ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ ООО «NAVTO»

РАБОЧАЯ ДОКУМЕНТАЦИЯ

Техническая документация

Пермь, 2022

**Содержание**

[1. Разработка базы данных 3](#_Toc122355056)

[1.1 Создание таблиц 3](#_Toc122355057)

[1.2 Добавление ограничений 4](#_Toc122355058)

[1.3 Создание ролей 5](#_Toc122355059)

[2. Разработка клиентской части 6](#_Toc122355060)

[2.1 Загрузчик таблиц 6](#_Toc122355061)

[2.2 Добавление данных 6](#_Toc122355062)

[2.3 Изменение данных 7](#_Toc122355063)

[2.4 Удаление данных 7](#_Toc122355064)

# 1. Разработка базы данных

## 1.1 Создание таблиц

Для определения таблиц БД использовались следующие SQL-запросы:

1. Таблица «Заявки»

CREATE TABLE `application` (

`id\_application` int(11) NOT NULL,

`id\_service` int(11) NOT NULL,

`id\_employee` int(11) NOT NULL,

`id\_client` int(11) NOT NULL,

`date\_registration` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

1. Таблица «Клиенты»

CREATE TABLE `client` (

`id\_client` int(11) NOT NULL,

`name\_client` varchar(50) NOT NULL,

`sname\_client` varchar(50) NOT NULL,

`patron\_client` varchar(50) NOT NULL,

`phone\_number` varchar(20) NOT NULL,

`email` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

1. Таблица «Сотрудники»

CREATE TABLE `employee` (

`id\_employee` int(11) NOT NULL,

`name\_employee` varchar(50) NOT NULL,

`sname\_employee` varchar(50) NOT NULL,

`patron\_employee` varchar(50) NOT NULL,

`date\_birth` varchar(30) NOT NULL,

`phone\_number` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

1. Таблица «Клиенты»

CREATE TABLE `service` (

`id\_service` int(11) NOT NULL,

`name\_service` varchar(50) NOT NULL,

`cost\_service` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## 1.2 Добавление ограничений

--

-- Индексы таблицы `application`

--

ALTER TABLE `application`

ADD PRIMARY KEY (`id\_application`);

--

-- Индексы таблицы `client`

--

ALTER TABLE `client`

ADD PRIMARY KEY (`id\_client`);

--

-- Индексы таблицы `employee`

--

ALTER TABLE `employee`

ADD PRIMARY KEY (`id\_employee`);

--

-- Индексы таблицы `service`

--

ALTER TABLE `service`

ADD PRIMARY KEY (`id\_service`);

--

-- AUTO\_INCREMENT для сохранённых таблиц

--

--

-- AUTO\_INCREMENT для таблицы `application`

--

ALTER TABLE `application`

MODIFY `id\_application` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=7;

--

-- AUTO\_INCREMENT для таблицы `client`

--

ALTER TABLE `client`

MODIFY `id\_client` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

-- AUTO\_INCREMENT для таблицы `employee`

--

ALTER TABLE `employee`

MODIFY `id\_employee` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=6;

--

-- AUTO\_INCREMENT для таблицы `service`

--

## 1.3 Создание ролей

1. Менеджер

GRANT SELECT ON \*.\* TO `manager`@`localhost` IDENTIFIED BY PASSWORD '\*89C6B530AA78695E257E55D63C00A6EC9AD3E977';

GRANT SELECT, INSERT ON `exam`.`application` TO `manager`@`localhost`;

GRANT SELECT ON `exam`.`service` TO `manager`@`localhost`;

GRANT SELECT ON `exam`.`client` TO `manager`@`localhost`;

GRANT SELECT ON `exam`.`employee` TO `manager`@`localhost`;

2. Старший менеджер

GRANT USAGE ON \*.\* TO `senior\_manager`@`localhost` IDENTIFIED BY PASSWORD '\*89C6B530AA78695E257E55D63C00A6EC9AD3E977';

GRANT SELECT, INSERT, UPDATE, DELETE ON `exam`.\* TO `senior\_manager`@`localhost`;

# 2. Разработка клиентской части

## 2.1 Загрузчик таблиц

private void LoadEmployee(MySqlConnection conn)

{

dataGridViewEmployee.Rows.Clear();

string st = "SELECT \* FROM employee";

MySqlCommand command = new MySqlCommand(st, conn);

MySqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[6]);

data[data.Count - 1][0] = reader[0].ToString();

data[data.Count - 1][1] = reader[1].ToString();

data[data.Count - 1][2] = reader[2].ToString();

data[data.Count - 1][3] = reader[3].ToString();

data[data.Count - 1][4] = reader[4].ToString();

data[data.Count - 1][5] = reader[5].ToString();

}

reader.Close();

foreach (string[] s in data)

dataGridViewEmployee.Rows.Add(s);

}

## 2.2 Добавление данных

private void button1\_Click(object sender, EventArgs e)

{

if ((textBox1.Text == "") || (textBox2.Text == "") || (textBox3.Text == "") ||

(textBox4.Text == "") || (maskedTextBox2.Text == ""))

{

MessageBox.Show("Заполните все поля корректно!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

string connect = "server=localhost; user=root; database=exam;";

MySqlConnection conn = new MySqlConnection(connect);

conn.Open();

string sql = "INSERT INTO client (name\_client, sname\_client, patron\_client, phone\_number, email)" + "VALUES(@name\_client, @sname\_client, @patron\_client, @phone\_number, @email)";

MySqlCommand command = new MySqlCommand(sql, conn);

command.Parameters.AddWithValue("name\_client", textBox2.Text);

command.Parameters.AddWithValue("sname\_client", textBox1.Text);

command.Parameters.AddWithValue("patron\_client", textBox3.Text);

command.Parameters.AddWithValue("email", textBox4.Text);

command.Parameters.AddWithValue("phone\_number", maskedTextBox2.Text);

command.ExecuteNonQuery();

conn.Close();

this.Close();

}

}

## 2.3 Изменение данных

private void button2\_Click(object sender, EventArgs e)

{

string connect = "server=localhost; user=root; database=exam;";

string str = this.dataGridViewEmployee.CurrentRow.Cells["id\_employee"].Value.ToString();

DialogResult check = MessageBox.Show($"Вы хотите изменить запись {str}?", "Изменение", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (check == System.Windows.Forms.DialogResult.Yes)

{

MySqlConnection conn = new MySqlConnection(connect);

conn.Open();

string s = this.dataGridViewEmployee.CurrentCell.OwningColumn.Name.ToString();

string sql = $"UPDATE employee SET {s} = @new WHERE employee.id\_employee = @num";

MySqlCommand command = new MySqlCommand(sql, conn);

command.Parameters.Add(new MySqlParameter("@new", this.dataGridViewEmployee.CurrentCell.Value));

command.Parameters.Add(new MySqlParameter("@num", this.dataGridViewEmployee.CurrentRow.Cells["id\_employee"].Value));

command.ExecuteNonQuery();

MessageBox.Show($"Запись {str} успешно изменена", "Изменение", MessageBoxButtons.OK, MessageBoxIcon.Information);

LoadEmployee(conn);

conn.Close();

}

}

## 2.4 Удаление данных

private void button3\_Click(object sender, EventArgs e)

{

string connect = "server=localhost; user=root; database=exam;";

string str = this.dataGridViewEmployee.CurrentRow.Cells["id\_employee"].Value.ToString();

DialogResult check = MessageBox.Show($"Вы хотите удалить запись {str}?", "Удаление", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (check == System.Windows.Forms.DialogResult.Yes)

{

MySqlConnection conn = new MySqlConnection(connect);

conn.Open();

string sql = "DELETE FROM employee WHERE id\_employee = @num";

MySqlCommand command = new MySqlCommand(sql, conn);

command.Parameters.Add(new MySqlParameter("@num", this.dataGridViewEmployee.CurrentRow.Cells["id\_employee"].Value));

command.ExecuteNonQuery();

MessageBox.Show($"Запись {str} успешно удалена", "Удаление", MessageBoxButtons.OK, MessageBoxIcon.Information);

LoadEmployee(conn);

conn.Close();

}

}