МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского»**

**Институт информационных технологий, математики и механики**

**Кафедра программной инженерии**

**ОТЧЕТ**

по лабораторной работе

на тему:

**«Платная поликлиника»**

**Выполнил:** студент группы 381708-2

Сазанов Д. Е.

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Проверил:** доцент каф. Программной инженерии ИТММ Шапошников Д.Е.

\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2019.

**Содержание**

[Формулировка задачи 3](#_Toc533091120)

[Анализ задания 4](#_Toc533091121)

[Программная реализация 5](#_Toc533091122)

[*Таблицы* 5](#_Toc533091123)

[*Триггер*](#_Toc533091123) 10

[*Функции*](#_Toc533091123) 11

[Результаты моелирования 12](#_Toc533091126)

[*Эксперимент №1* 12](#_Toc533091123)

[*Эксперимент №2* 13](#_Toc533091123)

[Выводы 14](#_Toc533091120)

[Список источников 15](#_Toc533091121)

Приложение............................................................................................................16

# Формулировка задания

1. Создать концептуальную схему по выбранной предметной области (с учетом её развития) и на её основе спроектировать структуру реляционной базы данных (для MySQL). Выделить обязательные поля, наложить условия целостности
2. Создать структуру базы данных (таблицы, представления, внешние ключи). Оформить структуру в виде скрипта на языке SQL.
3. Создать операторы языка SQL для вывода агрегатных данных (с использованием агрегатных функций и подзапросов).
4. Создать триггер INSERT для проверки правильности вводимых данных (на одно условие).
5. Создать процедуру для удаления родительской записи с соответствующими подчиненными (дочерними) записями в другой таблице.

# Анализ задания

### Описание предметной области

### Вы являетесь руководителем службы планирования платной поликлиники. Вашейзадачей является отслеживание финансовых показателей работы поликлиники.

### В поликлинике работают врачи различных специальностей, имеющие разнуюквалификацию. Каждый день в поликлинику обращаются больные. Все больныепроходят обязательную регистрацию, при которой в базу данных заносятсястандартные анкетные данные (фамилия, имя, отчество, год рождения). Каждыйбольной может обращаться в поликлинику несколько раз, нуждаясь в различноймедицинской помощи. Все обращения больных фиксируются, при этомустанавливается диагноз, определяется стоимость лечения, запоминается датаобращения.

### Развитие постановки задачи

В результате эксплуатации базы данных выяснилось, что при обращении в

поликлинику пациент обследуется и проходит лечение у разных специалистов. Общаястоимость лечения зависит от стоимости тех консультаций и процедур, которыеназначены пациенту. Кроме того, для определенных категорий гражданпредусмотрены скидки.Внести в структуру таблиц изменения, учитывающие эти факты, и изменитьсуществующие запросы. Добавить новые запросы.Программная реализация

### Таблицы:

* Врачи (Код врача, Фамилия, Имя, Отчество, Специальность, Категория).
* Была добавлена колонка – “Цена консультации” для реалезации развития постановки задачи.

|  |
| --- |
| -- Table: public.doctors |
|  |  |
|  | -- DROP TABLE public.doctors; |
|  |  |
|  | CREATE TABLE public.doctors |
|  | ( |
|  | did integer NOT NULL, |
|  | dsn character(120) NOT NULL, |
|  | dfn character(120) NOT NULL, |
|  | dtn character(120) NOT NULL, |
|  | dspec character(120) NOT NULL, |
|  | dcat integer NOT NULL, |
|  | cprice integer, |
|  | CONSTRAINT dfkey PRIMARY KEY (did), |
|  | CONSTRAINT did\_lim UNIQUE (did) |
|  | ) |
|  | WITH ( |
|  | OIDS=FALSE |
|  | ); |
|  | ALTER TABLE public.doctors |

|  |  |
| --- | --- |
|  | OWNER TO postgres; |

* Пациенты (Код пациента, Фамилия, Имя, Отчество, Год рождения).
* Была добавленна колонка – “Категория пациента” для реализации развития постановки задачи

|  |
| --- |
| -- Table: public.patients |
|  |  |
|  | -- DROP TABLE public.patients; |
|  |  |
|  | CREATE TABLE public.patients |
|  | ( |
|  | pid integer NOT NULL, |
|  | psn character(120) NOT NULL, |
|  | pfn character(120) NOT NULL, |
|  | ptn character(120) NOT NULL, |
|  | pdof date NOT NULL, |
|  | pcategory integer, |
|  | CONSTRAINT pfkey PRIMARY KEY (pid), |
|  | CONSTRAINT category FOREIGN KEY (pcategory) |
|  | REFERENCES public.sale (scategory) MATCH SIMPLE |
|  | ON UPDATE NO ACTION ON DELETE NO ACTION |
|  | ) |
|  | WITH ( |
|  | OIDS=FALSE |
|  | ); |
|  | ALTER TABLE public.patients |
|  | OWNER TO postgres; |

* Обращения (Код обращения, Код врача, Код пациента, Дата обращения, Диагноз, Стоимость лечения).

|  |
| --- |
|  |
|  |  |

|  |
| --- |
| -- Table: public.appeal |
|  |  |
|  | -- DROP TABLE public.appeal; |
|  |  |
|  | CREATE TABLE public.appeal |
|  | ( |
|  | aid integer NOT NULL, |
|  | did integer NOT NULL, |
|  | pid integer NOT NULL, |
|  | adate date NOT NULL, |
|  | diagnosis character(200) NOT NULL, |
|  | price integer NOT NULL, |
|  | CONSTRAINT aid PRIMARY KEY (aid), |
|  | CONSTRAINT appeal\_did\_fkey FOREIGN KEY (did) |
|  | REFERENCES public.doctors (did) MATCH SIMPLE |
|  | ON UPDATE NO ACTION ON DELETE NO ACTION, |
|  | CONSTRAINT appeal\_pid\_fkey FOREIGN KEY (pid) |
|  | REFERENCES public.patients (pid) MATCH SIMPLE |
|  | ON UPDATE NO ACTION ON DELETE NO ACTION, |
|  | CONSTRAINT aid\_lim UNIQUE (aid) |
|  | ) |
|  | WITH ( |
|  | OIDS=FALSE |
|  | ); |
|  | ALTER TABLE public.appeal |
|  | OWNER TO postgres; |
|  |  |
|  | -- Trigger: check\_a on public.appeal |
|  |  |
|  | -- DROP TRIGGER check\_a ON public.appeal; |
|  |  |
|  | CREATE TRIGGER check\_a |
|  | BEFORE INSERT |
|  | ON public.appeal |
|  | FOR EACH ROW |
|  | EXECUTE PROCEDURE public."check"(); |

### Процедуры (Название процедуры, цена процедуры).

|  |
| --- |
| -- Table: public.procedures |
|  |  |
|  | -- DROP TABLE public.procedures; |
|  |  |
|  | CREATE TABLE public.procedures |
|  | ( |
|  | procname character(120) NOT NULL, |
|  | procprice integer, |
|  | CONSTRAINT procfkey PRIMARY KEY (procname) |
|  | ) |
|  | WITH ( |
|  | OIDS=FALSE |
|  | ); |
|  | ALTER TABLE public.procedures |
|  | OWNER TO postgres; |

* Скидка (Категория пациента, процент скидки)

* Начало формы
* Конец формы
* Начало формы
* Конец формы

|  |  |
| --- | --- |
|  | -- Table: public.sale |
|  |  |
|  | -- DROP TABLE public.sale; |
|  |  |
|  | CREATE TABLE public.sale |
|  | ( |
|  | scategory integer NOT NULL, |
|  | spercent integer, |
|  | CONSTRAINT sfkey PRIMARY KEY (scategory), |
|  | CONSTRAINT scategory\_lim UNIQUE (scategory) |
|  | ) |
|  | WITH ( |
|  | OIDS=FALSE |
|  | ); |
|  | ALTER TABLE public.sale |
|  | OWNER TO postgres; |

* Результат по процедурам(Прививка, Горчичник, Укол, Операция, Таблетка, Результат цены, Код обращения)

|  |
| --- |
| -- Table: public.proc\_res |
|  |  |
|  | -- DROP TABLE public.proc\_res; |
|  |  |
|  | CREATE TABLE public.proc\_res |
|  | ( |
|  | grafit integer, |
|  | mp integer, |
|  | operation integer, |
|  | injection integer, |
|  | tablet integer, |
|  | res\_price integer, |
|  | aid integer NOT NULL, |
|  | CONSTRAINT fkey PRIMARY KEY (aid), |
|  | CONSTRAINT aid\_aid FOREIGN KEY (aid) |
|  | REFERENCES public.appeal (aid) MATCH SIMPLE |
|  | ON UPDATE NO ACTION ON DELETE NO ACTION |
|  | ) |
|  | WITH ( |
|  | OIDS=FALSE |
|  | ); |
|  | ALTER TABLE public.proc\_res |
|  | OWNER TO postgres; |

### Триггер:

Создан триггер INSERT для проверки правильности вводимых данных в таблице договоров “appeal”.

|  |
| --- |
| -- Function: public."check"() |
|  |  |
|  | -- DROP FUNCTION public."check"(); |
|  |  |
|  | CREATE OR REPLACE FUNCTION public."check"() |
|  | RETURNS trigger AS |
|  | $BODY$ BEGIN |
|  | IF NEW.aid < 10000 THEN |
|  | RAISE EXCEPTION 'Boundary aid cant be < 10000'; |
|  | END IF; |
|  |  |
|  | IF NEW.did < 1000 THEN |
|  | RAISE EXCEPTION 'boundary did cant be < 1000'; |
|  | END IF; |
|  |  |
|  | IF NEW.pid < 100 THEN |
|  | RAISE EXCEPTION 'boundary pid cant be < 100'; |
|  | END IF; |
|  |  |
|  | IF NEW.price < 0 THEN |
|  | RAISE EXCEPTION 'price cant be negative'; |
|  | END IF; |
|  |  |
|  | RETURN NEW; |
|  | END;$BODY$ |
|  | LANGUAGE plpgsql VOLATILE |
|  | COST 100; |
|  | ALTER FUNCTION public."check"() |
|  | OWNER TO postgres; |

### Функции:

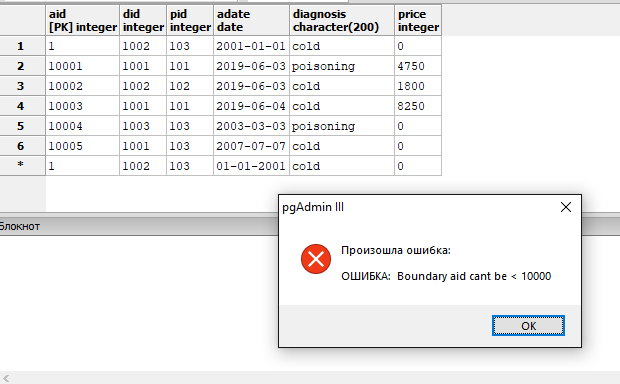
Функция удаления принимающая на вход “Код обращения” и удаляющая материнскую запись и все “Дочерние” записи

|  |
| --- |
| -- Function: public.del(integer) |
|  |  |
|  | -- DROP FUNCTION public.del(integer); |
|  |  |
|  | CREATE OR REPLACE FUNCTION public.del(aid\_del integer) |
|  | RETURNS void AS |
|  | $BODY$delete from proc\_res \* |
|  | where aid = aid\_del;  delete from appeal \*  where aid = aid\_del; |
|  | $BODY$ |
|  | LANGUAGE sql VOLATILE |
|  | COST 100; |
|  | ALTER FUNCTION public.del(integer) |
|  | OWNER TO postgres; |

Результаты моделирования

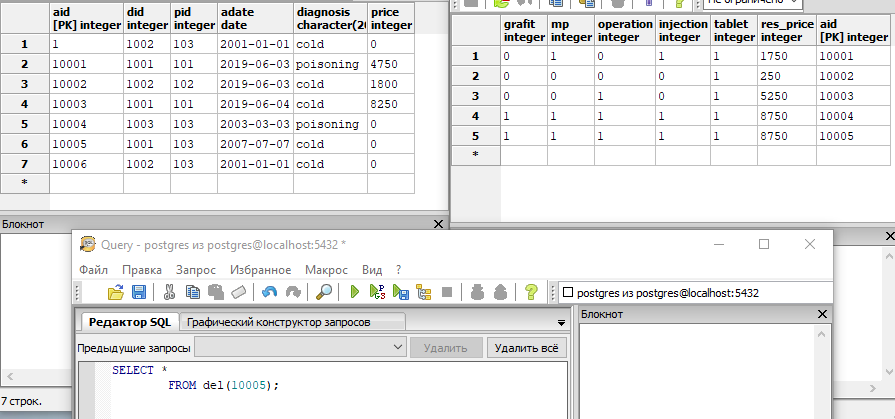
**Эксперимент №1**

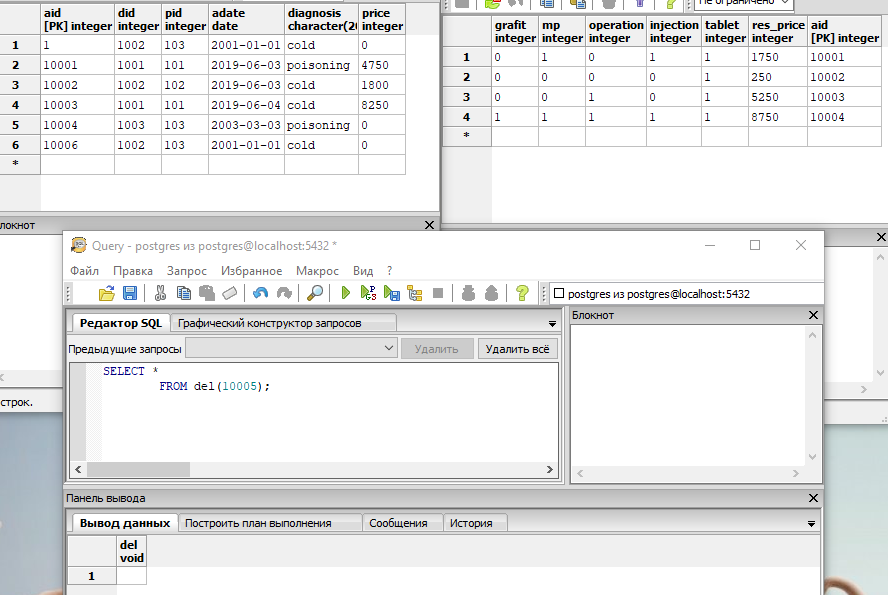
Проверка работы триггера, вводом неверного значения кода обращения



**Эксперимент №2**

Проверка работы функции удаления. Удаление записи агента влечет за собой удаление записи договра.





# Выводы

В ходе проделанной работы была реализована база данных, на основе имитации работы платной поликлиники из шести таблиц и были усановлены взаимоотношения между ними. Был создан триггер для проверки правильности данных, вводимых или заполняемых таблицы. А так же написаны функции удаления записей, что влекут за собой и удаление дочерних.

**Список источников**

1. Швецов В.И., Визгунов А.Н., Мееров И.Б. Базы данных. Н.Новгород: Изд-во ННГУ, 2004.
2. Карпова Т. Базы данных. Модели, разработка, реализация. С.-Петербург: Питер, 2001
3. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных. Учебник для вузов.- СПб: КОРОНА принт. 2000 – 416 с.
4. Диго С.М. Проектирование баз данных. - М.: Финансы и статистика, 1988.
5. Зеленков Ю.А. Введение в базы данных http://www.vsma.ac.ru/~pbarm/libraru/books/ db/toc.html.
6. Кириллов В.В. Основы проектирования баз данных. Учебное пособие. Сервер FORUM http://www.citforum.ru.
7. Кузнецов С.Д. Основы современных баз данных. http://www.citforum.ru.
8. Диго С.М. Проектирование баз данных. – М.: Финансы и статистика, 1988.
9. Четвериков В.Н., Ревунков Г.И., Самохвалов Э.Н. Базы и банки данных, ВШ 1986, 1992.

**Приложение**

<https://github.com/DimaSaz/TBD>