

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
Национальный исследовательский университет «МЭИ»

Типовой расчёт
по курсу «**Базы данных**»

Выполнил:
студент группы А-08-20
Харламов Андрей

Москва
2022

I. Тема типового расчёта

Предлагаемая тема — разработка базы данных для компании, занимающейся постройкой гражданского жилья. В требуемой базе данных необходимо хранить информацию о работах которые сейчас ведутся компанией, о сотрудниках компании а также хранить и обновлять информацию о количестве инструментов и материалов на складах компании и о том. сколько из них сейчас задействовано в работах.

II. Описание бизнес-процесса

Для успешного функционирования, компании необходим постоянный доступ к следующей информации:

- *Данные о сотрудниках компании* — ФИО, профессия, зарплата, в какой из работ принимает участие в данный момент
- *Данные о текущих работах, проводимых компанией* — её название, сроки, участвующие сотрудники, используемые инструменты и материал.
- *Данные о количестве материалов и инструментов на складе* — название, количество, цена.
- *Данные о новых поступлениях на склад материалов или инструментов* — сотрудника, который доставил товар, название товара, его количество, и дату привоза.

В случае поставки материала или инструмента необходимо обновлять баланс товара на складе.

III. Группы пользователей

1) Рабочие не имеют доступа к базе данных

Имеют возможность узнать информацию о сотруднике по его ФИО

2) Группа прорабов — имеют частичный доступ к базе данных.

Имеют следующие возможности:

- Получить информацию о задействованных на данный момент сотрудниках;
- Получить информацию о сотрудниках свободных в указанный период времени;
- Внести в базу данных информацию о новом сотруднике;
- Назначить сотрудника на выполнение определённой работы;
- Запросить инструменты и материалы со склада на рабочее место.

3) Технический отдел — имеют частичный доступ к базе данных

Имеют следующие возможности:

- Внести в базу данных информацию о новых поступлениях на склад;
- Добавить новые материалы и инструменты на склад;
- Получить данные о том, какие материалы и инструменты хранятся на складе сейчас.

4) Совет директоров компании — имеют полный доступ к базе данных

Имеют следующие возможности:

- Объединяют в себе функционал двух предыдущих групп пользователей;
- Добавить или удалить пользователя;
- Добавить новую работу;
- Получить информацию о цене всех инструментов и материалов на складе.

IV. Концептуальное проектирование. ER-диаграмма. Таблицы, индексы, ключи, связи, ограничения

Все данные, которые были описаны в пункте 2 необходимо организовать в виде таблиц, а именно сделать нормализацию, сделать данные атомарными, разделить на сущности и избавиться от избыточных данных и провести связи между таблицами.

Полученная ER-диаграмма БД Строительной компании представлена на рисунке 1.

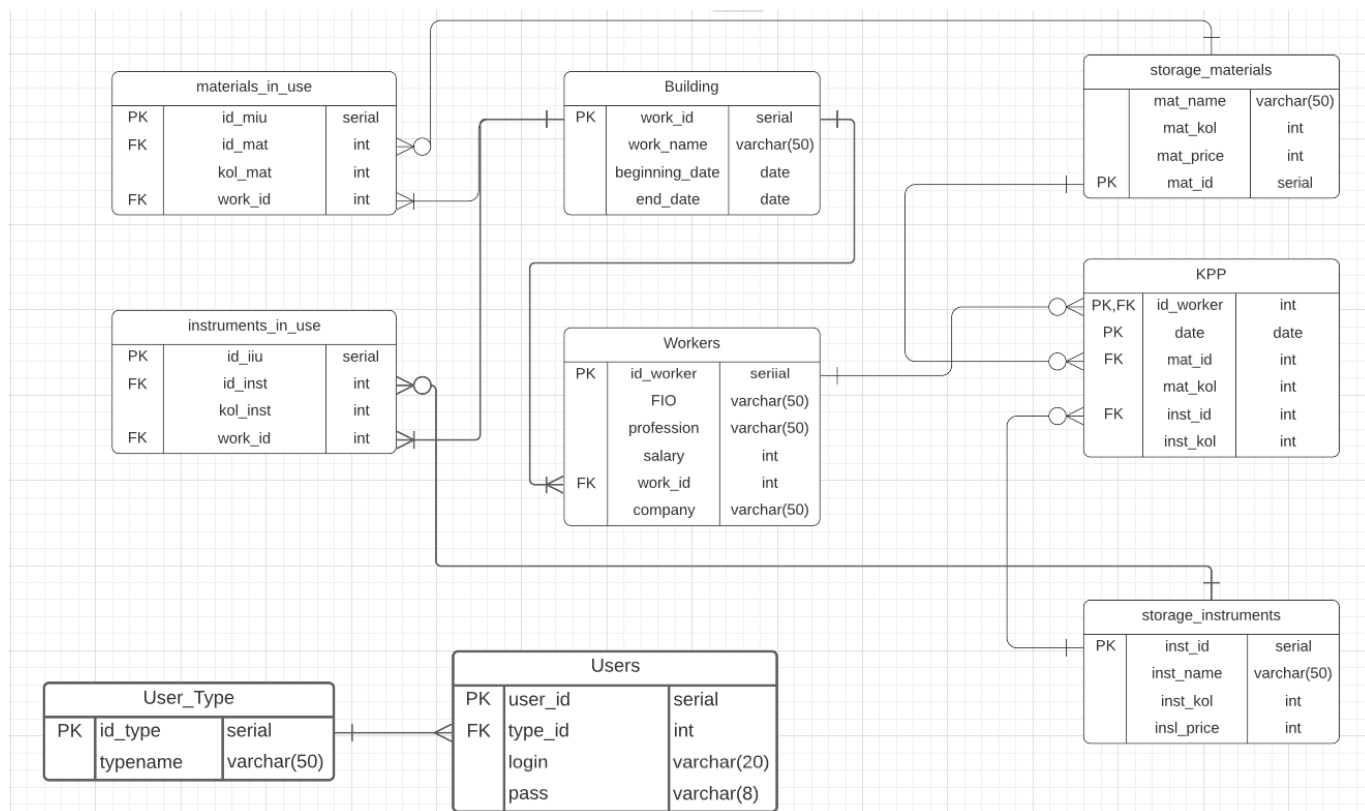


Рисунок 1. ER – диаграмма БД

V. Физическая модель БД

- Создание таблиц
-

1) Создание таблицы Building

```
create table tr.building (  
work_id serial primary key,  
work_name varchar(50),  
begin_date date not null,  
end_date date check (end_date > begin_date));
```

Рисунок 2. Building

В ней хранятся данные о текущих работах компании — id работы, название, и сроки начала и окончания этих работ.

2) Создание таблицы Workers

```
create table tr.workers (  
  worker_id serial primary key,  
  FIO varchar(80),  
  profession varchar(50) default 'разнорабочий',  
  salary int check (salary > 0),  
  company varchar(50) default 'MyCompany',  
  work_id int );
```

Рисунок 3. Workers

В ней хранятся данные обо всех сотрудниках занятых на работах компании — id, ФИО, профессия, зарплата, компания или бригада и id работы на которую назначен данный сотрудник.

3) Создание таблицы KPP

```
create table tr.kpp (  
  worker_id int not null,  
  kpp_date date not null,  
  mat_id int default null,  
  mat_kol int default null,  
  inst_id int default null,  
  inst_kol int default null,  
  primary key (worker_id, kpp_date));
```

Рисунок 4. KPP

В этой таблице хранится информация о поставках новых материалов и инструментов на склад — id работника который доставил, дата поставки, количество материала и его id, количество инструмента и его id.

4) Создание таблиц с данными со склада

```
-- таблица storage materials --  
create table tr.storage_mat(  
  mat_id serial primary key,  
  mat_kol int check (mat_kol >= 0),  
  mat_price int check (mat_price > 0),  
  mat_name varchar(50) not null);  
  
-- таблица storage instruments --  
create table tr.storage_inst(  
  inst_id serial primary key,  
  inst_kol int check (inst_kol >= 0),  
  inst_price int check (inst_price > 0),  
  inst_name varchar(50) not NULL);
```

Рисунок 5. Storage_mat и Storage_inst

Эти таблицы имеют идентичную структуру, и хранят в себе информацию о материалах и инструментах хранящихся в данный момент на складе — id, название, количество (в тоннах для материалов, и в штуках для инструментов), и цену (в рублях за тонну для материалов, и в рублях за штуку для инструментов).

5) Создание таблиц с данными о материалах/инструментах которые в данный момент используются в работе

```
-- таблица materials in use --
create table tr.mat_in_use(
  id_miu serial primary key,
  mat_id int not null,
  mat_kol int check (mat_kol >= 0),
  work_id int not null);

-- таблица instruments in use --
create table tr.inst_in_use(
  id_iiu serial primary key,
  inst_id int not null,
  inst_kol int check (inst_kol >= 0),
  work_id int not null);
```

Рисунок 6. *Materials_in_use* и *Instruments_in_use*

Данные таблицы также похожи друг на друга, В них содержится информация, о том на какой рабочей точке, какой материал/инструмент используется и в каком количестве (в тоннах для материалов, и в штуках для инструментов)

6) Создание таблиц с информацией о пользователях

```
-- таблица User type --
create table tr.user_type (
  id_type serial primary key,
  typename varchar(20));

-- таблица users --
create table tr.users(
  user_id serial primary key,
  type_id int,
  login varchar(20),
  pass varchar(8));
```

Рисунок 7. *User_type* и *Users*

User_type хранит в себе название группы пользователей и её id, а users — логин пароль и id группы к которой относиться данный пользователь.

- **Создание связей между таблицами**

```
-- Создание связей между таблицами --

-- building родительская для workers, тк сотрудники могут выполнять работы только на описанных в building проектах --
alter table tr.workers add foreign key (work_id) references tr.building on delete cascade;

-- building родительская для mat_in_use тк материалы используются только на работах описанных в building
alter table tr.mat_in_use add foreign key (work_id) references tr.building on delete cascade;

-- building родительская для inst_in_use тк инструменты используются только на работах описанных в building
alter table tr.inst_in_use add foreign key (work_id) references tr.building on delete cascade;

-- storage_mat родительская для mat_in_use тк в работе используются только те материалы которые описаны в storage_mat
alter table tr.mat_in_use add foreign key (mat_id) references tr.storage_mat on delete cascade;

-- storage_inst родительская для inst_in_use тк в работе используются только те инструменты которые описаны в storage_mat
alter table tr.inst_in_use add foreign key (inst_id) references tr.storage_inst on delete cascade;

/* таблица KPP дочерняя для таблиц workers, storage_mat и storage_inst
   тк в неё можно внести данные только с теми сотрудниками, материалами и инструментами
   которые описаны в вышеперечисленных таблицах */

alter table tr.kpp add foreign key (worker_id) references tr.workers on delete cascade;
alter table tr.kpp add foreign key (mat_id) references tr.storage_mat on delete cascade;
alter table tr.kpp add foreign key (inst_id) references tr.storage_inst on delete cascade;

-- связь таблиц где хранятся пароли доступа --
alter table tr.users add foreign key (type_id) references tr.user_type on delete cascade;
```

Рисунок 8. Создание связей между таблицами

VI. Скалярные функции. Табличные функции. Представления. Процедуры.

Скалярные функции

```
-- подсчитать цену всех материалов на складе --
create or replace function all_mat_price() returns int
as $$
begin
    return (
        select sum(mat_price * mat_kol) from tr.storage_mat);
end;
$$ language plpgsql;
end;

select all_mat_price();

-- подсчитать цену всех инструментов на складе --
create or replace function all_inst_price() returns int
as $$
begin
    return (
        select sum(inst_price * inst_kol) from tr.storage_inst);
end;
$$ language plpgsql;
end;

select all_inst_price();
```

Рисунок 9. Скалярные функции

Данные функции возвращают общую цену материалов/инструментов, которые в данный момент находятся на складе.

```

-- функция для получения группы пользователя (авторизация)--
create or replace function get_typename(l varchar(20), p varchar(8))
returns varchar(20)
as $$
begin
    return
    (select typename from tr.user_type
     join tr.users on tr.users.type_id = tr.user_type.id_type
     where login = l and pass = p);
end;
$$ language plpgsql;

select get_typename('superuser1', '00001111');

```

Рисунок 10. Функция get_typename

Функция необходимая при авторизации, возвращает группу пользователя по его логину и паролю.

Табличные функции

Функции реализующие функционал для группы пользователей - «рабочий»

```

create or replace function select_workers_data (wname varchar(80)) returns table ("work name" varchar(50),
FIO varchar(80), profession varchar(50), company varchar(50), salary int)
as $$
begin
    return query
    select * from workers_view
    where wname = workers_view.FIO;
end;
$$ language plpgsql;

```

Рисунок 11. Функция select_workers_data

Возвращает информацию о сотруднике (вид работ, ФИО, профессию, компанию и зарплату) принимая на вход имя.

Функции реализующие функционал для группы пользователей - «Тех. Отдел»

```

-- функции для получения данных о материалах на складе --
--DROP FUNCTION get_stor_mat_info()
create or replace function get_stor_mat_info() returns table (material varchar(50), kol int)
as $$
begin
    return query
    select mat_name, mat_kol from tr.storage_mat
    order by mat_name ;
end;
$$ language plpgsql;

--select * from get_stor_mat_info();

-- функции для получения данных об инструментах на складе --
create or replace function get_stor_inst_info() returns table (instrument varchar(50), kol int)
as $$
begin
    return query
    select inst_name, inst_kol from tr.storage_inst
    order by inst_name ;
end;
$$ language plpgsql;

```

Рисунок 12. Функции get_stor_mat_info и get_stor_inst_info

Данные функции возвращают название и количество материалов/инструментов на складе в данный момент.

Функции реализующие функционал для группы пользователей - «Прораб»

```
create or replace function get_all_works() returns table (work_name varchar(50), profession varchar(50),  
fio varchar(80), bd date, ed date)  
as $$  
begin  
    return query  
    select "Work name", allworks_now."profession" as profession, "Worker" as fio, "Begin" as bd,  
    "End" as ed from allworks_now  
    order by work_name;  
end;  
$$ language plpgsql;
```

Рисунок 13. Функция get_all_works

Данная функция возвращает список всех ведущихся сейчас работ с занятыми там сотрудниками.

```
create or replace function all_free_workers(bd date, ed date) returns table (profession varchar(50),  
fio varchar(80))  
as $$  
begin  
    return query  
    select workers.profession as profession, workers.fio as fio from tr.workers  
    join tr.building on building.work_id = workers.work_id  
    where building.begin_date > ed or building.end_date < bd;  
end;  
$$ language plpgsql;
```

Рисунок 14. Функция all_free_workers

Данная функция возвращает список всех свободных на определённый срок (принимает дату начала и дату окончания как входные параметры) работников — их ФИО и профессию.

```
create or replace function get_all_miu() returns table (material varchar(50), mat_kol int, work_name varchar(50))  
as $$  
begin  
    return query  
    select mat_name, mat_in_use.mat_kol, building.work_name from tr.mat_in_use  
    join tr.building on building.work_id = mat_in_use.work_id  
    join tr.storage_mat on storage_mat.mat_id = mat_in_use.mat_id  
    order by material;  
end  
$$ language plpgsql;
```

```
create or replace function get_all_iiu() returns table (instrument varchar(50), inst_kol int, work_name varchar(50))  
as $$  
begin  
    return query  
    select inst_name, inst_in_use.inst_kol, building.work_name from tr.inst_in_use  
    join tr.building on building.work_id = inst_in_use.work_id  
    join tr.storage_inst on storage_inst.inst_id = inst_in_use.inst_id  
    order by instrument;  
end  
$$ language plpgsql;
```

Рисунок 15. Функции get_all_miu и get_all_iiu

Возвращают информацию о материалах/инструментах которые используются в работах — название, количество, и название работы где он используется.

Процедуры

Процедуры реализующие функционал для группы пользователей - «Тех. Отдел»

```
-- Добавление записи в KPP --
create or replace procedure new_kpp (w int, kd date, mi int, mk int, ii int, ik int)
language sql
as $$
insert into tr.kpp (worker_id, kpp_date, mat_id, mat_kol, inst_id, inst_kol)
values (w, kd, mi, mk, ii, ik);
$$
```

Рисунок 16. Процедура new_kpp

Реализует добавление новой записи в таблицу kpp – добавление информации о новых поставках на склад.

```
-- процедура для загрузки нового материала в storage_mat --
create or replace procedure add_new_to_storage_mat (mn varchar(50), mp int)
as $$
begin
insert into tr.storage_mat (mat_kol, mat_price, mat_name)
values (0, mp, mn);
end;
$$ language plpgsql;

-- процедура для загрузки новых инструментов в storage_inst --
create or replace procedure add_new_to_storage_inst (inn varchar(50), ip int)
as $$
begin
insert into tr.storage_inst (inst_kol, inst_price, inst_name)
values (0, ip, inn);
end;
$$ language plpgsql;
```

Рисунок 17. Процедуры add_new_to_storage_mat и add_new_to_storage_inst

Реализуют добавление новой записи в таблицы storage_inst и storage_mat – добавление нового вида материала/инструмента на склад.

Процедуры реализующие функционал для группы пользователей - «Прораб»

```
-- процедура добавляющая нового сотрудника --
create or replace procedure add_new_worker(f varchar(80), p varchar(50), s int, c varchar(50), w int)
language sql
as $$
insert into tr.workers (fio, profession, salary, company, work_id)
values (f, p, s, c, w);
$$
```

Рисунок 18. Процедура add_new_worker

Реализует добавление нового сотрудника в базу данных.

```
create or replace procedure give_work_to_worker(f varchar(80), job varchar(50))
language sql
as $$
update tr.workers
--join tr.building on tr.workers.work_id = tr.building.work_id
set work_id = (select work_id from tr.building where work_name = job)
where tr.workers.fio = f;
$$
```

Рисунок 19. Процедура give_work_to_worker

Позволяет назначить имеющегося сотрудника на какой-либо вид работ.

Процедуры реализующие функционал для группы пользователей - «Директор»

```
-- процедура для загрузки данных в building --
create or replace procedure add_to_building(wn varchar(50), bd date, ed date)
language sql
as $$
insert into tr.building (work_name, begin_date, end_date)
values (wn, bd, ed);
$$
```

Рисунок 20. Процедура add_to_building

Позволяет добавить в базу данных новый вид работ.

```
-- процедура добавления нового пользователя --
create or replace procedure add_new_user(type_ varchar(20), l varchar(20), p varchar(8))
as $$
declare
ti int;
begin
ti = (select id_type from tr.user_type where typename = type_);
insert into tr.users (type_id, login, pass)
values (ti, l, p);
end;
$$ language plpgsql
```

Рисунок 21. Процедура add_new_user

Позволяет добавить нового пользователя в базу данных.

```
--Удаление пользователя
create or replace procedure delete_user(l varchar(20))
language sql
as $$
delete from tr.users
where login = l;
$$
```

Рисунок 22. Процедура delete_user

Осуществляет удаление пользователя.

Представления

```
create view allworks_now as
select work_name as "Work name", begin_date as "Begin", end_date as "End",
fio as "Worker", profession, company from tr.building
join tr.workers on workers.work_id = building.work_id
where current_date between begin_date and end_date
group by work_name, company, profession, fio, end_date, begin_date
order by end_date;
```

Рисунок 23. Представление allworks_now

Содержит информацию о работах, ведущихся в данный момент, и занятых в них сотрудниках.

```

create view workers_view as
select work_name as "Work name", fio as FIO, profession, company, salary from tr.workers
join tr.building on workers.work_id = building.work_id;

```

Рисунок 24. Представление workers_view

Содержит информацию обо всех сотрудниках которые когда-либо задействованы во всех работах которые описаны в БД.

VII. Триггеры

```

create or replace function new_mi_kol() returns trigger
as $tr_kpp$
declare
stor_kol_m real;
new_kol_m real;
kol_m real;
stor_kol_i int;
new_kol_i int;
kol_i int;
begin
if new.mat_id is not null then -- если привезли новые материалы
stor_kol_m = (select mat_kol from tr.storage_mat where mat_id = new.mat_id);
new_kol_m = new.mat_kol;
kol_m = stor_kol_m + new_kol_m;
call upd_sm (new.mat_id, kol_m);
end if;
if new.inst_id is not null then -- если привезли новые инструменты
stor_kol_i = (select inst_kol from tr.storage_inst where inst_id = new.inst_id);
new_kol_i = new.inst_kol;
kol_i = stor_kol_i + new_kol_i;
call upd_si (new.inst_id, kol_i);
end if;
return new;
end;
$tr_kpp$ language plpgsql;

create trigger tr_kpp after insert on tr.kpp
for each row execute procedure new_mi_kol();

```

Рисунок 25. Триггер tr_kpp и функция new_mi_kol

После того как добавляются данные в таблицу kpp (с помощью процедуры new_kpp), происходит обновление информации о количестве материала/инструментов на складе к уже имеющемуся там количеству материалов/инструментов добавляется то количество которое было доставлено на склад — т. е. количество записанное в kpp.

Вспомогательные процедуры:

```
-- update данных в таблице storage_mat--
create or replace procedure upd_sm (mi int, mk int)
language sql
as $$
update tr.storage_mat set mat_kol = mk where mat_id = mi;
$$

-- update данных в таблице storage_inst--
create or replace procedure upd_si (ii int, ik int)
language sql
as $$
update tr.storage_inst set inst_kol = ik where inst_id = ii;
$$
```

Рисунок 26. Процедуры upd_sm и upd_si

Отвечают за обновление данных о количестве того или иного материала или инструмента на складе.

```
-- триггер при создании запроса о новых материалах со склада-----
create or replace function new_miu() returns trigger
as $tr_miu$
begin
    -- если на складе материала меньше чем нужно для работы
    if new.mat_kol > (select mat_kol from tr.storage_mat where tr.storage_mat.mat_id = new.mat_id)
    then raise exception 'На складе не хватает материала';
end if;
    -- если на складе достаточно материала уменьшаем количество на складе и переносим сколько нужно в miu
    if new.mat_kol <= (select mat_kol from tr.storage_mat where tr.storage_mat.mat_id = new.mat_id)
    then update tr.storage_mat set mat_kol = mat_kol - new.mat_kol where mat_id = new.mat_id;
end if;
return new;
end;
$tr_miu$ language plpgsql;

create trigger tr_miu before insert on tr.mat_in_use
for each row execute procedure new_miu();
```

Рисунок 27. Триггер tr_miu и функция new_miu

Срабатывает при попытке запросить новые материалы для работы — если на складе достаточно материала, то забираем нужное количество оттуда и вписываем его в таблицу materials_in_use, иначе ошибка так как материала на складе не достаточно чтобы обеспечить данную работу.

```
-- триггер при создании запроса о новых инструментах со склада -----
create or replace function new_iiu() returns trigger
as $tr_iiu$
begin
    -- если на складе Инструментов меньше чем нужно для работы
    if new.inst_kol > (select inst_kol from tr.storage_inst where tr.storage_inst.inst_id = new.inst_id)
    then raise exception 'На складе не хватает инструментов';
end if;
    -- если на складе достаточно инструмента, уменьшаем количество на складе и переносим сколько нужно в iiu
    if new.inst_kol <= (select inst_kol from tr.storage_inst where tr.storage_inst.inst_id = new.inst_id)
    then update tr.storage_inst set inst_kol = inst_kol - new.inst_kol where inst_id = new.inst_id;
end if;
return new;
end;
$tr_iiu$ language plpgsql;

create trigger tr_iiu before insert on tr.inst_in_use
for each row execute procedure new_iiu();
```

Рисунок 27. Триггер tr_iiu и функция new_iiu

Данный триггер работает аналогично предыдущему, только работает с информацией об инструментах, а не материалах.

VIII. Приложение для работы с базой данных

Разработанное приложение имеет уникальный вид для различных групп пользователей, предусмотренных в ТЗ. Ниже представлены скриншоты приложения при различных сценариях использования.

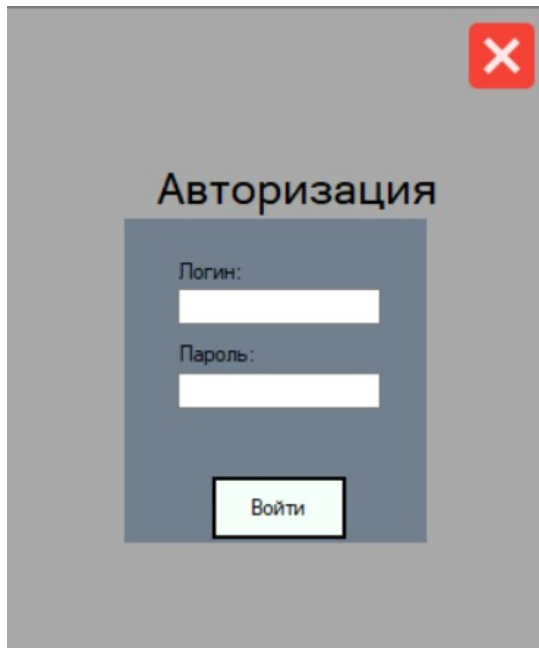
The image shows a window titled "Авторизация" (Authorization) with a red close button in the top right corner. Inside the window, there is a blue rectangular area containing two input fields: "Логин:" (Login) and "Пароль:" (Password). Below these fields is a button labeled "Войти" (Login).

Рисунок 28. Окно авторизации

Общее для групп пользователей окно: в соответствующие поля пользователь вводит свои логин и пароль, и по нажатию кнопки «Войти» открывается окно, соответствующей группы пользователей. Также на этом окне присутствует кнопка закрытия приложения.

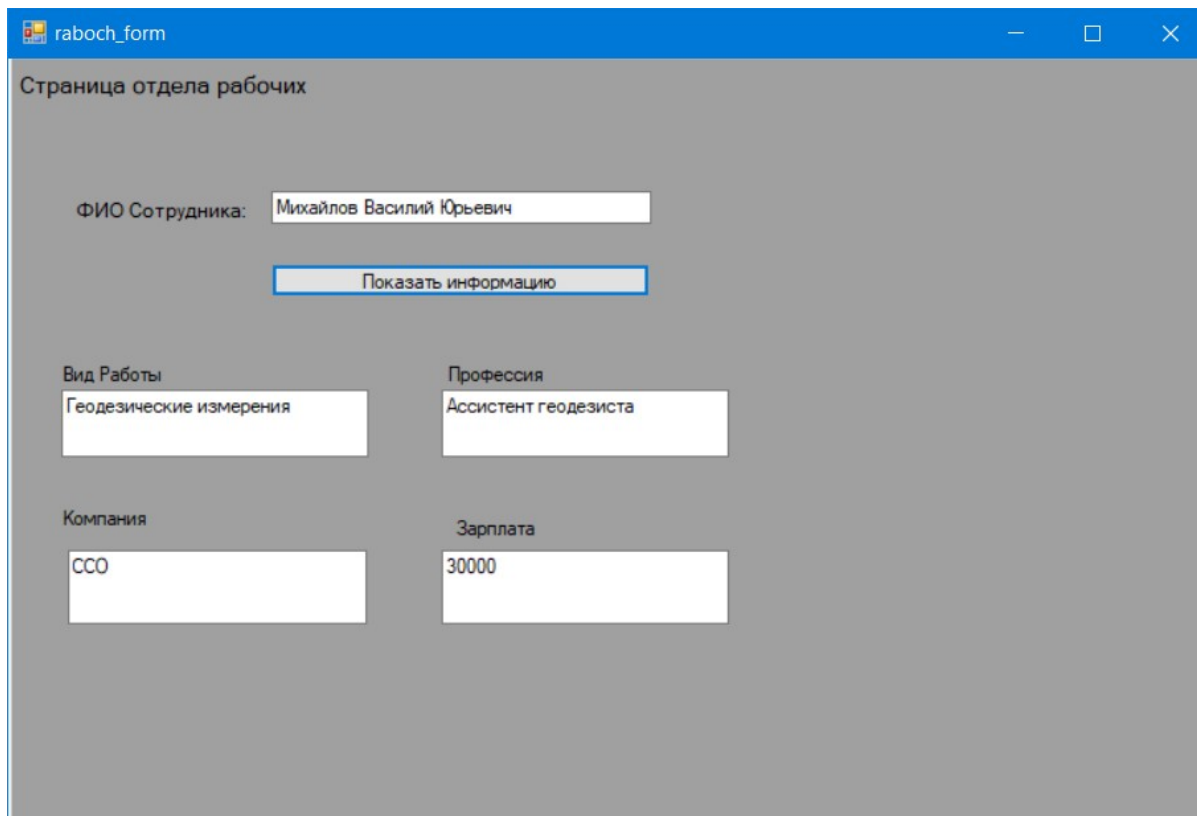
The image shows a window titled "raboch_form" with a blue header bar. The main content area is titled "Страница отдела рабочих" (Workers department page). It contains several input fields and a button. The fields are: "ФИО Сотрудника:" (Employee Name) with the value "Михайлов Василий Юрьевич", "Вид Работы" (Type of Work) with the value "Геодезические измерения", "Профессия" (Profession) with the value "Ассистент геодезиста", "Компания" (Company) with the value "ССО", and "Зарплата" (Salary) with the value "30000". There is a button labeled "Показать информацию" (Show information) below the first field.

Рисунок 29. Окно группы пользователей «Рабочие»

В данном окне сотрудник может ввести ФИО рабочего и при нажатии кнопки «Показать информацию» получить информацию о нём: текущую работу, профессию, компанию и зарплату.

tech_otd_form

Страница технического отдела

Новая запись в KPP

id поставщика
12

Дата
2022-12-10

id материала
10

Количество материала
15

id инструмента
11

Количество инструмента
3

Добавить KPP

Добавить новый материал на склад

Материал
щебень

Цена за тонну
20000

Добавить материал

Добавить новый инструмент на склад

Инструмент
сварочный аппарат

Цена за шт.
20500

Добавить инструмент

Материалы/инструменты на складе

Название	Кол-во
Бетон	16
Изоляционные материалы	1
Кабели и комплектующие	5
кирпич	9
Краски малярные	141
Металлочерепица	6
Профнастил	5
щебень	0
Кисти малярные	120
Кисти малярные 2	50
Лазерная рулетка	22
Лестница	16
Мультиметр	20
Набор инструментов	50
сварочный аппарат	0
Тахеометр	11
Уровень	20
Шуруповерт	40

Обновить информацию

Рисунок 30. Окно для группы пользователей «Тех. отдел»

Тех. отдел в основном работает с информации со склада, и фиксирует все поставки материалов и инструментов.

В правой части экрана можно добавить запись в таблицу kpp заполнив необходимые данные, в центральной части окна пользователь имеет возможность добавить новый материал или инструмент в список доступных на складе, количество добавленного инструмента/материала будет равно 0 до первой поставки. В левой части окна пользователю доступен список всех материалов и инструментов на складе и их количество.

prorab_form

Страница отдела прорабов

Задействованные сотрудники

Получить данные о задействованных сотрудниках

Работа	Профессия	ФИО сотрудника	Начало работ	Конец работ
Электромонтажные работы	разнорабочий	Телегин Данила Евгеньевич	15.11.2022 0:00:00	02.02.2023 0:00:00
Электромонтажные работы	слесарь	Потёмкин Степан Васильевич	15.11.2022 0:00:00	02.02.2023 0:00:00
Электромонтажные работы	Электрик	Муравьев Евгений Алексеевич	15.11.2022 0:00:00	02.02.2023 0:00:00
Электромонтажные работы	Электрик	Романов Георгий Павлович	15.11.2022 0:00:00	02.02.2023 0:00:00

Список свободных сотрудников

Показать данные

разнорабочий	Мамедов Порфирий Денисович
Ассистент геодезиста	Михайлов Василий Юрьевич
Геодезист	Иванов Пётр Михайлович
маляр	Козлов Иван Данилович

Начало работ

2022-12-03

Конец работ

2022-12-19

Добавить нового сотрудника

ФИО сотрудника: Потёмкин Степан Васильевич

Профессия: слесарь

Зарплата: 63000

Компания: бригада 1

id работы: 3

Добавить

Назначить сотрудника на работу

ФИО сотрудника: Мамедов Порфирий Денисович

Название работ: Отделочные работы

Назначить

Запрос материалов и инструментов со склада

Рисунок 31. Окно группы пользователей «Прораб»

В данном пользователь может добавить нового сотрудника и сразу же назначить его на конкретную работу, или же назначить уже существующего сотрудника на какую-либо работу и сразу же получить актуальную информацию о задействованных сейчас и свободных в определённый период времени сотрудниках. Также при нажатии кнопки «Запрос материалов и инструментов со склада» сотрудник переходит к следующему окну в котором можно запросить материалы и инструменты для определённой работы со склада компании.

Request

Запрос материала

Работа

Кровельные работы

Материал

Металлочерепица

Количество

2

Создать запрос

Запрос инструментов

Работа

Кровельные работы

Инструмент

сварочный аппарат

Количество

1

Создать запрос

Материалы на складе

Бетон	16
Изоляционные материалы	1
Кабели и комплектующие	5
кирпич	9
Краски малярные	141
Металлочерепица	4
Профнастил	5
щебень	15

Инструменты на складе

Кисти малярные	120
Кисти малярные 2	50
Лазерная рулетка	22
Лестница	16
Мультиметр	20
Набор инструментов	50
сварочный аппарат	2
Тахеометр	11
Уровень	20
Шуруповерт	40

Получить данные со склада

Используемые материалы

Отделочные работы	Металлочерепица	9
Отделочные работы	Бетон	4
Отделочные работы	Бетон	4
Кровельные работы	Профнастил	3
Отделочные работы	Изоляционные материалы	1
Отделочные работы	кирпич	3
Отделочные работы	кирпич	3
Кровельные работы	Металлочерепица	2

Используемые инструменты

Электромонтажные работы	Мультиметр	3
Отделочные работы	Уровень	9
малярные работы	Кисти малярные	50
малярные работы	Кисти малярные	50
Отделочные работы	Лестница	4
Кровельные работы	сварочный аппарат	1

Получить данные с рабочих мест

Рисунок 32. Окно запроса материалов и инструментов со склада

В данном окне пользователь может получить информацию о том, что в данный момент хранится на складе, и что используется в работах, и имеет возможность запросить часть материала или инструмента со склада на работу.

director_form

Страница отдела совета директоров

Добавить новую работу

Название
Реставрация фундамента

Начало работ
2022-12-05

Конец работ
2022-12-26

Добавить

Открыть страницу Прорабов

Открыть страницу Тех. отдела

Запрос со склада

Общая цена инструментов на складе
28217000

Общая цена материалов на складе
1354500

Рассчитать

Добавление пользователя

Тип пользователя прораб

Логин prorab123

Пароль 98988787

Добавить

Удаление пользователя

Имя удаляемого пользователя
prorab123

Удалить

Рисунок 33. Окно группы пользователей «Директор»

Пользователи этой группы имеют возможность добавить новую работу, которую будет осуществлять компания, получить информацию о стоимости содержимого складов и также добавлять и удалять пользователей из системы.

Также так как, согласно ТЗ, совет директоров имеет полный доступ ко всему функционалу БД, на их странице есть возможность открыть окно любой группы пользователей и совершать любые операции доступные им.

Приложение. Листинг программы на языке программирования C#.

Program.cs

```
using System;
using System.Windows.Forms;
namespace TR
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Authorization());
        }
    }
}
```

Authorization.cs

```
using System;
using System.Windows.Forms;
using Npgsql;

namespace TR
{
    public partial class Authorization : Form
    {
        public Authorization()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void button2_Click(object sender, EventArgs e) // кнопка выхода
        {
            this.Close();
        }

        private void button1_Click(object sender, EventArgs e) // кнопка "Войти"
        {
            string login_ = textBox1.Text;
            string pass_ = textBox2.Text;
            string role = "";

            NpgsqlConnection bdConnection = new
            NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres");

            NpgsqlCommand comand = new NpgsqlCommand("select get_typename(@log, @pass);",
            bdConnection);
            comand.Parameters.Add("@log", NpgsqlTypes.NpgsqlDbType.Varchar).Value = login_;
            comand.Parameters.Add("@pass", NpgsqlTypes.NpgsqlDbType.Varchar).Value = pass_;
            bdConnection.Open();
            var reader = comand.ExecuteReader();
            reader.Read();
            role = reader.GetValue(0).ToString();
            bdConnection.Close();
        }
    }
}
```

```

switch (role)
{
    case "директор":
    {
        director_form df1 = new director_form();
        df1.Show();
        break;
    }
    case "прораб":
    {
        prorab_form pf1 = new prorab_form();
        pf1.Show();
        break;
    }
    case "тех. отдел":
    {
        tech_otd_form tof1 = new tech_otd_form();
        tof1.Show();
        break;
    }
    case "рабочий":
    {
        raboch_form rf1 = new raboch_form();
        rf1.Show();
        break;
    }
    default:
    {
        MessageBox.Show("Неверный логин или пароль! \nПопробуйте ещё раз");
        textBox1.Text = "";
        textBox2.Text = "";
        break;
    }
};
}

private void panel1_Paint(object sender, PaintEventArgs e)
{
}

private void label2_Click(object sender, EventArgs e)
{
}
}
}

```

raboch_form.cs

```

using System;
using System.Windows.Forms;
using Npgsql;

namespace TR
{
    public partial class raboch_form : Form
    {
        public raboch_form()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)

```

```

{
    string fio = textBox2.Text;
    string[] info;
    NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres");

    NpgsqlCommand command = new NpgsqlCommand("select * from
select_workers_data(@wname);", bdConnection);
    command.Parameters.Add("@wname", NpgsqlTypes.NpgsqlDbType.Varchar).Value = fio;
    bdConnection.Open();

    using (var reader = command.ExecuteReader())
    {
        reader.Read();
        int FieldsCount = reader.FieldCount;
        info = new string[FieldsCount];
        for (int i = 0; i < FieldsCount; i++)
        {
            info[i] += reader.GetValue(i).ToString();
        }
        bdConnection.Close();
        textBox1.Text = info[0];
        textBox3.Text = info[2];
        textBox4.Text = info[3];
        textBox5.Text = info[4];
    }
}
}
}
}

```

tech_otd_form.cs

```

using System;
using System.Windows.Forms;
using Npgsql;

namespace TR
{
    public partial class tech_otd_form : Form
    {
        public tech_otd_form()
        {
            InitializeComponent();

            private void button1_Click(object sender, EventArgs e)
            {
                textBox1.Clear();
                textBox2.Clear();

                string[] info = new string[2];
                using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
                {
                    // получаем данные из таблицы с материалами
                    using (NpgsqlCommand command = new NpgsqlCommand("select * from
get_stor_mat_info();", bdConnection))
                    {
                        bdConnection.Open();
                        using (var reader = command.ExecuteReader())
                        {
                            while (reader.Read())
                            {
                                int FieldsCount = reader.FieldCount;
                                for (int i = 0; i < FieldsCount; i++)

```

```

        {
            info[i] = reader.GetValue(i).ToString();
        }
        textBox1.Text += info[0] + Environment.NewLine;
        textBox2.Text += info[1] + Environment.NewLine;
    }
}
// получаем данные из таблицы с инструментами
using (NpgsqlCommand command = new NpgsqlCommand("select * from
get_stor_inst_info()", bdConnection))
{
    using (var reader = command.ExecuteReader())
    {
        while (reader.Read())
        {
            int FieldsCount = reader.FieldCount;
            for (int i = 0; i < FieldsCount; i++)
            {
                info[i] = reader.GetValue(i).ToString();
            }
            textBox1.Text += info[0] + Environment.NewLine;
            textBox2.Text += info[1] + Environment.NewLine;
        }
    }
}
}

private void button2_Click(object sender, EventArgs e)
{
    string[] info = new string[6];
    info[0] = textBox3.Text;
    info[1] = textBox4.Text;
    info[2] = textBox5.Text;
    info[3] = textBox6.Text;
    info[4] = textBox7.Text;
    info[5] = textBox8.Text;

    using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
    {
        NpgsqlCommand command = new NpgsqlCommand("call new_kpp(@w, @d, @mi, @km,
@ii, @ki)", bdConnection);
        command.Parameters.Add("@w", NpgsqlTypes.NpgsqlDbType.Integer).Value =
int.Parse(info[0]);
        command.Parameters.Add("@d", NpgsqlTypes.NpgsqlDbType.Date).Value =
DateTime.Parse(info[1]);
        command.Parameters.Add("@mi", NpgsqlTypes.NpgsqlDbType.Integer).Value =
int.Parse(info[2]);
        command.Parameters.Add("@km", NpgsqlTypes.NpgsqlDbType.Integer).Value =
int.Parse(info[3]);
        command.Parameters.Add("@ii", NpgsqlTypes.NpgsqlDbType.Integer).Value =
int.Parse(info[4]);
        command.Parameters.Add("@ki", NpgsqlTypes.NpgsqlDbType.Integer).Value =
int.Parse(info[5]);
        bdConnection.Open();
        command.ExecuteNonQuery();
        bdConnection.Close();
    }
}

private void button3_Click(object sender, EventArgs e)
{
    string[] info = new string[2];

```

```

        info[0] = textBox9.Text;
        info[1] = textBox10.Text;

        using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
        {
            bdConnection.Open();
            NpgsqlCommand command = new NpgsqlCommand("call add_new_to_storage_mat(@mn,
@mp);", bdConnection);
            command.Parameters.Add("@mn", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
info[0];
            command.Parameters.Add("@mp", NpgsqlTypes.NpgsqlDbType.Integer).Value =
Convert.ToInt64(info[1]);
            command.ExecuteNonQuery();
            bdConnection.Close();
        }
    }

    private void button4_Click(object sender, EventArgs e)
    {
        string[] info = new string[2];
        info[0] = textBox11.Text;
        info[1] = textBox12.Text;

        using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
        {
            bdConnection.Open();
            NpgsqlCommand command = new NpgsqlCommand("call add_new_to_storage_inst(@in,
@ip);", bdConnection);
            command.Parameters.Add("@in", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
info[0];
            command.Parameters.Add("@ip", NpgsqlTypes.NpgsqlDbType.Integer).Value =
Convert.ToInt64(info[1]);
            command.ExecuteNonQuery();
            bdConnection.Close();
        }
    }
}

```

prorab_form.cs

```

using System;
using System.Windows.Forms;
using Npgsql;

namespace TR
{
    public partial class prorab_form : Form
    {
        public prorab_form()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            textBox5.Clear();
        }
    }
}

```

```

        string[] info = new string[5];
        using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
        {
            // получаем данные из таблицы с материалами
            using (NpgsqlCommand command = new NpgsqlCommand("select * from
get_all_works()", bdConnection))
            {
                bdConnection.Open();
                using (var reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        int FieldsCount = reader.FieldCount;
                        for (int i = 0; i < FieldsCount; i++)
                        {
                            info[i] = reader.GetValue(i).ToString();
                        }
                        textBox1.Text += info[0] + Environment.NewLine;
                        textBox2.Text += info[1] + Environment.NewLine;
                        textBox3.Text += info[2] + Environment.NewLine;
                        textBox4.Text += info[3] + Environment.NewLine;
                        textBox5.Text += info[4] + Environment.NewLine;
                    }
                }
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            textBox6.Clear();
            textBox9.Clear();
            string[] info = new string[2];
            info[0] = textBox7.Text;
            info[1] = textBox8.Text;

            using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
            {
                using (NpgsqlCommand command = new NpgsqlCommand("select * from
all_free_workers(@bd, @ed)", bdConnection))
                {
                    command.Parameters.Add("@bd", NpgsqlTypes.NpgsqlDbType.Date).Value =
Convert.ToDateTime(info[0]);
                    command.Parameters.Add("@ed", NpgsqlTypes.NpgsqlDbType.Date).Value =
Convert.ToDateTime(info[1]);
                    bdConnection.Open();
                    using (var reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            int FieldsCount = reader.FieldCount;
                            for (int i = 0; i < FieldsCount; i++)
                            {
                                info[i] = reader.GetValue(i).ToString();
                            }
                            textBox6.Text += info[0] + Environment.NewLine;
                            textBox9.Text += info[1] + Environment.NewLine;
                        }
                    }
                }
            }
        }
    }
}

```

```

private void button3_Click(object sender, EventArgs e)
{
    string[] info = new string[5];
    info[0] = textBox10.Text;
    info[1] = textBox11.Text;
    info[2] = textBox12.Text;
    info[3] = textBox13.Text;
    info[4] = textBox14.Text;

    using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
    {
        NpgsqlCommand command = new NpgsqlCommand("call add_new_worker(@f, @p, @s,
@c, @w);", bdConnection);
        command.Parameters.Add("@f", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
info[0]; // fio
        command.Parameters.Add("@p", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
info[1]; // profession
        command.Parameters.Add("@s", NpgsqlTypes.NpgsqlDbType.Integer).Value =
int.Parse(info[2]); // salary
        command.Parameters.Add("@c", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
info[3]; // company
        command.Parameters.Add("@w", NpgsqlTypes.NpgsqlDbType.Integer).Value =
int.Parse(info[4]); // work_id
        bdConnection.Open();
        command.ExecuteNonQuery();
        bdConnection.Close();
    }
}

private void button4_Click(object sender, EventArgs e)
{
    string[] info = new string[2];
    info[0] = textBox15.Text;
    info[1] = textBox16.Text;

    using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
    {
        NpgsqlCommand command = new NpgsqlCommand("call give_work_to_worker(@f,
@w);", bdConnection);
        command.Parameters.Add("@f", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
info[0]; // fio
        command.Parameters.Add("@w", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
info[1]; // work
        bdConnection.Open();
        command.ExecuteNonQuery();
        bdConnection.Close();
    }
}

private void button5_Click(object sender, EventArgs e)
{
    Request req = new Request();
    req.Show();
}
}

```

director_form.cs

```

using System;
using System.Windows.Forms;
using Npgsql;

```



```

namespace TR
{
    public partial class director_form : Form
    {
        public director_form()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string[] info = new string[3];
            info[0] = textBox2.Text;
            info[1] = textBox3.Text;
            info[2] = textBox4.Text;

            using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
            {
                bdConnection.Open();
                NpgsqlCommand command = new NpgsqlCommand("call add_to_building(@wn, @bd,
@ed);", bdConnection);
                command.Parameters.Add("@wn", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
info[0];
                command.Parameters.Add("@bd", NpgsqlTypes.NpgsqlDbType.Date).Value =
Convert.ToDateTime(info[1]);
                command.Parameters.Add("@ed", NpgsqlTypes.NpgsqlDbType.Date).Value =
Convert.ToDateTime(info[2]);
                command.ExecuteNonQuery();
                bdConnection.Close();
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
            {
                bdConnection.Open();
                NpgsqlCommand command = new NpgsqlCommand("select all_mat_price();",
bdConnection);
                var reader = command.ExecuteScalar();
                textBox1.Text = reader.ToString();

                command = new NpgsqlCommand("select all_inst_price();", bdConnection);
                reader = command.ExecuteScalar();
                textBox5.Text = reader.ToString();

                bdConnection.Close();
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            prorab_form pf1 = new prorab_form();
            pf1.Show();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            tech_otd_form tof1 = new tech_otd_form();
            tof1.Show();
        }
    }
}

```

```

private void button5_Click(object sender, EventArgs e)
{
    Request req = new Request();
    req.Show();
}

private void button6_Click(object sender, EventArgs e)
{
    string type_ = textBox6.Text;
    string log_ = textBox7.Text;
    string pass_ = textBox8.Text;

    using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
    {
        NpgsqlCommand comand = new NpgsqlCommand("call add_new_user(@tn, @l, @p);",
bdConnection);
        comand.Parameters.Add("@tn", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
type_;
        comand.Parameters.Add("@l", NpgsqlTypes.NpgsqlDbType.Varchar).Value = log_;
        comand.Parameters.Add("@p", NpgsqlTypes.NpgsqlDbType.Varchar).Value = pass_;
        bdConnection.Open();
        comand.ExecuteNonQuery();
        bdConnection.Close();
        MessageBox.Show("Пользователь успешно добавлен");
        textBox6.Text = "";
        textBox7.Text = "";
        textBox8.Text = "";
    }
}

private void button7_Click(object sender, EventArgs e)
{
    string log_ = textBox9.Text;

    using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
    {
        NpgsqlCommand comand = new NpgsqlCommand("call delete_user(@l);",
bdConnection);
        comand.Parameters.Add("@l", NpgsqlTypes.NpgsqlDbType.Varchar).Value = log_;
        bdConnection.Open();
        comand.ExecuteNonQuery();
        bdConnection.Close();
        MessageBox.Show("Пользователь успешно удалён");
        textBox9.Text = "";
    }
}
}

```

Request.cs

```

using System;
using System.Windows.Forms;
using Npgsql;

namespace TR
{
    public partial class Request : Form
    {
        public Request()
    }
}

```

```

{
    InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
    textBox1.Clear();
    textBox2.Clear();
    textBox3.Clear();
    textBox4.Clear();

    string[] info = new string[2];
    using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
    {
        // получаем данные из таблицы с материалами
        using (NpgsqlCommand command = new NpgsqlCommand("select * from
get_stor_mat_info()", bdConnection))
        {
            bdConnection.Open();
            using (var reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    int FieldsCount = reader.FieldCount;
                    for (int i = 0; i < FieldsCount; i++)
                    {
                        info[i] = reader.GetValue(i).ToString();
                    }
                    textBox1.Text += info[0] + Environment.NewLine;
                    textBox3.Text += info[1] + Environment.NewLine;
                }
            }
            // получаем данные из таблицы с инструментами
            using (NpgsqlCommand command = new NpgsqlCommand("select * from
get_stor_inst_info()", bdConnection))
            {
                using (var reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        int FieldsCount = reader.FieldCount;
                        for (int i = 0; i < FieldsCount; i++)
                        {
                            info[i] = reader.GetValue(i).ToString();
                        }
                        textBox2.Text += info[0] + Environment.NewLine;
                        textBox4.Text += info[1] + Environment.NewLine;
                    }
                }
            }
        }
    }

    private void button4_Click(object sender, EventArgs e)
    {
        textBox11.Clear();
        textBox12.Clear();
        textBox13.Clear();
        textBox14.Clear();
        textBox15.Clear();
        textBox16.Clear();

        string[] info = new string[3];

```

```

        using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
        {
            // получаем данные из таблицы с материалами
            using (NpgsqlCommand command = new NpgsqlCommand("select * from
get_all_miu()", bdConnection))
            {
                bdConnection.Open();
                using (var reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        int FieldsCount = reader.FieldCount;
                        for (int i = 0; i < FieldsCount; i++)
                        {
                            info[i] = reader.GetValue(i).ToString();
                        }
                        textBox11.Text += info[0] + Environment.NewLine;
                        textBox12.Text += info[1] + Environment.NewLine;
                        textBox15.Text += info[2] + Environment.NewLine;
                    }
                }
            }
            // получаем данные из таблицы с инструментами
            using (NpgsqlCommand command = new NpgsqlCommand("select * from
get_all_iiu()", bdConnection))
            {
                using (var reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        int FieldsCount = reader.FieldCount;
                        for (int i = 0; i < FieldsCount; i++)
                        {
                            info[i] = reader.GetValue(i).ToString();
                        }
                        textBox13.Text += info[0] + Environment.NewLine;
                        textBox14.Text += info[1] + Environment.NewLine;
                        textBox16.Text += info[2] + Environment.NewLine;
                    }
                }
            }
        }
    }

    private void button3_Click(object sender, EventArgs e)
    {
        string work_ = textBox8.Text;
        string inst_ = textBox9.Text;
        string kol_ = textBox10.Text;

        using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
        {
            NpgsqlCommand comand = new NpgsqlCommand("call new_iiu(@n, @ik, @wi);",
bdConnection);
            comand.Parameters.Add("@n", NpgsqlTypes.NpgsqlDbType.Varchar).Value = inst_;
            comand.Parameters.Add("@ik", NpgsqlTypes.NpgsqlDbType.Integer).Value =
int.Parse(kol_);
            comand.Parameters.Add("@wi", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
work_;

            bdConnection.Open();
            comand.ExecuteNonQuery();
            bdConnection.Close();
        }
    }
}

```

```

    }

    private void button2_Click(object sender, EventArgs e)
    {
        string work_ = textBox5.Text;
        string inst_ = textBox6.Text;
        string kol_ = textBox7.Text;

        using (NpgsqlConnection bdConnection = new
NpgsqlConnection("Host=localhost;Username=postgres;Password=db432goGA;Database=postgres"))
        {
            NpgsqlCommand comand = new NpgsqlCommand("call new_miu(@mn, @mk, @wi);",
bdConnection);
            comand.Parameters.Add("@mn", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
inst_;
            comand.Parameters.Add("@mk", NpgsqlTypes.NpgsqlDbType.Integer).Value =
int.Parse(kol_);
            comand.Parameters.Add("@wi", NpgsqlTypes.NpgsqlDbType.Varchar).Value =
work_;

            bdConnection.Open();
            comand.ExecuteNonQuery();
            bdConnection.Close();
        }
    }
}

```