

JDBC - zapytania SQL

java.sql.Statement

- **zapytanie** - do bazy danych
- **statyczne** - proste zapytania, bez parametrów
- **interfejs** - nie można go wprost utworzyć
- **Statement.executeQuery(q)** - pobranie danych: SELECT
- **Statement.executeUpdate(q)** - aktualizowanie danych: UPDATE, INSERT...
- **try-with-resources** - zawsze zamykamy na koniec

java.sql.ResultSet

- **odpowiedź** - z wynikami z bazy danych
- **interfejs** - nie można go wprost utworzyć
- **try-with-resources** - jak zamkniemy Statement to i ResultSet
- **użycie:**

```
try(Statement statement = connection.createStatement())
{
    String query = "INSERT INTO table_1(id, name)
                    VALUES(1, 'Bob')";
    statement.executeUpdate(query);

    query = "SELECT id FROM table_1";
    ResultSet result = statement.executeQuery(query);
    while (result.next()) {
        int id = result.getInt("id");
        logger.info("id: {}", id);
    }
}
```

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

- **konwersja 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 2:

1. Stwórz nową bazę danych w lokalnej bazie MySQL o nazwie: **sda_courses**
2. W katalogu **resources** dodaj nowy plik z parametrami do nowo stworzonej bazy danych (np.: 'sda-courses-database.properties')
3. Otwórz klasę **CoursesManager** i uzupełnij metodę: *createCoursesTable()*. Korzystając z klasy **ConnectionFactory** (wykorzystaj plik z parametrami z pkt 2) pobierz nowy obiekt **Connection** i za jego pomocą wywołaj zapytanie do bazy danych. Struktura tabeli:

courses		
Nazwa kolumny	Typ SQL	
id	INT	PK, AUTO_INCREMENT
name	VARCHAR(50)	
place	VARCHAR(50)	
start_date	DATE	
end_date	DATE	

W tej samej metodzie dodaj kod, który wstawi kilka wierszy z danymi do tabelki **courses** - wykorzystaj polecenie SQL INSERT.

Wywołaj metodę *createCoursesTable()* z poziomu metody *main()* i sprawdź czy działa.

4. W podobny sposób uzupełnij metody *createStudentsTable()* i *createAttendanceListTable()*, struktura tabel:

students		
Nazwa kolumny	Typ SQL	
id	INT	PK, AUTO_INCREMENT
name	VARCHAR(50)	
course_id	INT	FK
description	VARCHAR(200)	
seat	VARCHAR(10)	

W kolumnie **description** dodaj krótki opis (dlaczego kurs Java?)

W kolumnie **seat** dodaj miejsce w którym siedzi student 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**

attendance_list		
Nazwa kolumny	Typ SQL	
id	INT	PK, AUTO_INCREMENT
student_id	VARCHAR(50)	FK
course_id	INT	FK
date	DATETIME	

5. Uzupełnij metodę: *dropAllTables()* - metoda ma usunąć wszystkie tabelki z bazy danych. Uważaj na kolejność usuwanych tabel!
6. Dodaj do klasy **CoursesManager** metody: *printAllStudents()* i *printAllCourses()* Metody mają wypisać (użyj logger'a) wszystkie wiersze z poszczególnych tabel - wykorzystaj polecenie SQL SELECT.
7. W metodzie *printAllStudents()* dodaj informacje o kursie(nazwa) do którego jest zapisany student lub napis "brak kursu" - wykorzystaj polecenie SQL SELECT i połącz dwie tabelki za pomocą JOINa
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)** Liquibase:
 - a. Usuń wszystkie tabele z bazy **sda_courses**, użyj metody *dropAllTables()*
 - b. Zaznacz profil Maven'a o nazwie: run-liquibase (w panelu Maven Projects)
 - c. Uruchom z tego samego paneli polecenie mavena: compile
 - d. Sprawdź czy w lokalnej bazie pojawiły się tabele.

Zapoznaj się z biblioteką liquibase do zarządzania strukturą bazy danych.
Spróbuj stworzyć nowe tabele (dowolne) w tej technologii.

<https://www.liquibase.org/quickstart.html>