

НИУ «ИТМО»

Отчёт о
лабораторной работе №1
по дисциплине «Базы данных и знаний»

Выполнил
студент Крайнов.В.А

Проверил
преподаватель Субботин А. Н.

Санкт-Петербург
2024

individuals
id
name
fathurname
surname
passport
tin
ssn
driver_license
extradocs
note

individual_loans
id
individual_id
sum
percent
rate
duration
terms
note

entity_loans
id
entity_id
individual_id
sum
duration
percent
terms
note



individual_loans
123 id
123 individual_id
ABC sum
123 percent
123 rate
123 duration
ABC terms
ABC note

individuals
123 id
ABC name
ABC fathername
ABC surname
ABC passport
ABC tin
ABC ssn
ABC driver_license
ABC extradocs
ABC note

entity_loans
123 id
123 entity_id
123 individual_id
ABC sum
123 duration
123 percent
ABC terms
ABC note



individual_loans
123 id
123 individual_id
ABC sum
123 percent
123 rate
123 duration
ABC terms
ABC note

individuals
123 id
ABC name
ABC fathename
ABC surname
ABC passport
ABC tin
ABC ssn
ABC driver_license
ABC extradocs
ABC note
123 loaner_id

loaners
123 id
ABC tin
123 is_entity
ABC address
ABC sum
ABC terms
ABC law_note
ABC documents

entity_loans
123 id
123 entity_id
123 individual_id
ABC sum
123 duration
123 percent
ABC terms
ABC note

```

// работа с базой данных
const mysql = require('mysql2/promise');
const connection = mysql.createPool({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'bank'
});

// обработка параметров из формы
const qs = require('querystring');
async function reqPost (request, response) {
  if (request.method !== 'POST') return;

  let rawBody = '';

  request.on('data', chunk => {
    rawBody += chunk.toString();
  })

  request.on('end', async() => {
    const body = qs.parse(rawBody);

    const sql = "INSERT INTO persons (text, description, keywords) VALUES
(\\""+body['col1']+"\\",\\""+body['col2']+"\\",\\""+body['col3']+"\\")";
    await connection.query(sql);
  });
}

// выгрузка массива данных
async function ViewSelect(response) {
  const [result1] = await connection.query('SHOW COLUMNS FROM persons');
  response.write('<tr>');
  for(const res of result1) {
    response.write(`<td>${res.Field}</td>`);
  }
  response.write('</tr>');

  const [result2] = await connection.query('SELECT * FROM persons WHERE id>14 ORDER
BY id DESC');
  for(const res of result2) {
    response.write(`<tr><td>${res.id.toString()}</td><td>${res.text}</td><td>${
res.description}</td><td>${res.keywords}</td></tr>`)
  }
}

async function ViewVer(response) {
  const [result] = await connection.query('SELECT VERSION() AS ver');
  response.write(result[0].ver);
}

// создание ответа в браузер, на случай подключения
const http = require('http');
const fs = require('fs');
const path = require('path');
const { exit } = require('process');
const server = http.createServer(async(req, res) => {
  reqPost(req, res);
  console.log('Loading...');
});

```

```

    res.statusCode = 200;
    // res.setHeader('Content-Type', 'text/plain');

    // чтение шаблона в каталоге со скриптом
    const selectHtml = path.join(__dirname, 'select.html');
    const html = fs.readFileSync(selectHtml).toString().split("\n");
    for(const row of html) {
        // подстановка
        if (row.trim() === '@tr') await ViewSelect(res);
        else if (row.trim() === '@ver') await ViewVer(res);
        else res.write(row);
    }
    res.end();
    console.log('1 User Done.');
```

});

```

// запуск сервера, ожидание подключений из браузера.
const hostname = '127.0.0.1';
const port = 3000;
server.listen(port, hostname, () => {
    console.log(`Server running at http://${hostname}:${port}/`);
});
```