

JDBC - zapytania SQL

- JDBC API:

java.sql.Statement (wykonywanie prostych/statycznych zapytań SQL bez parametrów i pobranie wyników)
java.sql.ResultSet (obiekt przechowujący zestaw danych zwróconych przez bazę po wykonaniu zapytania SQL)

<https://docs.oracle.com/javase/tutorial/jdbc/basics/processingsqlstatements.html>

- Proste zapytanie i pobranie wyników:

```
try(Statement statement = connection.createStatement()) {  
    ResultSet resultSet =  
        statement.executeQuery("SELECT id FROM table_1;");  
    while (resultSet.next()) {  
        int id = resultSet.getInt("id");  
        logger.info("id: {}", id);  
    }  
}
```

- Statement.executeQuery() - wyciąganie danych, Statement.executeUpdate() - aktualizowanie danych (dodawanie, aktualizacja, usuwanie)
- Statement i ResultSet również zamykamy jak tylko przestaniemy ich używać! Jak zamkniemy Statement, ResultSet zostanie automatycznie zamknięty
- Mapowanie typów (odpowiada za to sterownik):

SQL Type	Java Type
VARCHAR	java.lang.String
INTEGER	int
DATE	java.sql.Date
TIMESTAMP	java.sql.Timestamp

więcej: http://www.java2s.com/Tutorials/Java/JDBC/0070_JDBC_Data_Types.htm

Zadanie nr 4:

1. Stwórz klasę CoursesManager
2. W metodzie main (wykorzystując klasę ConnectionFactory i plik 'remote-database.properties') pobierz obiekt Connection
3. Wykorzystując informacje zdobyte w zadaniu nr 3 dodaj metodę która wyświetli id i nazwę wszystkich kursów jakie są zapisane w bazie, znajdź id kursu dla 'JavaGda11' - wykorzystaj polecenie SQL SELECT
4. Wykorzystując informacje zdobyte w zadaniu nr 3 dodaj metodę która doda twoje dane do tabelki ze studentami: imię, id kursu 'JavaGda11', krótki opis (dlaczego kurs Java?) - wykorzystaj polecenie SQL INSERT
5. Dodaj metodę która wyświetli wszystkich studentów wraz z nazwą kursów do którego są zapisani - wykorzystaj polecenie SQL SELECT i połącz dwie tabelki za pomocą JOINa
6. Zaktualizuj swój rekord (wykorzystaj polecenie SQL UPDATE) w tabelce ze studentami i dodaj miejsce na sali w którym aktualnie siedzisz w formacie:

{column}.{row}.{seat}

gdzie (patrzac przodem do klasy):

- column to kolumna ławek patrząc od lewej opisana literami: A, B, C...
- row to ilość ławek patrząc od początku opisana numerami: 1, 2, 3...
- seat to miejsce w ławce (krzesło) od lewej: 1, 2,...

np. **A.3.1**, **C.5.2**

7. Dodaj metodę która doda do tabelki z listą obecności swoją obecność na kursie, dane:
 - id kursu 'JavaGda11',
 - twoje id studenta,
 - aktualna data w formacie '2018-06-25'wykorzystaj polecenie SQL INSERT
8. (dla chętnych) Dodaj metodę która wyświetli listę obecności z nazwami studentów i nazwą kursów w których uczestniczą - wykorzystaj polecenie SQL SELECT i połącz tabelki za pomocą JOINa
9. (dla chętnych) Dodaj metodę która wyświetli listę studentów siedzących w konkretnym miejscu:

`listStudentsBySeat(String column, String row, String seat)`

Metoda powinna obsługiwać takie argumenty:

- `listStudentsBySeat(A, null, null)` - wszyscy studenci z kolumny A
- `listStudentsBySeat(null, 1, null)` - wszyscy studenci z rzędu 1
- `listStudentsBySeat(null, null, 2)` - wszyscy studenci z krzesła 2
- `listStudentsBySeat(A, 1, 5)` - student siedzący w miejscu A.1.5

Użyj szukania po wzorcu:

<https://dev.mysql.com/doc/refman/8.0/en/pattern-matching.html>