

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Базы данных»**  
**Тема: Реализация базы данных с использованием ORM**

Студент гр. 1303

\_\_\_\_\_

Кузнецов Н.А.

Преподаватель

\_\_\_\_\_

Заславский М.М.

Санкт-Петербург

2023

### **Цель работы.**

Создание базы данных с использованием Object-Relational Mapping (ORM).

### **Текст задания**

#### Вариант 6

- Описать в виде моделей Sequelize таблицы из 1-й лабораторной работы
- Написать скрипт заполнения тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
- Написать запросы к БД, отвечающие на вопросы из 1-й лабораторной работы с использованием ORM. Вывести результаты в консоль (или иной человек-читаемый вывод)
- Запустить в репозиторий исходный код проекта, соблюсти .gitignore, убрать исходную базу из проекта (или иные сгенерированные данные бд если они есть).
- Описать процесс запуска: команды, зависимости
- В отчёте описать цель, текст задания в соответствии с вариантом, выбранную ORM, инструкцию по запуску, скриншоты (код) моделей ORM, скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ), ссылку на PR в приложении, вывод

## Выполнение работы

Для установки необходимых библиотек использовались следующие команды:

```
npm install sequelize
npm install pg
```

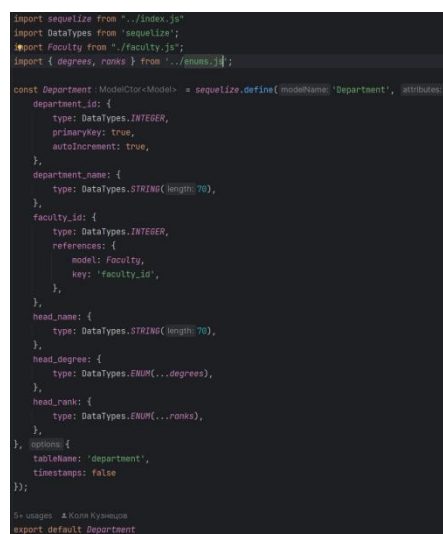
В файле index.js происходит настройка соединения с базой данных:

```
const sequelize = new Sequelize('university', 'postgres', '1', {
  host: 'localhost',
  dialect: 'postgres',
  define: {
    allowNull: false,
  },
});
```

Здесь создается объект sequelize, который представляет собой соединение с базой данных. В конструкторе Sequelize передаются следующие параметры:

- university: Название базы данных.
- postgres: Имя пользователя базы данных.
- 1: Пароль пользователя базы данных.
- { host: 'localhost', dialect: 'postgres' }: Дополнительные опции, указывающие хост (localhost) и используемый диалект (PostgreSQL).

Реализованные модели:



```
import sequelize from '../index.js'
import DataTypes from 'sequelize';
import Faculty from './faculty.js';
import { degrees, ranks } from './enums.js';

const Department = ModelCtor<Model> = sequelize.define<Model>('Department', {
  department_id: {
    type: DataTypes.INTEGER,
    primaryKey: true,
    autoIncrement: true,
  },
  department_name: {
    type: DataTypes.STRING(70),
  },
  faculty_id: {
    type: DataTypes.INTEGER,
    references: {
      model: Faculty,
      key: 'faculty_id',
    },
  },
  head_name: {
    type: DataTypes.STRING(70),
  },
  head_degree: {
    type: DataTypes.ENUM(...degrees),
  },
  head_rank: {
    type: DataTypes.ENUM(...ranks),
  },
}, {
  timestamps: false,
});

export default Department
```

Рисунок 1. – Структура созданной БД.

```

import sequelize from "../index.js"
import DataTypes from "sequelize";
import {reportTypes} from "../enums.js";

const Discipline : ModelCtor<Model> = sequelize.define(modelName: 'Discipline', attributes: {
  discipline_id: {
    type: DataTypes.INTEGER,
    primaryKey: true,
    autoIncrement: true,
  },
  discipline_name: {
    type: DataTypes.STRING( length: 70),
  },
  semesters: {
    type: DataTypes.INTEGER,
  },
  lesson_hours: {
    type: DataTypes.DECIMAL( precision: 3, scale: 1),
  },
  practice_hours: {
    type: DataTypes.DECIMAL( precision: 3, scale: 1),
  },
  lab_hours: {
    type: DataTypes.DECIMAL( precision: 3, scale: 1),
  },
  cw_hours: {
    type: DataTypes.DECIMAL( precision: 3, scale: 1),
  },
  report_type: {
    type: DataTypes.ENUM(...reportTypes),
  },
}, options: {
  tableName: 'discipline',
  timestamps: false
});

5+ usages  ⚡ Коля Кузнецов
export default Discipline;

```

Рисунок 2. – Структура созданной БД.

```

import sequelize from "../index.js"
import DataTypes from "sequelize";
import Discipline from "../discipline.js";
import Specialization from "../specialization.js";

const DisciplineSpecialization : ModelCtor<Model> = sequelize.define(modelName: 'DisciplineSpecialization', attributes: {
  discipline_id: {
    type: DataTypes.INTEGER,
    references: {
      model: Discipline,
      key: 'discipline_id',
    },
  },
  specialization_id: {
    type: DataTypes.INTEGER,
    references: {
      model: Specialization,
      key: 'specialization_id',
    },
  },
}, options: {
  tableName: 'discipline_specialization',
  timestamps: false
});

5+ usages  ⚡ Коля Кузнецов
export default DisciplineSpecialization;

```

Рисунок 3. – Структура созданной БД.

```

import sequelize from "../index.js"
import DataTypes from "sequelize";

const Faculty : ModelCtor<Model> = sequelize.define(modelName: 'Faculty', attributes: {
  faculty_id: {
    type: DataTypes.INTEGER,
    primaryKey: true,
    autoIncrement: true,
  },
  faculty_name: {
    type: DataTypes.STRING( length: 70),
  },
}, options: {
  tableName: 'faculty',
  timestamps: false
});

5+ usages  ⚡ Коля Кузнецов
export default Faculty

```

Рисунок 4. – Структура созданной БД.

```
import {studyForms} from '../enums.js';

const Specialization: ModelCtor<Model> = sequelize.define(modelName: 'Specialization', attributes: {
  specialization_id: {
    type: DataTypes.INTEGER,
    primaryKey: true,
    autoIncrement: true,
  },
  specialization_name: {
    type: DataTypes.STRING(length: 70),
  },
  department_id: {
    type: DataTypes.INTEGER,
    references: {
      model: Department,
      key: 'department_id',
    },
  },
  specialization_code: {
    type: DataTypes.STRING(length: 20),
  },
  qualification: {
    type: DataTypes.STRING(length: 70),
  },
  duration: {
    type: DataTypes.DECIMAL(precision: 3, scale: 2),
  },
  study_form: {
    type: DataTypes.ENUM(...studyForms),
  },
}, options: {
  tableName: 'specialization',
  timestamps: false
});

5+ usages  ▲ Коля Кузнецов
export default Specialization
```

Рисунок 5. – Структура созданной БД.

```
import sequelize from '../index.js'
import DataTypes from 'sequelize';
import Department from './department.js';

const Telephone: ModelCtor<Model> = sequelize.define(modelName: 'Telephone', attributes: {
  telephone_id: {
    type: DataTypes.INTEGER,
    primaryKey: true,
    autoIncrement: true,
    allowNull: false,
  },
  telephone_number: {
    type: DataTypes.STRING(length: 20),
  },
  department_id: {
    type: DataTypes.INTEGER,
    references: {
      model: Department,
      key: 'department_id',
    },
  },
}, options: {
  tableName: 'telephone',
  timestamps: false
});

4 usages  ▲ Коля Кузнецов
export default Telephone
```

Рисунок 6. – Структура созданной БД.

Выполним запросы к БД предложенные вариантом:

```

await Discipline.findAll({options: {
  attributes: ['discipline_name'],
  where: {
    semesters: {[Op.gt]: 1}
  },
}})

.then((res : Model[] ) : void => {
  console.log('\n', '1) Названия дисциплин, которые читаются более одного семестра?')
  console.log(JSON.stringify(res, {replacer: null, space: 2}), "\n")
})

```

Рисунок 7. – Запрос 1.

```

1) Названия дисциплин, которые читаются более одного семестра?
[
  {
    "discipline_name": "Математический анализ"
  },
  {
    "discipline_name": "Физика"
  },
  {
    "discipline_name": "Программирование на C++"
  }
]

```

Рисунок 7. – Результат запроса 1.

```

await sequelize.query(sql: '' +
  'SELECT SUM(discipline.lab_hours) AS "lab_hours", department.name\n' +
  'FROM discipline\n' +
  '  JOIN discipline_specialization on discipline.discipline_id = discipline_specialization.discipline_id\n' +
  '  JOIN specialization on discipline_specialization.specialization_id = specialization.specialization_id\n' +
  '  JOIN department on specialization.dep_id = department.dep_id\n' +
  'GROUP BY department.name\n',
  {options: {type: QueryTypes.SELECT}}
)

.then(res : object[] => {
  console.log('\n', '2) Общее количество часов, отводимых на лабораторные работы в одном из семестров, проведение которых обеспечивает определенная кафедра?');
  console.log(JSON.stringify(res, {replacer: null, space: 2}), "\n")
})

```

Рисунок 8 – Запрос 2.

```

2) Общее количество часов, отводимых на лабораторные работы в одном из семестров, проведение которых обеспечивает определенная кафедра?
[
  {
    "lab_hours": "35.0",
    "name": "РЭС"
  },
  {
    "lab_hours": "35.0",
    "name": "МЭК"
  },
  {
    "lab_hours": "40.0",
    "name": "СО"
  },
  {
    "lab_hours": "105.0",
    "name": "МОЗБМ"
  },
  {
    "lab_hours": "50.0",
    "name": "САУ"
  },
  {
    "lab_hours": "30.0",
    "name": "МБЗ"
  },
  {
    "lab_hours": "30.0",
    "name": "САПР"
  }
]

```

Рисунок 8 – Результат запроса 2.

```

await Discipline.findAll(options: {
  attributes: ["discipline_name"],
  where: {lab_hours: {[Op.gt]: 0}},
  include: [{
    model: DisciplineSpecialization,
    attributes: [],
    include: [{
      model: Specialization,
      include: [{
        model: Department,
        include: [{
          model: Faculty,
        }]
      }]
    }]
  }]
})
})
.then((res: Model[]) : void => {
  console.log('\n', "3) Название дисциплин, по которым проводятся лабораторные работы на факультете?")
  console.log(JSON.stringify(res, replacer: null, space: 2), "\n")
})

```

Рисунок 9 – Запрос 3.

```

3) Название дисциплин, по которым проводятся лабораторные работы на факультете?
[
  {
    "discipline_name": "Математический анализ"
  },
  {
    "discipline_name": "Теория электрических цепей"
  },
  {
    "discipline_name": "Программирование на C++"
  },
  {
    "discipline_name": "Инженерная графика"
  },
  {
    "discipline_name": "Электроника и микроэлектроника"
  },
  {
    "discipline_name": "Механика сплошных сред"
  },
  {
    "discipline_name": "Основы автоматизации и управления"
  },
  {
    "discipline_name": "Базы данных"
  },
  {
    "discipline_name": "WEB-технологии"
  }
]

```

Рисунок 9 – Результат запроса 3.

```

await discipline.findAll(options: {
  attributes: ["discipline_name", [fn(fn 'ABS', sequelize.literal(`${'lab_hours' - practice_hours}`), 'difference')]],
  include: [{
    attributes: [],
    model: DisciplineSpecialization,
    required: true,
    include: [{
      model: Specialization,
      required: true,
      include: [{
        model: Department,
        required: true,
        include: [{
          model: Faculty,
          where: {faculty_name: {[Op.like]: "ФКИМ"}},
        }]
      }]
    }]
  }]
})
})
.then((res: Model[]) : void => {
  console.log('\n', "4) Разница в часах, отведенных по каждой дисциплине на лабораторные и практические занятия в одном из семестров на заданной факультете?")
  console.log(JSON.stringify(res, replacer: null, space: 2), "\n")
})

```

Рисунок 10 – Запрос 4.

```

4) Разница в часах, отведенных по каждой дисциплине на лабораторные и практические занятия в одном из семестров на заданном факультете?
[
  {
    "discipline_name": "Математический анализ",
    "difference": "10.0"
  },
  {
    "discipline_name": "Физика",
    "difference": "25.0"
  },
  {
    "discipline_name": "Теория электрических цепей",
    "difference": "5.0"
  },
  {
    "discipline_name": "Базы данных",
    "difference": "15.0"
  },
  {
    "discipline_name": "WEB-технологии",
    "difference": "20.0"
  }
]

```

Рисунок 10 – Результат запроса 4.

```

await Discipline.findAll({
  attributes: ["discipline_name"],
  where: {
    cw_hours: { [Op.gt]: 0 },
  },
  include: [{
    attributes: [],
    required: true,
    model: DisciplineSpecialization,
    include: [{
      model: Specialization,
      required: true,
      where: {
        specialization_name: {[Op.in]: ['ПИ', 'ПМИ'] },
      },
    ]
  }]
})
.then((res: Model[]).void => {
  console.log('\n', "5) Дисциплины, по которым выполняет курсовые работы студенты указанной специальности?")
  console.log(JSON.stringify(res, {replacer: null, space: 2}), "\n")
})

```

Рисунок 11 – Запрос 5.

```

5) Дисциплины, по которым выполняют курсовые работы студенты указанной специальности?
[
  {
    "discipline_name": "Математический анализ"
  },
  {
    "discipline_name": "Физика"
  },
  {
    "discipline_name": "WEB-технологии"
  }
]

```

Рисунок 12 – Результат запроса 5.

```

await Discipline.findAll({
  attributes: ["discipline_name"],
  where: {
    discipline_name: {[Op.in]: ['WEB-технологии', 'Математический анализ'] },
  },
  include: [{
    attributes: ["id"],
    model: DisciplineSpecialization,
    include: [{
      model: Specialization,
      attributes: ["specialization_name"],
    }]
  }]
})
.then((res: Model[]).void => {
  console.log('\n', "6) Для каких специальностей читается указанная дисциплина?")
  console.log(JSON.stringify(res, {replacer: null, space: 2}), "\n")
})

```

Рисунок 13 – Запрос 6.



```

6) Для каких специальностей читается указанная дисциплина?
[
  {
    "discipline_name": "Математический анализ",
    "DisciplineSpecializations": [
      {
        "id": 1,
        "Specialization": {
          "specialization_name": "ПИ"
        }
      }
    ]
  },
  {
    "discipline_name": "WEB-технологии",
    "DisciplineSpecializations": [
      {
        "id": 10,
        "Specialization": {
          "specialization_name": "ПМИ"
        }
      }
    ]
  }
]

```

Рисунок 14 – Результат запроса 6.

```

await Discipline.findAll({
  attributes: ["discipline_name"],
  include: [{
    model: DisciplineSpecialization,
    attributes: ["id"],
    required: true,
    include: [{
      model: Specialization,
      required: true,
      where: {
        specialization_name: {[Op.in]: ['ПИ']},
      },
      attributes: ["specialization_name", "duration"],
    }]
  }]
})

.then((res: Model[]) => {
  console.log('\n', "7) Какое количество дисциплин входит в учебный план подготовки студентов по указанной специальности, и сколько лет осуществляется подготовка?")
  console.log(JSON.stringify(res, replacer: null, space: 2), "\n")
})

```

Рисунок 15 – Запрос 7.

```

7) Какое количество дисциплин входит в учебный план подготовки студентов по указанной специальности, и сколько лет осуществляется подготовка?
[
  {
    "discipline_name": "Математический анализ",
    "DisciplineSpecializations": [
      {
        "id": 1,
        "Specialization": {
          "specialization_name": "ПИ",
          "duration": "4.00"
        }
      }
    ]
  },
  {
    "discipline_name": "Базы данных",
    "DisciplineSpecializations": [
      {
        "id": 9,
        "Specialization": {
          "specialization_name": "ПИ",
          "duration": "4.00"
        }
      }
    ]
  }
]

```

Рисунок 16 – Результат запроса 7.

## Выводы.

В данной лабораторной работе освоена работа с ORM для Node.js - Sequelize.

## **ПРИЛОЖЕНИЕ А**

Pull request: <https://github.com/moevm/sql-2023-1303/pull/42>