**Лабораторная работа №4, №5**

**SEQUALIZE NODE JS**

**Цель:** изучить генерацию моделей, создание связей 1:1, 1:N, N:M.

*Генерация модели:*

В терминале:

Установим пакет *npm* и драйвер к mysql.

npm install --save sequelize

$ npm install --save mysql2

Создали пустой проект, используя команду:

npx sequelize-cli init

Это создаст следующие папки:

config;

models;

migrations;

seeders.

В файле config.json устанавливаем соответствующие настройки базы данных (рис. 1)

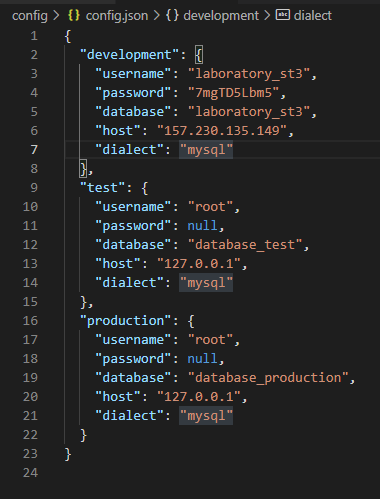


Рис.1

Создаем миграцию и модель

npx sequelize-cli model:generate --name User --attributes firstName:string,lastName:string,email:string

npx sequelize-cli model:generate --name Comment --attributes id\_user:string comment:string

npx sequelize-cli model:generate --name Login --attributes id\_user:string login:string

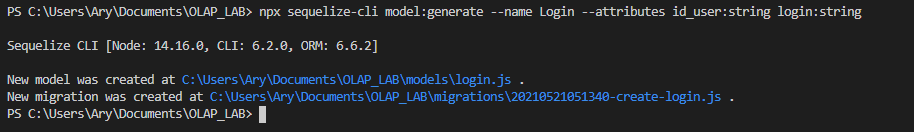


Рис. 2

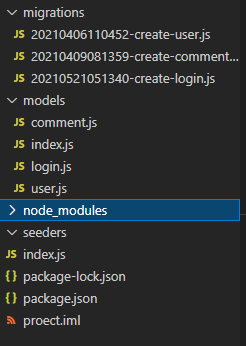


Рис.3

Создали модели в папке models;

Создали миграцию в файле …-create-user.js в папке migrations.

Теперь необходимо добавить созданные модели в базу данных. Для добавления таблицы в БД выполнили команду:

npx sequelize-cli db:migrate

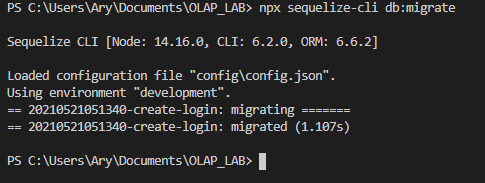


Рис.4

Листинг

Добавление записи и связь 1:1

User.create({

        firstName: "Alexandra",

        lastName: "Fedotova",

        email: "labeofernatus@srspu.ru"

    }).then(user=>{

    // Добавляем логин

        Login.create({name:" alexa"}).then(login\_user=>{

        // устнавливаем логин для пользователя

        user.setLogin(login\_user).catch(err=>console.log(err));

    });

}).catch(err=>console.log(err));

Добавили пользователя с id = 23

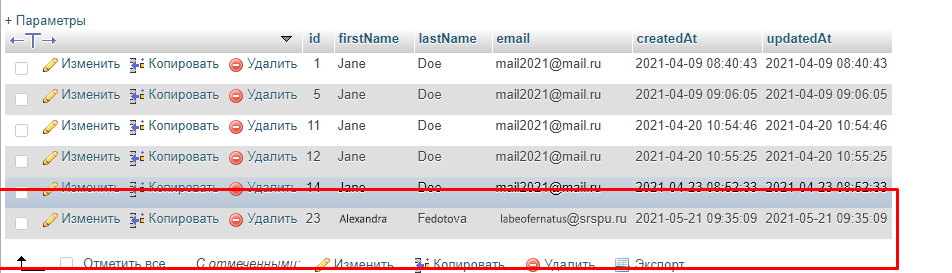


Рис.5

Добавили логин для этого пользователя

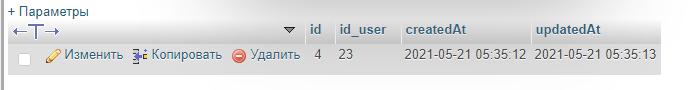


Рис.6

Листинг . Удаление записи

User.destroy({

        where: {

            id: 28

        }

    })

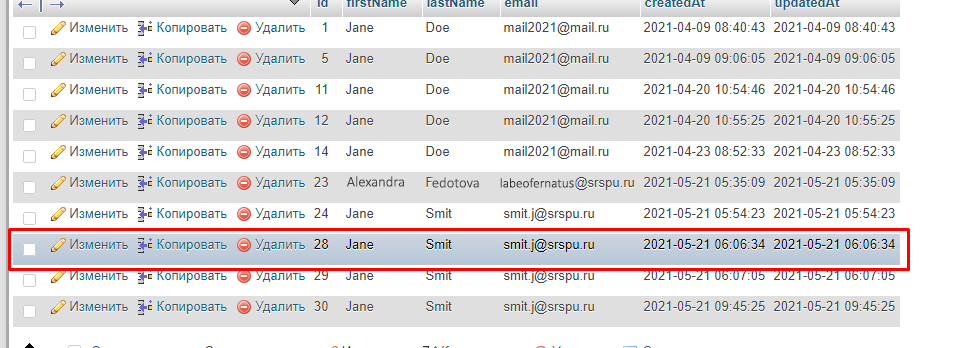


Рис.7

Листинг

Работа со связью 1:N

    const alexa = await User.findByPk(23);

    console.log("alexa's auto-generated ID:", alexa.id);

    //Добавить комментарий

    const comment = await Comment.create({user\_id: alexa.id, comment: "comment" });

    const comment\_1 = await Comment.create({user\_id: alexa.id, comment: "можно 5 автоматом?" });

    const comment\_2 = await Comment.create({user\_id:alexa.id, comment: "барманен бабанер не чарчапен" });

    const comment\_delete = await Comment.create({user\_id:24, comment:"удалить"});

    //Удалить комментарий

    await Comment.destroy({

        where:{

            id:9,

            user\_id:jane.id

            //comment: "На удаление"

        }

 });

    //Найти комментарии

        User.findAll({

        include: [{

          model: Comment,

          as: 'Comments'

        }],

        attributes:{ exclude: ['createdAt','updatedAt'] }

      }).then((users)=>{

        console.log(users)

        });;

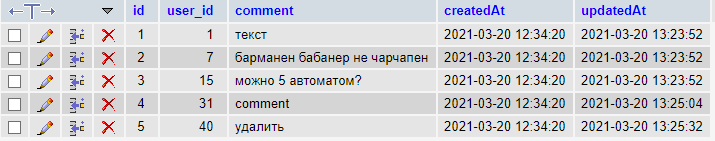


Рис.8 – Создание комментариев

C:\Users\Alexandra\Desktop\1.png

Рис.9 –Удаление комментария

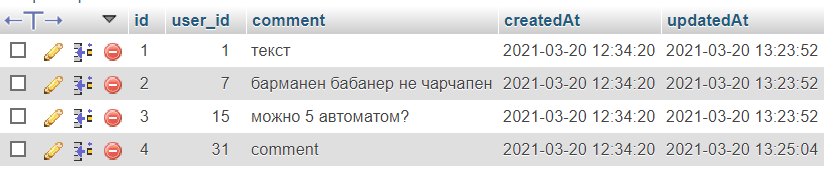


Рис.10 – Удаление комментария

Листинг

Работа со связью M:M

Устанавливаем внешние ключи в таблицах User и Courses.

В user.js:

User.belongsToMany(models.Course,{through:'enty'});

В course.js:

Course.belongsToMany(models.User,{through:'enty'});

Создаем записи в этих таблицах и связываем таблицы, добавляя запись типа связи

Листинг index.js

User.create({

        first\_name:"Имя1",

        last\_name:"Фамилия1",

        email:"email2"

    });

    User.create({

        first\_name:"Имя2",

        last\_name:"Фамилия2",

        email:"email2"

    });

    User.create({

        firstName:"Имя3",

        lastName:"Фамилия3",

        email:"email3"

    });

    Course.create({

        name:"name1"

    });

    Course.create({

        name:"name2"

    });

    Course.create({

        name:"name3"

    });

     //нашли пользователя

    const user = await User.findByPk(23);

//нашли курс

    const course = await Course.findOne({

        where:{name:"name2"}

    });

//добавили запись в таблицу связку

    await user.addCourse(course, { through: { row\_1: "false", row\_2:"true", row\_3:"кошмар" } });

        //вывели результат работы

User.findOne({where: {id:23}})

        .then(user=>{

            if(!user) return;

            user.getCourses().then(courses=>{

                for(course of courses){

                    console.log(user.lastName + " " + course.name);

                }

            });

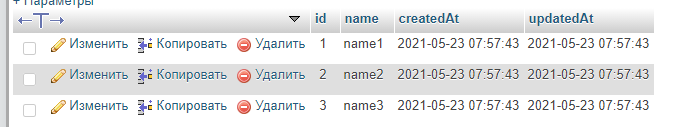


Рис.11

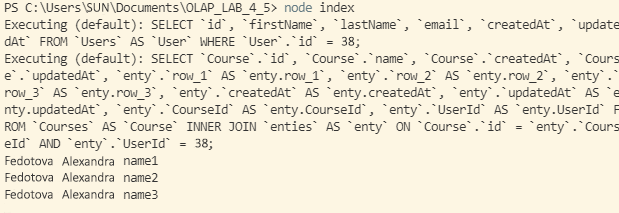


Рис.12



**Вывод:** изучили генерацию моделей, выполнение CREATE, DELETE и SELECT – операций.