# ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

# "ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА к курсовой работе по курсу «Базы данных»

Тема работы: «Типографии города Донецка»

Руководители: Щедрин С.В Незамова Л.В Рычка О.В Разработал: ст.гр. ПИ-186

Моргунов А.Г

#### Донецк — 2020

#### РЕФЕРАТ

Пояснительная записка содержит: 90 страниц, 64 рисунка, 4 источника, 4 приложения.

Объектами исследования являются типографии города Донецка.

Цель работы - разработка системы для учета деятельности типографий города Донецка.

Результатом работы является база данных и клиентское приложение для взаимодействия с ней. Система может выполнять такие функции как: добавление, удаление, поиск записей в таблицах и справочниках БД, составление однотабличного и многотабличного отчетов в Microsoft Excel, составление HTML публикации на основе таблицы, создание различных типов диаграмм основанных на статистических данных, выполнение запросов различного уровня сложности.

Для разработки системы использована СУБД PostgreSQL.

SQL, C#, POSTGRESQL, ТАБЛИЦА, БАЗА ДАННЫХ, СУБД, ТИПОГРАФИЯ, SQL-ЗАПРОС, ВИДЫ БУМАГИ, БАНКИ

# СОДЕРЖАНИЕ

введение	5
1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ	6
1.1 Проблемы автоматизации учета деятельности типографий	горда
Донецка	6
1.2 Разработка информационной системы «Учет деятельности Типо	ографий
города Донецка»	6
1.2.1 Анализ предметной области	6
1.2.2 Описание входных документов	7
1.2.3 Описание выходных документов	7
1.2.4 Список ограничений	7
1.3 Назначение и функции системы	8
2 ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ	9
2.1 Инфологическое проектирование	9
2.2 Даталогическое проектирование	10
2.3 Физическое проектирование	10
3 РАЗРАБОТКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ	15
3.1 Создание проекта	15
3.2 Создание базы данных	15
3.3 Создание таблиц	15
3.4 Схема БД	17
3.5 Разработка пользовательского интерфейса	17
3.6 Разработка процедур обработки данных (модулей)	26
3.7 Тестирование и отладка БД и ИС	28
3.8 Разработка эксплуатационной документации БД и ИС	29
ВЫВОДЫ	30
СПИСОК ЛИТЕРАТУРЫ	31
ПРИЛОЖЕНИЕ А. Техническое задание	32
ПРИЛОЖЕНИЕ Б. SQL-запросы	33

ПРИЛОЖЕНИЕ В. Руководство пользователя	50
ПРИЛОЖЕНИЕ Г. Текст программы	.51

#### ВВЕДЕНИЕ

В современном мире технологии все больше и больше вливаются в жизнь людей. В том числе и в бизнес — сферу. Многие предприниматели и организации используют современные методы обработки информации. Основным средством для быстрой обработки информации являются базы данных. Это средство позволяет обрабатывать, выводить, хранить и манипулировать огромными объемами данных. Поэтому такое средство востребовано и используется повсеместно.

Система управления базами данных(СУБД) — это программный комплекс, обеспечивающий централизованное хранение данных и предоставляющий приложениям услуги по обработке данных.

Совокупность данных, хранимых под управлением СУБД, называется базой данных. В оригинальном английском варианте словосочетание data base означает «основание, состоящее из данных». Этот смысл несколько искажается в русском словосочетании «база данных». На самом деле это – фундамент, на котором строятся приложения и который состоит из данных. Действительно, данные (а следовательно, база данных) являются очень существенной частью практически любой информационной системы[1]

Целью разработки является создание базы данных типографий города Донецка с помощью СУБД PostgreSQL и создание клиентского приложение для взаимодействия с этой базой данных на языке программирования С#

Разработанная система может применятся для учета деятельности типографий города Донецка

#### 1. ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ

1.1 Проблемы автоматизации учета деятельности типографий горда Донецка

Работа типографий подразумевает учет огромного числа данных. В них входят: информация о типографиях, работниках, заказчиках, заказах, информация о различных изделиях и их видах и т.д. Также финансовый вопрос стоит не на последнем месте. Из-за ограниченного бюджета необходимо максимально сократить расходы на учет информации, но при этом не потерять надежность и качество хранения информации. Без автоматизации невероятно сложно и дорого следить за всем. Следовательно автоматизация упрощает и уменьшает затраты на обработку и хранение информации. Также благодаря переходу на хранение данных с помощью БД облегчается доступ и контроль за информацией.

1.2 Разработка информационной системы «Учет деятельности Типографий города Донецка

# 1.2.1 Анализ предметной области

Благодаря анализу можно заключить, что для учета деятельности типографий нуобходимы такие данные название типографии, тип собственности, район, адрес типографии, телефон типографии, год открытия, фио заказчика (физ.или юр. лицо), адрес заказчика. дата рождения/регистрации, телефон заказчика, расчетный счет заказчика (номер р/с), банк размещения р/с, вид изделия заказчика, название издания, титульная страница (эскиз автора), фио принявшего заказ в типографии, цена экземпляра изделия, тираж, количество листов в изделии, формат листов, тип бумаги, плотность бумаги, дата принятия заказа, дата выполнения заказа(план), дата выполнения заказа(факт), предоплата (принятый аванс), дополнительные сведения о заказе.

#### 1.2.2 Описание входных документов

Ввод данных производится с помощью формы добавления записи, а также генерации записей. Для изменения данных используется форма обновления записи.

# 1.2.3 Описание выходных документов

Входные данные могут быть различными:

- Excel таблица с результатом запроса
- -HTML-страница с выбранными данными
- -Диаграммы

# 1.2.4 Список ограничений

При разработке базы данных были выявлены следующие ограничения:

- -Год открытия типографии должен быть больше 1980 и меньше 2020
- -Дата принятия заказа не может быть меньше года основания типографии
- -Дата выполнения заказа (план) и (факт) должны быть больше даты принятия заказа
  - Тираж заказа начинается с 50 штук
  - -Цена одного экземпляра изделия начинается с 50 у/е
  - -Количество листов издания больше 0
  - -Предоплата больше или равна 0
  - -Дата рождения заказчика должна быть меньше 2002 года
  - -Плотность бумаги должна быть больше 40 и меньше 350

# 1.3 Назначение и функции системы

Разрабатываемая программа предназначена для учета деятельности типографий города Донецка. Благодаря своим возможностям она облегчает работу с данными.

К функциям системы относятся: хранение данных, поиск данных, управление данными, группировка данных, сохранение целостности данных, генерация псевдоинформации.

#### 2. ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

## 2.1 Инфологическое проектирование

Инфологическая модель (информационно-логическая модель) ориентированная на человека и не зависимая от типа СУБД модель предметной области, определяющая совокупности информационных объектов, их атрибутов и отношений между объектами, динамику изменений предметной области, а информационных потребностей также характер пользователей. Инфологическая модель предметной области может быть описана моделью "сущность—связь" (моделью Чена), в основе которой лежит деление реального мира на отдельные различимые сущности, находящиеся в определенных связях друг с другом, причем обе категории — сущность и связь полагаются первичными, неопределенными понятиями.[2]

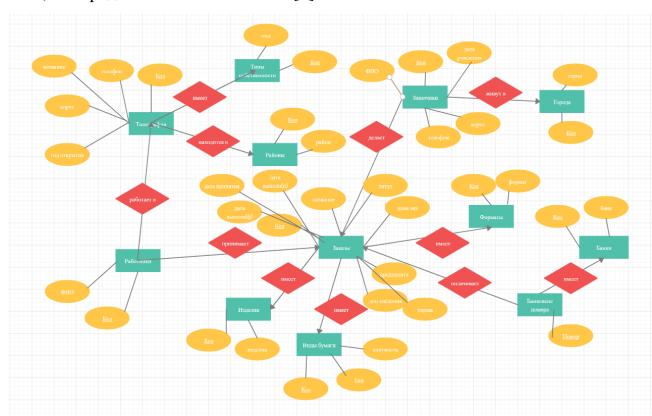


Рисунок 2.1 – ER диаграмма

#### 2.2 Даталогическое проектирование

Даталогическое проектирование является проектированием логической структуры БД, что означает определение всех информационных единиц и связей между ними, задание их имен и типов, а также некоторых количественных характеристик (например, длины поля).

При проектировании логической структуры БД, осуществляется преобразование исходной инфологической модели в модель данных, поддерживаемую конкретной СУБД, и проверка адекватности полученной даталогической модели отображаемой предметной области.

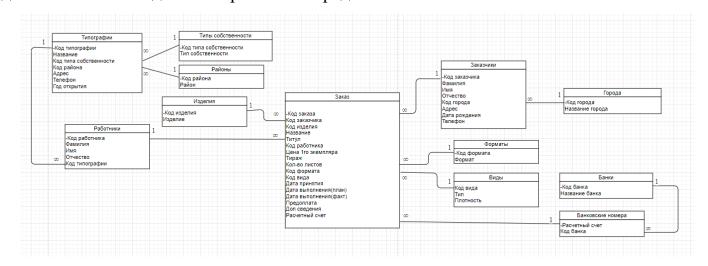


Рисунок 2.2 – логическая схема

# 2.3 Физическое проектирование

Физическое проектирование — создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.[3]

С учетом возможностей и особенностей PostgreSQL было проведено физическое проектирования. В результате которого были выведены скрипты на языке SQL, которые описывают все ограничения, типы и связи между таблицами. (Рис 2.3-2.18)

```
CREATE TABLE public.accounts

(
    account character(20) COLLATE pg_catalog."default" NOT NULL,
    bank_id integer NOT NULL,
    CONSTRAINT accounts_pkey PRIMARY KEY (account),
    CONSTRAINT accounts_bank_id_fkey FOREIGN KEY (bank_id)
    REFERENCES public.banks (id) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
```

Рисунок 2.3 Срипт SQL описывающий таблицу банковские счета

```
CREATE TABLE public.banks
(
    id bigint NOT NULL DEFAULT nextval('banks_id_seq'::regclass),
    bank character varying(30) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT banks_pkey PRIMARY KEY (id)
)
```

Рисунок 2.4 Срипт SQL описывающий таблицу банки

```
CREATE TABLE public.cities
(
   id bigint NOT NULL DEFAULT nextval('cities_id_seq'::regclass),
   city character varying(30) COLLATE pg_catalog."default" NOT NULL,
   CONSTRAINT cities_pkey PRIMARY KEY (id)
)
```

Рисунок 2.5 Срипт SQL описывающий таблицу города

```
CREATE TABLE public.customers

(
    id bigint NOT NULL DEFAULT nextval('customers_id_seq'::regclass),
    first_name character varying(15) COLLATE pg_catalog."default" NOT NULL,
    second_name character varying(15) COLLATE pg_catalog."default" NOT NULL,
    third_name character varying(15) COLLATE pg_catalog."default" NOT NULL,
    city_id integer NOT NULL,
    address character varying(40) COLLATE pg_catalog."default" NOT NULL,
    birthday birthday_check NOT NULL,
    phone character(10) COLLATE pg_catalog."default",
    CONSTRAINT customers_pkey PRIMARY KEY (id),
    CONSTRAINT city_cascade_del FOREIGN KEY (city_id)
        REFERENCES public.cities (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT customers_birthday_check CHECK (birthday::date < '2002-01-01'::date) NOT VALID,
    CONSTRAINT customer_birthday_check2 CHECK (birthday::date > '1920-01-01'::date) NOT VALID
)
```

Рисунок 2.6 Срипт SQL описывающий таблицу заказчики

```
CREATE TABLE public.districts
(
   id bigint NOT NULL DEFAULT nextval('districts_id_seq'::regclass),
   district character varying(15) COLLATE pg_catalog."default" NOT NULL,
   CONSTRAINT districts_pkey PRIMARY KEY (id)
)
```

Рисунок 2.7 Срипт SQL описывающий таблицу районы

```
CREATE TABLE public.formats
(
    id bigint NOT NULL DEFAULT nextval('formats_id_seq'::regclass),
    format character varying(20) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT formats_pkey PRIMARY KEY (id)
)
```

Рисунок 2.8 Срипт SQL описывающий таблицу форматы

```
CREATE TABLE public.prints
    id bigint NOT NULL DEFAULT nextval('prints_id_seq'::regclass),
    print character varying(20) COLLATE pg_catalog."default" NOT NULL,
    type_id integer NOT NULL,
    district_id integer NOT NULL,
    address character varying(40) COLLATE pg_catalog."default" NOT NULL,
    phone character(10) COLLATE pg_catalog."default" NOT NULL,
    year integer NOT NULL,
    CONSTRAINT prints_pkey PRIMARY KEY (id),
    CONSTRAINT prints_district_id_fkey FOREIGN KEY (district_id)
        REFERENCES public.districts (id) MATCH SIMPLE ON UPDATE CASCADE
        ON DELETE CASCADE,
        REFERENCES public.types (id) MATCH SIMPLE ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT year_check2021 CHECK (year < 2021),
    CONSTRAINT prints_year_check CHECK (year > 1980)
```

Рисунок 2.9 Срипт SQL описывающий таблицу типографии

```
CREATE TABLE public.orders
   id bigint NOT NULL DEFAULT nextval('orders_id_seq'::regclass),
   customer_id integer NOT NULL,
   producttype_id integer NOT NULL,
   publication character varying(20) COLLATE pg_catalog."default",
   picture bytea,
   worker_id integer NOT NULL,
   price integer NOT NULL,
   calculation integer NOT NULL,
   count integer,
   format_id integer NOT NULL,
   papertype_id integer NOT NULL,
   datastart date NOT NULL,
   dataplan date NOT NULL,
   datafact date NOT NULL,
   cost integer NOT NULL,
   comment text COLLATE pg_catalog."default",
   account character(20) COLLATE pg_catalog."default",
   CONSTRAINT orders_pkey PRIMARY KEY (id),
   CONSTRAINT orders_account_fkey FOREIGN KEY (account)
        REFERENCES public.accounts (account) MATCH SIMPLE
       ON UPDATE CASCADE
       ON DELETE CASCADE,
   CONSTRAINT orders_customer_id_fkey FOREIGN KEY (customer_id)
       REFERENCES public.customers (id) MATCH SIMPLE
       ON UPDATE CASCADE
       ON DELETE CASCADE,
   CONSTRAINT orders_format_id_fkey FOREIGN KEY (format_id)
        REFERENCES public.formats (id) MATCH SIMPLE
       ON UPDATE CASCADE
       ON DELETE CASCADE,
   CONSTRAINT orders_papertype_id_fkey FOREIGN KEY (papertype_id)
        REFERENCES public.papertypes (id) MATCH SIMPLE
        ON UPDATE CASCADE
       ON DELETE CASCADE,
   CONSTRAINT orders_producttype_id_fkey FOREIGN KEY (producttype_id)
       REFERENCES public.producttypes (id) MATCH SIMPLE
       ON UPDATE CASCADE
       ON DELETE CASCADE,
   CONSTRAINT orders_worker_id_fkey FOREIGN KEY (worker_id)
        REFERENCES public.workers (id) MATCH SIMPLE
       ON UPDATE CASCADE
       ON DELETE CASCADE
```

Рисунок 2.10 Срипт SQL описывающий таблицу заказы

```
CONSTRAINT orders_calculation_check CHECK (calculation >= 50),

CONSTRAINT orders_check CHECK (dataplan > datastart),

CONSTRAINT orders_checkl CHECK (datafact > datastart AND datafact < CURRENT_DATE),

CONSTRAINT orders_cost_check CHECK (cost >= 0),

CONSTRAINT orders_count_check CHECK (count > 0),

CONSTRAINT orders_price_check CHECK (price >= 50),

CONSTRAINT orders_price_check CHECK (datastart > '1980-01-01'::date AND datastart < '2021-01-01'::date),

CONSTRAINT orders_cost_check2 CHECK (cost <= (calculation * price)) NOT VALID

)
```

Рисунок 2.10, лист 2

```
CREATE TABLE public.papertypes
(
   id bigint NOT NULL DEFAULT nextval('papertypes_id_seq'::regclass),
   papertype character varying(30) COLLATE pg_catalog."default" NOT NULL,
   density integer NOT NULL,
   CONSTRAINT papertypes_pkey PRIMARY KEY (id),
   CONSTRAINT papertypes_density_check CHECK (density > 39 AND density < 351)
)</pre>
```

Рисунок 2.11 Срипт SQL описывающий таблицу виды бумаги

```
CREATE TABLE public.types
(
   id bigint NOT NULL DEFAULT nextval('types_id_seq'::regclass),
   type character varying(15) COLLATE pg_catalog."default" NOT NULL,
   CONSTRAINT types_pkey PRIMARY KEY (id)
)
```

Рисунок 2.12 Срипт SQL описывающий таблицу типы собственности

```
CREATE TABLE public.workers
(
    id bigint NOT NULL DEFAULT nextval('workers_id_seq'::regclass),
    first_name character varying(15) COLLATE pg_catalog."default" NOT NULL,
    second_name character varying(15) COLLATE pg_catalog."default" NOT NULL,
    third_name character varying(15) COLLATE pg_catalog."default" NOT NULL,
    print_id integer NOT NULL,
    CONSTRAINT workers_pkey PRIMARY KEY (id),
    CONSTRAINT workers_print_id_fkey FOREIGN KEY (print_id)
        REFERENCES public.prints (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)
```

Рисунок 2.13 Срипт SQL описывающий таблицу работники

# 3. РАЗРАБОТКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

# 3.1 Создание проекта

Для разработки клиентской части системы выбран язык С# и среда разработки Visual Studio 2019.

Процесс создания и подготовки проекта для работой с БД включает в себя. Создание нового проекта на языке С#. Подключение библиотеки Npgsql для связывания серверной части системы с клиентской.

#### 3.2 Создание БД

Для работы с сервером используется среда PG admin 4. В ней упрощены и автоматизированы некоторые взаимодействия с базой данных.

Для начала работы нужно авторизоваться. После этого создаем новую базу данных. После того как база данных создана можно приступать к созданию таблиц

#### 3.3 Создание таблиц

При помощи инструмента Query Tool мы можем воспользоваться командой CREATE TABLE, которая позволяет создавать таблицы.

Благодаря физическому проектированию у нас уже есть код для создания каждой таблицы. Остается только запустить его. Важно что сначала создаются таблицы без внешних ключей, а уже потом таблицы, связанные с ними.

После создания таблиц база данных выглядит так(Рис 3.1)

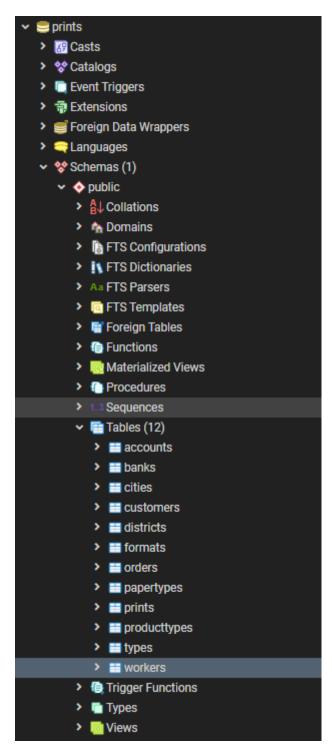


Рисунок 3.1 – База данных после создания таблиц

# 3.4 Схема БД

# Схема БД создается чтобы сохранить целостность данных(Рис 3.2)

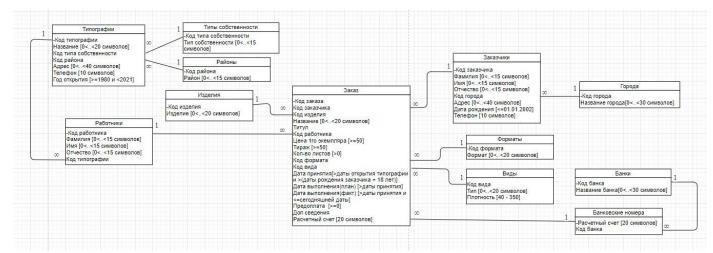


Рисунок 3.2 – схема БД

# 3.5 Разработка пользовательского интерфейса

При запуске программы пользователя встречает окно авторизации (Рис 3.3)

Авторизация	×					
Типографии города Донецка						
Логин:						
Пароль:						
Курсовая работа студента группы ПИ-18б Моргунова Арсения						

Рисунок 3.3 -форма авторизации

После авторизации пользователь попадает на главную форму, которая по умолчанию заполняется таблицей типографии (Рис 3.4)

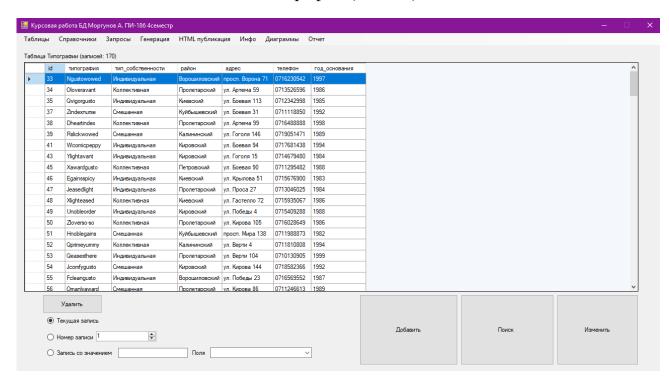


Рисунок 3.4 – Главная форма

Для выбора отображаемой таблицы в пунктах меню есть кнопки таблицы и справочники, при нажатии на которые появится выпадающий список таблиц и справочников соответственно (Рис 3.5-3.6)

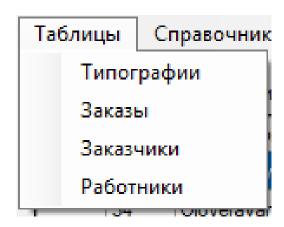


Рисунок 3.5 – Выпадающий список таблиц

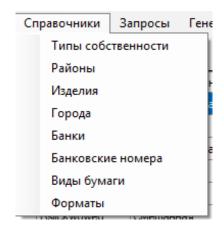


Рисунок 3.6 – Выпадающий список справочников

При нажатии на кнопку добавить появится форма с добавлением записи в ту таблицу, которая сейчас отображается на главной форме. Формы добавления представлены ниже(Рис 3.7-3.12)

🖳 Добавление элемента		_	×
Таблица Типографии (запи	исей: 170)		
Название типографии			
Тип собственности		~	
Район		~	
Адрес			
Телефон			
Год открытия			
Добавить	Закрыть		

Рисунок 3.7 – Добавление в таблицу типографии

🔛 Добавление элемента			_		×
Таблица Заказы (записей: 1500)					
Заказчик	~	Тираж	0	+	
Изделие	~	Количество листов в изделии	0	-	
Название		Формат листов		~	
Титиульная страница		Вид бумаги		~	
		Дата принятия заказа	2020-05-28	-	
Выбрать		Дата выполнения заказа (план)	2020-05-28	■▼	
Отчистить		Дата выполнения заказа (факт)	2020-05-28	■▼	
Работник	~	Предоплата	0	<u> </u>	
Цена экземпляра	A .	Банковский счет		~	
Доп сведения					
		Добавить	Назад		

Рисунок 3.8 – Добавление в таблицу заказы

Lenemou				
🔛 Добавление элеме	нта	_		×
Таблица Зака	взчики (записей: 1500)			
Имя				
Фамилия				
Отчество				
Город			~	
Адрес				
Дата рождения	2020-05-28		■▼	
Телефон				
Добавить		Ha	эзад	

Рисунок 3.9 – Добавление в таблицу заказчики

🔛 Добавление элемента	– 🗆 X
Таблица Работники (за	іписей: 1000)
Имя	
Фамилия	
Отчество	
Тпография	~
Добавить	Назад

Рисунок 3.10 – Добавление в таблицу работники

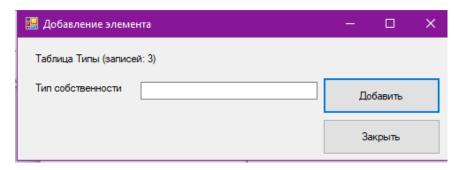


Рисунок 3.11 – Добавление в таблицу типы собственности

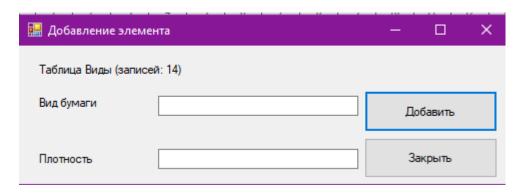


Рисунок 3.12 — Добавление в таблицу виды собственности

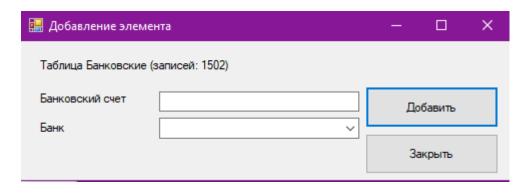


Рисунок 3.13 – Добавление в таблицу банковские счета

Для изменения записи нужно выделить ее на главной форме и нажать кнопку изменить. После этого появится форма добавления записей, но уже заполненная(Рис 3.14). Для изменения данных нужно нажать кнопку изменить.

🖳 Изменение элемен	та	_		×
Таблица Зака	эзчики (записей: 1500)			
Имя	Василий			
Фамилия	Бобров			
Отчество	Вадимович			
Город	3. Харцизск		~	
Адрес	ул. Боевая 105			
Дата рождения	1989-04-07			
Телефон	0716672767			
Изменить		Н	азад	

Рисунок 3.14 — Форма изменения записи таблицы заказчики

Удаление происходит с помощью кнопки удалить на главной форме. При этом можно выбрать желаемый тип удаления(Рис 3.15)

130 I BODNC I BOODOR	Эдурович	FOCTOR	HIDOCH, DODOHA 44	107.03.1373	10/103
Удалить					
<ul><li>Текущая запись</li></ul>					
О Номер записи 1	-				
О Запись со значением		По	ля		~

Рисунок 3.15 – Кнопка удалить и переключатели типа удаления

Реализованы запросы. Чтобы открыть форму запросов необходимо нажать на пункт верхнего меню запросы. Откроется форма, в которой можно выбрать запрос(Рис 3.16)

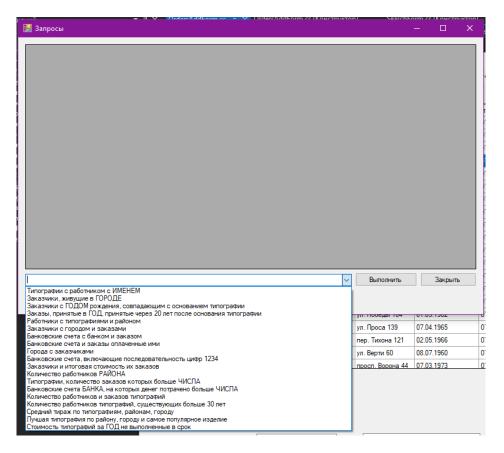


Рисунок 3.16 – Форма запросов и выбор запроса

Для того чтобы открыть диаграмму нужно нажать на пункт верхнего меню диаграммы и выбрать интересующую. Вид диаграмм(Рис 3.17-3.19)

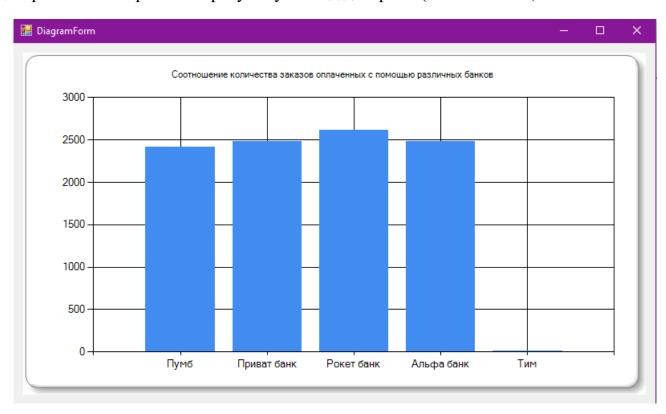


Рисунок 3.17 – Диаграмма количество заказов на банк

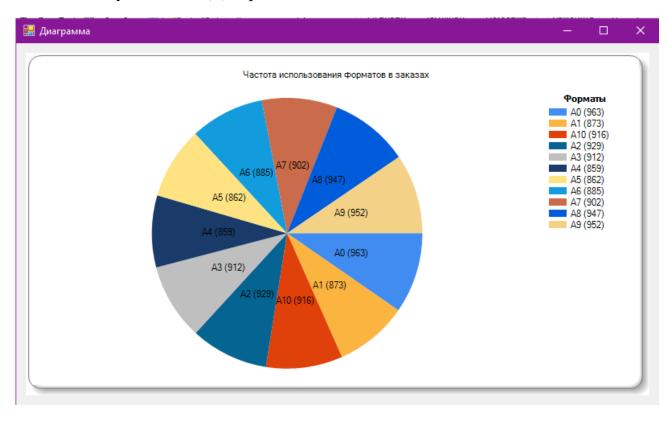


Рисунок 3.18 – Диаграмма частота использования форматов

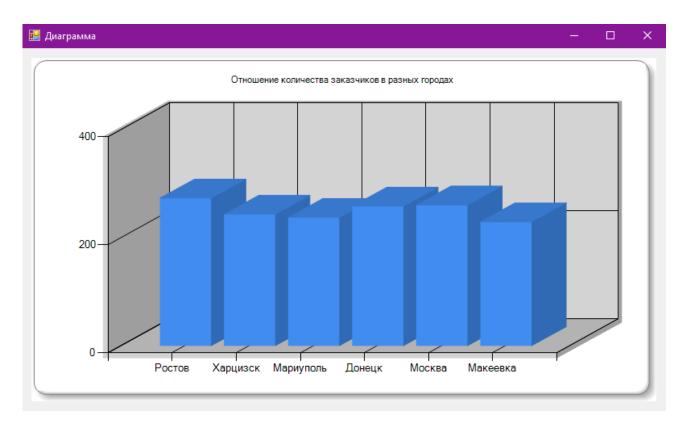


Рисунок 3.19 – Диаграмма количество заказчиков в городах

Для вызова html публикации нажмите на пункт верхнего меню html-публикация и выберите таблицу. Результат(Рис 3.20)

[	id	типография	тип собственности	район	адрес	телефон	год основания
Ш	=	Ngustowowed	Индивидуальная		просп. Ворона 71	0716230542	
	34	Oloveravant		_		0713526596	
Н	_	Oloveravant	Коллективная	Пролетарский	ул. Артема 59	0/13326396	1980
Ш	35	Qvigorgusto	Индивидуальная	Киевский	ул. Боевая 113	0712342998	1985
	37	Zindexnurse	Смешанная	Куйбышевский	ул. Боевая 31	0711118850	1992
П	38	Dheartindex	Коллективная	Пролетарский	ул. Артема 99	0716488888	1998
	39	Rslickwowed	Смешанная	Калининский	ул. Гоголя 146	0719051471	1989
	41	Wcomicpeppy	Индивидуальная	Кировский	ул. Боевая 94	0717681438	1994
	43	Ylightavant	Индивидуальная	Кировский	ул. Гоголя 15	0714679480	1984
Ш	45	Xawardgusto	Коллективная	Петровский	ул. Боевая 90	0711295482	1988
	46	Egainsspicy	Индивидуальная	Киевский	ул. Крылова 51	0715676900	1983
Ш	47	Jeasedlight	Индивидуальная	Пролетарский	ул. Проса 27	0713046025	1984
	48	Xlighteased	Коллективная	Киевский	ул. Гастелло 72	0715935067	1986
	49	Unobleorder	Индивидуальная	Кировский	ул. Победы 4	0715409288	1988
	50	Zloverso-so	Коллективная	Пролетарский	ул. Кирова 105	0716028649	1986
	51	Hnoblegains	Смешанная	Куйбышевский	просп. Мира 138	0711988873	1982
	52	Qprimeyummy	Коллективная	Калининский	ул. Верти 4	0711810808	1994

Рисунок 3.20 – html – публикация таблицы типографии

Также реализованы однотабличные (Рис 3.21) и многотабличные (Рис 3.22) Excel отчеты

A	1	2	3	4
1	номер_заказа	заказ	дата_выполнения_факт	дата_выполнения_план
2	9022	Alphaado	23.08.2018	18.03.2017
3	6914	Alphaavar	24.05.2018	15.01.2017
4	3212	Alphaavar	15.09.2018	26.08.2017
5	331	Alphaavar	14.03.2018	10.03.2018
6	7062	Alphaawa	15.06.2018	24.04.2018
7	5480	Alphacivil	16.04.2018	24.01.2018
8	3459	Alphacivil	18.05.2018	28.08.2017
9	6008	Alphacivil	20.01.2018	17.01.2018
10	3045	Alphaclea	12.01.2018	21.01.2017
11	2532	Alphaclea	20.06.2018	14.04.2017
12	4750	Alphacom	19.05.2018	25.08.2017
13	5524	Alphacom	14.05.2018	14.01.2017
14	5019	Alphacom	20.03.2018	19.06.2017
15	9484	Alphacris	26.05.2018	26.06.2017
16	7063	Alphacris	28.07.2018	21.01.2017
17	1539	Alphadefi	16.02.2018	26.01.2018
18	4628	Alphadefi	13.09.2018	19.02.2018
19	450	Alphadefi	28.09.2018	14.01.2018
20	4747		22.22.2242	25 27 2247

Рисунок 3.21 – Однотабличный отчет

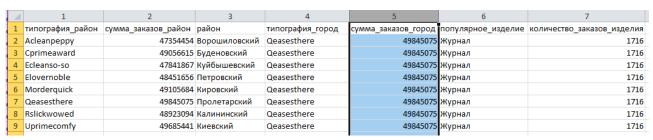


Рисунок 3.22 – Многотабличный отчет

# 3.6 Процедуры обработки данных

Для вывода информации на главную форму используется метод Select()(Рис.3.23)

```
public void Select()
{
    try
    {
        conn.Open();
        cmd = new NpgsqlCommand(selectSql, conn);
        dt = new DataTable();
        dt.Load(cmd.ExecuteReader());

        conn.Close();

        dgvData.DataSource = null;
        dgvData.DataSource = dt;
    }
    catch(Exception ex)
    {
        conn.Close();
        MessageBox.Show("Error: " + ex.Message);
    }
}
```

Рисунок 3.23 – Meтод Select()

Для экспорта в html(Рис.3.24) также создан отдельный метод

```
public void CreateHTMLFile(string table,string name)
{
    string path = System.IO.Directory.GetCurrentDirectory() + @"\" +name +".html";
    var sbHTML = new StringBuilder();
    var sw = new StreamWriter(System.IO.Directory.GetCurrentDirectory() + @"\" +name +".html");
    sbHTML.AppendLine("<DOCTYPE html>");
    sbHTML.AppendLine("<html>");
    sbHTML.AppendLine("<html>");
    sbHTML.AppendLine("<itile>");
    sbHTML.AppendLine("</itile>");
    sbHTML.AppendLine("</html>");
    sbHTML.AppendLine("</head>");
    sbHTML.AppendLine("</head>");
    sbHTML.AppendLine("</bddy>");
    sbHTML.AppendLine("</hdalp");
    sbHTML.AppendLine("</html>");
    sw.Write(sbHTML);
    sw.Close();
    Process.Start(System.IO.Directory.GetCurrentDirectory() + @"\" + name + ".html");
}
```

Рисунок 3.24 – Экспорт в html

Функциональность остальных возможностей системы заключается в комбинировании функционала отдельных методов и кода.

# 3.7 Тестирование и отладка БД и ИС

При вводе некорректных данных программа уведомляет пользователя и выдает ошибку(Рис 3.25)

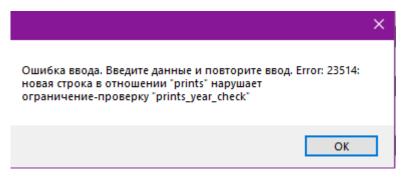


Рисунок 3.25 – неправильный ввод

Если пользователь ввел не все данные, то система оповестит его об этом(Рис 3.26).

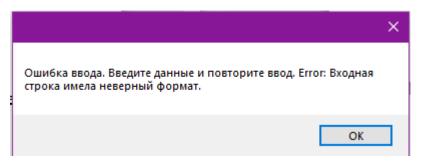


Рисунок 3.26 – неполный ввод

При попытке пользователя сгенерировать справочник система выдаст ошибку(Рис 3.27)

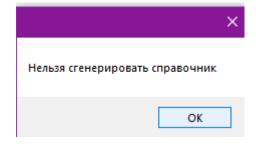


Рисунок 3.27 – ошибка при генерации справочника

3.8 Разработка эксплуатационной документации БД и ИС

В приложении В присутствует руководство пользователя, в котором описан функционал и взаимодействие с системой.

#### ВЫВОДЫ

Результат работы — система, реализующая систему для учета деятельности типографий города Донецка, реализованная при помощи языка программирования С# и системы управления базами данных PostgreSQL.

К преимуществам программы относятся: удобный пользовательский интерфейс, реализация и управление базой данных с помощью клиентского интерфейса, возможность добавлять, удалять, редактировать, выбирать данные из таблиц. Возможен экспорт данных в html и excel. Безопасность для пользователя от случайного удаления данных. Возможен вывод диаграмм со статистическими данными. Просмотр результатов запросов различной сложности.

Недостатки: при слишком большом количестве обрабатываемых записей заметны подтормаживания системы.

#### СПИСОК ЛИТЕРАТУРЫ

- 1. Основы технологий баз данных: учеб. пособие/ Б.А. Новиков, Е.А.Горшкова; под ред. Е.В. Рогова. – М.:ДМК Пресс, 2019. -240c
- 2. Инфологическая модель [Электронный ресурс] // Инфологическая модель. режим доступа:
  <a href="http://wiki.mvtom.ru/index.php/%D0%98%D0%BD%D1%84%D0%BE%D0%BB%D0%BB%D0%BE%D0%BB%D0%BD%D0%BB%D0%BD%D0%BB%D0%BD%D0%BD%D0%BD%D0%BD%D0%BD%D0%BD%D0%BD%D0%BD%D0%BD%D0%BD%D0%BD%D0%BD%D0%BD%D0%

0%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5

3. Физическое проектирование [Электронный ресурс] // Физическое проектирование. — режим доступа:

https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1
%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5
\_%D0%B1%D0%B0%D0%B7\_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B
%D1%85#%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D
0%BA%D0%BE%D0%B5\_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1
%82%D0%B8%D1%80%D0%BE%D0%B5%D0%B8%D0%B5

# ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ

#### ПРИЛОЖЕНИЕ Б SQL-ЗАПРОСЫ

#### Симметричное внутреннее соединение с условием:

# 1.Заказчики с указанным годом рождения, который совпадает с годом

# основания типографии

select cust.first\_name as имя, cust.second\_name as фамилия, cust.third\_name as отчество, extract(year from cust.birthday) as год\_рождения, p.year as год\_основаия, p.print as типография from customers as cust, orders as o, workers as w, prints as p where cust.id = o.customer\_id and o.worker\_id=w.id and w.print\_id=p.id and extract(year from cust.birthday)=p.year and  $p.year = \Gamma O \underline{\mathcal{I}}$ order by p.print

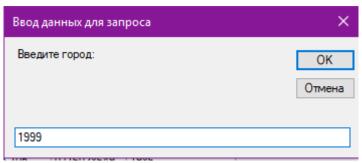


Рисунок Б.1 – Ввод данных для запроса 1

Запрос с параметром 1999

	имя	фамилия	отчество	типография
•	Илья	Макаров	Викторович	Cspicylover
	Жерар	Иванов	Викторович	Hoomicroomy
	Николай	Петухов	Назарович	Heomicroomy
	Василий	Макаров	Густавович	Hoomicroomy
	Густав	Панов	Игнатович	Heomicroomy
	Назар	Карпов	Карлович	Hgustoslick
	Карл	Ершов	Геннадиевич	Hgustoslick
	Марат	Федоров	Игнатович	Mprotogains
	Альберт	Рябов	Михаилович	Qeasesthere
	Егор	Карпов	Захарович	Vexactoh my
	Егор	Смирнов	Густавович	Vexactoh my
	Марат	Карпов	Игнатович	Vexactoh my

Рисунок Б.2 – Запрос 1

# 2.Заказы с указанным годом принятия заказа, принятые через 20 лет после основания типографии

select
o.id as номер\_заказа,
o.publication as название
from workers as w,
prints as p,
orders as o
where p.id=w.print\_id and
o.worker\_id=w.id and
(extract(year from o.datastart)-20)=p.year and
extract(year from o.datastart)=<u>ГОЛ</u>
order by p.year

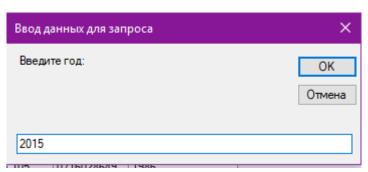


Рисунок Б.3 – Ввод данных для запроса 2

Запрос с параметром 2015

	номер_заказа	название	
<b>&gt;</b>	20	Omicronoh my Y	
	108	Deltathere P	
	165	Zetapeace I	
	234	Lambdapeace D	
	316	Alphagusto I	
	340	Upsilonready H	
	346	Nuyummy F	
	395	Thetasharp S	
	414	Sigmaclean X	
	417	Rhoeases R	
	492	Xiyummy W	
	511	Etacomic V	
	525	Omegaloved B	
	581	Tauso-so M	
	601	Xiadore K	
	602	Upsilonready V	
	633	Engilonoivil T	

Рисунок Б.4 – Запрос 2

# 3.Заказчики, которые живут в указанном городе

select cust.first\_name as имя, cust.second\_name as фамилия, cust.third\_name as отчество, city.city as город from customers as cust, cities as city where cust.city\_id=city.id and city.city=<u>ГОРОЛ</u> order by 4

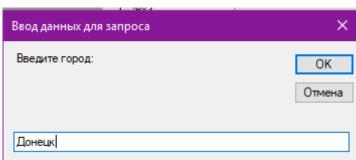


Рисунок Б.5 – Ввод данных для запроса 3

Запрос с параметром «Донецк»

	имя	фамилия	отчество	город
	Жерар	Панов	Борисович	Донецк
	Захар	Иванов	Николаевич	Донецк
	Альберт	Рябов	Заурович	Донецк
	Захар	Макаров	Захарович	Донецк
	Назар	Зайцев	Густавович	Донецк
	Виктор	Рябов	Михаилович	Донецк
	Николай	Зайцев	Назарович	Донецк
	Илья	Макаров	Егорович	Донецк
	Артур	Петров	Альбертович	Донецк
	Иван	Осипов	Васильевич	Донецк
	Гавриил	Макаров	Альбертович	Донецк
	Геннадий	Петров	Игоревич	Донецк
	Гавриил	Калинин	Жерарович	Донецк
	Густав	Бобров	Заурович	Донецк
	Богдан	Иванов	Артурович	Донецк
	Геннадий	Осипов	Абрамович	Донецк
	Мори	Фоловов	Менатории	Пология

Рисунок Б.6 – Запрос 3

# 4. Типографии, в которых есть работник с указанным именем

select p.print as типография, count(w.first\_name) as количество\_работников\_c\_именем from prints as p, workers as w where p.id=w.print\_id and w.first\_name=<u>ИМЯ</u> group by p.print order by 1

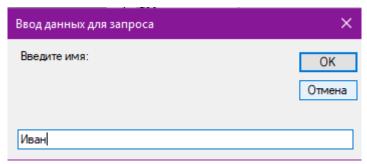


Рисунок Б.7 – Ввод данных для запроса 4

Запрос с параметром «Иван»

	типография	количество_рабо
•	Cawardcivil	1
	Cyummycivil	1
	Egainsspicy	1
	Fprotogains	1
	Gcomicmanly	1
	Geasessharp	1
	Hprotonurse	1
	lindexdefix	1
	Jeasesexact	1
	Nawardcomfy	1
	Ngustowowed	1
	Oloveravant	1
	Omanlyaward	2
	Qcomfyasset	2
	Qlovedready	1
	Qprimeyummy	1
	Releamente Driaymore E 9	201100 4

Рисунок Б.8 – Запрос 4

#### Симметричное внутреннее соединение без условия:

# 1. Работники с типографиями в которых они работают и районом расположения типографии

select
w.first\_name as имя,
w.second\_name as фамилия,
w.third\_name as отчество,
p.print as типография ,
d.district as район
from workers as w,
prints as p,
districts as d
where p.id=w.print\_id and
p.district\_id=d.id
order by 1,2,3

	имя	фамилия	отчество	типография	район
<b>•</b>	Абрам	Бобров	Богданович	Xcleanwowed	Куйбышевский
	Абрам	Бобров	Вадимович	Oloveravant	Пролетарский
	Абрам	Зайцев	Богданович	Yslickclean	Кировский
	Абрам	Зайцев	Борисович	Lquickso-so	Буденовский
	Абрам	Зайцев	Вадимович	Ceasedquick	Калининский
	Абрам	Иванов	Абрамович	Geasessharp	Кировский
	Абрам	Иванов	Николаевич	Ceasedquick	Калининский
	Абрам	Калинин	Вадимович	Fgustomanly	Киевский
	Абрам	Калинин	Густавович	Rnurseexact	Буденовский
	Абрам	Калинин	Игнатович	Fprotogains	Калининский
	Абрам	Карпов	Артурович	Pjazzynurse	Калининский
	Абрам	Карпов	Богданович	lindexdefix	Ворошиловский
	Абрам	Карпов	Вадимович	Jawardasset	Ворошиловский
	Абрам	Макаров	Игоревич	Sdefixindex	Киевский
	Абрам	Морозов	Абрамович	Wlightpeppy	Калининский
	Абрам	Морозов	Абрамович	Jexactexact	Пролетарский
	Лбозм	Monogon	Альбертовии	Zindevareet	Кировский

Рисунок Б.9 – Запрос 5

#### 2. заказчики с городами проживания и количеством заказов

select cust.id,
cust.first\_name as имя
,cust.second\_name as фамилия,
cust.third\_name as отчество,
c.city as город,
count(o.\*) as количество\_заказов
from customers as cust,
cities as c,
orders as o
where cust.city\_id=c.id and
o.customer\_id=cust.id
group by 1,2,3,4,5
order by 6 desc

	id	имя	фамилия	отчество	город	количество_заказ
<b>•</b>	315	Игорь	Петров	Михаилович	Макеевка	18
	1438	Густав	Осипов	Назарович	Макеевка	17
	1168	Гавриил	Шаров	Альбертович	Мариуполь	17
	997	Геннадий	Шаров	Захарович	Мариуполь	17
	1119	Михаил	Суворов	Густавович	Ростов	16
	385	Василий	Морозов	Захарович	Мариуполь	15
	1376	Артур	Ершов	Абрамович	Харцизск	15
	127	Вадим	Рябов	Гавриилович	Ростов	15
	236	Густав	Макаров	Михаилович	Москва	15
	1288	Михаил	Фролов	Игоревич	Мариуполь	14
	1426	Альберт	Макаров	Богданович	Москва	14
	746	Вадим	Рябов	Маратович	Москва	14
	110	Николай	Петухов	Густавович	Харцизск	14
	918	Егор	Федоров	Карлович	Донецк	13
	575	Игорь	Смирнов	Густавович	Москва	13
	847	Назар	Панов	Игоревич	Харцизск	13
	1081	Гистав	Fours	Fronceidu	Ростов	13

Рисунок Б.10 – Запрос 6

#### 3. банковские номера с банком и количеством заказов оплаченных ими

select
a.account as банковский\_счет,
b.bank as банк,
count(o.\*) as количество\_заказов
from accounts as a,
banks as b,
orders as o
where a.account=o.account and
a.bank\_id = b.id
group by 1,2
order by 3 desc

	банковский счет	банк	количество_заказ
<b>—</b>	6067170401313	Рокет банк	19
<u></u>	7501469119589	Пумб	18
	/301403113303	ТІУМО	10
	3177883652220	Альфа банк	16
	4153225895864	Рокет банк	16
	6980356510233	Пумб	16
	5366513470948	Альфа банк	15
	4614965063195	Приват банк	14
	8232447509820	Приват банк	14
	4916933489822	Рокет банк	14
	1653972955181	Приват банк	14
	6970518621734	Альфа банк	14
	6436661849934	Пумб	14
	7800263028265	Пумб	13
	8606379410426	Альфа банк	13
	1667571363498	Рокет банк	13
	4026290581703	Приват банк	13
	C2CCEQ20C0740	An an 6 mm	12

Рисунок Б.11 – Запрос 7

## Левое внешнее соединение номера бановских счетов и заказы которые были оплачены ими

select
a.account as банковский\_счет,
o.id as номер\_заказа
from accounts as a
left join orders as o on o.account=a.account
order by 2 desc

	банковский_счет	номер_заказа
<b>•</b>	6181421612382	
	7858163834632	10010
	8978135141644	10009
	7800263028265	10008
	7085085322233	10007
	2324229971203	10006
	7468249077833	10005
	9912581351970	10004
	5469983756661	10003
	5358261519981	10002
	2128143730305	10001
	5996411561682	10000
	7078749918702	9999
	3856579271611	9998
	3985083776192	9997
	6653467529539	9996
	8757267986316 E 12	9995

Рисунок Б.12 – Запрос 8

### Правое внешнее соединение

#### все города и заказчики в городах

select c.id,c.city as Город, cust.first\_name as имя, cust.second\_name as фамилия, cust.third\_name as отчество from customers as cust right join cities as c on c.id=cust.city\_id order by 1 desc

	id	Город	имя	фамилия	отчество
<b>•</b>	7	Вашингтон			
	6	Мариуполь	Густав	Петухов	Игоревич
	6	Мариуполь	Заур	Федоров	Михаилович
	6	Мариуполь	Илья	Тарасов	Игоревич
	6	Мариуполь	Богдан	Макаров	Маратович
	6	Мариуполь	Назар	Фролов	Васильевич
	6	Мариуполь	Артур	Зайцев	Егорович
	6	Мариуполь	Богдан	Рябов	Жерарович
	6	Мариуполь	Захар	Бобров	Васильевич
	6	Мариуполь	Жерар	Зайцев	Заурович
	6	Мариуполь	Абрам	Смирнов	Геннадиевич
	6	Мариуполь	Василий	Смирнов	Михаилович
	6	Мариуполь	Виктор	Калинин	Альбертович
	6	Мариуполь	Артур	Фролов	Иванович
	6	Мариуполь	Борис	Орлов	Вадимович
	6	Мариуполь	Гавриил	Калинин	Маратович
	6	Мариуроль	Виктор	Смирнов	Викторовии

Рисунок Б.13 – Запрос 9

Запрос на запросе по принципу левого соединения информация о банковских счетах, которые включают в себя

последовательность цифр 1234

select aa.account,

b.bank

from (select account,bank\_id from accounts where account like '%1234%') as aa left join banks as b on aa.bank\_id=b.id

	account	bank
<b>&gt;</b>	1234255552341	Рокет банк
	1234123417341	Приват банк
	1234123412341	Тим
	1912346562500	Рокет банк
	5460123464801	Приват банк
	8390192885864	Рокет банк
*		

Рисунок Б.14 – Запрос 10

Итоговый запрос без условия Общая стоимость заказов каждого заказчика select cust.id as номер\_заказчика, cust.first\_name as имя, cust.second\_name as фамилия,

cust.third\_name as отчество, sum(o.cost\*o.calculation) as денег\_потрачено from customers as cust, orders as o where o.customer\_id=cust.id group by 1,2,3,4

order by 5

	_				
	id	имя	фамилия	отчество	денег_потрачено
•	986	Артур	Фролов	Густавович	84348
	204	Иван	Зайцев	Артурович	2152500
	588	Гавриил	Смирнов	Маратович	2354345
	1059	Василий	Рябов	Карлович	2684736
	951	Гавриил	Федоров	Карлович	2737202
	634	Абрам	Рябов	Назарович	3545705
	1141	Заур	Калинин	Борисович	3585969
	389	Артур	Суворов	Михаилович	3891052
	1314	Борис	Калинин	Егорович	4242640
	162	Василий	Тарасов	Михаилович	4957027
	430	Заур	Суворов	Захарович	5015008
	384	Артур	Петухов	Вадимович	5060790
	207	Михаил	Рябов	Гавриилович	6306124
	969	Иван	Петров	Маратович	6535324
	1211	Борис	Зайцев	Михаилович	6885721
	1145	Игорь	Орлов	Николаевич	6907508
	1/11/	Генналий	Tapacoe	Жераровии	7328442

Рисунок Б.15 – Запрос 11

Итоговый запрос с условием на данные Количество работников задействованных в типографиях данного

#### <u>РАЙОНА</u>

select d.district as район,
count (w.\*) as работников
from prints as p,
workers as w,
districts as d
where d.id=p.district\_id and

p.id=w.print\_id and d.district=<u>PAЙOH</u> group by 1 order by 2

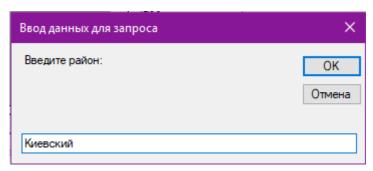


Рисунок Б.16 – Ввод данных для запроса 12

результат запроса с параметром: Киевский

	район	работников
<b>&gt;</b>	Киевский	137
*		

Рисунок Б.17 – Запрос 12

Итоговый запрос с условием на группы Типографии, количество заказов в которых больше **ЧИСЛА** select p.id as номер\_типографии, p.print as типография, count(o.\*) as количество\_заказов from prints as p, workers as w, orders as o where p.id=w.print\_id and o.worker\_id=w.id group by 1,2 having(count(o.\*)>**ЧИСЛО**) order by 3

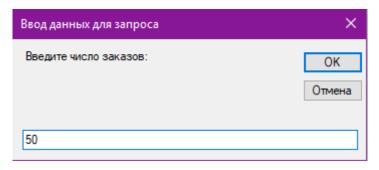


Рисунок Б.18 – Ввод данных для запроса 13

#### Результат запроса с параметром 50

	номер_типогра	афи типография	количество_заказ
•	97	Ysharkindex	51
	139	Mprotogains	53
	176	Cawardcivil	53
	103	Ceasedquick	54
	84	Moh myyummy	54
	179	Rmanlycomic	55
	175	Qcleanproto	55
	115	Lexacteases	56
	83	Jexactdefix	56
	60	Wavantgreet	56
	96	Rjazzyheart	56
	55	Fcleangusto	58
	172	Ycleanindex	58
	101	Xcomfyeases	59
	66	Neomfycomic	59
	89	Jlovermanly	59
	162	Renicumomy	60

Рисунок Б.19 – Запрос 13

Итоговый запрос с условием на данные и на группы Банковские счета **БАНКА**, на которых денег потрачено больше **ЧИСЛА** select a.account as банковский\_счет, sum(o.cost\*o.calculation) as денег\_потрачено from accounts as a,

banks as b,

orders as o

where o.account=a.account and

a.bank\_id = b.id and

b.bank=**<u>BAHK</u>** 

group by 1

having (sum(o.cost\*o.calculation)><u>ЧИСЛО</u>)

order by 2

запрос с параметрами Пумб, 100000000

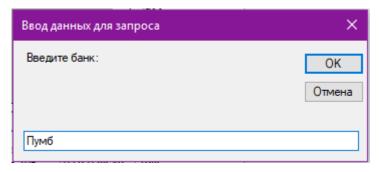


Рисунок Б.20 – Ввод данных для запроса 14

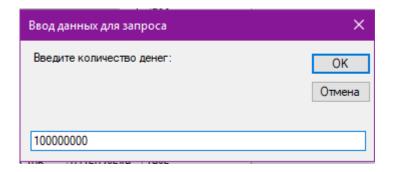


Рисунок Б.21 – Ввод данных для запроса 14

	банковский_счет	денег_потрачено
<b>&gt;</b>	9516377743726	101225124
	6880098974923	101381123
	1521953819950	101389732
	8082929674181	101830417
	8327811340636	103205484
	5045328127556	103912799
	3350285849499	103913928
	3463323774400	104269135
	7531320265606	104414857
	5297225317761	104498539
	7737593496346	105253156
	3080268871155	105346227
	4781557000749	105636201
	5377231844172	106130566
	4092784916294	106412174
	7100516001292	106799063
	2182483437620	106839617

Рисунок Б.22 – Запрос 14

```
Запрос на запросе по принципу итогового запроса Количество работников и количество заказов всех типографий select p.id ав номер_типографии, p.print as типография , count (w.*) ав количество_работников, p.возраст, p.год_основания from workers as w, (select id,print,year as год_основания,(extract(year from current_date)-year) as возраст from prints where (extract(year from current_date)-year>=30)) as p where p.id=w.print_id group by 1,2,4,5 order by 4 desc,3
```

	номер_типографи	типография	количество_работ	количество_заказ
•	34	Oloveravant	14	139
	94	Ecleanso-so	14	133
	153	Morderquick	12	121
	91	Rroomyprime	10	113
	99	Vexactoh my	11	110
	85	Kexactso-so	12	110
	173	Hoomicroomy	9	108
	72	Clightroomy	9	108
	51	Hnoblegains	10	108
	123	Cslickaward	10	108
	102	Olovedavant	10	107
	105	Teasesgusto	11	105
	104	lindexdefix	10	104
	42	Qspicypeace	10	103
	76	Uwowedsaucy	10	102
	119	Znoblewowed	9	101
	54	Icomfugueto	q	100

Рисунок Б.23 – Запрос 15

#### Запрос с подзапросом

Количество работников, которые работают в типографиях, существующих более 30 лет select p.id as номер\_типографии,

p.print as типография,

count (w.\*) as количество работников,

р.возраст,

р.год\_основания

from workers as w,

(select id,print,year as год\_основания,(extract(year from current\_date)-year) as возраст from prints where (extract(year from current\_date)-year>=30)) as p

where p.id=w.print\_id

group by 1,2,4,5

order by 4 desc,3

	номер_типографи	типография	количество_работ	возраст	год_основания
•	134	Rnurseexact	3	39	1981
	133	Hprotonurse	4	39	1981
	84	Moh myyummy	5	39	1981
	128	Vprimeasset	5	39	1981
	118	Rsharkindex	7	39	1981
	144	Jnobleeased	7	39	1981
	122	Fgustomanly	7	39	1981
	87	Ulightlover	5	38	1982
	161	Osharklight	7	38	1982
	177	Tcomicroomy	7	38	1982
	163	Gcomicmanly	8	38	1982
	116	Ysharkvigor	8	38	1982
	160	Qlovedready	9	38	1982
	51	Hnoblegains	10	38	1982
	62	Bexactexact	3	37	1983
	83	Jexactdefix	5	37	1983
	129	Oeasedob my	7	37	1983

Рисунок Б.24 – Запрос 16

#### ПРИЛОЖЕНИЕ В РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При запуске пользователю необходимо авторизоваться.

После этого пользователь попадает на главную форму.

Для того, чтобы вывести любую таблицу на главную форму нужно нажать пункт меню таблицы и выбрать нужную таблицу в выпадающем списке.

Для того, чтобы вывести любой справочник на главную форму нужно нажать пункт меню справочники и выбрать нужный справочник в выпадающем списке.

Для того чтобы увидеть результаты запросов нужно нажать пункт меню запросы. В открывшейся форме выбрать запрос и нажать кнопку выполнить.

Для того, чтобы сгенерировать записи в таблицу выберите любую таблицу(не справочник) и нажмите кнопку генерация. Введите нужное число и подтвердите действие.

Для того чтобы сделать html публикацию нужно нажать на пункт меню html публикация и выбрать таблицу в выпадающем списке

Для того чтобы посмотреть на диаграммы нажмите на пункт меню диаграммы. И выберите из выпадающего списка.

Для того чтобы экспортировать запрос в Excel нужно нажать на пункт меню отчет и выбрать запрос в выпадающем списке

Для того чтобы добавить запись нужно выбрать таблицу и нажать кнопку добавить. После этого заполнить поля и нажать кнопку добавить.

Для того чтобы изменить запись нужно выбрать запись нажать кнопку изменить и изменить поля с данными на те, что вы хотите. После нажмите кнопку изменить.

Для того, чтобы выполнить поиск нужно нажать на кнопку поиск. Откроется форма поиска. В ней выберите нужное поле и введите значение для поиска

Для удаления записи выберите режим удаления. При текущей записи вы удалите выделенную запись. При номере записи вы ужалите запись с номером указанным в поле ввода. При записи со значением вы удалите поля со значениями равными указанным.

#### ПРИЛОЖЕНИЕ Г ТЕКСТ ПРОГРАММЫ

```
case "Таблица":
using Npgsql;
using System;
                                                                {
using System.Collections.Generic;
                                                                  return true;
using System.ComponentModel;
                                                                  break;
using System.Data;
using System.Drawing;
                                                             case "Справочник":
using System.Ling;
                                                                  return false;
using System. Text;
using System. Threading. Tasks;
                                                                  break;
using System. Windows. Forms;
using System.Reflection;
using Excel = Microsoft.Office.Interop.Excel;
using System.Web.UI;
using System.Web.UI.HtmlControls;
                                                           return true;
using System.IO;
using System. Diagnostics;
namespace Prints_Of_Donets_KursovayaBD_
                                                         public string FindTable() //показывает
                                                    какая таблица показана пользователю
  public partial class MainForm: Form
                                                           string[] words = label1.Text.Split(' ');
                                                           switch (words[1])
    private string connstring =
String.Format("Server={0};Port={1};"
                                                             case "Типографии":
       + "User
Id={2};Password={3};Database={4}",
                                                                  return "prints";
"localhost", 5432, "postgres", "sava000",
                                                                  break;
"prints");
                                                             case "Заказы":
    public NpgsqlConnection conn;
    public string sql;
    public string selectSql;
                                                                  return "orders";
    public NpgsqlCommand cmd;
                                                                  break;
    public DataTable dt;
    private int rowIndex = -1;
                                                             case "Заказчики":
    public MainForm()
                                                                  return "customers";
       InitializeComponent();
                                                                  break;
                                                             case "Работники":
     }
                                                                  return "workers";
    public bool
                                                                  break;
FindTableType()//показывает тип таблицы
показанной пользователю true - таблица
                                                             case "Банковские":
false - справочник
                                                                  return "accounts";
       string[] words = label1.Text.Split(' ');
                                                                  break;
       switch (words[0])
                                                             case "Типы":
       {
```

"(select district from districts where

```
{
              return "types";
                                                   id=district id) as район, address as адрес,
                                                   phone as телефон, year as год_основания
              break;
                                                   from prints";
         case "Районы":
                                                          Select();
                                                          //select
              return "districts";
              break;
                                                          //вывод количества записей и
                                                   названия таблицы с которой работаем на
         case "Города":
                                                   форму
                                                          sql = @"select count(*) from prints";
              return "cities";
                                                          ChangeLabel("Таблица
                                                   Типографии");
              break;
                                                          //конец вывод количества записей и
         case "Банки":
                                                   названия таблицы с которой работаем на
                                                   форму
              return "banks";
                                                          //заполнение ComboBox
                                                          int i = 1:
              break;
                                                          cellComboBox.Items.Clear();
                                                          foreach (DataColumn item in
         case "Изделия":
                                                   dt.Columns)
              return "producttypes";
              break;
                                                            if (CheckBytea(i - 1))
         case "Виды":
                                                               cellComboBox.Items.Add(i + ") "
                                                   + item);
              return "papertypes";
              break;
                                                            i++;
         case "Форматы":
                                                          //заполнение ComboBox
              return "formats";
                                                        public void ChangeLabel(string
                                                   tableName)//записывает с какой таблицей
              break:
                                                   работаем и сколько в ней записей
                                                          try
       return "";
                                                            conn.Open();
                                                            cmd = new NpgsqlCommand(sql,
                                                   conn);
                                                            label1.Text = tableName +"
    private void Form1_Load(object sender,
EventArgs e)//при загрузке главной формы
                                                   (записей: " + cmd.ExecuteScalar() + ")";
                                                            conn.Close();
     {
       conn = new
NpgsqlConnection(connstring);//присоедине
                                                          catch (Exception ex)
ние к БД
                                                            conn.Close();
                                                            MessageBox.Show("Ошибка при
       //select
       selectSql = @ "select id,print as
                                                   вычислении количества записей. Error: " +
типография,(select type from types where
                                                   ex.Message);
id=type_id) as тип_собственности," +
                                                          }
```

```
}
                                                   + ex.Message);
    public void Select()
                                                                  return "print";
                                                                  case "тип_собственности":
       try
                                                               return "type_id";
         conn.Open();
                                                             case "год_основания":
                                                               return "year";
         cmd = new
NpgsqlCommand(selectSql, conn);
                                                             default:
         dt = new DataTable();
                                                               break:
         dt.Load(cmd.ExecuteReader());
                                                           return column;
         conn.Close();
                                                        public bool CheckBytea(int i)//проверка
         dgvData.DataSource = null;
         dgvData.DataSource = dt;
                                                   типа столбца на bytea
       catch(Exception ex)
                                                           switch
                                                   (dt.Columns[i].DataType.ToString())
         conn.Close();
         MessageBox.Show("Error: " +
                                                             case "System.Int64":
ex.Message);
                                                                  break;
    public string
NameConverterForDelete(string column)
                                                             case "System.Int32":
       switch (column)
                                                                  break;
         case "типография":
                                                             case "System.String":
              try
                conn.Open();
                                                                  break;
                cmd = new
                                                             case "System.DateTime":
NpgsqlCommand(@"select count(*) filter
                                                               {
(where print_id="+""+") from workers ",
conn);
                                                                  break;
                                                             default:
                conn.Close();
                                                                  return false;
              catch (Exception ex)
                                                               break;
                conn.Close();
                                                           return true;
                MessageBox.Show("Error: "
```

```
public string CheckComboBox(int
                                                   нажатии на строку в таблице
і)//проверка типа вводимого значения для
ComboBox на MainForm
                                                          if (e.RowIndex >= 0)
       //valueTextBox.Text =
dt.Columns[i].DataType.ToString();
                                                            rowIndex = e.RowIndex;
       switch
(dt.Columns[i].DataType.ToString())
                                                            //nameTextBox.Text =
                                                   dgvData.Rows[e.RowIndex].Cells["name"].V
         case "System.Int64":
                                                   alue.ToString();
                                                          }
              return valueTextBox.Text;
              break;
                                                       private void insertButton_Click(object
         case "System.Int32":
                                                   sender, EventArgs e)
                                                          PrintsAddForm addForm = new
              return valueTextBox.Text;
              break;
                                                   PrintsAddForm();
                                                          addForm.Visible = true;
           }
         case "System.String":
                                                          //int result = 0;
                                                         //rowIndex = -1;
                                                         //try
              return ""+
                                                         //{
valueTextBox.Text+"";
                                                          // conn.Open();
                                                         // sql = @"select * from
              break;
                                                   print_insert(:_name)";
         case "System.DateTime":
                                                         // cmd = new NpgsqlCommand(sql,
                                                   conn);
                                                          // //
              return """ + valueTextBox.Text
                                                   cmd.Parameters.AddWithValue("_name",
                                                   nameTextBox.Text);
                                                         // result = (int)cmd.ExecuteScalar();
              break;
                                                         // conn.Close();
         default:
                                                         // if (result == 1)
                                                          //
                                                         //
              valueTextBox.Text =
                                                                MessageBox.Show("Inserted
dt.Columns[i].DataType.ToString();
                                                   succesfully");
              return "error";
                                                                Select();
                                                         //
                                                          //
           break;
                                                         // else
                                                         //
                                                                MessageBox.Show("Inserted
                                                   fail");
       //return
dt.Columns[i].DataType.ToString();
                                                         //
                                                            }
    private void dgvData_CellClick(object
                                                          //}
sender, DataGridViewCellEventArgs e)//при
                                                          //catch (Exception ex)
```

```
//{
                                                   Error: " + ex.Message);
       // conn.Close();
      // MessageBox.Show("Inserted fail.
                                                          //result = 0;
Error: " + ex.Message);
      //}
                                                        private void deleteButton_Click(object
     }
                                                   sender, EventArgs e)
    private void updateButton_Click(object
sender, EventArgs e)
                                                          if (rowIndex < 0)
       //int result = 0;
                                                            MessageBox.Show("Выберите
                                                   строку для удаления");
       //if (rowIndex < 0)
                                                            return;
       // MessageBox.Show("Please choose
                                                          }
student to update");
                                                          try
      // return;
       //}
                                                            conn.Open();
       //try
                                                            sql = @"select * from
                                                   st_delete(:_id)";
       //{
       // conn.Open();
                                                            cmd = new NpgsqlCommand(sql,
      // sql = @"select * from
                                                   conn);
print_update(:_id,:_name)";
       // cmd = new NpgsqlCommand(sql,
                                                   cmd.Parameters.AddWithValue("_id",
                                                   int.Parse(dgvData.Rows[rowIndex].Cells["id"
conn);
                                                   ].Value.ToString()));
      //
cmd.Parameters.AddWithValue("_id",
                                                            if ((int)cmd.ExecuteScalar() == 1)
int.Parse(dgvData.Rows[rowIndex].Cells["id"
].Value.ToString()));
                                                               MessageBox.Show("Удалено
       //
                                                   успешно");
//cmd.Parameters.AddWithValue("_name",
                                                              rowIndex = -1;
nameTextBox.Text);
       // result = (int)cmd.ExecuteScalar();
       // conn.Close();
                                                            conn.Close();
                                                            Select();
       // if (result == 1)
                                                          catch (Exception ex)
       //
       //
             MessageBox.Show("Updated
succesfully");
                                                            conn.Close();
                                                            MessageBox.Show("Ошибка при
       //
             Select();
                                                   удалении. Error: " + ex.Message);
       //
       //
          else
      //
                                                        }
             MessageBox.Show("Updated
fail");
      // }
                                                        private void
                                                   MainForm_FormClosed(object sender,
       //}
                                                   FormClosedEventArgs e)//при закрытии
       //catch (Exception ex)
                                                   главной формы
       //{
       // conn.Close();
                                                        {
          MessageBox.Show("Update fail.
                                                          Application.Exit();
```

```
}
                                                          }
    private void
buttonSearch_Click_1(object sender,
EventArgs e)
       string[] words = label1.Text.Split(' ');
                                                        private void
       SearchForm searchForm;
                                                   типографииToolStripMenuItem_Click(object
                                                   sender, EventArgs e)
       switch (words[1])
         case "Типографии":
                                                          //select
                                                          selectSql = @"select id,print as
                                                   типография, (select type from types where
                                                   id=type_id) as тип_собственности,"+
              searchForm = new
                                                             "(select district from districts where
SearchForm(dt,cellComboBox.Items,
FindTable(),label1.Text,dgvData.Columns);
                                                   id=district_id) as район, address as адрес,
              searchForm.Show();
                                                   phone as телефон, year as год_основания
              break;
                                                   from prints";
                                                          Select();
         case "Заказы":
                                                          //select
              searchForm = new
                                                          //count
SearchForm(dt, cellComboBox.Items,
                                                          sql = @"select count(*) from prints";
FindTable(), label1.Text, dgvData.Columns);
                                                          ChangeLabel("Таблица
              searchForm.Show();
                                                   Типографии");
                                                          //count
              break;
         case "Заказчики":
                                                          //заполнение ComboBox
                                                          int i = 1:
              searchForm = new
                                                          cellComboBox.Items.Clear();
SearchForm(dt, cellComboBox.Items,
                                                          foreach (DataColumn item in
FindTable(), label1.Text, dgvData.Columns);
                                                   dt.Columns)
              searchForm.Show();
              break;
                                                            if (CheckBytea(i-1))
         case "Работники":
                                                               cellComboBox.Items.Add(i + ") "
                                                   + item);
              searchForm = new
SearchForm(dt,
                                                            i++;
cellComboBox.Items,FindTable(),
label1.Text, dgvData.Columns);
                                                          //конец заполнение ComboBox
              searchForm.Show();
              break;
                                                        }
                                                        private void
         default:
                                                   заказыToolStripMenuItem_Click(object
                                                   sender, EventArgs e)
              MessageBox.Show("Поиск
                                                        {
выполняется только для таблиц");
                                                          //select
              break;
                                                          selectSql = @"select id, customer_id
                                                   as индекс_заказчика, (select producttype
            }
```

" publication as название_издания, picture as индежс_работника, price as индежс_работник, price as understand as yellowns, private void runsicofect sender, EventArgs e) {  // salonomentus_dakar, cost as inpedonara, comment as yellowns, private void  runsicofect sender, EventArgs e)  // select // select // select // select count(*) from orders", changeLabel("Cnpabountury Tunis) // count // salonomentus_dakar, cost as inpedonara, comment as yellowns, private void  runsicofect sender, EventArgs e)  // select // select // select // select count(*) from types";  Select(s); // select // select select select id, producttype as understand under	from producttypes where id=producttype_id)	Select();
название_издания_picture as   rmryльная_сграница, worker_id as   sql = @"select count(*) from   sql = @"select count(*) from orders';   changeLabel("Cnpaboчник   selectSql = @"select count(*) from   sql = @"select count(*) from orders';   changeLabel("Cnpaboчник   sql = @"select count(*) from orders';   changeLabel("Cnpabovnuk   sql = @"select count(*) from types";   selectSql = @	as изделие," +	//select
яд  = @"select count(*) from districts";  спара жажампара, "+	" publication as	<pre>cellComboBox.Items.Clear();</pre>
видае работника, price as     districts";       поста разахимлара, " + "calculation as тираж, count as количество_листов, (select format from formats where id=format_id) as формат, (select papertype from papertypes where id = papertype_id) as вид_бумати" + ", datastart as дата_выполнения_план, datafact as дата_выполнения_план, cost as предоплата, comment as доп_сведения" + ", account as foankobsckий_nowep from orders limit 1000"; Select();	название_издания, picture as	//count
пцена_эжэемпляра, "-k" calculation as тираж, count as количество_листов, (select format from formats where id=format_id) as формат, (select papertype from papertypes where id = papertype_id) as вид_бумаги" + "., datastart as дата_выполнения_план, datafact as дата_выполнения_план, datafact as дата_выполнения_план, datafact as ".account as банковский_помер from orders limit 1000"; Select(); //select select count(*) from orders"; ChangeLabel("Ta6лина Заказы"); //count sql = @"select count(*) from orders"; ChangeLabel("Ta6лина Заказы"); //count sql = @"select count(*) from types"; ChangeLabel("Cправочник Типы cofcrвенности"); //count sql = @"select count(*) from types"; ChangeLabel("Cправочник Типы cofcrвенности"); //count sql = @"select count(*) from types"; Select(); //select cellComboBox.Items.Clear(); //count sql = @"select count(*) from types"; Select(); //select cellComboBox.Items.Clear(); //count sql = @"select count(*) from types"; Select(); //count sql = @"select count(*) from types sql = @"select count(*) from types sql = @"select co	титульная_страница, worker_id as	sql = @"select count(*) from
"calculation as тираж, count as количество_листов, (select format from formats where id=format; id) as бромат, (select papertype from papertypes where id = papertype_id) as вид_бумати" + ". datastart as дата_принятия, dataplan as дата_выполнения_план, datafact as дата_выполнения_план, datafact as дата_выполнения_план, datafact as дата_выполнения_план, cost as предоплата, comment as доп_сведения" + ".account as банковский_номер from orders limit 1000"; Select();	индекс_работника, price as	districts";
count as количество_листов, (select format from formats where id—format_id) as формат, (select papertype from papertypes where id = papertype_id) as вид_бумати" +	цена_экземпляра, " +	ChangeLabel("Справочник
from formats where id=format_id) as формат, (select papertype from papertypes where id = papertype_id) as вид_бумаги" + ", datastart as дата_принятия, dataplan as дата_выполнения_пан, datafact as дата_выполнения_пан, datafact as ", account as gl = @"select count(*) from orders'; ChangeLabel("Taблица Заказы"); // count	"calculation as тираж,	Районы");
from formats where id=format_id) as формат, (select papertype from papertypes where id = papertype_id) as вид_бумаги" + ", datastart as дата_принятия, dataplan as дата_выполнения_пан, datafact as дата_выполнения_пан, datafact as ", account as gl = @"select count(*) from orders'; ChangeLabel("Taблица Заказы"); // count	count as количество_листов, (select format	//count
(select papertype from papertypes where id = papertype_id) as вид_бумаги" + ", datastart as дата_принятия, dataplan as дата_выполнения_план, datafact as дата_выполнения_факт, cost as предоплата, comment as доп_сведения" + ", account as банковский_номер from orders limit 1000"; Select(); //select select(); //select (cellComboBox.ltems.Clear(); //count sql = @"select count(*) from orders"; ChangeLabel("Taблица Заказы"); //count sql = @"select ComboBox int i = 1; cellComboBox.ltems.Clear(); foreach (DataColumn item in dt.Columns) { // sanonnenue ComboBox for cellComboBox.ltems.Add(i + ") " + item); // select selectSql = @"select id,type as Tun_Co6crвенности from types"; Select(); //select cellComboBox.ltems.Clear(); //count sql = @"select count(*) from types"; ChangeLabel("Cправочник Типы co6crвенности"); //count sql = @"select count(*) from types"; Select(); //select selectSql = @"select id,producttype as UзделияТооlStripMenuItem_Click(object sender, EventArgs e) { // select selectSql = @"select id,producttype as UзделияТооlStripMenuItem_Click(object sender, EventArgs e) { // select cellComboBox.ltems.Clear(); // count sql = @"select count(*) from producttypes"; ChangeLabel("Cправочник Uзделия"); // count sql = @"select id,producttype as UзделияТооlStripMenuItem_Click(object sender, EventArgs e) { // select cellComboBox.ltems.Clear(); // count sql = @"select count(*) from producttypes"; ChangeLabel("Cправочник Uзделия"); // count sql = @"select id,producttype as Uзделия from producttypes"; ChangeLabel("Cправочник Usdentypes"; ChangeLabel("Cправочник Usdentypes"; ChangeLabel("Cnpasovnuk Usdentypes Usdentypes Usdentypes Usdentypes Usdentypes Usdentype		}
papertype_id) as вил_бумаги" +		,
", dafastart as дата_принятия, dataplan as дата_выполнения_план, datafact as дата_выполнения_план, datafact as дата_выполнения_план, datafact as дата_выполнения_факт, cost as предоплата, comment as доп_сведения" + ", account as Gahkobckuil_home from orders limit 1000";		private void
дата_принятия, dataplan as дата_выполнения_план, datafact as дата_выполнения_факт, cost as предоплата, comment as доп_сведения" +		<u>=</u>
дата_выполнения_план, datafact as дата_выполнения_факт, cost as предоплата, comment as доп_сведения" +	•	<u> </u>
дата_выполнения_факт, cost as предоплата, comment as доп_евсдения" +		{
comment as доп_сведения" +		//select
",account as банковский_номер from orders limit 1000"; Select(); Select  //select  //count sql = @"select count(*) from orders"; ChangeLabel("Tаблица Заказы"); //count  //saполнение ComboBox int i = 1; cellComboBox.Items.Clear(); foreach (DataColumn item in  dt.Columns)  {     if (CheckBytea(i-1))     {         if (CheckBytea(i-1))         }         i++;         i++;         }         it++;     }         ip-         inthick the private void  //saполнение ComboBox int i = 1; cellComboBox.Items.Add(i + ") "  + item);  // select selectSql = @"select id,producttype as Usageлия from producttypes"; Select(); //select selectSql = @"select count(*) from types"; ChangeLabel("Cправочник Типы coocriterinoctru"); //count //count //count sql = @"select id,producttype as Usagenue from producttypes"; Select(); //count //count //count //count //select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //select //select cellComboBox.Items.Clear(); //count //select sql = @"select count(*) from //sanonhehue ComboBox //select cellComboBox.Items.Clear(); //select selectSql = @"select id,producttype as Usagenue from producttypes"; ChangeLabel("Справочник //select cellComboBox.Items.Clear(); //count //select sql = @"select count(*) from //select cellComboBox.Items.Clear(); //select cellComboBo		
банковский_номер from orders limit 1000"; Select(); //selectSelect(); //select cellComboBox.Items.Clear(); //count sql = @"select count(*) from orders"; ChangeLabel("Tаблица Заказы"); //countchangeLabel("Cправочник Типы coбственности"); //count//заполнение ComboBox int i = 1; cellComboBox.Items.Clear(); foreach (DataColumn item in dt.Columns)private void изделияToolStripMenuItem_Click(object selectSql = @"select id,producttype as Изделия from producttypes"; Select(); //count//select cellComboBox.Items.Add(i + ") " + item);//select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count } ChangeLabel("Справочник Изделия"); //count ) //select private void ropoдaToolStripMenuItem_Click(object sender, EventArgs e)		*
Select();//select//select//selectcellComboBox.Items.Clear();//countsql = @"select count(*) from orders";ChangeLabel("Справочник ТипыChangeLabel("Таблица Заказы");coбственности");//count}//заполнение ComboBoxprivate voidint i = 1;private voidcellComboBox.Items.Clear();uajæлияToolStripMenuItem_Click(objectforeach (DataColumn item inselectSql = @"select id,producttype asdt.Columns){{//selectecllComboBox.Items.Add(i + ") "selectSql = @"select id,producttype asHajæлия from producttypes";Select();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear();//selectcellComboBox.Items.Clear(); <td></td> <td>· · · · · · · · · · · · · · · · · · ·</td>		· · · · · · · · · · · · · · · · · · ·
//select  //count  sql = @"select count(*) from orders"; ChangeLabel("Таблица Заказы"); //count  //заполнение ComboBox int i = 1; cellComboBox.Items.Clear(); foreach (DataColumn item in dt.Columns)  {     if (CheckBytea(i-1))     {         cellComboBox.Items.Add(i + ") "     + item);         cellComboBox     }      //заполнение ComboBox      if (CheckBytea(i-1))     {         cellComboBox.Items.Add(i + ") "         + item);         cellComboBox.Items.Add(i + ") "         private void         producttypes";         cellComboBox.Items.Clear();         //select         cellComboBox.Items.Clear();         //sel		<b>V</b> ,
//count sql = @"select count(*) from orders"; ChangeLabel("Таблица Заказы"); //count sql = @"select count(*) from types"; ChangeLabel("Справочник Типы собственности"); //count }  //заполнение ComboBox int i = 1; cellComboBox.Items.Clear(); foreach (DataColumn item in dt.Columns) {  if (CheckBytea(i-1)) {  if (CheckBytea(i-1)) {  it++; select cellComboBox.Items.Add(i + ") "  + item); Select(); //select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //select cellComboBox.Items.Clear(); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count sql = @"select count(*) from producttypes"; //count sql = @"select count(*) from producttypes (*) from p		
//countsql = @"select count(*) from orders"; ChangeLabel("Таблица Заказы"); //countsql = @"select count(*) from types"; ChangeLabel("Справочник Типы coбственности"); //count//заполнение ComboBox int i = 1; cellComboBox.Items.Clear(); foreach (DataColumn item in dt.Columns)private void изделия ToolStripMenuItem_Click(object sender, EventArgs e)(**GelComboBox.Items.Add(i + ") " + item);{ //select select(t); //select cellComboBox.Items.Add(i + ") " //select \$ //заполнение ComboBox**GelComboBox.Items.Clear(); //count sql = @"select count(*) from producttypes"; Select(t); //select cellComboBox.Items.Clear(); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count } //count } //sanonhehue ComboBox***Private void païoныToolStripMenuItem_Click(object sender, EventArgs e)private void ropoдaToolStripMenuItem_Click(object sender, EventArgs e)	//Scicct	
sql = @"select count(*) from orders"; ChangeLabel("Таблица Заказы"); //count //заполнение ComboBox int i = 1; cellComboBox.Items.Clear(); foreach (DataColumn item in dt.Columns) {     if (CheckBytea(i-1))     {         cellComboBox.Items.Add(i + ") "     + item);     //saполнение ComboBox      int i = 1;     cellComboBox.Items.Clear();     if (CheckBytea(i-1))     {	//aount	
ChangeLabel("Таблица Заказы"); //countсобственности"); //count//заполнение ComboBox int i = 1; cellComboBox.Items.Clear(); foreach (DataColumn item inprivate void изделияТоolStripMenuItem_Click(object sender, EventArgs e)dt.Columns) { if (CheckBytea(i-1)) { cellComboBox.Items.Add(i + ") " + item);{ //select selectSql = @"select id,producttype as Изделие from producttypes"; Select(); //select cellComboBox.Items.Clear(); //count sql = @"select count(*) from} //заполнение ComboBoxproducttypes"; ChangeLabel("Справочник Изделия"); //count } //count } Private void ropoдaToolStripMenuItem_Click(object sender, EventArgs e){ //select selectSql = @"select id,district asprivate void ropoдaToolStripMenuItem_Click(object sender, EventArgs e)		
//count //cou		• • •
//заполнение ComboBox int i = 1;		7°
int i = 1; cellComboBox.Items.Clear(); foreach (DataColumn item in  dt.Columns)  {     if (CheckBytea(i-1))	//count	//COUIIL
int i = 1; cellComboBox.Items.Clear(); foreach (DataColumn item in  dt.Columns)  {     if (CheckBytea(i-1))	//	}
cellComboBox.Items.Clear(); изделияТoolStripMenuItem_Click(object foreach (DataColumn item in dt.Columns) {     {		• , • • •
foreach (DataColumn item in dt.Columns) {		•
dt.Columns) {		<del>_</del>
{	`	sender, EventArgs e)
if (CheckBytea(i-1))	•	{
{ Изделие from producttypes"; cellComboBox.Items.Add(i + ") " Select();   + item);		
cellComboBox.Items.Add(i + ") " Select(); + item);	if (CheckBytea(i-1))	
+ item);//select}cellComboBox.Items.Clear();i++;//count}sql = @"select count(*) from//заполнение ComboBoxproducttypes";ChangeLabel("Справочник}Изделия");//count}районы ToolStripMenuItem_Click(objectsender, EventArgs e)private void{ropoдaToolStripMenuItem_Click(object//selectsender, EventArgs e)selectSql = @"select id,district as{	{	
} i++; 3 //sаполнение ComboBoxcellComboBox.Items.Clear(); //count sql = @"select count(*) from producttypes"; ChangeLabel("Справочник} Изделия"); //count private void paйоныToolStripMenuItem_Click(object sender, EventArgs e)В private void ropoдaToolStripMenuItem_Click(object sender, EventArgs e){ //select selectSql = @"select id,district asFor podaToolStripMenuItem_Click(object sender, EventArgs e)	cellComboBox.Items.Add(i + ") "	
i++;//count}sql = @"select count(*) from//заполнение ComboBoxproducttypes";ChangeLabel("Справочник}Изделия");//count//countprivate void}paйоны ToolStripMenuItem_Click(objectsender, EventArgs e)private void{ropoдa ToolStripMenuItem_Click(object//selectsender, EventArgs e)selectSql = @"select id,district as{	+ item);	
\$\text{sql} = @"select count(*) from producttypes"; ChangeLabel("Справочник Изделия"); //count private void \$\text{paŭohisToolStripMenuItem_Click(object sender, EventArgs e)} private void {\text{private void ropoдaToolStripMenuItem_Click(object sender, EventArgs e)} }\text{private void ropodaToolStripMenuItem_Click(object sender, EventArgs e)} \$\text{select sender, EventArgs e}\$ \$\text{sender, EventArgs e}\$ \$sen	}	<pre>cellComboBox.Items.Clear();</pre>
//заполнение ComboBox  producttypes";  ChangeLabel("Справочник  Изделия");  //count  private void  paйоныToolStripMenuItem_Click(object  sender, EventArgs e)  {  ropogaToolStripMenuItem_Click(object  select  //select  selectSql = @"select id,district as  producttypes";  ChangeLabel("Справочник  Изделия");  //count  private void  ropogaToolStripMenuItem_Click(object  sender, EventArgs e)  {	i++;	//count
ChangeLabel("Справочник  Изделия"); //count private void paйоныToolStripMenuItem_Click(object sender, EventArgs e)  {	}	sql = @"select count(*) from
} Изделия"); //count private void paйоныToolStripMenuItem_Click(object sender, EventArgs e) private void {	//заполнение ComboBox	producttypes";
//count private void paйоныToolStripMenuItem_Click(object sender, EventArgs e)  {		ChangeLabel("Справочник
private void } районыToolStripMenuItem_Click(object sender, EventArgs e) private void { ropoдaToolStripMenuItem_Click(object	}	Изделия");
районыToolStripMenuItem_Click(object sender, EventArgs e)  {		//count
sender, EventArgs e)private void{городаToolStripMenuItem_Click(object//selectsender, EventArgs e)selectSql = @"select id,district as{	private void	}
{ ropoдaToolStripMenuItem_Click(object sender, EventArgs e) selectSql = @"select id,district as {	районыToolStripMenuItem_Click(object	
{ ropoдaToolStripMenuItem_Click(object sender, EventArgs e) selectSql = @"select id,district as {	sender, EventArgs e)	private void
//select sender, EventArgs e) selectSql = @"select id,district as {	{	
selectSql = @"select id,district as {	//select	
		{
	Район from districts";	//select

selectSql = @"select id,city as I ород from cities";	Вид_бумаги,density from papertypes"; Select();
Select();	//select
//select	
	cellComboBox.Items.Clear();
cellComboBox.Items.Clear();	//count
//count	sql = @"select count(*) from
sql = @"select count(*) from cities";	papertypes";
ChangeLabel("Справочник Города");	ChangeLabel("Справочник Виды
//count	бумаги");
}	//count
	}
private void	•
банкиToolStripMenuItem_Click(object	private void
sender, EventArgs e)	форматы Tool Strip Menu Item_Click (object
{	sender, EventArgs e)
//select	sender, Eventrings ey
	//select
selectSql = @"select id,bank as Банк	
from banks";	selectSql = @"select id,format as
Select();	Формат from formats";
//select	Select();
cellComboBox.Items.Clear();	//select
//count	<pre>cellComboBox.Items.Clear();</pre>
<pre>sql = @"select count(*) from banks";</pre>	//count
ChangeLabel("Справочник Банки");	<pre>sql = @"select count(*) from</pre>
//count	formats";
}	ChangeLabel("Справочник
J	Форматы");
private void	//count
банковскиеНомераToolStripMenuItem_Clic	//count
<u>.</u>	}
k(object sender, EventArgs e)	• • •
{	private void
//select	заказчикиToolStripMenuItem_Click(object
selectSql = @"select * from	sender, EventArgs e)
accounts_select()";	{
Select();	//select
//select	selectSql = @"select id,first_name as
cellComboBox.Items.Clear();	имя, second_name as фамилия, third_name
//count	as отчество,(select city from cities where
sql = @"select count(*) from	id=city_id) as город," +
accounts";	"address as адрес,
ChangeLabel("Справочник	birthday as дата_рождения, phone as
Банковские номера");	телефон from customers";
± /*	<u> </u>
//count	Select();
}	//select
maissata ssaid	//2020#
private void	//count
видыБумагиToolStripMenuItem_Click(objec	sql = @"select count(*) from
t sender, EventArgs e)	customers";
{	ChangeLabel("Таблица Заказчики"):
//select	//count
selectSql = @"select id,papertype as	//заполнение ComboBox

```
int i = 1;
                                                        private void insertButton_Click_1(object
       cellComboBox.Items.Clear();
                                                   sender, EventArgs e)
       foreach (DataColumn item in
dt.Columns)
                                                          string[] words = label1.Text.Split(' ');
                                                          switch (words[1])
         if (CheckBytea(i - 1))
                                                            case "Типографии":
            cellComboBox.Items.Add(i + ") "
+ item);
                                                                 //return "prints";
                                                                 PrintsAddForm printsAdd =
                                                   new PrintsAddForm();
         i++:
                                                                 printsAdd.isUpdate = false;
                                                                 printsAdd.Visible = true;
       //конец заполнение ComboBox
     }
                                                   //MessageBox.Show(""+rowIndex);
    private void
                                                                 break;
работникиToolStripMenuItem_Click(object
sender, EventArgs e)
                                                            case "Заказы":
    {
                                                                 //return "orders";
       //select
       selectSql = @"select id, first_name as
                                                                 OrdersAddForm ordersAdd =
имя, second_name as фамилия, third_name
                                                   new OrdersAddForm();
as отчество,(select print from prints where id
                                                                 ordersAdd.isUpdate = false;
= print_id) as типография from workers";
                                                                 ordersAdd.Visible = true;
       Select();
                                                                 break;
       //select
                                                               }
                                                            case "Заказчики":
       //count
                                                                 //return "customers";
       sql = @"select count(*) from
workers";
       ChangeLabel("Таблица Работники");
                                                                 CustomersAddForm
                                                   customersAdd = new CustomersAddForm();
       //заполнение ComboBox
                                                                 customersAdd.isUpdate =
       int i = 1;
                                                   false;
       cellComboBox.Items.Clear();
                                                                 customersAdd.Visible = true;
       foreach (DataColumn item in
dt.Columns)
                                                                 break;
                                                               }
                                                            case "Работники":
         if (CheckBytea(i - 1))
           cellComboBox.Items.Add(i + ") "
                                                                 //return "workers";
+ item);
                                                                 WorkersAddForm workersAdd
                                                   = new WorkersAddForm();
                                                                 workersAdd.isUpdate = false;
         i++:
                                                                 workersAdd.Visible = true;
       //конец заполнение ComboBox
                                                                 break;
                                                            case "Банковские":
                                                                 //return "accounts";
```

	DirectoryAddForm	{		
directoryAdd	= new		//return "banks";	
•	lForm(true,true,false,"Банковск		DirectoryAddForm	
-	нк","", words[1],	directoryAdd = new DirectoryAddForm(true,		
FindTable(),0		false, false, "Банк", "", words[1],		
1 111010(),0	directoryAdd.isUpdate = false;	FindTable(), (		
		ring rabic(), (		
	directoryAdd.Visible = true;		directoryAdd.isUpdate = false;	
	directoryAdd.directory =		directoryAdd.Visible = true;	
FindTable();		<b>—</b> — 1 1 0	directoryAdd.directory =	
	break;	FindTable();		
}			break;	
case	е "Типы":	}		
{		case	е "Изделия":	
	//return "types";	{		
	DirectoryAddForm		//return "producttypes";	
directoryAdd	= new DirectoryAddForm(true,		DirectoryAddForm	
•	Гип собственности", "", "",	directorvAdd	= new DirectoryAddForm(true,	
	dTable(), 0, "");		Ізделие", "", "", words[1],	
words[1], 1 m	directoryAdd.isUpdate = false;	FindTable(), (		
	directoryAdd.Visible = true;	i marabic(), (	directoryAdd.isUpdate = false;	
	•		· -	
E: 4T-1-1-().	directoryAdd.directory =		directoryAdd.Visible = true;	
FindTable();	1 1		directoryAdd.directory =	
	break;	<pre>FindTable();</pre>		
}			break;	
case	е "Районы":	}		
{		case	е "Виды":	
	//return "districts";	{		
	DirectoryAddForm		//return "papertypes";	
directoryAdd	= new DirectoryAddForm(true,		DirectoryAddForm	
•	Район", "", "", words[1],	directoryAdd	= new DirectoryAddForm(true,	
FindTable(),			ид бумаги", "", "Плотность",	
,	directoryAdd.isUpdate = false;		dTable(), 0, "");	
	directoryAdd.Visible = true;		directoryAdd.isUpdate = false;	
	directoryAdd.directory =		directoryAdd.Visible = true;	
FindTable();	directory/Add.directory =		directoryAdd.directory =	
rilia i able(),	hanalra	EindToblo()	directoryAdd.directory =	
1	break;	<pre>FindTable();</pre>	1 1	
}	WE	,	break;	
case	е "Города":	}		
{		case	е "Форматы":	
	//return "cities";	{		
	DirectoryAddForm		//return "formats";	
directoryAdd	= new DirectoryAddForm(true,		DirectoryAddForm	
false, false, "I	Город", "", "", words[1],	directoryAdd	= new DirectoryAddForm(true,	
FindTable(), (		false, false, "d	Рормат", "", "", words[1],	
· · · · · · · · · · · · · · · · · · ·	directoryAdd.isUpdate = false;	FindTable(), (	±	
	directoryAdd.Visible = true;		directoryAdd.isUpdate = false;	
	directoryAdd.directory =		directoryAdd.Visible = true;	
FindTable();	directory radianectory –		directoryAdd.directory =	
i marabic(),	brank	FindTabla()	directory Add. directory –	
1	break;	FindTable();	huaalra	
}	"F "	,	break;	
case	е "Банки":	}		

```
записей.";
       }
                                                              MessageBox.Show(a);
                                                              //конец проверка каскадного
    private void
deleteButton_Click_1(object sender,
                                                  удаления для таблицы заказы
EventArgs e)
                                                              //удаление записи
       if (radioButton1.Checked ==
                                                              //sql = @ "delete from prints
true)//если помечено удаление текущей
                                                  where id="+
                                                  dgvData.Rows[rowIndex].Cells["id"].Value.T
                                                  oString();
         if (rowIndex < 0)//если не выбрана
                                                              //cmd = new
                                                  NpgsqlCommand(sql, conn);
запись
                                                              //dt.Load(cmd.ExecuteReader());
           MessageBox.Show("Выберите
                                                              //MessageBox.Show("Удаление
типографию которую хотите удалить");
                                                  успешно");
           return;
                                                              //конец удаление записи
         }
                                                              conn.Close();
         try
                                                              rowIndex = -1;//переход к
                                                  состоянию "ни одна запись не выбрана"
           conn.Open();
           //проверка каскадного
                                                              //select
удаления для таблицы работники
                                                              selectSql = @ "select id,print as
           cmd = new
                                                  типография,(select type from types where
NpgsqlCommand("select count(*) from
                                                  id=type_id) as тип_собственности," +
workers as w, prints as p where
                                                            "(select district from districts where
p.id=w.print_id and p.id=" +
                                                  id=district_id) as район, address as адрес,
                                                  phone as телефон, year as год_основания
dgvData.Rows[rowIndex].Cells["id"].Value.T
                                                  from prints";
oString(), conn);
                                                              Select();
           string a = "";
                                                              //select
           а += "Из таблицы работники
будет удалено "+cmd.ExecuteScalar()+"
                                                           catch (Exception ex)
записей. ":
           //конец проверка каскадного
                                                              conn.Close();
                                                              MessageBox.Show("Ошибка
удаления для таблицы работники
                                                  при удалении. Error: " + ex.Message);
           //проверка каскадного
                                                           }
удаления для таблицы заказы
           cmd = new
                                                         else if (radioButton2.Checked ==
NpgsqlCommand("select count(*) from
                                                  true)//если помечено удаление по номеру
workers as w, prints as p, orders as o where
                                                  записи
p.id=w.print_id and w.id=o.worker_id and
                                                         {
p.id="+
                                                           try
dgvData.Rows[rowIndex].Cells["id"].Value.T
                                                              conn.Open();
oString(), conn);
           а+= "Из таблицы заказы будет
                                                              //удаление записи по id
удалено " + cmd.ExecuteScalar() + "
                                                              sql = @ "delete from prints where
```

```
id=" + numericUpDown1.Value;
                                                 conn);
           cmd = new NpgsqlCommand(sql,
                                                             dt.Load(cmd.ExecuteReader());
                                                             MessageBox.Show("Удаление
conn);
           dt.Load(cmd.ExecuteReader());
                                                 успешно");
           MessageBox.Show("Удаление
                                                             //конец удаление поля со
успешно");
                                                 значением
           //конец удаление записи по id
                                                             conn.Close();
           conn.Close();
           //select
                                                 CheckComboBox(cellComboBox.SelectedInd
           selectSql = @"select id,print as
                                                 ex);
типография, (select type from types where
id=type_id) as тип_собственности," +
                                                             //select
         "(select district from districts where
                                                             selectSql = @ "select id,print as
id=district_id) as район, address as адрес,
                                                 типография, (select type from types where
                                                 id=type_id) as тип_собственности," +
phone as телефон, year as год_основания
from prints";
                                                           "(select district from districts where
                                                 id=district_id) as район, address as адрес,
           Select();
           //select
                                                 phone as телефон, year as год_основания
                                                 from prints";
         }
         catch (Exception ex)
                                                             Select();
                                                             //select
           conn.Close();
           MessageBox.Show("Ошибка
                                                           catch (Exception ex)
при удалении. Error: " + ex.Message);
                                                             conn.Close();
                                                             MessageBox.Show("Ошибка
                                                 при удалении. Error: " + ex.Message);
      else//если помечено удаление со
значением поля
                                                        }
                                                      }
         try
           conn.Open();
                                                      private void
                                                 генерацияToolStripMenuItem_Click(object
                                                 sender, EventArgs e)//-----
           //удаление поля со значением
                                                 генерация-----
           MessageBox.Show("delete from
prints where " + NameConverterForDelete(
dt.Columns[cellComboBox.SelectedIndex].C
                                                        string[] streets = { "просп. Мира",
                                                 "ул. Артема", "ул. Кирова", "ул. Победы",
olumnName) +
              "=" +
                                                 "ул. Проса", "ул. Верти", "просп. Ворона",
                                                           "пер. Тихона", "ул. Боевая", "ул.
CheckComboBox(cellComboBox.SelectedInd
ex));
                                                 Гоголя", "ул. Гастелло", "ул. Крылова"
                                                 };//улицы для случайной генерации
           sql = @ "delete from prints where
" +
                                                        string[] first_names = { "Борис",
NameConverterForDelete(dt.Columns[cellCo
                                                 "Абрам", "Альберт", "Богдан", "Вадим",
mboBox.SelectedIndex].ColumnName) +
             "=" +
                                                 "Василий", "Гавриил", "Виктор", "Густав",
                                                 "Геннадий", "Егор", "Игорь", "Заур",
CheckComboBox(cellComboBox.SelectedInd
                                                 "Игнат", "Иван", "Жерар"
ex);
           cmd = new NpgsqlCommand(sql,
```

```
,"Захар","Илья","Карл","Назар","Николай",
                                                            {
"Артур", "Михаил", "Марат" };//имена
                                                              case "Банковские":
       string[] second names = { "Рябов",
                                                                   string result =
"Смирнов", "Иванов", "Петров",
                                                    Microsoft.VisualBasic.Interaction.InputBox("
"Морозов", "Зайцев", "Тарасов",
                                                     Сколько записей сгенерировать:",
"Федоров", "Орлов", "Бобров", "Фролов",
                                                     "Генерация справочника банковские
"Ершов", "Карпов", "Шаров", "Панов",
                                                     номера", "");
         "Суворов", "Осипов", "Петухов",
                                                                   if (result != "")//если нажали
"Калинин", "Макаров" };//фамилии
                                                    отмена и строка не пустая
       string[] third names = { "Борисович",
"Абрамович", "Альбертович",
                                                                      //формирование sql
"Богданович", "Вадимович", "Васильевич",
                                                    запроса insert
"Гавриилович", "Викторович",
                                                                      if (true)
"Густавович", "Геннадиевич",
                                                                      {
         "Егорович", "Игоревич",
                                                                        sql = @"insert into
"Заурович", "Игнатович", "Иванович",
                                                    accounts( account,bank_id) values";
"Жерарович", "Захарович", "Карлович", "Наз
                                                                        sql += "("" +
                                                    random.Next(11111, 99999) +
арович", "Николаевич", "Артурович", "Миха
илович", "Маратович" }; //отчества
                                                    random.Next(11111, 99999) +
                                                                          random.Next(11111,
       string[] alphabet = { "A", "B", "C",
                                                    99999) + random.Next(11111, 99999) + "'," +
"D", "E", "F", "G", "H", "I", "J", "K", "L",
                                                    random.Next(1, 5) + ")";
"M", "N", "O", "P", "Q", "R", "S", "T", "U",
                                                                        for (int i = 1; i <
"V", "W", "X", "Y", "Z" };//Англ алфавит
                                                    int.Parse(result); i++)
       string[] greekAlphabet = { "Alpha",
                                                                          sql += ",("" +
"Beta", "Gamma", "Delta", "Epsilon", "Zeta",
                                                    random.Next(11111, 99999) +
"Eta", "Theta", "Kappa", "Lambda", "Mu", "Nu", "Xi", "Omicron", "Pi",
                                                    random.Next(11111, 99999) +
                                                                             random.Next(11111,
"Rho", "Sigma", "Tau", "Upsilon",
                                                    99999) + random.Next(11111, 99999) + "'," +
                                                    random.Next(1, 5) + ")";
"Omega" \};//греческий алфавит
       string[] nameParts = { "ta-da", "so-
so", "ready", "loved", "lover", "order",
                                                                      //конец формирование sql
"slick", "sharp", "shark", "civil", "wowed",
                                                    запроса insert
"eased", "adore", "roomy", "oh my"
              , "gusto", "peppy", "vigor",
                                                                      try
"comic", "dress", "heart", "light", "exact",
"index", "peace", "noble", "eases", "gains",
                                                                        conn.Open();
"award", "quick", "prime"
, "eases", "there", "yummy", "crisp", "jazzy", "saucy", "greet", "proto",
                                                                        //запрос insert
                                                                        cmd = new
"clean", "avant", "defix", "asset", "comfy",
                                                    NpgsqlCommand(sql, conn);
"spicy", "nurse", "manly" } ;//части названий
                                                     dt.Load(cmd.ExecuteReader());
       string[] words = label1.Text.Split('
                                                                        //конец запрос insert
');//проверка названия таблицы
       var random = new Random();
                                                                        conn.Close();
       switch (words[1])
```

```
//select
                                                                          streets[random.Next(0,
                   selectSql = @"select
                                                    streets.Length)] + " " + random.Next(1, 150)
account as банковский_номер, (select bank
                                                    + "', " +//address
                                                                          "'071" +
from banks where id=bank_id) as bank from
accounts";
                                                    random.Next(0, 10) + random.Next(0, 10) +
                   Select();
                                                    random.Next(0, 10) + random.Next(0, 10) +
                                                    random.Next(0, 10) + random.Next(0, 10) +
                   //select
                                                    random.Next(0, 10) + "', " +//phone
                 }
                 catch (Exception ex)
                                                                          random.Next(1981,
                                                    2000) + ")";//year
                   conn.Close();
                                                                        for (int i = 2; i <
                                                    int.Parse(result) + 1; i++)
MessageBox.Show("Ошибка при удалении.
                                                                          sql += ",("" +
Error: " + ex.Message);
                                                     alphabet[random.Next(0, 26)] +
                                                     nameParts[random.Next(1,
              break;
                                                    nameParts.Length)] +
                                                    nameParts[random.Next(1,
         case "Типографии":
                                                     nameParts.Length)] + "', " +//print
                                                                          random.Next(1, 4) + "
                                                     " +//type id
              string result =
                                                                          random.Next(1, 9) + ",
Microsoft. Visual Basic. Interaction. Input Box ("
                                                     " +//district id
Сколько записей сгенерировать:",
                                                                          streets[random.Next(0,
"Генерация таблицы Типографии", "");
                                                    streets.Length)] + " " + random.Next(1, 150)
              if (result != "")//если нажали
                                                    + "', " +//address
                                                                           "'071" +
отмена и строка не пустая
                                                    random.Next(0, 10) + random.Next(0, 10) +
                                                    random.Next(0, 10) + random.Next(0, 10) +
                                                    random.Next(0, 10) + random.Next(0, 10) +
                                                    random.Next(0, 10) + "', " +//phone
                                                                          random.Next(1981,
                //формирование sql
                                                    2000) + ")";//year
запроса insert
                                                                        }
                 if (true)
                                                                     //конец формирование sql
                   sql = "INSERT INTO
                                                    запроса insert
prints( print, type_id, district_id, address,
phone, year) values" +
                                                                     try
                                                                        conn.Open();
alphabet[random.Next(0, 26)] +
nameParts[random.Next(1,
                                                                        //запрос insert
                                                                        sql = @"" + sql + ";";
nameParts.Length)] +
nameParts[random.Next(1,
                                                                        cmd = new
nameParts.Length)] + "', " +//print
                                                    NpgsqlCommand(sql, conn);
                      random.Next(1, 4) + ",
" +//type id
                                                    dt.Load(cmd.ExecuteReader());
                      random.Next(1, 9) + ",
                                                                        //конец запрос insert
"" +//district id
```

```
conn.Close();
                                                                  int o;//счетчик для циклов
                                                   foreach
                   //select
                                                                  string result =
                                                   Microsoft.VisualBasic.Interaction.InputBox("
                   selectSql = @"select
id,print as типография,(select type from
                                                   Сколько записей сгенерировать?",
types where id=type_id) as
                                                   "Генерация таблицы Заказы", "");
                                                                  if (result != "")//если нажали
тип_собственности," +
                      '(select district from
                                                   ок(и строка не пустая)
districts where id=district_id) as район,
address as адрес, phone as телефон, year as
год from prints";
                   Select();
                                                                    try
                   //select
                                                                      if (true)//заполнение
                   //count
                                                   массивов с индексами
                   sql = "select count(*)
from prints";
                                                                         conn.Open();
                   ChangeLabel("Таблица
Заказы");
                                                                         //вытягивание
                   //count
                                                   аккаунотов из справочника аккаунты
                                                                         dataTable = new
                                                   DataTable();
MessageBox.Show("Генерация успешно
                                                                         sql = @ "select account
выполнена");
                                                   from accounts";
                                                                         cmd = new
                catch (Exception ex)
                                                   NpgsqlCommand(sql, conn);
                   conn.Close();
                                                   dataTable.Load(cmd.ExecuteReader());
                                                                         acc = new
MessageBox.Show("Ошибка при генерации.
                                                   string[dataTable.Rows.Count];
Error: " + ex.Message);
                                                                         foreach (DataRow dr in
                                                   dataTable.Rows)
              break;
                                                                           acc[o] =
         case "Заказы":
                                                   dr.ItemArray[0].ToString();
              string[] acc = \{ "a" \};//HOMEPa
                                                   //MessageBox.Show("acc["+i+"]: "+acc[i]);
аккаунтов
                                                                           o += 1;
              DataTable
dataTable;//промежуточная таблица данных
                                                                         //конец вытягивание
              int[] customers= { 0};//массив
                                                   аккаунотов из справочника аккаунты
индексов пользователей
              int[] producttypes = \{ 0 \}
                                                                         //вытягивание id из
                                                   таблицы заказчики
};//массив видов изделий
              int[] workers = \{0\};//maccub
                                                                         cmd = new
                                                   NpgsqlCommand("select id from customers",
работников
              int[] formats = \{ 0 \}
                                                   conn);
                                                                         dataTable = new
};//форматов
              int[] papertypes = \{ 0 \}
                                                   DataTable();
};//видов бумаги
```

```
dataTable.Load(cmd.ExecuteReader());
                                                                       workers = new
                     customers = new
                                                  int[dataTable.Rows.Count];
int[dataTable.Rows.Count];
                                                                       o = 0;
                                                                       foreach (DataRow dr in
                     0 = 0:
                     foreach (DataRow dr in
                                                  dataTable.Rows)
dataTable.Rows)
                                                                          workers[o] =
                                                  int.Parse(dr.ItemArray[0].ToString());
                       customers[o] =
int.Parse(dr.ItemArray[0].ToString());
                                                  //MessageBox.Show("worekrs[" + o + "]: " +
//MessageBox.Show("customers[" + o + "]: "
                                                  workers[o]);
+ customers[o]);
                                                                          o += 1;
                       o += 1;
                                                                       //конец вытягивание
                                                  id из таблицы работники
                     //конец вытягивание
id из таблицы заказчики
                                                                       //вытягивание id из
                     //вытягивание id из
                                                  таблицы форматы
таблицы изделия
                                                                       cmd = new
                     cmd = new
                                                  NpgsqlCommand("select id from formats",
NpgsqlCommand("select id from
                                                  conn):
producttypes", conn);
                                                                       dataTable = new
                     dataTable = new
                                                  DataTable();
DataTable();
                                                  dataTable.Load(cmd.ExecuteReader());
dataTable.Load(cmd.ExecuteReader());
                                                                       formats = new
                     producttypes = new
                                                  int[dataTable.Rows.Count];
int[dataTable.Rows.Count];
                     0 = 0:
                                                                       foreach (DataRow dr in
                     foreach (DataRow dr in
                                                  dataTable.Rows)
dataTable.Rows)
                                                                          formats[o] =
                       producttypes[o] =
                                                  int.Parse(dr.ItemArray[0].ToString());
int.Parse(dr.ItemArray[0].ToString());
                                                  //MessageBox.Show("formats[" + o + "]: " +
//MessageBox.Show("producttypes[" + o + "]:
                                                  formats[0]);
" + producttypes[o]);
                                                                          o += 1;
                       o += 1;
                                                                       //конец вытягивание
                     //конец вытягивание
                                                  id из таблицы форматы
id из таблицы изделия
                                                                       //вытягивание id из
                     //вытягивание id из
                                                  таблицы виды бумаги
таблицы работники
                                                                       cmd = new
                     cmd = new
                                                  NpgsqlCommand("select id from papertypes",
NpgsqlCommand("select id from workers",
                                                  conn);
conn);
                                                                       dataTable = new
                     dataTable = new
                                                  DataTable();
DataTable();
                                                  dataTable.Load(cmd.ExecuteReader());
dataTable.Load(cmd.ExecuteReader());
                                                                       papertypes = new
```

```
int[dataTable.Rows.Count];
                                                    alphabet[random.Next(0, alphabet.Length)] +
                                                    "', " +//publication
                     foreach (DataRow dr in
dataTable.Rows)
                                                    workers[random.Next(0, workers.Length)] +
                                                    ", " +//worker id
                        papertypes[o] =
int.Parse(dr.ItemArray[0].ToString());
                                                                          random.Next(100,
                                                    5000) + ", " +//price
//MessageBox.Show("papertypes[" + o + "]: "
                                                                          random.Next(100,
                                                    5000) + ", " +//calculation
+ papertypes[o]);
                        o += 1;
                                                                          random.Next(100,
                                                    5000) + ", " +//count
                     //конец вытягивание
id из таблицы виды бумаги
                                                    formats[random.Next(0, formats.Length)] + ",
                                                    " +//format id
                      conn.Close();
                   }//заполнение массивов
                                                    papertypes[random.Next(0,
                                                    papertypes.Length)] + ", " +//papertype id
с индексами
                                                                          random.Next(2015,
                                                    2017) + "-0" + random.Next(1, 10) + "-" +
                 catch (Exception ex)
                                                    random.Next(10, 29) + "', "' +//datastart
                   conn.Close();
                                                                          random.Next(2017,
                                                    2020) + "-0" + random.Next(1, 10) + "-" +
                                                    random.Next(10, 29) + "', '" +//dataplan
MessageBox.Show("Deleted fail. Error: " +
                                                                          random.Next(2017,
ex.Message);
                                                    2020) + "-0" + random.Next(1, 10) + "-" +
                                                    random.Next(10, 29) + "', " +//datafact
                                                                          random.Next(0, 10000)
                                                    + ", "" +//cost
                //формирование sql
запроса insert
                                                                          acc[random.Next(0,
                 if (true)
                                                    acc.Length)] + "')";//account
                                                                        for (int i = 2; i <
                   sql = "INSERT INTO
                                                    int.Parse(result) + 1; i++)
orders(customer_id, producttype_id,
                                                                          sql += ",(" + " " +//id
publication, worker_id, price, calculation,
count, format id, papertype id, datastart,
dataplan, datafact, cost, account) VALUES"
                                                    customers[random.Next(0,
                                                    customers.Length)] + ", " +//customer id
                      "(" + " " +//id
                                                    producttypes[random.Next(0,
                                                    producttypes.Length)] + ", "' +//producttype
customers[random.Next(0,
customers.Length)] + ", " +//customer id
                                                    id
producttypes[random.Next(0,
                                                    greekAlphabet[random.Next(0,
producttypes.Length)] + ", "' +//producttype
                                                    greekAlphabet.Length)] +
                                                    nameParts[random.Next(0,
id
                                                    nameParts.Length)] + " " +
greekAlphabet[random.Next(0,
greekAlphabet.Length)] +
                                                    alphabet[random.Next(0, alphabet.Length)] +
nameParts[random.Next(0,
                                                    "', " +//publication
```

nameParts.Length)] + " " +

```
workers[random.Next(0, workers.Length)] +
                                                   id=producttype_id) as изделие," +
", " +//worker id
                                                                         " publication as
                                                   название издания, picture as
                     random.Next(100,
5000) + ", " +//price
                                                   титульная страница," +
                                                                       "calculation as тираж,
                     random.Next(100,
5000) + ", " +//calculation
                                                   count as количество листов, (select format
                                                   from formats where id=format_id) as формат,
                     random.Next(100,
5000) + ", " +//count
                                                   (select papertype from papertypes where id =
                                                   papertype_id) as вид_бумаги" +
formats[random.Next(0, formats.Length)] + ",
                                                                       ", datastart as
" +//format id
                                                   дата_принятия, dataplan as
                                                   дата_выполнения_план, datafact as
papertypes[random.Next(0,
                                                   дата_выполнения_факт, cost as предоплата,
papertypes.Length)] + ", " +//papertype id
                                                   comment as доп_сведения" +
                     random.Next(2015,
                                                                       ",account as
2017) + "-0" + random.Next(1, 10) + "-" +
                                                   банковский_номер from orders limit 1000";
random.Next(10, 29) + "', '" +//datastart
                                                                      Select();
                     random.Next(2017,
                                                                      //select
2020) + "-0" + random.Next(1, 10) + "-" +
random.Next(10, 29) + "', '" +//dataplan
                                                                      //count
                     random.Next(2017,
                                                                      sql = "select count(*)
2020) + "-0" + random.Next(1, 10) + "-" +
                                                   from orders";
random.Next(10, 29) + "", " +//datafact
                                                                      ChangeLabel("Таблица
                     random.Next(0, 10000)
                                                   Заказы");
+ ", " +//cost
                                                                      //count
                     acc[random.Next(0,
acc.Length)] + "')";//account
                                                   MessageBox.Show("Генерация успешно
                                                   выполнена");
                //конец формирование sql
запроса insert
                                                                    catch (Exception ex)
                                                                      conn.Close();
                try
                                                   MessageBox.Show("Ошибка при генерации.
                                                   Error: " + ex.Message):
                   conn.Open();
                   //запрос insert
                   sql = @"" + sql + ";";
                                                                 else//если нажали отмена или
                   cmd = new
                                                   строка пустая
NpgsqlCommand(sql, conn);
dt.Load(cmd.ExecuteReader());
                                                   MessageBox.Show("Генерация отменена!",
                                                   "Оповещение", MessageBoxButtons.OK,
                  //конец запрос insert
                                                   MessageBoxIcon.Information);
                   conn.Close();
                                                                 break;
                   //select
                   selectSql = @"select id,
                                                             case "Заказчики":
customer id as индекс заказчика, (select
producttype from producttypes where
                                                                 //"prints";
```

```
string result =
                                                     10) + random.Next(0, 10) + random.Next(0, 10)
Microsoft. Visual Basic. Interaction. Input Box ("
                                                     10) + "') ";//phone
Сколько записей сгенерировать:",
                                                                        for (int i = 2; i <
"Генерация таблицы Заказчики", "");
                                                     int.Parse(result) + 1; i++)
              if (result != "")//если нажали
                                                                           sql += ",(" + " "" +//id
отмена и строка не пустая
                                                     first_names[random.Next(0,
                 try
                                                     first_names.Length)] + "', "' +//first name
                   selectSql = @"select *
                                                     second names[random.Next(0,
from customers":
                                                     second_names.Length)] + "', "" +//second
                   Select();
                                                     name
                 catch (Exception ex)
                                                     third_names[random.Next(0,
                                                     third_names.Length)] + "', " +// third name
                   conn.Close();
                                                                           random.Next(1, 7) + ",
MessageBox.Show("Ошибка при отчистке
                                                     "" +//city id
таблицы. Error: " + ex.Message);
                                                                           streets[random.Next(0,
                                                     streets.Length)] + " " + random.Next(1, 150)
                                                     + "', " +//address
                 //формирование sql
запроса insert
                                                     random.Next(1950, 2002) + "-0" +
                 if (true)
                                                     random.Next(1, 10) + "-0" + random.Next(1, 10)
                                                     28) + "', " +//birthday
                                                                           "'071" +
                   sql = "INSERT INTO
customers(first_name, second_name,
                                                     random.Next(0, 10) + random.Next(0, 10) +
third_name, city_id, address, birthday, phone)
                                                     random.Next(0, 10) + random.Next(0, 10) +
       VALUES" +
                                                     random.Next(0, 10) + random.Next(0, 10) +
                    "(" + " '" +//id
                                                     random.Next(0, 10) + "') ";//phone
first_names[random.Next(0,
                                                                         }
first_names.Length)] + "', "' +//first name
                                                                      //конец формирование sql
second_names[random.Next(0,
                                                     запроса insert
second names.Length)] + "', "' +//second
name
                                                                      try
third_names[random.Next(0,
                                                                        conn.Open();
third_names.Length)] + "', " +// third name
                   random.Next(1, 7) + ", ""
                                                                        //запрос insert
                                                                        sql = @"" + sql + ";";
+//city id
                   streets[random.Next(0,
                                                                        cmd = new
streets.Length)] + " " + random.Next(1, 150)
                                                     NpgsqlCommand(sql, conn);
+ "', " +//address
                   """ + random.Next(1950,
                                                     dt.Load(cmd.ExecuteReader());
2002) + "-0" + random.Next(1, 10) + "-0" +
                                                                        //конец запрос insert
random.Next(1, 28) + "', " +//birthday
                   "'071" + random.Next(0,
                                                                        conn.Close();
10) + random.Next(0, 10) + random.Next(0, 10)
10) + random.Next(0, 10) + random.Next(0, 10)
                                                                        //select
```

```
selectSql = @"select
                                                                   {
id,first_name as имя, second_name as
                                                                     conn.Open();
фамилия, third_name as отчество,(select city
                                                                     //sql = @"select count(*)
from cities where id=city_id) as город," +
                                                  from workers";
                     "address as адрес,
                                                                     //cmd = new
birthday as дата_рождения, phone as
                                                  NpgsqlCommand(sql, conn);
телефон from customers";
                  Select();
                                                  //MessageBox.Show("Будет удалено " +
                                                  cmd.ExecuteScalar() + " записей из таблицы
                  //select
                                                  Работники", "Предупреждение",
                                                  MessageBoxButtons.OKCancel);
                  //count
                  sql = "select count(*)
                                                                     sql = @"select count(*)
                                                  from prints";
from customers";
                  ChangeLabel("Таблица
                                                                     cmd = new
Заказчики");
                                                  NpgsqlCommand(sql, conn);
                                                                     printsCount =
                  //count
                                                  int.Parse(cmd.ExecuteScalar().ToString());
MessageBox.Show("Генерация успешно
                                                  //MessageBox.Show(printsCount.ToString());
выполнена");
                                                                     conn.Close();
                                                                     //conn.Open();
                                                                     //sql = @"delete from
                catch (Exception ex)
                                                  workers";
                  conn.Close();
                                                                     //cmd = new
                                                  NpgsqlCommand(sql, conn);
MessageBox.Show("Ошибка при генерации.
Error: " + ex.Message);
                                                  //dt.Load(cmd.ExecuteReader());
                                                  ////MessageBox.Show("Delete succesfully");
              else//если нажали отмена или
                                                                     //conn.Close();
строка пустая
                                                                     //select
                                                                     selectSql = @"select *
MessageBox.Show("Генерация отменена!",
                                                  from workers";
"Оповещение", MessageBoxButtons.OK,
                                                                     Select():
MessageBoxIcon.Information);
                                                                     //select
              break;
                                                                   catch (Exception ex)
         case "Работники":
                                                                     conn.Close();
                                                  MessageBox.Show("Ошибка при отчистке
              string result =
                                                  таблицы. Error: " + ex.Message);
Microsoft.VisualBasic.Interaction.InputBox("
Сколько записей сгенерировать:",
"Генерация таблицы Работники", "");
                                                                   //формирование sql
              if (result != "")//если нажали
                                                  запроса insert
отмена и строка не пустая
                                                                  if (true)
                int printsCount=0;
                                                                     sql = "INSERT INTO
                                                  public.workers( first_name, second_name,
                try
```

```
third_name, print_id) VALUES " +
                                                    id,first_name as имя, second_name as
                      "(" + " "" +//id
                                                    фамилия, third name as отчество, (select
                                                    print from prints where id=print_id) as
                                                    типография from workers";
first_names[random.Next(0,
first_names.Length)] + "', "' +//first name
                                                                       Select();
                                                                       //select
second_names[random.Next(0,
second_names.Length)] + "', "' +//second
                                                                       //count
name
                                                                       sql = "select count(*)
                                                    from workers";
third_names[random.Next(0,
                                                                       ChangeLabel("Таблица
third_names.Length)] + "', " +// third name
                                                    Работники");
                     random.Next(32, 32 +
                                                                       //count
printsCount) + ")";//print id
                   for (int i = 2; i <
                                                    MessageBox.Show("Генерация успешно
int.Parse(result) + 1; i++)
                                                    выполнена");
                     sql += ",(" + """ +//id
                                                                    catch (Exception ex)
first_names[random.Next(0,
first names.Length)] + "', "' +//first name
                                                                       conn.Close();
                                                    MessageBox.Show("Ошибка при генерации.
second_names[random.Next(0,
second_names.Length)] + "', "' +//second
                                                    Error: " + ex.Message);
name
                                                                     }
                                                                  else//если нажали отмена или
third_names[random.Next(0,
third\_names.Length)] + "", " + \!\!/\!/ third \ name
                                                    строка пустая
                     random.Next(32, 32 +
printsCount) + ")";//print id
                                                    MessageBox.Show("Генерация отменена!",
                                                    "Оповещение", MessageBoxButtons.OK,
                                                    MessageBoxIcon.Information);
                //конец формирование sql
запроса insert
                                                                  break;
                try
                                                             default:
                   conn.Open();
                                                                {
                   //запрос insert
                   sql = @"" + sql + ";";
                                                                    MessageBox.Show("Нельзя
                                                    сгенерировать справочник");
                   cmd = new
NpgsqlCommand(sql, conn);
                                                                  break;
dt.Load(cmd.ExecuteReader());
                                                                }
                   //конец запрос insert
                                                           }
                   conn.Close();
                   //select
                   selectSql = @"select
```

```
= new OrdersAddForm();
                                                                    ordersAdd.isUpdate = true;
                                                                    ordersAdd.id =
                                                   int.Parse(dt.Rows[rowIndex].ItemArray[0].To
                                                   String());
                                                                    ordersAdd.Visible = true;
                                                                    break;
     }
                                                               case "Заказчики":
    private void panel1_Paint(object sender,
PaintEventArgs e)
                                                                    //return "customers";
                                                                    CustomersAddForm
                                                    customersAdd = new CustomersAddForm();
                                                                    customers Add. is Update = \\
                                                   true;
    private void
                                                                    customersAdd.id =
cellComboBox_SelectedIndexChanged(object
                                                   int.Parse(dt.Rows[rowIndex].ItemArray[0].To
sender, EventArgs e)
                                                   String());
                                                                    customersAdd.Visible =
                                                   true;
Check Combo Box (cell Combo Box. Selected Ind\\
ex);
     }
                                                                    break;
    private void
                                                               case "Работники":
updateButton_Click_1(object sender,
EventArgs e)
                                                                    //return "workers";
                                                                    WorkersAddForm
     {
       string[] words = label1.Text.Split(' ');
                                                   workersAdd = new WorkersAddForm();
       if (rowIndex >= 0)
                                                                    workersAdd.isUpdate =
         switch (words[1])
                                                   true;
                                                                    workersAdd.id=
         {
                                                    int.Parse(dt.Rows[rowIndex].ItemArray[0].To
                                                    String());
            case "Типографии":
                                                                    workersAdd.Visible = true:
                                                                    break;
                //return "prints";
                PrintsAddForm addForm =
                                                               case "Банковские":
new PrintsAddForm();
                                                                    //return "accounts";
                addForm.isUpdate = true;
                                                                    DirectoryAddForm
                addForm.id =
int.Parse(dt.Rows[rowIndex].ItemArray[0].To
                                                    directoryAdd = new DirectoryAddForm(true,
                                                    true, false, "Банковский счет", "Банк", "",
String());
                                                    words[1], FindTable(),0,
                addForm.Visible = true;
                                                    dt.Rows[rowIndex].ItemArray[0].ToString());
                break;
                                                                    directoryAdd.isUpdate =
              }
            case "Заказы":
                                                   true;
                                                                    directoryAdd.Visible = true;
                //return "orders";
                OrdersAddForm ordersAdd
                                                                    directoryAdd.directory =
```

<pre>FindTable();</pre>		<pre>FindTable();</pre>		
	break;		break;	
}		}		
case	: "Типы":	case	: "Банки":	
{		{		
	//return "types";		//return "banks";	
	DirectoryAddForm		DirectoryAddForm	
directoryAdd = r	new DirectoryAddForm(true,	directoryAdd = r	new DirectoryAddForm(true,	
	г собственности", "", "",	false, false, "Бан	к", "", "", words[1],	
words[1], FindT	able(),	FindTable(),		
int.Parse(dt.Row	s[rowIndex].ItemArray[0].To	int.Parse(dt.Row	s[rowIndex].ItemArray[0].To	
String()), "");	• • • • • • • • • • • • • • • • • • • •	String()), "");		
<i>5</i> (/// //	directoryAdd.isUpdate =	<i>5</i> (/// //	directoryAdd.isUpdate =	
true;	and the second of the second o	true;		
	directoryAdd.Visible = true;	,	directoryAdd.Visible = true;	
	directory rad. Visitore drae,		directoryAdd.directory =	
	directoryAdd.directory =	<pre>FindTable();</pre>	directory/radianectory =	
<pre>FindTable();</pre>	directory/tud.directory =	i marabic(),	break;	
Tille Lable(),	break;	ı	oreak,	
ı	bicak,	Case	: "Изделия":	
3000	: "Районы":	Case	нзделия.	
Case	ганоны.	1	//motives "man divietty mas".	
1	//naturan "diataiata".		//return "producttypes";	
	//return "districts";	1: 4 A 1.1	DirectoryAddForm	
مانات مدمس ۸ ماما	DirectoryAddForm	directoryAdd = new DirectoryAddForm(true, false, false, "Изделие", "", "", words[1],		
•	new DirectoryAddForm(true,		teлиe, , , words[1],	
	юн", "", "", words[1],	FindTable(),	F	
FindTable(),	F		s[rowIndex].ItemArray[0].To	
	s[rowIndex].ItemArray[0].To	String()), "");		
String()), "");			directoryAdd.isUpdate =	
	directoryAdd.isUpdate =	true;		
true;			directoryAdd.Visible = true;	
	directoryAdd.Visible = true;		directoryAdd.directory =	
		<pre>FindTable();</pre>		
	directoryAdd.directory =		break;	
<pre>FindTable();</pre>		}		
	break;	case	е "Виды":	
}		{		
case	: "Города":		//return "papertypes";	
{			DirectoryAddForm	
	//return "cities";	directoryAdd = r	new DirectoryAddForm(true,	
	DirectoryAddForm	false, true, "Вид	бумаги", "", "Плотность",	
directoryAdd = r	new DirectoryAddForm(true,	words[1], FindT	able(),	
false, false, "Γορ	юд", "", "", words[1],	int.Parse(dt.Row	s[rowIndex].ItemArray[0].To	
FindTable(),		String()), "");		
int.Parse(dt.Row	s[rowIndex].ItemArray[0].To		directoryAdd.isUpdate =	
String()), "");	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	true;	, ,	
	directoryAdd.isUpdate =	,	directoryAdd.Visible = true;	
true;	, r		directoryAdd.directory =	
7	directoryAdd.Visible = true;	<pre>FindTable();</pre>		
		,,	break;	
	directoryAdd.directory =	}	7	
	-	,		

```
case "Форматы":
                                                       private void
                                                  количестоЗаказовНаБанкТoolStripMenuIte
                //return "formats";
                                                  m_Click(object sender, EventArgs e)
                DirectoryAddForm
directoryAdd = new DirectoryAddForm(true,
                                                         DiagramForm diagramForm = new
false, false, "Формат", "", "", words[1],
                                                  DiagramForm();
FindTable(),
                                                         diagramForm.type = 1;
int.Parse(dt.Rows[rowIndex].ItemArray[0].To
                                                         diagramForm.Show();
String()),"");
                directoryAdd.isUpdate =
true:
                directoryAdd.Visible = true;
                                                       private void
                directoryAdd.directory =
                                                  городаСЗаказчикамиToolStripMenuItem Cl
                                                  ick(object sender, EventArgs e)
FindTable();
                break;
              }
                                                         string path =
                                                  System.IO.Directory.GetCurrentDirectory() +
       else MessageBox.Show("Выберите
                                                  @"\" + "Save_value.xlsx";
запись");
                                                         Excel.Application excelapp = new
    }
                                                  Excel.Application();
                                                         Excel.Workbook workbook =
                                                  excelapp.Workbooks.Add();
    private void
запросыToolStripMenuItem_Click(object
                                                         Excel.Worksheet worksheet =
sender, EventArgs e)
                                                  workbook.ActiveSheet;
       SelectForm selectForm = new
                                                         for (int i= 1; i < dgvData.RowCount +
SelectForm();
                                                  1; i++)
       selectForm.Show();
                                                           for (int j = 1; j <
                                                  dgvData.ColumnCount+1; j++)
    private void
изделияToolStripMenuItem2_Click(object
                                                              worksheet.Rows[i].Columns[j] =
                                                  dgvData.Rows[i - 1].Cells[j - 1].Value;
sender, EventArgs e)
       DiagramForm diagramForm = new
DiagramForm();
                                                         excelapp.AlertBeforeOverwriting =
       diagramForm.type = 3;
                                                  false;
                                                         workbook.SaveAs(path);
       diagramForm.Show();
    }
                                                         excelapp.Quit();
                                                       }
    private void
частотаИспользованияФорматовToolStrip
                                                      private void
MenuItem_Click(object sender, EventArgs e)
                                                  средий Размер Тиражей По Типографии Tool
                                                  StripMenuItem_Click(object sender,
       DiagramForm diagramForm = new
                                                  EventArgs e)//Средний тираж по
DiagramForm();
                                                  типографиям, районам, городу
       diagramForm.type = 2;
       diagramForm.Show();
                                                         DataTable dt4 = new DataTable();
                                                         string path =
    }
                                                  System.IO.Directory.GetCurrentDirectory() +
```

```
@"\" + "Средний_тираж.xlsx";
                                                          Excel.Application excelapp = new
                                                   Excel.Application();
       sql = @"select p.print as
                                                          Excel.Workbook workbook =
типография, avg (o. calculation) as
                                                   excelapp.Workbooks.Add();
средний_тираж_типография,d.district as
                                                          Excel.Worksheet worksheet =
район,dist.cpeдний_тираж_район,cit.cpeдни
                                                   workbook.ActiveSheet:
й_тираж_город" +
                 " from districts as d, prints
                                                          for (int j = 1; j < dt4.Columns.Count +
as p, workers as w, orders as o, (
                                   select
                                                   1; j++)
d.district,avg(o.calculation) as
средний тираж район
                            from districts as
                                                             worksheet.Rows[1].Columns[j] =
d," +
                                                   dt4.Columns[j - 1].ColumnName;
                     prints as p, workers as
                     where p.id=w.print id
                                                          for (int i = 2; i < dt4.Rows.Count + 2;
       orders as o
w.
and
       d.id=p.district_id and o.worker_id =
                                                   i++)
       group by 1
                     order by 1) as dist," +
w.id
                     (select
                                                             for (int j = 1; j < dt4.Columns.Count
avg(o.calculation) as средний_тираж_город
                                                   +1; j++)
       from districts as d,
                            prints as p,
workers as w, orders as o
                            where
                                                               worksheet.Rows[i].Columns[j] =
p.id=w.print id and "+
                                                   dt4.Rows[i - 2].ItemArray[j - 1];
                     d.id=p.district_id and
       o.worker_id = w.id
                           order by 1) as cit
where p.id=w.print_id and d.id=p.district_id
                                                          excelapp.AlertBeforeOverwriting =
and o.worker_id = w.id and dist.district =
                                                   false;
d.district" +
                                                          workbook.SaveAs(path);
                " group by 1,3,4,5 order by
                                                          excelapp.Quit();
3,1";
       try
                                                        private void
                                                   стреднийРазмерТиражейПоРайонуToolStri
AuthorizationForm.mainForm.conn.Open();
                                                   pMenuItem_Click(object sender, EventArgs
                                                   е)//Лучшая типография по району, городу
         AuthorizationForm.mainForm.cmd
                                                   и самое популярное изделие
= new NpgsqlCommand(sql,
AuthorizationForm.mainForm.conn);
                                                          DataTable dt4 = new DataTable();
         dt4 = new DataTable();
                                                          string path =
                                                   System.IO.Directory.GetCurrentDirectory() +
                                                   @"\" + "Лучшая_типография.xlsx";
dt4.Load(AuthorizationForm.mainForm.cmd.
ExecuteReader());
                                                          sql = @"select p.print as
                                                   типография_район,о.cost*o.calculation as
AuthorizationForm.mainForm.conn.Close();
                                                   сумма_заказов_район,d.district as
                                                   район, maxxx.print as типография_город," +
       catch (Exception ex)
                                                                    "maxxx.maxcd as
                                                   сумма_заказов_город,ргтах.изделие аs
                                                   популярное_изделие,
AuthorizationForm.mainForm.conn.Close();
                                                   ргтах.количество заказов изделия
         MessageBox.Show("Error: " +
                                                          from districts as d," +
                                                                        prints as p, workers as
ex.Message);
                                                          orders as o, (
                                                                               select
       }
                                                   w,
```

```
d.district,max(o.cost*o.calculation) as maxed
              from districts as d," +
                                                    AuthorizationForm.mainForm.conn.Open();
                      prints as p, workers as
                                   where
                                                              AuthorizationForm.mainForm.cmd
              orders as o
p.id=w.print_id and
                                                    = new NpgsqlCommand(sql,
       d.id=p.district_id and
                                                    AuthorizationForm.mainForm.conn);
       o.worker_id = w.id " +
                                                              dt4 = new DataTable();
                      group by 1
       order by 1) as maxd,
                                                    dt4.Load(AuthorizationForm.mainForm.cmd.
                                                    ExecuteReader());
       select
p.print,max(o.cost*o.calculation) as maxed
       from districts as d, "+
                       prints as p, workers as
                                                    AuthorizationForm.mainForm.conn.Close();
              orders as o.(
w.
       select max(o.cost*o.calculation) as
                                                           catch (Exception ex)
                     from districts as d," +
maxcc
                         prints as p, workers
                     orders as o
                                                    AuthorizationForm.mainForm.conn.Close();
as w,
                                                              MessageBox.Show("Error: " +
       where p.id=w.print_id and
       d.id=p.district_id and" +
                                                    ex.Message);
                         o.worker_id = w.id
                     order by 1) as maxc
                                                           Excel.Application excelapp = new
       where p.id=w.print_id and
                                                    Excel.Application();
       d.id=p.district_id and
                                                           Excel.Workbook workbook =
       o.worker_id = w.id and" +
                                                    excelapp.Workbooks.Add();
                                                           Excel.Worksheet worksheet =
                                                    workbook.ActiveSheet;
maxc.maxcc=(o.cost*o.calculation)
       group by 1
                            order by 1) as
                     (select pr.producttype
                                                           for (int j = 1; j < dt4.Columns.Count +
maxxx,
as изделие, o.producttype_id, count(o.*) as
                                                    1; j++)
количество_заказов_изделия
       from districts as d, " +
                                                              worksheet.Rows[1].Columns[j] =
                         prints as p, workers
                                                    dt4.Columns[j - 1].ColumnName;
                     orders as o,
as w.
producttypes as pr
                                                           for (int i = 2; i < dt4.Rows.Count + 2;
                                   where
p.id=w.print id and
                                                    i++)
       d.id=p.district_id and
       o.worker_id = w.id and" +
                                                              for (int j = 1; j < dt4.Columns.Count
                                                    +1; j++)
o.producttype_id=pr.id
                                                                worksheet.Rows[i].Columns[j] =
       group by 1,2
                                   order by
                                    " +
                                                    dt4.Rows[i - 2].ItemArray[j - 1];
3 desc
              limit 1) as prmax
                     where p.id=w.print_id
       d.id=p.district_id and o.worker_id =
and
w.id and
                                                           excelapp.AlertBeforeOverwriting =
       maxd.maxcd=(o.cost*o.calculation)
                                                    false:
       maxd.district=d.district
and
                                   order by
                                                           workbook.SaveAs(path);
1";
                                                           excelapp.Quit();
       try
                                                         private void
```

```
стреднийРазмерТиражейПоГородуТооlStri
                                                        for (int j = 1; j < dt4.Columns.Count +
pMenuItem Click(object sender, EventArgs
                                                 1; j++)
е)//Стоимость заказов типографий за ГОД
не выполненные в срок
                                                           worksheet.Rows[1].Columns[j] =
                                                 dt4.Columns[j - 1].ColumnName;
      DataTable dt4 = new DataTable();
                                                        for (int i = 2; i < dt4.Rows.Count + 2;
      string path =
System.IO.Directory.GetCurrentDirectory() +
                                                 i++)
@"\" + "Стоимость_заказов.xlsx";
                                                           for (int j = 1; j < dt4.Columns.Count
      string result =
                                                 +1; j++)
Microsoft.VisualBasic.Interaction.InputBox("
Введите год:", "Ввод данных для запроса",
                                                             worksheet.Rows[i].Columns[j] =
""):
                                                 dt4.Rows[i - 2].ItemArray[j - 1];
       sql = @"select id as
номер_заказа, publication заказ, datafact
дата_выполнения_факт,dataplan as
                                                        excelapp.AlertBeforeOverwriting =
дата выполнения план from orders" +
                                                 false:
         " where datafact>dataplan and
                                                        workbook.SaveAs(path);
extract(year from datafact) =" + result + "
                                                        excelapp.Quit();
order by 2";
      try
                                                      public void CreateHTMLFile(string
                                                 table, string name)
AuthorizationForm.mainForm.conn.Open();
                                                        string path =
                                                 System.IO.Directory.GetCurrentDirectory() +
         AuthorizationForm.mainForm.cmd
= new NpgsqlCommand(sql,
                                                 @"\" +name +".html";
AuthorizationForm.mainForm.conn);
                                                        var sbHTML = new StringBuilder();
         dt4 = new DataTable();
                                                        var sw = new
                                                 StreamWriter(System.IO.Directory.GetCurren
                                                 tDirectory() + @ "\" +name +".html");
dt4.Load(AuthorizationForm.mainForm.cmd.
                                                        sbHTML.AppendLine("<DOCTYPE
ExecuteReader());
                                                 html>");
                                                        sbHTML.AppendLine("<html>");
                                                        sbHTML.AppendLine("<head>");
AuthorizationForm.mainForm.conn.Close();
                                                        sbHTML.AppendLine("<title>");
                                                        sbHTML.AppendLine(name);
      catch (Exception ex)
                                                        sbHTML.AppendLine("</title>");
                                                        sbHTML.AppendLine("</head>");
                                                        sbHTML.AppendLine("<body>");
AuthorizationForm.mainForm.conn.Close();
         MessageBox.Show("Error: " +
                                                        sbHTML.AppendLine(table);
ex.Message);
                                                        sbHTML.AppendLine("</body>");
                                                        sbHTML.AppendLine("</html>");
                                                        sw.Write(sbHTML);
      Excel.Application excelapp = new
Excel.Application();
                                                        sw.Close();
       Excel.Workbook workbook =
excelapp.Workbooks.Add();
                                                 Process.Start(System.IO.Directory.GetCurren
      Excel.Worksheet worksheet =
                                                 tDirectory() + @"\" + name + ".html");
workbook.ActiveSheet;
```

```
private void
                                                  титульная_страница, worker_id as
типографииToolStripMenuItem1_Click(obje
                                                  индекс работника, price as
ct sender, EventArgs e)
                                                  цена_экземпляра, "+
                                                                    "calculation as тираж,
       var dt = new DataTable();
                                                  count as количество_листов, (select format
       sql = @"select id,print as
                                                  from formats where id=format id) as формат,
типография, (select type from types where
                                                  (select papertype from papertypes where id =
id=type id) as тип собственности," +
                                                  papertype_id) as вид_бумаги" +
                                                                    ", datastart as
  "(select district from districts where
id=district_id) as район, address as адрес,
                                                  дата_принятия, dataplan as
phone as телефон, year as год_основания
                                                  дата_выполнения_план, datafact as
from prints";
                                                  дата_выполнения_факт, cost as предоплата,
                                                  comment as доп_сведения" +
       try
                                                                    ",account as
                                                  банковский_номер from orders limit 1000";
AuthorizationForm.mainForm.conn.Open();
                                                         try
         AuthorizationForm.mainForm.cmd
= new NpgsqlCommand(sql,
                                                  AuthorizationForm.mainForm.conn.Open();
AuthorizationForm.mainForm.conn);
         dt = new DataTable():
                                                           AuthorizationForm.mainForm.cmd
                                                  = new NpgsqlCommand(sql,
                                                  AuthorizationForm.mainForm.conn);
dt.Load(AuthorizationForm.mainForm.cmd.E
xecuteReader());
                                                           dt = new DataTable();
                                                  dt.Load(AuthorizationForm.mainForm.cmd.E
AuthorizationForm.mainForm.conn.Close();
                                                  xecuteReader());
       catch (Exception ex)
                                                  AuthorizationForm.mainForm.conn.Close();
AuthorizationForm.mainForm.conn.Close();
                                                         catch (Exception ex)
         MessageBox.Show("Error: " +
ex.Message);
                                                  AuthorizationForm.mainForm.conn.Close();
       }
                                                           MessageBox.Show("Error: " +
CreateHTMLFile(Extensions.GetHtml(dt),"ти
                                                  ex.Message);
пографии");
                                                         }
                                                  CreateHTMLFile(Extensions.GetHtml(dt),
    private void
                                                  "заказы");
заToolStripMenuItem_Click(object sender,
                                                       }
EventArgs e)
                                                       private void
                                                  заказчикиToolStripMenuItem1_Click(object
       var dt = new DataTable();
       sql = @"select id, customer_id as
                                                  sender, EventArgs e)
индекс_заказчика, (select producttype from
producttypes where id=producttype_id) as
                                                         var dt = new DataTable();
изделие," +
                                                         sql = @"select id,first_name as имя,
                   " publication as
                                                  second name as фамилия, third name as
                                                  отчество,(select city from cities where
название_издания, picture as
```

```
id=city_id) as город," +
                                                 dt.Load(AuthorizationForm.mainForm.cmd.E
                                                 xecuteReader());
                  "address as адрес,
birthday as дата_рождения, phone as
телефон from customers";
                                                 AuthorizationForm.mainForm.conn.Close();
      try
                                                        catch (Exception ex)
AuthorizationForm.mainForm.conn.Open();
                                                 AuthorizationForm.mainForm.conn.Close();
         AuthorizationForm.mainForm.cmd
                                                          MessageBox.Show("Error: " +
= new NpgsqlCommand(sql,
AuthorizationForm.mainForm.conn);
                                                 ex.Message);
         dt = new DataTable();
dt.Load(AuthorizationForm.mainForm.cmd.E
                                                 CreateHTMLFile(Extensions.GetHtml(dt),
xecuteReader());
                                                 "работники");
AuthorizationForm.mainForm.conn.Close();
                                                      private void
                                                 типыСобственностиТооlStripMenuItem1_Cl
      catch (Exception ex)
                                                 ick(object sender, EventArgs e)
                                                        var dt = new DataTable();
AuthorizationForm.mainForm.conn.Close();
                                                        sql = @"select id,type as
         MessageBox.Show("Error: " +
                                                 Тип_Собственности from types";
ex.Message);
                                                        try
       }
CreateHTMLFile(Extensions.GetHtml(dt),
                                                 AuthorizationForm.mainForm.conn.Open();
"заказчики");
                                                          AuthorizationForm.mainForm.cmd
                                                 = new NpgsqlCommand(sql,
                                                 AuthorizationForm.mainForm.conn);
    private void
работникиToolStripMenuItem1_Click(object
                                                          dt = new DataTable();
sender, EventArgs e)
                                                 dt.Load(AuthorizationForm.mainForm.cmd.E
       var dt = new DataTable();
                                                 xecuteReader());
      sql = @ "select id, first name as имя,
second_name as фамилия, third_name as
отчество, (select print from prints where id =
                                                 AuthorizationForm.mainForm.conn.Close();
print_id) as типография from workers";
      try
                                                        catch (Exception ex)
AuthorizationForm.mainForm.conn.Open();
                                                 AuthorizationForm.mainForm.conn.Close();
                                                          MessageBox.Show("Error: " +
                                                 ex.Message);
         AuthorizationForm.mainForm.cmd
= new NpgsqlCommand(sql,
                                                        }
AuthorizationForm.mainForm.conn);
         dt = new DataTable();
                                                 CreateHTMLFile(Extensions.GetHtml(dt),
                                                 "типы_собственности");
```

```
}
                                                 AuthorizationForm.mainForm.conn);
                                                           dt = new DataTable();
    private void
районыToolStripMenuItem1_Click(object
                                                 dt.Load(AuthorizationForm.mainForm.cmd.E
sender, EventArgs e)
                                                 xecuteReader());
       var dt = new DataTable();
      sql = @"select id,district as Район
                                                 AuthorizationForm.mainForm.conn.Close();
from districts";
      try
                                                        catch (Exception ex)
AuthorizationForm.mainForm.conn.Open();
                                                 AuthorizationForm.mainForm.conn.Close();
                                                           MessageBox.Show("Error: " +
         AuthorizationForm.mainForm.cmd
                                                 ex.Message);
= new NpgsqlCommand(sql,
                                                        }
AuthorizationForm.mainForm.conn);
         dt = new DataTable();
                                                 CreateHTMLFile(Extensions.GetHtml(dt),
                                                 "изделия");
dt.Load(AuthorizationForm.mainForm.cmd.E
xecuteReader()):
                                                      private void
                                                 городаToolStripMenuItem1_Click(object
AuthorizationForm.mainForm.conn.Close();
                                                 sender, EventArgs e)
       catch (Exception ex)
                                                        var dt = new DataTable();
                                                        sql = @"select id,city as Город from
                                                 cities";
AuthorizationForm.mainForm.conn.Close();
                                                        try
         MessageBox.Show("Error: " +
ex.Message);
                                                 AuthorizationForm.mainForm.conn.Open();
       }
CreateHTMLFile(Extensions.GetHtml(dt),
                                                           AuthorizationForm.mainForm.cmd
"районы");
                                                 = new NpgsqlCommand(sql,
                                                 AuthorizationForm.mainForm.conn);
                                                           dt = new DataTable();
    private void
изделияToolStripMenuItem1_Click(object
                                                 dt.Load(AuthorizationForm.mainForm.cmd.E
sender, EventArgs e)
                                                 xecuteReader());
       var dt = new DataTable();
      sql = @ "select id,format as Формат
                                                 AuthorizationForm.mainForm.conn.Close();
from formats";
                                                        catch (Exception ex)
      try
AuthorizationForm.mainForm.conn.Open();
                                                 AuthorizationForm.mainForm.conn.Close();
                                                           MessageBox.Show("Error: " +
         AuthorizationForm.mainForm.cmd
                                                 ex.Message);
= new NpgsqlCommand(sql,
```

```
CreateHTMLFile(Extensions.GetHtml(dt),
                                                          AuthorizationForm.mainForm.cmd
"города");
                                                 = new NpgsqlCommand(sql,
                                                 AuthorizationForm.mainForm.conn);
    }
                                                          dt = new DataTable();
    private void
банкиToolStripMenuItem1_Click(object
                                                 dt.Load(AuthorizationForm.mainForm.cmd.E
sender, EventArgs e)
                                                 xecuteReader());
      var dt = new DataTable();
      sql = @"select id,bank as Банк from
                                                 AuthorizationForm.mainForm.conn.Close();
banks";
                                                        catch (Exception ex)
      try
AuthorizationForm.mainForm.conn.Open();
                                                 AuthorizationForm.mainForm.conn.Close();
                                                          MessageBox.Show("Error: " +
         AuthorizationForm.mainForm.cmd
                                                 ex.Message);
= new NpgsqlCommand(sql,
AuthorizationForm.mainForm.conn);
         dt = new DataTable():
                                                 CreateHTMLFile(Extensions.GetHtml(dt),
                                                 "банковские_ноиера");
dt.Load(AuthorizationForm.mainForm.cmd.E
xecuteReader());
                                                      private void
                                                 видыБумагиТoolStripMenuItem1_Click(obje
                                                 ct sender, EventArgs e)
AuthorizationForm.mainForm.conn.Close();
      catch (Exception ex)
                                                        var dt = new DataTable();
                                                        sql = @"select id,papertype as
                                                 Вид_бумаги, density from papertypes";
AuthorizationForm.mainForm.conn.Close();
                                                        try
         MessageBox.Show("Error: " +
ex.Message);
                                                 AuthorizationForm.mainForm.conn.Open();
       }
CreateHTMLFile(Extensions.GetHtml(dt),
                                                          AuthorizationForm.mainForm.cmd
"банки");
                                                 = new NpgsqlCommand(sql,
                                                 AuthorizationForm.mainForm.conn);
                                                          dt = new DataTable();
    private void
банковскиеНомераToolStripMenuItem1_Cli
                                                 dt.Load(AuthorizationForm.mainForm.cmd.E
ck(object sender, EventArgs e)
                                                 xecuteReader());
       var dt = new DataTable();
      sql = @"select * from
                                                 AuthorizationForm.mainForm.conn.Close();
accounts_select()";
                                                        catch (Exception ex)
      try
AuthorizationForm.mainForm.conn.Open();
                                                 AuthorizationForm.mainForm.conn.Close();
```

```
MessageBox.Show("Error: " +
                                                 new StringWriter())
ex.Message);
                                                          using (HtmlTextWriter htmlWriter
       }
                                                 = new HtmlTextWriter(stringWriter))
CreateHTMLFile(Extensions.GetHtml(dt),
"виды_бумаги");
                                                             using (var htmlTable = new
                                                 HtmlTable())
                                                               htmlTable.Border = 1;
    private void
форматыToolStripMenuItem1_Click(object
                                                               // Add table header row
sender, EventArgs e)
                                                               using (var headerRow = new
                                                 HtmlTableRow())
       var dt = new DataTable();
       sql = @"select id,format as Формат
                                                                 foreach (DataColumn
from formats";
                                                 dataColumn in dataTable.Columns)
      try
                                                                    using (var htmlColumn =
                                                 new HtmlTableCell())
AuthorizationForm.mainForm.conn.Open();
                                                                      htmlColumn.InnerText
         AuthorizationForm.mainForm.cmd
                                                 = dataColumn.ColumnName:
= new NpgsqlCommand(sql,
AuthorizationForm.mainForm.conn);
                                                 headerRow.Cells.Add(htmlColumn);
         dt = new DataTable();
dt.Load(AuthorizationForm.mainForm.cmd.E
xecuteReader());
                                                 htmlTable.Rows.Add(headerRow);
                                                               // Add data rows
                                                               foreach (DataRow row in
AuthorizationForm.mainForm.conn.Close();
                                                 dataTable.Rows)
      catch (Exception ex)
                                                                 using (var htmlRow = new
                                                 HtmlTableRow())
AuthorizationForm.mainForm.conn.Close();
         MessageBox.Show("Error: " +
                                                                    foreach (DataColumn
ex.Message);
                                                 column in dataTable.Columns)
                                                                      using (var htmlColumn
                                                 = new HtmlTableCell())
CreateHTMLFile(Extensions.GetHtml(dt),
"форматы");
    }
                                                 htmlColumn.InnerText =
  public static class Extensions
                                                 row[column].ToString();
    public static String GetHtml(this
                                                 htmlRow.Cells.Add(htmlColumn);
DataTable dataTable)
       StringBuilder sbControlHtml = new
StringBuilder();
                                                 htmlTable.Rows.Add(htmlRow);
      using (StringWriter stringWriter =
                                                                 }
```