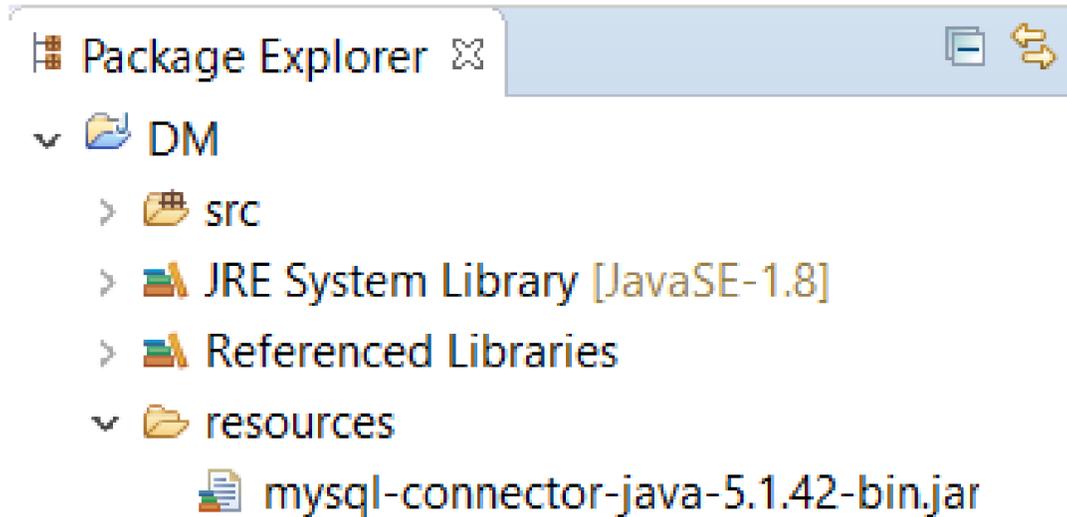
- ...und dann in den resources-Ordner ziehen...



- Dann rechter Mausklick auf die jar, Build Path -> Add to Build Path



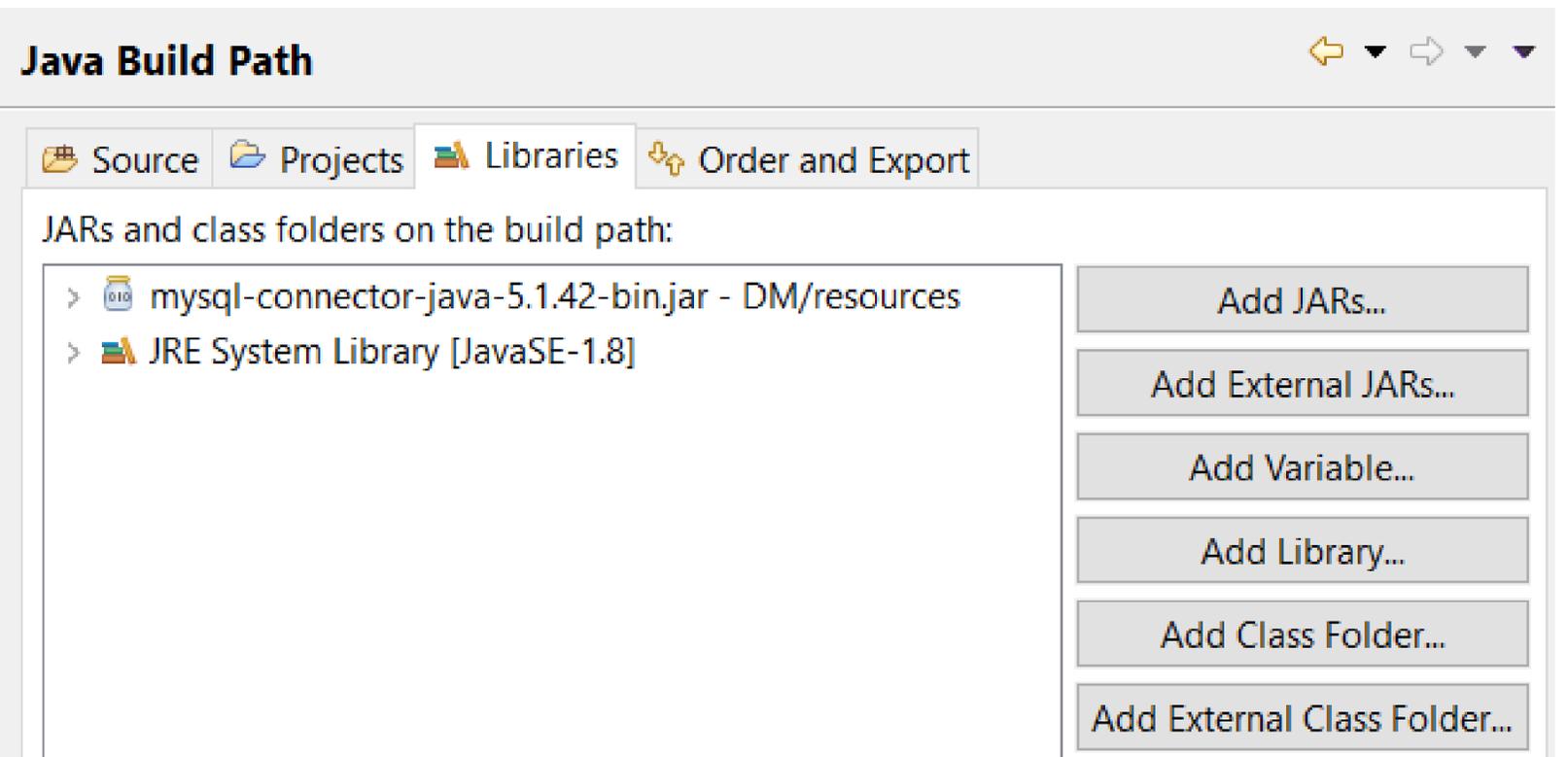
Einbinden des JDBC-Connectors



Properties for DM

type filter text

- > Resource Builders
- Java Build Path**
- > Java Code Style
- > Java Compiler
- > Java Editor
- Javadoc Location
- Project Facets
- Project References
- Refactoring History
- Run/Debug Settings
- Server



Der erste lesende Zugriff via JDBC... Die bereits existierende Datenbank in phpMyAdmin

The screenshot shows the phpMyAdmin interface. The left sidebar displays a tree view of databases, with 'dm' selected. The main area shows the 'Struktur' (Structure) view for the 'produkt' table. The table structure is as follows:

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion
1	produktnummer	bigint(20)			Nein	kein(e)		AUTO_INCREMENT	Bearbeiten Löschen
2	name	varchar(50)	utf8mb4_general_ci		Nein	kein(e)			Bearbeiten Löschen
3	beschreibung	varchar(500)	utf8mb4_general_ci		Nein	kein(e)			Bearbeiten Löschen
4	standardpreis	decimal(6,2)			Nein	kein(e)			Bearbeiten Löschen
5	lagerbestand	int(11)			Nein	kein(e)			Bearbeiten Löschen

Below the table structure, there are various actions and options, including 'Alle auswählen', 'markierte:', 'Anzeigen', 'Bearbeiten', 'Löschen', 'Primärschlüssel', 'Unique', 'Index', 'Räumlich', 'Volltext', 'Zu zentralen Spalten hinzufügen', and 'Zentrale Spalten entfernen'. There is also a 'Drucken' button and a 'Tabellenstruktur analysieren' button.

The 'Indizes' section is visible, showing a table with the following columns: Aktion, Schlüsselname, Typ, Unique, Gepackt, Spalte, Kardinalität, Kollation, Null, and Kommentar. The primary key is listed as follows:

Aktion	Schlüsselname	Typ	Unique	Gepackt	Spalte	Kardinalität	Kollation	Null	Kommentar
Bearbeiten Umbenennen Löschen	PRIMARY	BTREE	Ja	Nein	produktnummer	20	A	Nein	

At the bottom, there is a form to create an index: 'Erstelle Index über 1 Spalten OK'.

Der erste lesende Zugriff via JDBC... Die bereits existierende Datenbank in phpMyAdmin

Server: 127.0.0.1 » Datenbank: dm » Tabelle: produkt

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Rechte Mehr

✓ Zeige Datensätze 0 - 19 (20 insgesamt, Die Abfrage dauerte 0,0006 Sekunden.)

```
SELECT * FROM `produkt`
```

Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

Alles anzeigen | Anzahl der Datensätze: 25 | Zeilen filtern: Diese Tabelle durchsuchen | Nach Schlüssel sortieren: keine

Zusätzliche Optionen

	produktnummer	name	beschreibung	standardpreis	lagerbestand
<input type="checkbox"/> Bearbeiten Kopieren Löschen	1	Intel Core i7-13700K	13. Generation Intel Core Prozessor, 16 Kerne, 5,4...	419.99	120
<input type="checkbox"/> Bearbeiten Kopieren Löschen	2	AMD Ryzen 9 7900X	High-End-Prozessor mit 12 Kernen, 24 Threads, 4,7 ...	549.99	80
<input type="checkbox"/> Bearbeiten Kopieren Löschen	3	NVIDIA GeForce RTX 4070 Ti	Grafikkarte mit 12 GB GDDR6X Speicher und Raytraci...	899.99	50
<input type="checkbox"/> Bearbeiten Kopieren Löschen	4	Corsair Vengeance DDR5 32GB	32 GB DDR5 RAM mit 5600 MHz Taktfrequenz	189.99	200

Der erste lesende Zugriff via JDBC...

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Lesen_01 {

    public static void main(String[] args) {
        // Datenbank-Parameter:
        String url = "jdbc:mysql://localhost:3306/dm"; // IP, TCP-Port und Datenbankname
        String user = "root"; // MySQL-Benutzername
        String password = ""; // Passwort dieses Benutzers

        // Verbindung herstellen:
        try {
            Connection conn = DriverManager.getConnection(url, user, password);
            Statement stmt = conn.createStatement();
            // SQL-Select absetzen:
            ResultSet rs = stmt.executeQuery("SELECT * FROM produkt");

            // Ergebnisse durchgehen und ausgeben
            while (rs.next()) {
                System.out.println("Produktnummer: " + rs.getInt("produktnummer"));
                System.out.println("Name: " + rs.getString("name"));
                System.out.println("Preis: " + rs.getString("standardpreis"));
                System.out.println("Bestand: " + rs.getString("lagerbestand"));
                System.out.println();
            }
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

```
Produktnummer: 1
Name: Intel Core i7-13700K
Preis: 419.99
Bestand: 120

Produktnummer: 2
Name: AMD Ryzen 9 7900X
Preis: 549.99
Bestand: 80

Produktnummer: 3
Name: NVIDIA GeForce RTX 4070 Ti
Preis: 899.99
Bestand: 50

Produktnummer: 4
Name: Corsair Vengeance DDR5 32GB
Preis: 189.99
Bestand: 200

Produktnummer: 5
Name: Samsung 970 EVO Plus 1TB
Preis: 99.99
Bestand: 300

Produktnummer: 6
Name: ASUS ROG Strix Z790-E
Preis: 349.99
Bestand: 75
```

Der erste schreibende Zugriff via JDBC...

```
import java.math.BigDecimal;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class Schreiben_01 {

    public static boolean schreibe_Produkt(
        String name, String beschreibung, BigDecimal preis, int lagerbestand) {
        // Datenbank-Parameter:
        String url = "jdbc:mysql://localhost:3306/dm"; // IP, TCP-Port und Datenbankname
        String user = "root"; // MySQL-Benutzername
        String password = ""; // Passwort dieses Benutzers
        // Verbindung herstellen:
        try {
            Connection conn = DriverManager.getConnection(url, user, password);
            Statement stmt = conn.createStatement();
            // SQL-Select absetzen:
            String sql = "INSERT INTO produkt (name,beschreibung,standardpreis,lagerbestand) ";
            sql += "VALUES (";
            sql += "'" + name + "', ";
            sql += "'" + beschreibung + "', ";
            sql += preis + ", ";
            sql += lagerbestand + ")";
            stmt.execute(sql);
            return true;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    public static void main(String[] args) {
        boolean erfolg = schreibe_Produkt("Test-Name","Test-Beschreibung",new BigDecimal(200.99),100);
        System.out.println("Erfolg: " + erfolg);
    }
}
```

...bei jedem Parameter,
der von außen kommt!

ACHTUNG!

```
<terminated> Schreiben_01
Erfolg: true
```

Der erste schreibende Zugriff via JDBC: Test des Erfolgs via phpMyAdmin

Server: 127.0.0.1 » Datenbank: dm » Tabelle: produkt

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Rechte Operationen Nachverfolgung Trigger

```
SELECT * FROM `produkt`
```

Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

Alles anzeigen | Anzahl der Datensätze: 100 | Zeilen filtern: Diese Tabelle durchsuchen | Nach Schlüssel sortieren: keine

Zusätzliche Optionen

	produktnummer	name	beschreibung	standardpreis	lagerbestand
<input type="checkbox"/> Bearbeiten Kopieren Löschen	1	Intel Core i7-13700K	13. Generation Intel Core Prozessor, 16 Kerne, 5,4...	419.99	120
<input type="checkbox"/> Bearbeiten Kopieren Löschen	2	AMD Ryzen 9 7900X	High-End-Prozessor mit 12 Kernen, 24 Threads, 4,7 ...	549.99	80
<input type="checkbox"/> Bearbeiten Kopieren Löschen	3	NVIDIA GeForce RTX 4070 Ti	Grafikkarte mit 12 GB GDDR6X Speicher und Raytraci...	899.99	50
...					
<input type="checkbox"/> Bearbeiten Kopieren Löschen	19	WD Black SN850X 2TB	NVMe SSD mit bis zu 7300 MB/s Lesegeschwindigkeit	249.99	70
<input type="checkbox"/> Bearbeiten Kopieren Löschen	20	Gigabyte AORUS FI32U	32-Zoll-4K-Monitor mit 144 Hz und 1 ms Reaktionsze...	799.99	30
<input type="checkbox"/> Bearbeiten Kopieren Löschen	21	Test-Name	Test-Beschreibung	200.99	100

- Schreibende Zugriffe sind u.a. INSERT INTO, UPDATE, DELETE