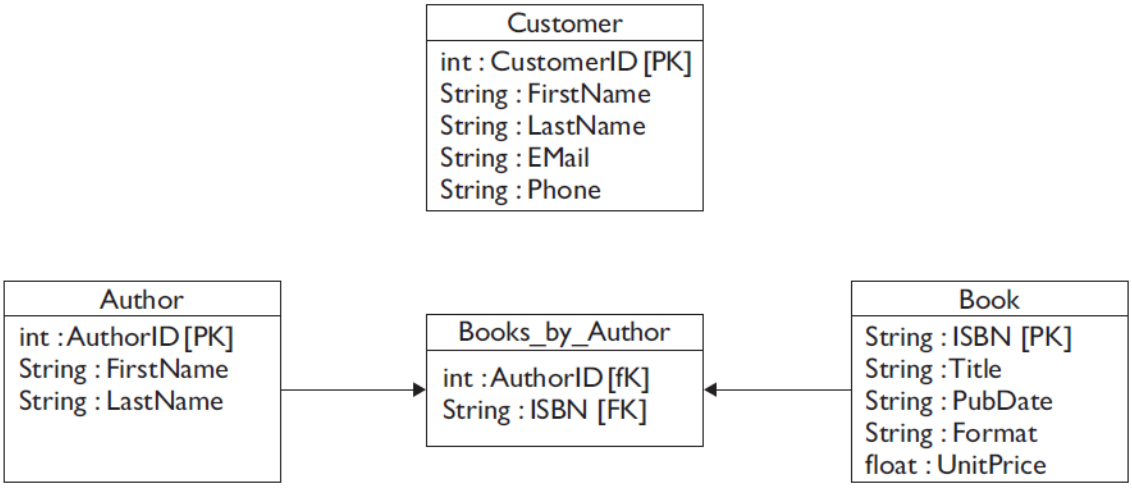
1. Zainicjalizuj strukturę bazodanową w oparciu o przygotowany skrypt.



2. Utwórz nowy projekt maven w środowisku programistycznym Intellij. W tym celu dodaj następujące zależności do projektu:

```
<dependency>
        <groupId>com.mysql</groupId>
        <artifactId>mysql-connector-j</artifactId>
        <version>8.1.0</version>
</dependency>
<dependency>
        <groupId>ch.qos.logback</groupId>
        <artifactId>logback-classic</artifactId>
        <version>1.4.11</version>
</dependency>
<dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.18.28</version>
        <scope>provided</scope>
</dependency>
```

3. Utwórz klasę **JDBCConnectionProvider** oraz zaimplementuj metody umożliwiające nawiązywanie połączenie z bazą danych:

```
public static Connection getConnection() throws SQLException {
        …
}

public static Connection getConnectionBasedOnProperties() throws SQLException {
        …
}

public static Connection getConnectionBasedOnURLParameters() throws SQLException {
        …
}
```

Przetestuj nawiązywanie oraz zamykanie połączenia z bazą danych na podstawie prawidłowo zdefiniowanego adresu URL bazy danych, nazwy użytkownika oraz hasła.

4. Utwórz klasę **DataSourceProvider** oraz zaimplementuj następujące metody:

```
public static DataSource getDataSource() {
        MysqlDataSource dataSource = new MysqlDataSource();
        Properties properties = loadProperties();

        …

        return dataSource;
}

private static Properties loadProperties() {
        Properties properties = new Properties();

        …

        return properties;
}
```

Przetestuj nawiązywanie połączenia przy pomocy **DataSource**. Zwróć uwagę na prawidłowe zamykanie połączeń (**try-with-resources**, **try-catch-finally**).

Następnie, przenieś parametry bazy danych do pliku **db.properties**
(w tym celu utwórz plik w katalogu **resources**):

W pliku db.properties powinny znaleźć się pary **klucz=wartość** dla każdego parametru:

pl.sda.jdbc.db. url={*db_url*}

pl.sda.jdbc.db.username={*username*}

pl.sda.jdbc.db.password={*password*}

Przekaż nazwę pliku jako argument metody tworzącej obiekt Properties oraz DataSource. Wczytaj parametry z pliku za pomocą klasy **java.util.Properties** i metody **Properties.load(InputStream inStream)**. Obiekt InputStream można pobrać wykorzystując właściwości *classloadera*:
InputStream inputStream = ClassLoader.getSystemResourceAsStream("db.properties")
Uruchom i sprawdź, czy program działa.

5*.

Connection - pobieranie danych o bazie

- Klasa Connection umożliwia pobieranie danych o połączonej bazie danych za pomocą:

```
DatabaseMetaData metaData = connection.getMetaData();
```

- W metadanych bazy znajdziemy m.in. informacje z jaką bazą jesteśmy połączeni (nazwa, wersja), jakim sterownikiem połączyliśmy się itp
- Ponadto możemy uzyskać informację o strukturze bazy danych, najłatwiej zrobić to poprzez kod:

```
ResultSet rs = metaData.getColumns(null, null, null, null);
while(rs.next()) {
   String tableName = rs.getString("TABLE_NAME");
   logger.info("table: {}", tableName);
}
```

- ResultSet zawiera dużo więcej informacji - nazwy wszystkich parametrów są opisane w dokumentacji metody DatabaseMetaData.getColumns(). Wystarczy najechać myszką na nazwę metody i nacisnąć Ctrl+Q

Utwórz klasę **DatabaseDiscovery**. W metodzie main pobierz obiekt **Connection** i wykorzystując metodę **getMetaData()** pobierz z niego obiekt **DatabaseMetaData**.

Wyświetl w konsoli (za pomocą odpowiednio przygotowanego loggera) następujące informacje: **nazwa i wersja bazy danych**, **nazwa użytkownika, nazwa i wersja sterownika do bazy danych** - w obiekcie **DatabaseMetaData** znajdziesz metody dostępowe do każdej z tych informacji.

Za pomocą kodu opisanego we wstępie pobierz i wyświetl w konsoli:
nazwę tabeli, nazwę kolumny oraz nazwę typu, odpowiednio dla wszystkich tabel znajdujących się w bazie danych.

(**dla chętnych**) Spróbuj poeksperymentować z metodą

DatabaseMetaData.getColumns(null, null, null, null), sprawdź dokumentację i spróbuj pobrać dane dla konkretnej bazy danych albo dla konkretnej tabeli. W tym celu zamiast wartości nieokreślonej **null** trzeba podać odpowiednie argumenty w metodzie **getColumns**().

6. Utwórz klasę reprezentującą tabelę **Customer**. Wykorzystując odpowiednie zapytanie typu **select**, pobierz z bazy danych wszystkich klientów (tworząc kolekcję obiektów). Pobierz wartości atrybutów poszczególnych krotek przy pomocy parametrów nazwanych. Posortuj kolekcję obiektów.
Następnie, wykonaj to samo zadanie wykorzystując parametry porządkowe. Zwróć uwagę na sposób przypisywania wartości obiektów null. Poeksperymentuj z wartościami null.
W tym celu, pobierz i przetwórz zawartość tabeli **Book** zwracając szczególną uwagę na atrybut **price**.
Zapoznaj się z metodą: **resultSet.wasNull()**.

Wykorzystując metodę **Statement.executeUpdate()** obniż cenę o połowę wszystkich książek w formacie „*Hardcover*”. Wypisz, ile książek zostało zaktualizowanych. Wyznacz sumę cen wszystkich

książek.

Wstaw nowy rekord do tabeli **book**.
Usuń wszystkie książki, których tytuł zaczyna się od frazy: „*The white*”.

Utwórz tabelę łączącą książki z autorami. Dodaj wybrane rekordy tabeli łączące autorów z ich książkami. Zauważ, iż mamy do czynienia z relacją wiele do wielu.

Wykonaj następujące zapytanie z poziomu JDBC:
String SQL = "select customer_id from customer order by created_at"
Przetwórz wyniki zapytania wykorzystując w tym celu metody:
resultSet.next()
resultSet.previous()
resultSet.absolute(3)
resultSet.relative(1)
resultSet.first()
resultSet.last()
resultSet.beforeFirst()
resultSet.afterLast()

W tym celu, odpowiednio zainicjalizuj obiekt Statement:
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY)

Spróbuj wykonać następujące zapytanie:
select not_a_column from customer
Czy potrafisz wyjaśnić dlaczego wykonanie programu kończy się błędem?
Wypisz na ekranie (przy pomocy odpowiedniego loggera) komunikat błędu, kod błędu oraz status.

7. Zamykanie połączeń.
Wykonaj następujące zapytanie z poziomu JDBC:
select * from customer book
Po przetworzeniu wyników, zamknij (zwolnij) zasoby wykorzystując konstrukcję try, catch, finally.

8. Wyznacz sumę książek z uwzględnieniem odpowiednich warunków na format książki oraz datę publikacji. Format książki = „Paperback”, data publikacji <= 2006.12.31.

Porównaj czas wykonywania polecenia SQL realizowane w oparciu o **Statement** oraz **PreparedStatement**:
insert into author(first_name, last_name) values(?, ?)

W celu uzyskania lepszych rezultatów, spróbuj wykonać to zapytanie w pętli 20 000 razy.

9. **SQLInjections** - wstrzyknięcie w normalne zapytanie SQL złośliwego kodu, który ma na celu zniszczyć dane albo uzyskać dane, których użytkownik nie powinien posiadać. **PreparedStatement** zapobiega tego typu atakom.

https://www.w3schools.com/sql/sql_injection.asp
https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet

Utwórz tabelę na podstawie poniższej definicji:

```
public void createTable() throws SQLException {
    try (Connection connection = new ConnectionFactory().getConnection();
         Statement statement = connection.createStatement()) {

        statement.execute("CREATE TABLE admins (" +
                "  id INT NOT NULL AUTO_INCREMENT, " +
                "  login VARCHAR(50) NOT NULL, " +
                "  password VARCHAR(45) NOT NULL, " +
                "PRIMARY KEY (id))"
        );
    }
}
```

Zaimplementuj funkcję wyszukującą administratorów na podstawie następującego zapytania:
SELECT id, login, password FROM admins WHERE login='%s' AND password='%s';

Wykorzystaj w tym celu odpowiedni formatter:
String sql = String.format("SELECT id, login, password FROM admins WHERE login='%s' AND password='%s';", login, password);

Następnie, zaimplementuj funkcję dodającą administratora wykorzystując odpowiedni formatter:
String sql = String.format("INSERT INTO admins(login, password) VALUES('%s', '%s');", login, password);

Czy widzisz jakieś niebezpieczeństwa?

10. Wywołaj funkcję bazodanową *hello_sda_function* i wypisz wynika na ekranie.