

# Курсовая работа на тему: “База данных стоматологической клиники”

Студент: Ионов Т. Р. ИУ9-61Б

Научный руководитель: Вишняков И. Э.

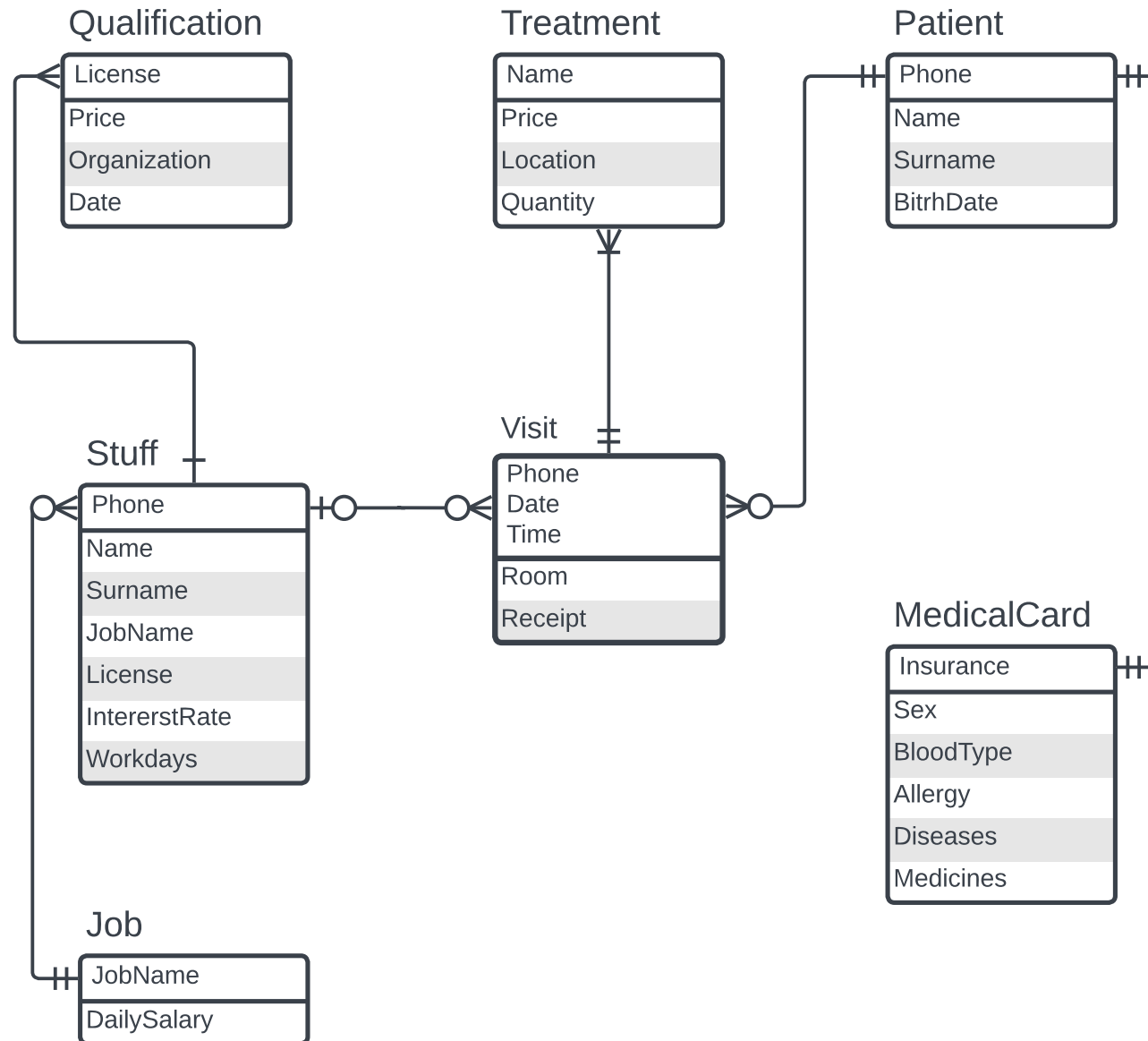
# Цель курсовой работы

- Разработать базу данных стоматологической клиники и приложение, позволяющее вносить посещения врача и рассчитывать заработную плату сотрудников.

# Актуальность

- Автоматизация бизнес-процессов;
- Прозрачность бизнес-процессов малого бизнеса.

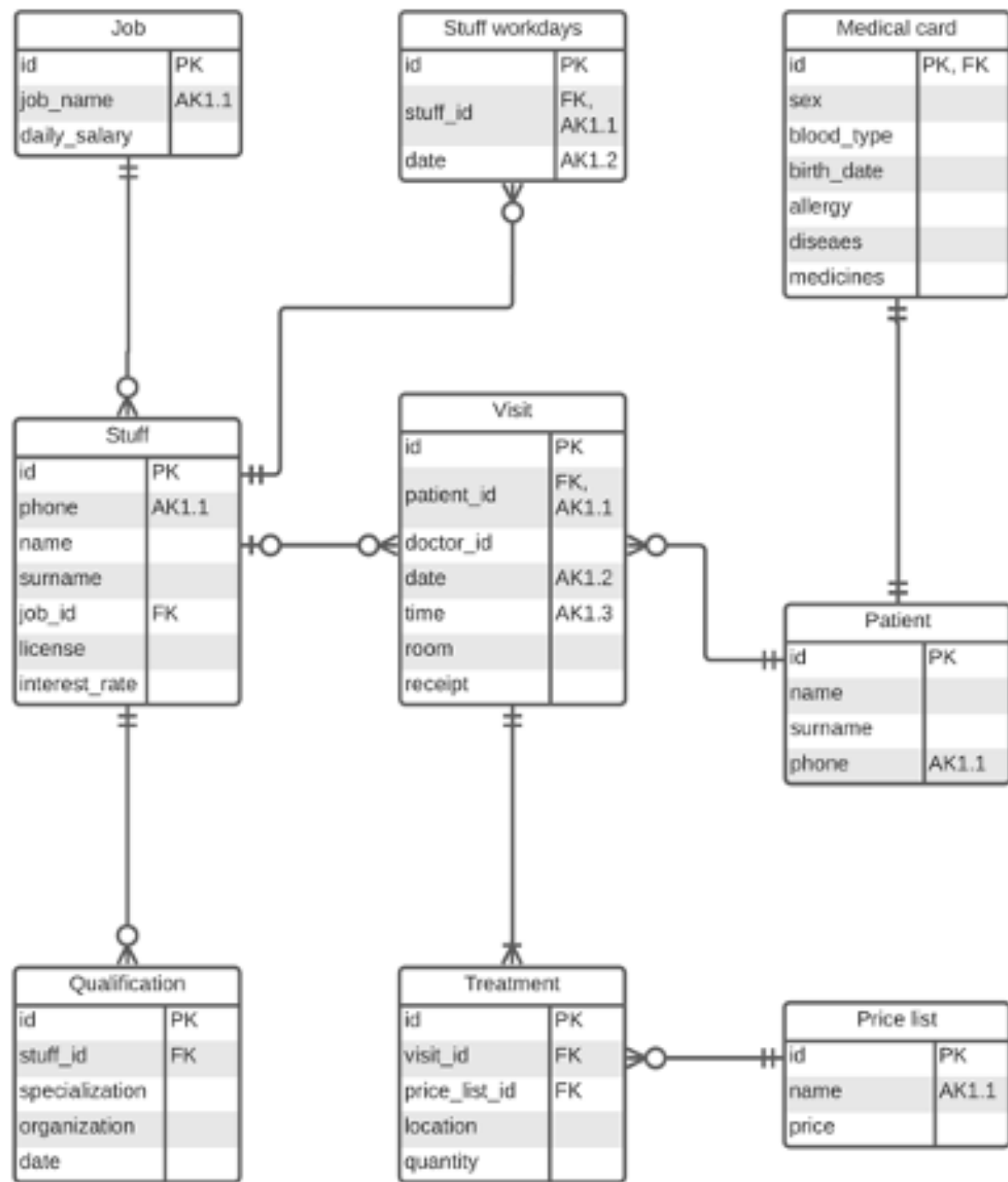
# Модель “сущность-связь”



# Преобразование в реляционную модель

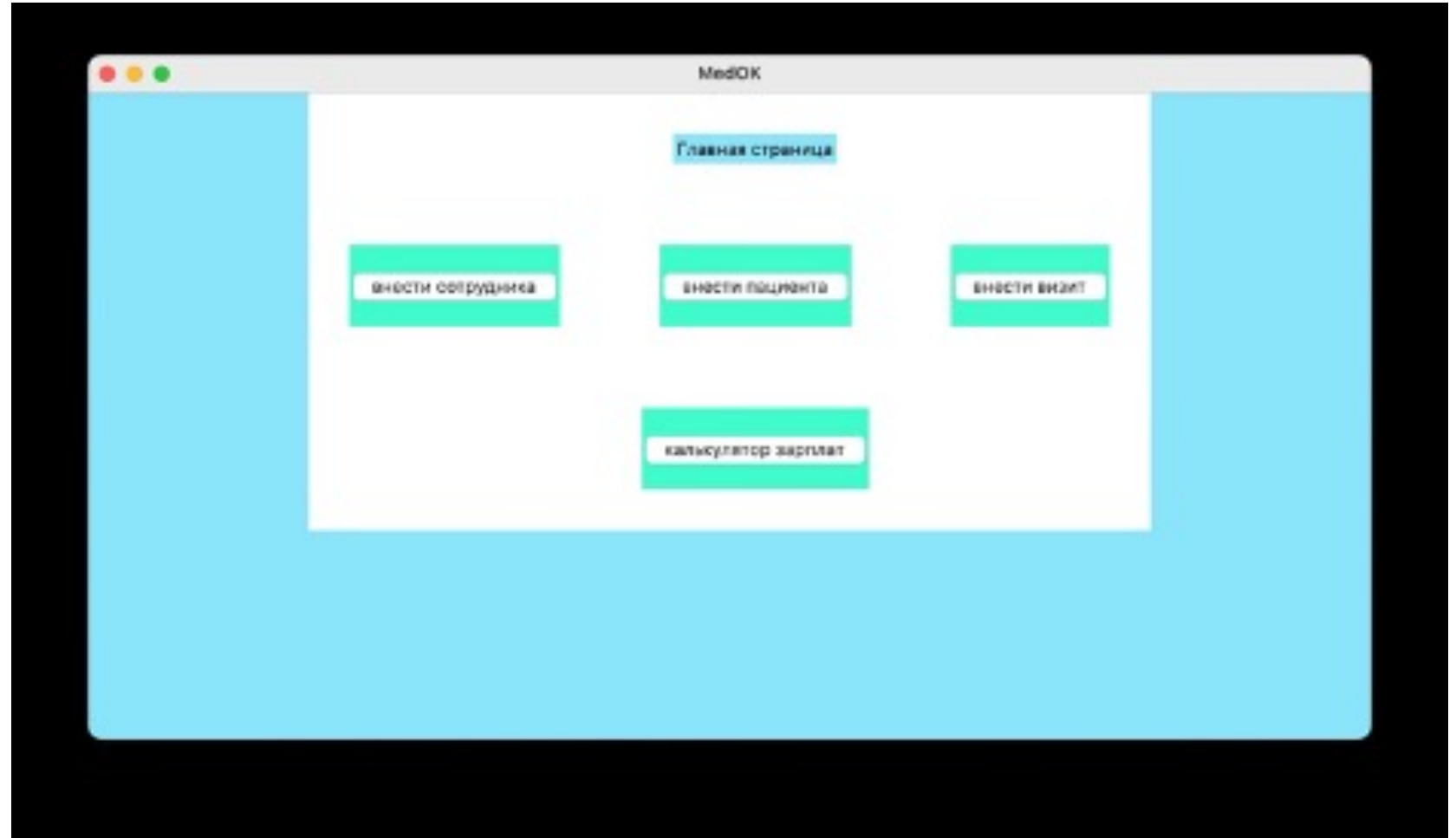
- 1. Создание таблицы для каждой сущности:
  - 1.1. определение первичного ключа (возможно, суррогатного);
  - 1.2. определение ключей кандидатов;
  - 1.3. определение свойств каждого столбца:
    - 1.3.1. тип данных;
    - 1.3.2. возможность неопределенного значения;
    - 1.3.3. значение по умолчанию;
    - 1.3.4. ограничений на значения.
  - 1.4. проверка нормализации.
- 2. Создание связей с помощью внешних ключей:
  - 2.1. Между сильными сущностями (1:1, 1:N, N:M);
  - 2.2. Для идентификационно-зависимых сущностей;
  - 2.3. Для слабых сущностей;
  - 2.4. Для сущностей тип-подтип.
- 3. Обеспечение условий минимальной кардинальности.

# Реляционная модель

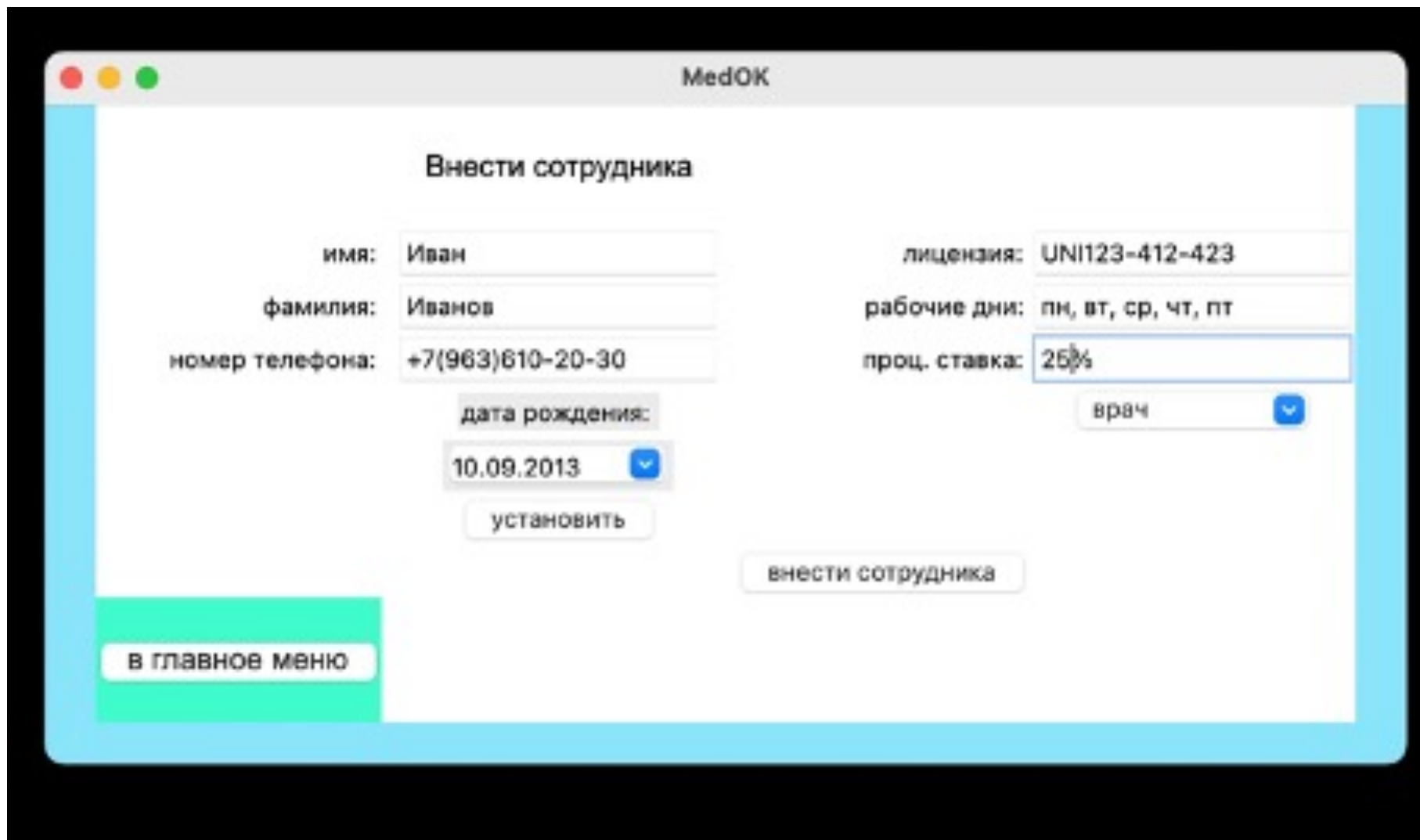


# Приложение

- Главное меню



# Внесение сотрудника



The screenshot shows a web application window titled "MedOK". Inside, there is a form titled "Внести сотрудника". The form contains several input fields and buttons. The fields are arranged in two columns. The left column contains fields for "имя:" (Ivan), "фамилия:" (Ivanov), "номер телефона:" (+7(963)610-20-30), and "дата рождения:" (10.09.2013). The right column contains fields for "лицензия:" (UNI123-412-423), "рабочие дни:" (пн, вт, ср, чт, пт), and "проц. ставка:" (25%). There is also a dropdown menu for "врач" with a blue arrow icon. At the bottom left, there is a green button labeled "в главное меню". At the bottom right, there is a button labeled "внести сотрудника".

MedOK

Внести сотрудника

имя: Иван

фамилия: Иванов

номер телефона: +7(963)610-20-30

лицензия: UNI123-412-423

рабочие дни: пн, вт, ср, чт, пт

проц. ставка: 25%

дата рождения: 10.09.2013

врач

установить

внести сотрудника

в главное меню



# Внесение пациента

MedOK


## Внести пациента

имя:	Валерий	аллергия:	арахис
фамилия:	Шорохов	хронические заболевания:	
номер телефона:	7(966)742-65-36	принимаемые препараты:	
выбрано:	1966-09-19	M	▼
	19.09.1966	AB+	▼

установить

внести пациента

в главное меню



успешно внесен пациент

OK

# Страница расчета зарплат

врач Арина Жук

администратор Саша Доля  
администратор Маша Флоря  
медсестра Оля Орлова  
медсестра Ксения Фролова  
врач Иван Сергев  
врач Арина Жук

Расчет зарплаты сотрудника

врач Иван Иванов

сотрудник: врач Иван Иванов  
смена: 3000.0, 25.0%

начальная дата: 2022-09-19

19.09.2022

установить

конечная дата: 2022-09-25

25.09.2022

установить

расчет

итого за смены: 15000

в главное меню

# Страница внесения визита

01.06.2022

Июня 2022

	пн	вт	ср	чт	пт	сб	вс
22	30	31	1	2	3	4	5
23	6	7	8	9	10	11	12
24	13	14	15	16	17	18	19
25	20	21	22	23	24	25	26
26	27	28	29	30	1	2	3
27	4	5	6	7	8	9	10

MedOK

### Внести визит

врач: Иван Иванов

пациент: Валерий Шорохов

кабинет: 1

лечение кармеса: ☒

количество:

код зуба:

добавить

добавлено!  
процедура: лечение кармеса  
количество: 3  
код зуба: 1

список процедур:  
анестезия x 1  
удаление зуба x 1  
лечение кармеса x 3

рецепт:

время начала приема:

выбрано: 2022-09-20

установить

ВНЕСТИ ВИЗИТ

в главное меню

# Запросы к базе данных

```
def insert(entity: Entity, conn=get_connection()):
    table_name = entity.__class__.__name__.lower()
    query = f'insert into {table_name}'
    d = entity.get_data()
    fields = d.keys()
    values = list(d.values())
    query += ' (' + ','.join(fields) + ')'
    query += ' values (' + ','.join(['%s'] * len(values)) + ');'
    try:
        with conn.cursor() as cur:
            cur.execute(query, values)
            conn.commit()
        return 0
    except Exception as ex:
        conn.rollback()
        print(f"Exeption in insert: {ex} for table {table_name}+
        f\"with entity {entity}\"")
        return ex
```

# Реализация триггера

```
create or replace function check_stuff()  
    returns trigger  
as $check_stuff$  
begin  
    if not exists(select id, job_id from stuff where id=new.doctor_id and job_id=3) then  
        raise exception 'Only doctor can hold a visit';  
    end if;  
    if new.date not in(  
        select date from stuff_workdays s where s.stuff_id=new.doctor_id  
    ) then  
        raise exception 'Doctor cannot hold a visit on a non-working day';  
    end if;  
    return new;  
end;  
$check_stuff$ language plpgsql;
```

create trigger check\_stuff before insert or update on visit  
for each row execute function check\_stuff();

# Тестирование

```
insert into job (id, daily_salary, job_name) values
    (3, 3000, 'врач');

insert into stuff (id, name, surname, job_id, license, phone,
interest_rate) values
    (6, 'Арина', 'Жук', 3, 'DOC123-4124', '89617391777',
0.25);

insert into stuff_workdays (stuff_id, date)
select 6, * from generate_series('2022-06-02'::date,
'2022-06-23'::date, '2 day'::interval);

insert into price_list values
    (1, 'анестезия', 1000),
    (2, 'удаление зуба', 2500),
    (3, 'лечение кариеса', 3000),
    (4, 'установка коронки', 5000);
commit;

insert into visit (patient_id, doctor_id, date) values
    (2, 6, '2022-06-02');

insert into treatment (visit_id, code, quantity) values
    (2, 1, 2),
    (2, 3, 2);
commit;
```

# Пример работы калькулятора

## Расчет зарплаты сотрудника

врач Арина Жук



сотрудник: врач Арина Жук  
смена: 3000.0, 25.0%

начальная дата: 2022-06-01

01.06.2022



установить

конечная дата: 2022-06-27

27.06.2022



установить

расчет

итого за смены: 33000  
итого за визиты: 2000  
итого: 35000

в главное меню

# Демонстрация изменений расчета

сотрудник: врач Иван Иванов  
смена: 3000.0, 25.0%

начальная дата: 2022-09-19  
19.09.2022 

конечная дата: 2022-09-25  
25.09.2022 

итого за смену: 15000  
итого за визиты: 3125  
итого: 18125



# Заключение

- Изучена предметная область;
- Разработаны модель “сущность-связь” и реляционная модель;
- Реализовано кросс-платформенное оконное приложение;
- Протестированы сценарии взаимодействия пользователя и базы данных.

# Дальнейшая разработка

- Авторизация, разделение возможностей приложений
- Внесение новых процедур