|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ»

ДИСЦИПЛИНА «БАЗЫ ДАННЫХ»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_Информационная система \_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_отделения ГИБДД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Руководитель курсового проекта **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М. М. Фомин**

(Подпись, дата)

Студент ИУ6-43 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.А. Шашкин**

(Подпись, дата)

*2018 г*

**Оглавление**

Техническое задание……………………………………………………………...3

Глава 1. Проектирование…………………………………………………………5

1.1 Анализ предметной области…………………………………………..5

1.2 Бизнес процессы пользователей информационной системы…….…6

1.3 Выделение сущностей…………………………………………………7

1.4 Проектирование схемы базы данных…………………………………8

1.5 Определение ролей пользователей……………………………………9

Глава 2. Реализация………………………………………………………...……10

2.1 Пользовательская аутентификация………………………………….10

2.2 Создание отчетов……………………………………………………...11

2.3 Добавление нового инспектора………………………………………13

2.4 Создание штрафа и протокола ДТП…………………………………15

2.5 Постраничная авторизация…………………………………………..16

Глава 3. Инструкция пользователя «Инспектор»……………………………..18

Заключение………………………………………………………………………24

Приложения……………………………………………………………………...25

Приложение А…………………………………………………………….25

Приложение Б……………………………………………………………..35

Приложение В……………………………………………………………..38

**Техническое задание**

1. Введение
   1. Полное название

Информационная система отделения ГИДББ

* 1. Общие сведения

Система предназначена для автоматизации сбора данных о сотрудниках отделения, ПДД, правонарушениях и ДТП, облегчения регистрации ДТП, а также создания отчётов и эффективного сбора статистики.

2.Основание для разработки

Учебный план кафедры ИУ6 МГТУ им. Баумана.

1. Назначения и цели создания системы

Система должна стать инструментом информационной поддержки сотрудников ГАИ, обеспечить возможность регистрации ДТП, позволить собирать эффективную статистику о ДТП и их участниках, вести учёт штрафов, выписанных сотрудниками ГИБДД.

Основными целями создания системы являются:

* Создание единый базы водителей и инспекторов
* Создание полной базы правил дорожного движения
* Формирование статистики ДТП
* Формирование протокола ДТП
* Снижение времени регистрации ДТП

1. Общие требования к системе

4.1 Требования к функциональным характеристикам

Система должна обеспечивать автоматизацию следующих функций:

* Регистрация правонарушений, формирование протокола ДТП и штрафов, а так же их вывод на печать – только для инспектора
* Формирование статистики о ДТП (по округу или району, правонарушениям), об инспекторах (по количеству зарегистрированных ДТП и выписанных штрафах), о штрафах (по округу или району, правонарушениям, определенным инспекторам) – только руководитель отделения
* Ввод в систему новых инспекторов– только руководитель отделения
* Заносить в базу данных новых водителей и новые транспортные средства – только инспектор-оператор
  1. Требования к надёжности

Система должна обеспечить надежное хранение информации (защиту от сбоев, защиту от несанкционированного доступа).

* 1. Требования к составу и параметрам технических программных средств

Компьютер поддерживающий операционную систему Windows 7 и выше.

1. Требования к программной документации

В состав документации входят пояснительная записка, руководство пользователя, альбом печатных форм.

1. Порядок оформления и предъявления результатов работы

Результаты представляются в виде отлаженной информационной системы, выполняющей функции данного ТЗ.

**Глава 1. Проектирование**

**1.1 Анализ предметной области**

Проанализируем техническое задание для понимания, какие бизнес-процессы будут реализованы в данной системе и какие данные должны находиться в базе данных. Воспользуемся примером справки о ДТП (рис. 1).

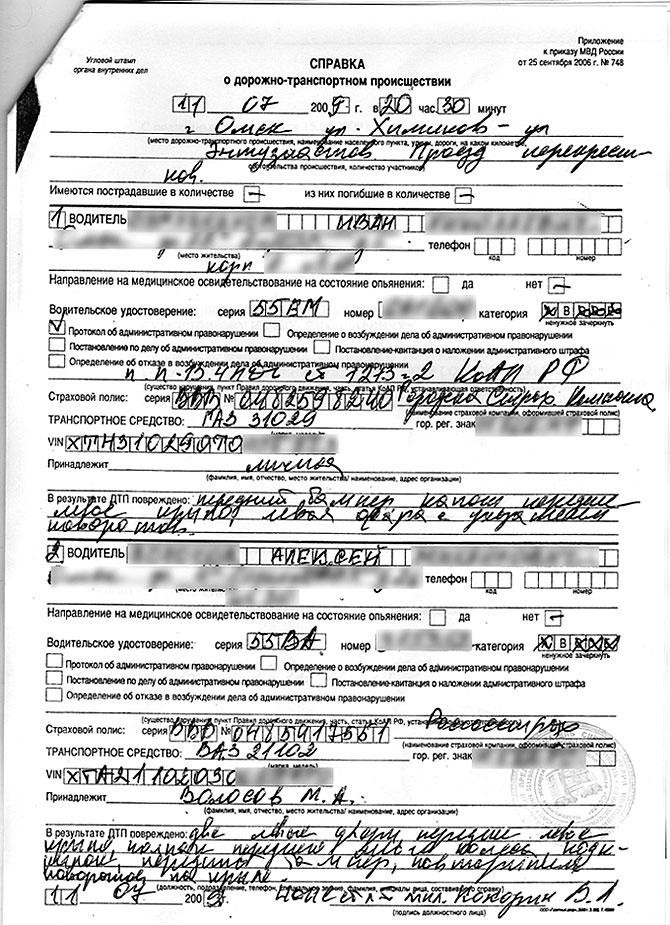


Рис. 1 Справка о ДТП

* + 1. ПДД

В данном разделе необходима таблица правил дорожного движения, чтобы инспектор при составлении протокола ДТП имел возможность удостовериться, какой именно пункт ПДД нарушили те или иные участники дорожного движения.

Кроме того, необходима отдельная таблица со штрафами (т.к. наказания за нарушения ПДД прописаны в КоАП) для того, чтобы инспектор мог выбирать по какому пункту выписывается штраф и сумму штрафа из предложенных программой, а не самостоятельно вбивать эту информацию, что значительно увеличит скорость выписки штрафа.

* + 1. Водители

Для учета водителей необходима одна таблица, в которой будет содержаться информация о водителях, их паспортных данных и водительских удостоверениях.

* + 1. Транспортные средства

В данном разделе будет вестись учет транспортных средств, номерных знаков, владельцев автомобилей и страховых полисов.

* + 1. Инспектор ГИБДД

В базе данных должна храниться информация о сотруднике ГИБДД: номер трудовой книжки, звание, зарплата.

* + 1. Штрафы и протоколы

В данном разделе необходим учет штрафов, выписанных инспекторами. Кроме того, необходимо хранение протоколов ДТП.

* 1. **Бизнес процессы пользователей информационной системы**

Для выделения сущностей базы данных необходим анализ бизнес процессов пользователей.

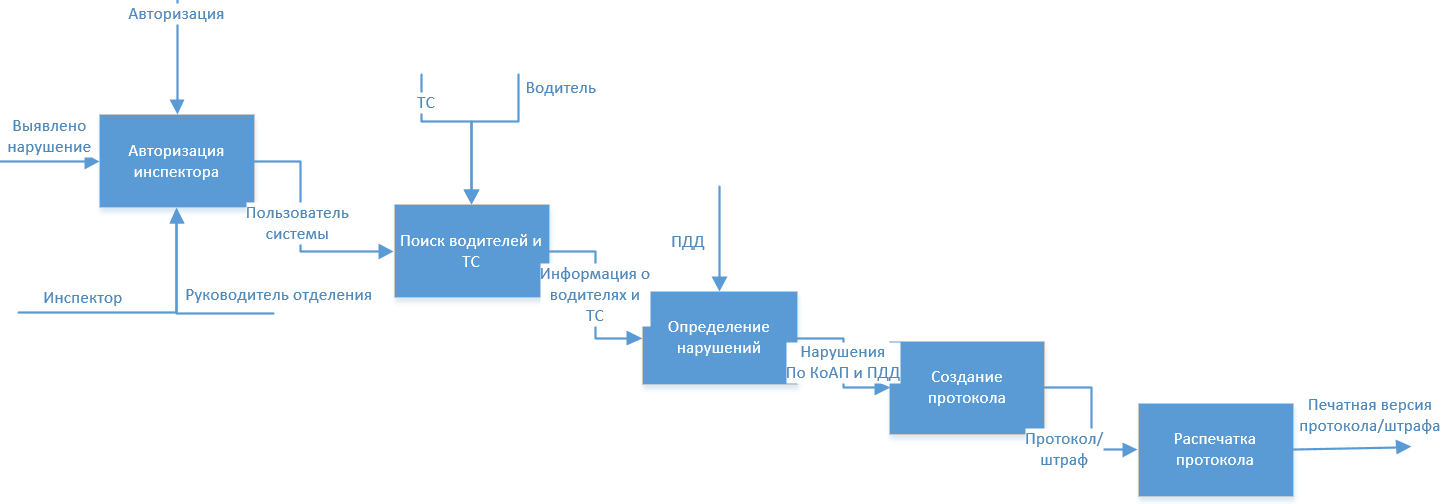


Рис. 2 БП «Создание протокола ДТП и выписка штрафов»

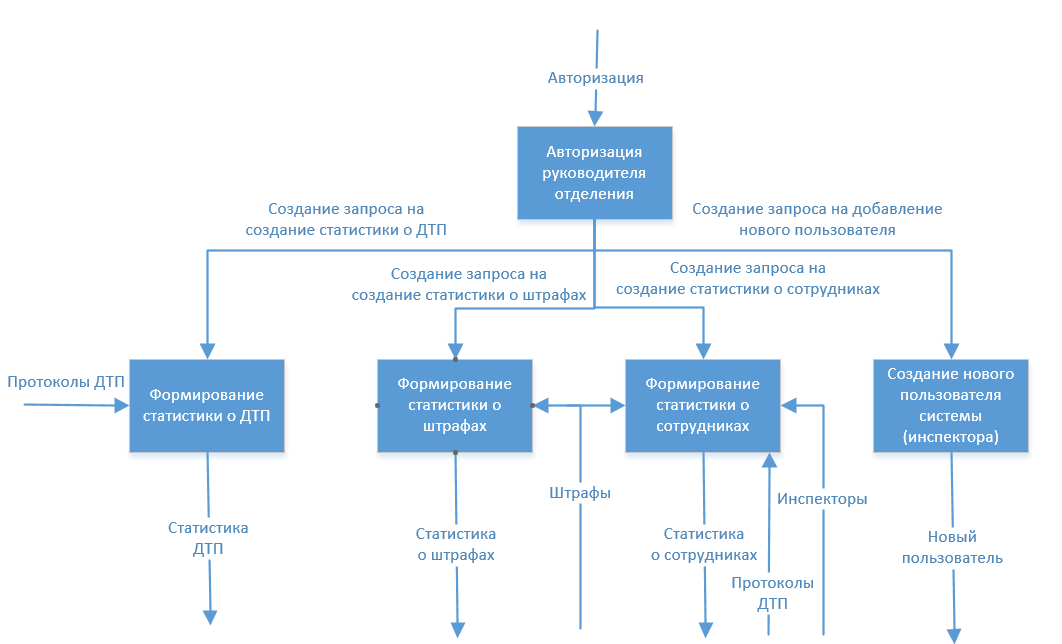


Рис. 3 Диаграмма уникальных возможностей пользователя «Руководитель отделения»

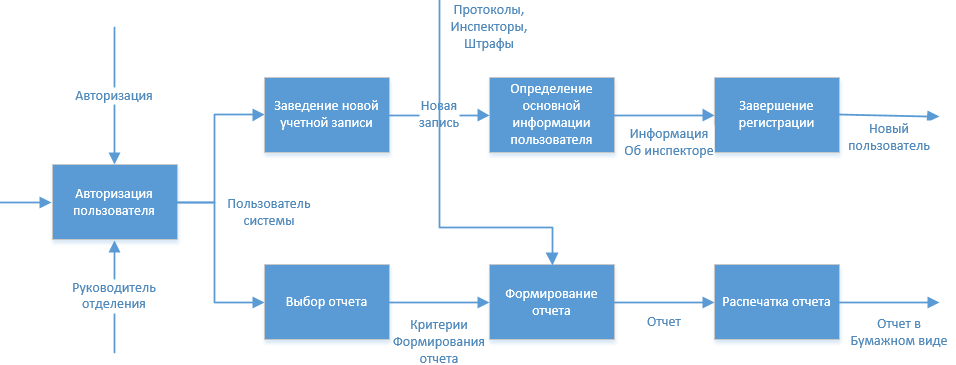


Рис. 4 БП пользователя «Руководитель отделения»

* 1. **Выделение сущностей**

По бизнес процессам на рис. 2 и 3 можно выделить следующие сущности, с которыми работает инспектор и руководитель отделения:

* Транспортные средства
* Водители
* Инспекторы
* ПДД
* КоАП
* Штрафы
* Протоколы ДТП

В сущностях «водители» и «инспекторы» будет встречаться повторяющаяся группа атрибутов (ФИО, паспортные данные, номер водительского удостоверения), поэтому оставим эту группу в сущности «водитель», а в «инспекторы» будет храниться только информация с места работы.

В нескольких сущностях будет встречается информация о регионе правонарушения, поэтому он будет выделен в отдельную сущность «регионы».

Кроме того, для создания сервиса авторизации и аутентификации понадобятся отдельные сущности: «пользователь-роль» и «роль-страница» (для того, чтобы только определенный пользователь имел доступ только к определенным страницам).

* 1. **Проектирование схемы базы данных**

Во время проектирования схемы выяснилось, что регион правонарушения встречается в нескольких таблицах, поэтому была создана новая таблица, чтобы не хранить одни и те же данные несколько раз.

Во всех таблицах первичные ключи-суррогатные.

На рисунке 5 представлена итоговая схема базы данных. Код формирования всех объектов БД представлена в приложении А.

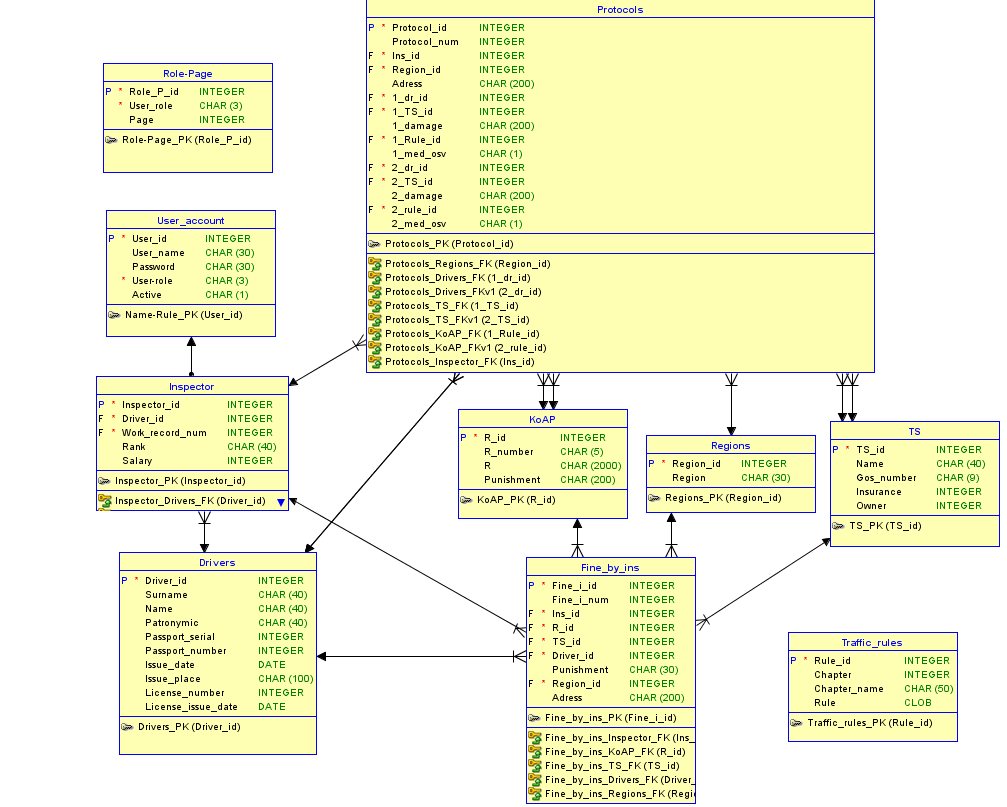


Рис. 5 Схема базы данных

**1.5 Определение ролей пользователей**

Техническое задание подразумевает три типа пользователей:

1). Инспектор – может выписывать штрафы, составлять протоколы, получать их печатные формы.

2). Руководитель отделения(администратор) – получает доступ к статистике, оговоренной в ТЗ и получать их в печатном виде, может создавать нового пользователя

3). Оператор – может заносить в базу данных новых водителей и новые транспортные средства.

Кроме того, все пользователи имею доступ к справочнику КоАП и ПДД

**Глава 2. Реализация**

**2.1 Пользовательская аутентификация**

В связи с тем, выбрана среда разработки – ORACLE APEX 5.1.4 необходимо создать в программе пользовательскую аутентификацию, так как стандартная – работает только с пользователя APEX, что не удовлетворяет требованиям к разрабатываемой системе.

Страница аутентификации (101 Login page) была сгенерирована автоматически при создании программы, поэтому воспользуемся ей. Создание новой таблицы не обязательно, так как при проектировании мы предусмотрели таблицу USER\_ACCOUNT, в которой хранятся все необходимые данные.

Далее, создадим пакет для обработки логина и пароля пользователя (код представлен приложении Б). Кроме того, добавим функцию, которая покажет пользователю причину, по которой он не смог получить доступ к своему аккаунту.

Затем, добавим в систему сообщения об ошибках, добавим на страницу блок для вывода сообщения об ошибке (рис. 6) и изменим процесс Login (рис. 7) на новую схему авторизации

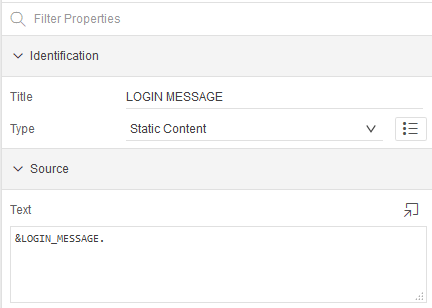


Рис. 6 Настройки блока вывода ошибки

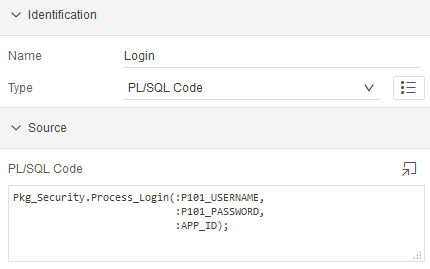


Рис. 7 Настройка процесса Login

**2.2 Создание отчетов**

Все отчеты, которые доступны администратору, создаются по одному и тому же алгоритму, поэтому возможно описание только одного из них. Рассмотрим страницу «Отчет по инспекторам». Создаем Classic report и заполняем SQL Query так, чтобы нам выводилась необходимая статистика (рис. 8)

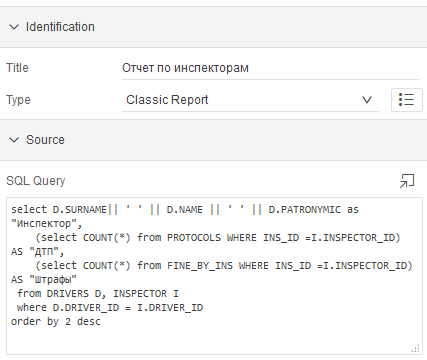


Рис. 8 Настройки блока «отчет по инспекторам»

Для реализации возможности создания печатных форм воспользуемся программой Oracle BI Publisher. Подключим к APEX сервер преобразования документов из формата XML в формат PDF, используемый для печати. Далее во вкладке «Shared components» добавим очередь и шаблон для создания формы (рис. 9). Теперь, создав кнопку со ссылкой на шаблон, мы можем получить печатную форму (рис. 10).

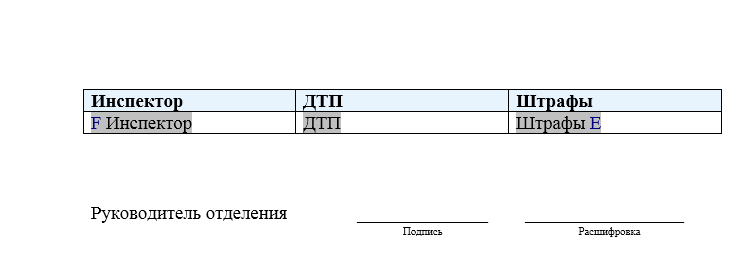


Рис. 9 Шаблон «Отчет по инспекторам»

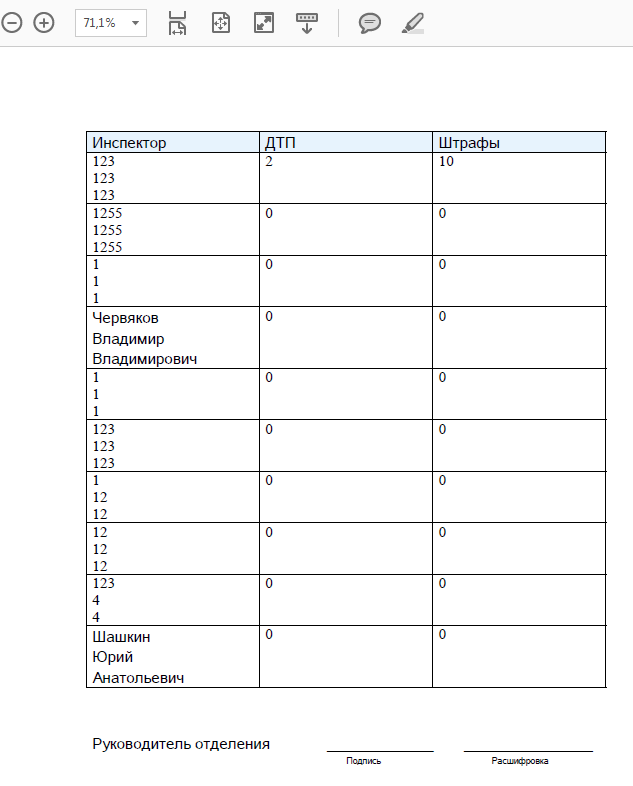


Рис. 10 Печатная форма отчета

**2.3 Добавление нового инспектора**

Для добавления нового инспектора создана страница (№6) на который есть возможность добавить нового инспектора с условием, что в БД уже есть информация о водителе (рис. 11), а так же добавить инспектора, который получал водительское удостоверение в другом отделении ГИБДД (рис. 12).

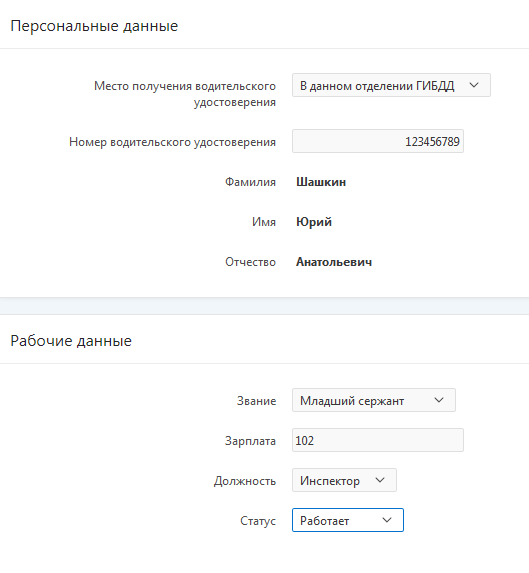


Рис. 11 Добавление инспектора при условии получения вод. уд. в данном ГИБДД

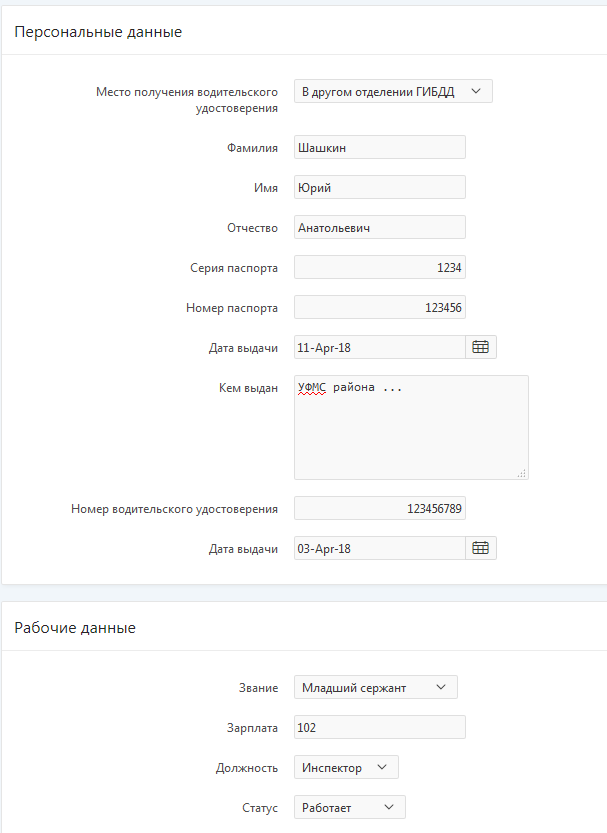


Рис. 12 Добавление инспектора при условии получения вод. уд. в другом ГИБДД

Процесс добавления инспектора в БД реализуется процессом New, код которого представлен на рисунке 13

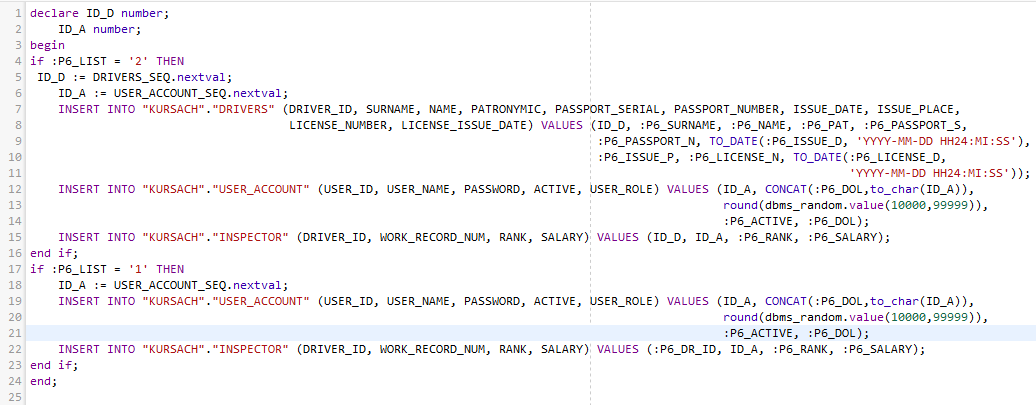


Рис. 13 Код процесса добавления нового инспектора в БД

**2.4 Создание штрафа и протокола ДТП**

В связи с тем, что логика создания обеих страниц одинакова, рассмотрим создание только одной – страницы штрафов.

На данной странице реализован выбор статьи, которую нарушил водитель, с получением полного текста статьи и возможных наказаний (рис. 14). Затем инспектору необходимо ввести номер автомобиля нарушителя, затем инспектору предоставляется выбор водителя, находящегося за рулем ТС, так как за рулем может находиться другой водитель. Данная функция реализуется через «Dynamic action», за счет которого мы можем изменять свойства и значения полей страницы.

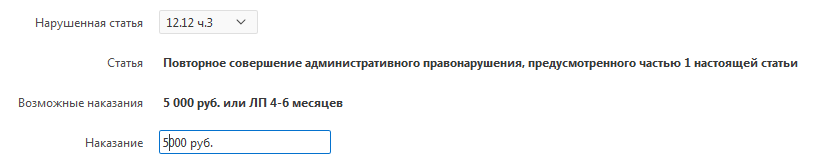


Рис. 14 Выбор статьи КоАП и наказания

После создания штрафа пользователь должен быть перенаправлен на другую страницу для получения печатной формы штрафа. Страница реализуется через «Classic report» (рис. 15), а получение печатной формы – через сервер BI Publisher.

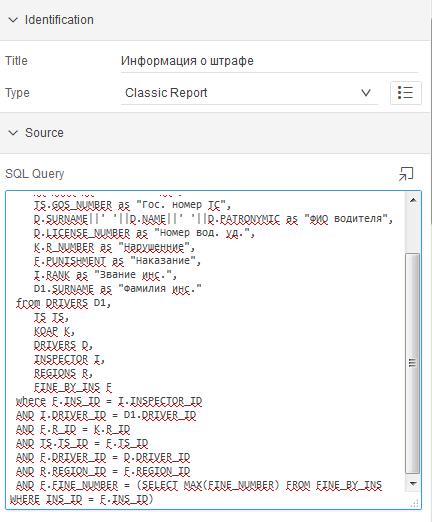


Рис. 15 Свойства таблицы с отчетом о штрафе

**2.5 Постраничная авторизация**

Для создания постраничной авторизации была создана таблица «Role-Page», где показано, какому пользователю разрешен доступ к выбранной странице. Воспользуемся APEX представлением APEX\_APPLICATION\_LIST\_ENTRIES, в котором есть информация, какие страницы прикреплены к каким объектам навигации. Для того, чтобы не поддерживать отдельно еще одну таблицу со связью Роль -> Объект навигации, создадим представление, связывающее роли и List Entries на основе связи Роль –> Страница (код представлен в приложении Б). Затем изменим навигационное меню в компонентах программы, чтобы каждый пользователь видел только те страницы, к котором имеет доступ. Для примера, на рисунке 16 представлено навигационное меню руководителя отделения.

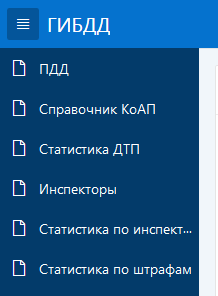


Рис. 16 Навигационное меню руководителя отделения

**Глава 3. Инструкция пользователя «Инспектор»**

**Введение**

Данная программа разработана для облегчения регистрации ДТП, выписывания штрафов и просмотра справочников ПДД и КоАП. Для доступа к ресурсам программы Вам необходимо выполнить аутентификацию на странице авторизации (рис. 1) с использованием логина и пароля, которые Вам были выданы руководителем отделения. Если логин или пароль введены неверно или имеются какие-либо проблемы с Вашей учетной записью, Вы получите сообщение с дальнейшими указаниями для решения проблемы.

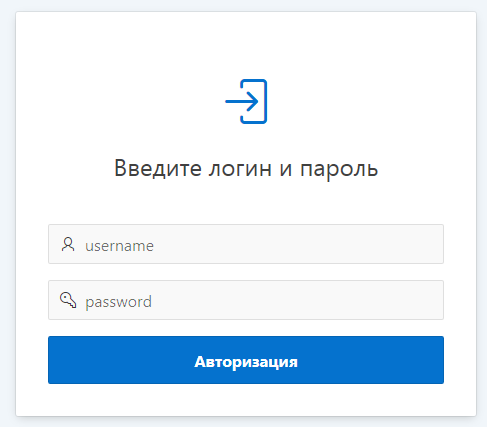


Рис. 1 Страница авторизации

После успешного выполнения аутентификации вы увидите начальную страницу и меню с Вашими возможностями в левой части страницы (рис. 2)

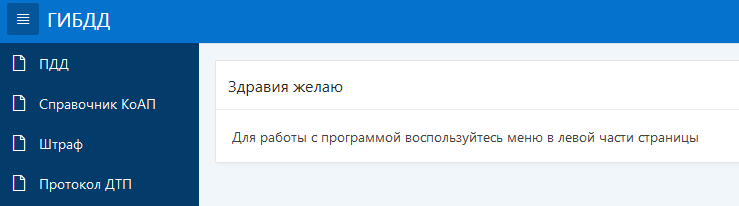


Рис. 2 Начальная страница и навигационное меню

**Использование справочников**

Для использования справочника ПДД и КоАП нажмите на соответствующую кнопку навигационного меню (рис. 2). Затем, вы попадете на страницу справочника, где представлена таблица, а так же строка поиска по таблице (рис. 3). Для использования поиска просто введите фразу для поиска и нажмите кнопку «Go»

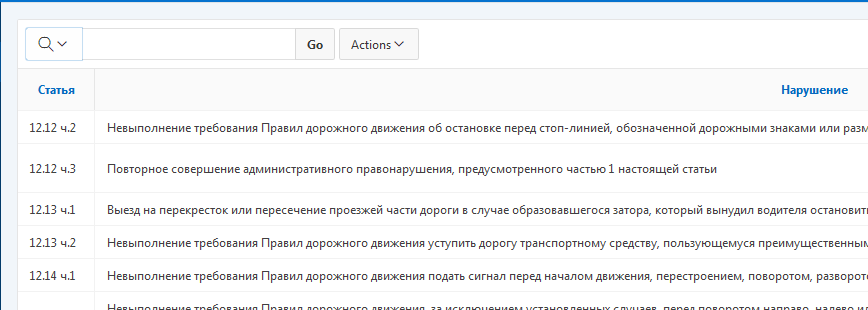


Рис. 3 Поиск по справочнику

**Выписка штрафа**

Для того, чтобы попасть на страницу выписки штрафа нажмите на соответствующую вкладку в навигационном меню (рис. 2). Далее, Вы попадаете на страницу штрафа. Заполнять страницу следует в следующем порядке:

1. Выбор нарушенной статьи и штрафа. Вам следует выбрать нарушенную водителем статью, затем Вам будет представлена формулировка статьи и возможное наказания (рис. 4).

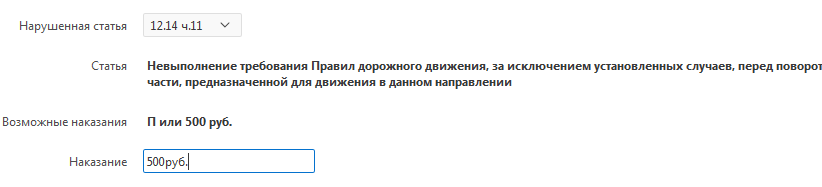


Рис. 4 1 пункт заполнения страницы штрафов

2. Ввод данных о ТС и водителе. Вам следует заполнить поле «Номер ТС», чтобы получить информацию о человеке, на которого зарегистрирован данный автомобиль. Если за рулем находился данный человек, то оставьте следующие поля без изменений, в ином случае поставьте галочку в поле «водитель» на строку «За рулем находился другой водитель» и заполните поле «Номер водительского удостоверения» (рис. 5).

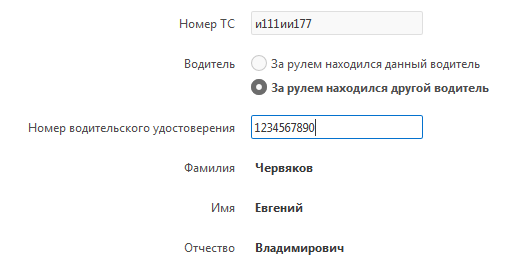


Рис. 5 2 пункт заполнения страницы штрафов

3. Ввод адреса. Выберете округ правонарушения, а затем введите полный адрес (рис. 6).

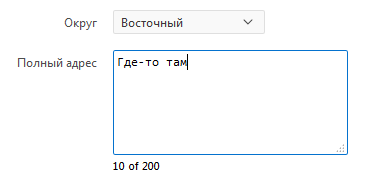


Рис. 6 3 пункт заполнения страницы штрафов

После заполнения страницы штрафа, нажмите на кнопку «Сохранить». Вы попадете на страницу выписки штрафа (рис. 7), где вы увидите информацию о выписанном штрафе и кнопку для получения печатной формы. При нажатии на кнопку «Печать» вы получите штраф в формате PDF, который можно распечатать.

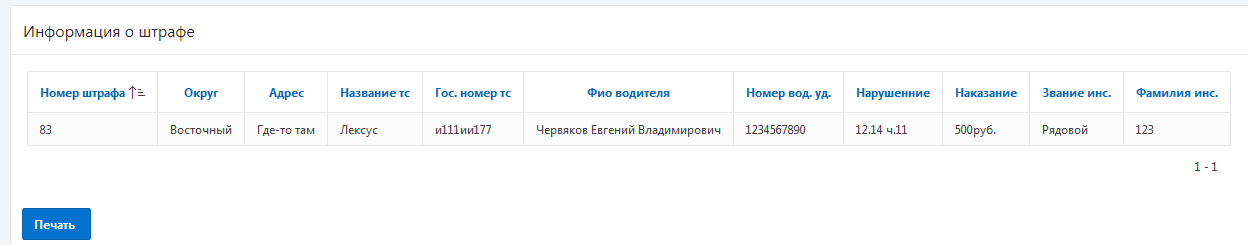


Рис. 7 Выписка штрафа

**Протокол ДТП**

Для того, чтобы попасть на страницу протокола ДТП нажмите на соответствующую вкладку в навигационном меню (рис. 2). Далее, Вы попадаете на страницу регистрации ДТП. Заполнять страницу следует в следующем порядке:

1. Заполнить блок с общей информацией – вид ДТП (с участием одного или двух водителей), округ и адрес (рис. 8).

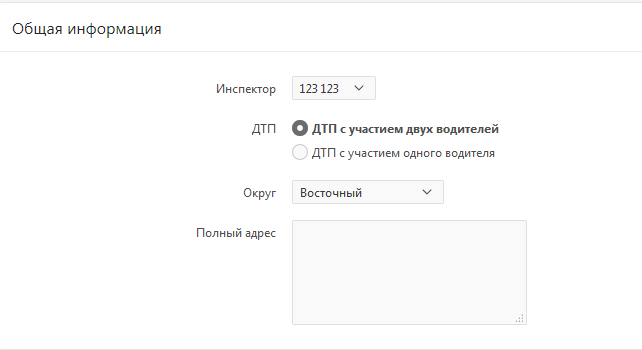


Рис. 8 Блок с общей информацией

2. Заполнить блок с информацией о первом водителе (рис.9). В первую очередь необходимо выбрать, нарушил ли водитель ПДД. Если да, то заполняется информация о нарушенном правиле (по схеме, описанной в «выписке штрафа» в пункте 1), ТС и водителе (по схеме, описанной в «выписке штрафа» в пункте 2). Если нет, то заполняется только информация о ТС и водителе. Затем выставляется статус направления водителя на медицинское освидетельствование и полный список поврежденных деталей автомобиля.

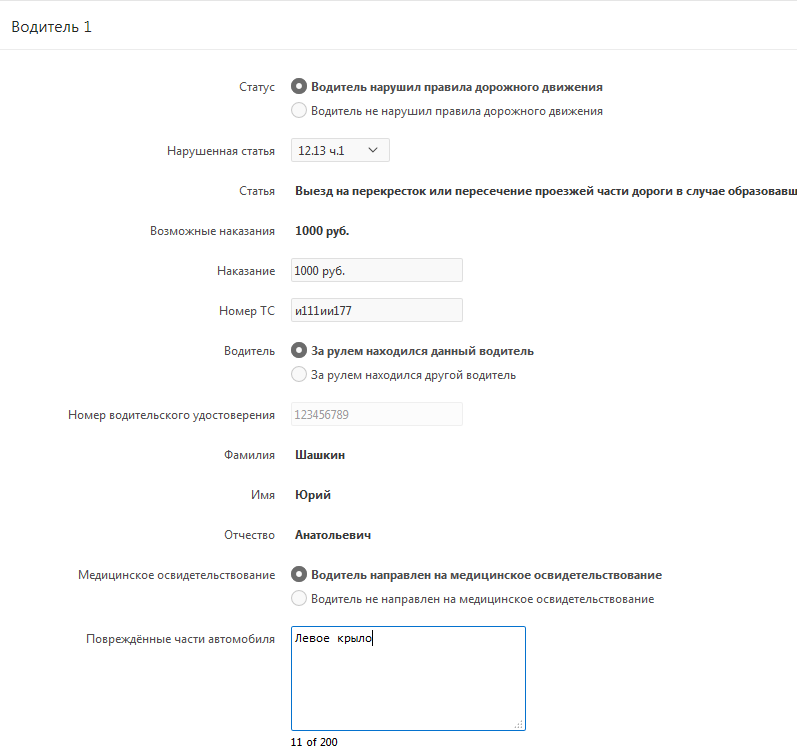


Рис. 9 Блок с информацией о первом водителе

3. Заполнить блок с информацией о втором водителе/поврежденных объектах. Если в ДТП участвовали два водителя, то информация о втором водителе заполняется по тому же алгоритму, что и о первом. Если ДТП с участием одного водителя, то заполняется только информация о поврежденных объектах местности.

4. Для завершения регистрации ДТП необходимо нажать на кнопку сохранить. После нажатия вы попадете на страницу с протоколом ДТП. Для получения печатной версии протокола нажмите на кнопку печать. Пример печатной версии протокола представлен на рисунке 10.

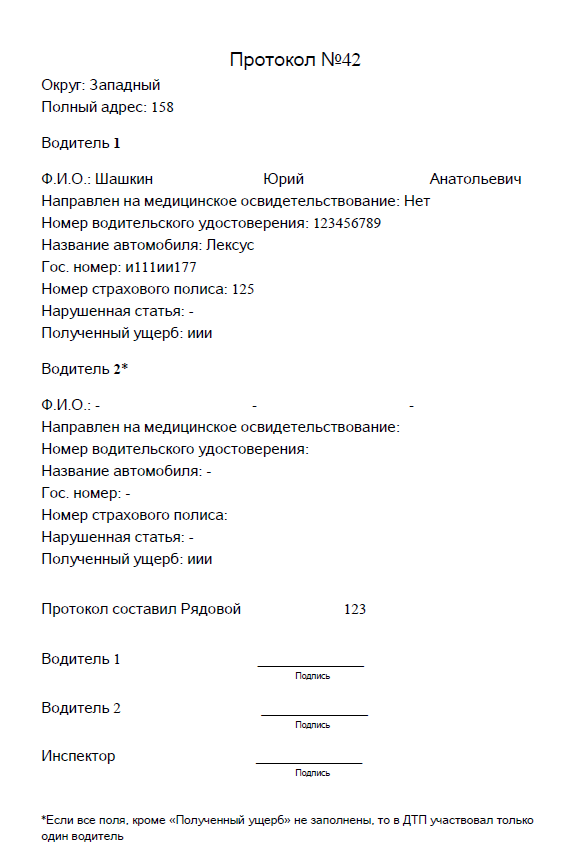


Рис. 10 Печатная версия протокола ДТП

**Заключение**

В результате работы была создана информационная система отделения ГИБДД, в основе которой находится база данных. Основа работы базы данных – СУБД Oracle Database Express Edition 11g, интерфейсы разработаны при помощи Oracle APEX версии 5.1, а печатные формы разработаны с помощью Oracle BI Publisher.

* Во время выполнения работы были проведены следующие действия:
* Исследование предметной области, изучение возможных сущностей и связей между ними;
* Проектирование базы данных, её реализация и доработка;
* Разработка интерфейсов пользователей;
* Разработка методов авторизации и доступа для разных типов пользователей;
* Подключение и настройка сервиса для создания печатных форм;

Использование данной системы значительно облегчит работу отделения ГИБДД, в котором она будет введена, тем самым использование данной системы достаточно перспективно. Однако для полного функционала отделения ГИБДД данной системы недостаточно, так как требуются и другие функции, которых в данной работе нет (например, регистрация мототранспорных средств и выдача водительских удостоверений). Следовательно, можно сделать вывод, что использование данной системы возможно, но её функций недостаточно для полноценной работы отделения ГИБДД.

**Приложения**

**Приложение А**

Скрипт, формирующий объекты БД

-- Generated by Oracle SQL Developer Data Modeler 17.2.0.188.1059

-- at: 2018-04-12 18:05:00 MSK

-- site: Oracle Database 11g

-- type: Oracle Database 11g

CREATE TABLE drivers (

driver\_id INTEGER NOT NULL,

surname CHAR(40),

name CHAR(40),

patronymic CHAR(40),

passport\_serial INTEGER,

passport\_number INTEGER,

issue\_date DATE,

issue\_place CHAR(100),

license\_number INTEGER,

license\_issue\_date DATE

);

ALTER TABLE drivers ADD CONSTRAINT drivers\_pk PRIMARY KEY ( driver\_id );

CREATE TABLE fine\_by\_ins (

fine\_i\_id INTEGER NOT NULL,

fine\_i\_num INTEGER,

ins\_id INTEGER NOT NULL,

r\_id INTEGER NOT NULL,

ts\_id INTEGER NOT NULL,

driver\_id INTEGER NOT NULL,

punishment CHAR(30),

region\_id INTEGER NOT NULL,

adress CHAR(200)

);

ALTER TABLE fine\_by\_ins ADD CONSTRAINT fine\_by\_ins\_pk PRIMARY KEY ( fine\_i\_id );

CREATE TABLE inspector (

inspector\_id INTEGER NOT NULL,

driver\_id INTEGER NOT NULL,

work\_record\_num INTEGER NOT NULL,

rank CHAR(40),

salary INTEGER

);

ALTER TABLE inspector ADD CONSTRAINT inspector\_pk PRIMARY KEY ( inspector\_id );

CREATE TABLE koap (

r\_id INTEGER NOT NULL,

r\_number CHAR(5),

r CHAR(2000),

punishment CHAR(200)

);

ALTER TABLE koap ADD CONSTRAINT koap\_pk PRIMARY KEY ( r\_id );

CREATE TABLE protocols (

protocol\_id INTEGER NOT NULL,

protocol\_num INTEGER,

ins\_id INTEGER NOT NULL,

region\_id INTEGER NOT NULL,

adress CHAR(200),

"1\_dr\_id" INTEGER NOT NULL,

"1\_TS\_id" INTEGER NOT NULL,

"1\_damage" CHAR(200),

"1\_Rule\_id" INTEGER NOT NULL,

"1\_med\_osv" CHAR(1),

"2\_dr\_id" INTEGER NOT NULL,

"2\_TS\_id" INTEGER NOT NULL,

"2\_damage" CHAR(200),

"2\_rule\_id" INTEGER NOT NULL,

"2\_med\_osv" CHAR(1)

);

ALTER TABLE protocols ADD CONSTRAINT protocols\_pk PRIMARY KEY ( protocol\_id );

CREATE TABLE regions (

region\_id INTEGER NOT NULL,

region CHAR(30)

);

ALTER TABLE regions ADD CONSTRAINT regions\_pk PRIMARY KEY ( region\_id );

CREATE TABLE "Role-Page" (

role\_p\_id INTEGER NOT NULL,

user\_role CHAR(3) NOT NULL,

page INTEGER

);

ALTER TABLE "Role-Page" ADD CONSTRAINT "Role-Page\_PK" PRIMARY KEY ( role\_p\_id );

CREATE TABLE traffic\_rules (

rule\_id INTEGER NOT NULL,

chapter INTEGER,

chapter\_name CHAR(50),

rule CLOB

);

ALTER TABLE traffic\_rules ADD CONSTRAINT traffic\_rules\_pk PRIMARY KEY ( rule\_id );

CREATE TABLE ts (

ts\_id INTEGER NOT NULL,

name CHAR(40),

gos\_number CHAR(9),

insurance INTEGER,

owner INTEGER

);

ALTER TABLE ts ADD CONSTRAINT ts\_pk PRIMARY KEY ( ts\_id );

CREATE TABLE user\_account (

user\_id INTEGER NOT NULL,

user\_name CHAR(30),

password CHAR(30),

"User-role" CHAR(3) NOT NULL,

active CHAR(1)

);

ALTER TABLE user\_account ADD CONSTRAINT "Name-Rule\_PK" PRIMARY KEY ( user\_id );

ALTER TABLE fine\_by\_ins

ADD CONSTRAINT fine\_by\_ins\_drivers\_fk FOREIGN KEY ( driver\_id )

REFERENCES drivers ( driver\_id );

ALTER TABLE fine\_by\_ins

ADD CONSTRAINT fine\_by\_ins\_inspector\_fk FOREIGN KEY ( ins\_id )

REFERENCES inspector ( inspector\_id );

ALTER TABLE fine\_by\_ins

ADD CONSTRAINT fine\_by\_ins\_koap\_fk FOREIGN KEY ( r\_id )

REFERENCES koap ( r\_id );

ALTER TABLE fine\_by\_ins

ADD CONSTRAINT fine\_by\_ins\_regions\_fk FOREIGN KEY ( region\_id )

REFERENCES regions ( region\_id );

ALTER TABLE fine\_by\_ins

ADD CONSTRAINT fine\_by\_ins\_ts\_fk FOREIGN KEY ( ts\_id )

REFERENCES ts ( ts\_id );

ALTER TABLE inspector

ADD CONSTRAINT inspector\_drivers\_fk FOREIGN KEY ( driver\_id )

REFERENCES drivers ( driver\_id );

ALTER TABLE inspector

ADD CONSTRAINT inspector\_user\_account\_fk FOREIGN KEY ( work\_record\_num )

REFERENCES user\_account ( user\_id );

ALTER TABLE protocols

ADD CONSTRAINT protocols\_drivers\_fk FOREIGN KEY ( "1\_dr\_id" )

REFERENCES drivers ( driver\_id );

ALTER TABLE protocols

ADD CONSTRAINT protocols\_drivers\_fkv1 FOREIGN KEY ( "2\_dr\_id" )

REFERENCES drivers ( driver\_id );

ALTER TABLE protocols

ADD CONSTRAINT protocols\_inspector\_fk FOREIGN KEY ( ins\_id )

REFERENCES inspector ( inspector\_id );

ALTER TABLE protocols

ADD CONSTRAINT protocols\_koap\_fk FOREIGN KEY ( "1\_Rule\_id" )

REFERENCES koap ( r\_id );

ALTER TABLE protocols

ADD CONSTRAINT protocols\_koap\_fkv1 FOREIGN KEY ( "2\_rule\_id" )

REFERENCES koap ( r\_id );

ALTER TABLE protocols

ADD CONSTRAINT protocols\_regions\_fk FOREIGN KEY ( region\_id )

REFERENCES regions ( region\_id );

ALTER TABLE protocols

ADD CONSTRAINT protocols\_ts\_fk FOREIGN KEY ( "1\_TS\_id" )

REFERENCES ts ( ts\_id );

ALTER TABLE protocols

ADD CONSTRAINT protocols\_ts\_fkv1 FOREIGN KEY ( "2\_TS\_id" )

REFERENCES ts ( ts\_id );

CREATE SEQUENCE drivers\_driver\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER drivers\_driver\_id\_trg BEFORE

INSERT ON drivers

FOR EACH ROW

BEGIN

:new.driver\_id := drivers\_driver\_id\_seq.nextval;

END;

/

CREATE SEQUENCE fine\_by\_ins\_fine\_i\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER fine\_by\_ins\_fine\_i\_id\_trg BEFORE

INSERT ON fine\_by\_ins

FOR EACH ROW

BEGIN

:new.fine\_i\_id := fine\_by\_ins\_fine\_i\_id\_seq.nextval;

END;

/

CREATE SEQUENCE inspector\_inspector\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER inspector\_inspector\_id\_trg BEFORE

INSERT ON inspector

FOR EACH ROW

BEGIN

:new.inspector\_id := inspector\_inspector\_id\_seq.nextval;

END;

/

CREATE SEQUENCE koap\_r\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER koap\_r\_id\_trg BEFORE

INSERT ON koap

FOR EACH ROW

BEGIN

:new.r\_id := koap\_r\_id\_seq.nextval;

END;

/

CREATE SEQUENCE protocols\_protocol\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER protocols\_protocol\_id\_trg BEFORE

INSERT ON protocols

FOR EACH ROW

BEGIN

:new.protocol\_id := protocols\_protocol\_id\_seq.nextval;

END;

/

CREATE SEQUENCE regions\_region\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER regions\_region\_id\_trg BEFORE

INSERT ON regions

FOR EACH ROW

BEGIN

:new.region\_id := regions\_region\_id\_seq.nextval;

END;

/

CREATE SEQUENCE "Role-Page\_Role\_P\_id\_SEQ" START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER "Role-Page\_Role\_P\_id\_TRG" BEFORE

INSERT ON "Role-Page"

FOR EACH ROW

BEGIN

:new.role\_p\_id := "Role-Page\_Role\_P\_id\_SEQ".nextval;

END;

/

CREATE SEQUENCE traffic\_rules\_rule\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER traffic\_rules\_rule\_id\_trg BEFORE

INSERT ON traffic\_rules

FOR EACH ROW

BEGIN

:new.rule\_id := traffic\_rules\_rule\_id\_seq.nextval;

END;

/

CREATE SEQUENCE ts\_ts\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER ts\_ts\_id\_trg BEFORE

INSERT ON ts

FOR EACH ROW

BEGIN

:new.ts\_id := ts\_ts\_id\_seq.nextval;

END;

/

CREATE SEQUENCE user\_account\_user\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER user\_account\_user\_id\_trg BEFORE

INSERT ON user\_account

FOR EACH ROW

BEGIN

:new.user\_id := user\_account\_user\_id\_seq.nextval;

END;

/

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE 10

-- CREATE INDEX 0

-- ALTER TABLE 25

-- CREATE VIEW 0

-- ALTER VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 10

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 10

-- CREATE MATERIALIZED VIEW 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 0

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

--

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

--

-- ERRORS 0

-- WARNINGS 0

**Приложение Б**

Пакет для обработки логина и пароля пользователя

Create Or Replace Package Pkg\_Security Is

Function Authenticate\_User(p\_User\_Name Varchar2

,p\_Password Varchar2) Return Boolean;

-----

Procedure Process\_Login(p\_User\_Name Varchar2

,p\_Password Varchar2

,p\_App\_Id Number);

End Pkg\_Security;

/

Create Or Replace Package Body Pkg\_Security Is

Function Authenticate\_User(p\_User\_Name Varchar2

,p\_Password Varchar2) Return Boolean As

v\_Password User\_Account.Password%Type;

v\_Active User\_Account.Active%Type;

Begin

If p\_User\_Name Is Null Or p\_Password Is Null Then

-- Write to Session, Notification must enter a username and password

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'Please enter Username and password.');

Return False;

End If;

----

Begin

Select u.Active

,u.Password

Into v\_Active

,v\_Password

From User\_Account u

Where u.User\_Name = p\_User\_Name;

Exception

When No\_Data\_Found Then

-- Write to Session, User not found.

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'User not found');

Return False;

End;

If v\_Password <> p\_Password Then

-- Write to Session, Password incorrect.

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'Password incorrect');

Return False;

End If;

If v\_Active <> 'Y' Then

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'User locked, please contact admin');

Return False;

End If;

---

-- Write user information to Session.

--

Apex\_Util.Set\_Session\_State('SESSION\_USER\_NAME'

,p\_User\_Name);

---

---

Return True;

End;

--------------------------------------

Procedure Process\_Login(p\_User\_Name Varchar2

,p\_Password Varchar2

,p\_App\_Id Number) As

v\_Result Boolean := False;

Begin

v\_Result := Authenticate\_User(p\_User\_Name

,p\_Password);

If v\_Result = True Then

-- Redirect to Page 1 (Home Page).

Wwv\_Flow\_Custom\_Auth\_Std.Post\_Login(p\_User\_Name -- p\_User\_Name

,p\_Password -- p\_Password

,v('APP\_SESSION') -- p\_Session\_Id

,p\_App\_Id || ':1' -- p\_Flow\_page

);

Else

-- Login Failure, redirect to page 101 (Login Page).

Owa\_Util.Redirect\_Url('f?p=&APP\_ID.:101:&SESSION.');

End If;

End;

End Pkg\_Security;

/

Представление, связывающее роли и List Entries

SET DEFINE OFF;

CREATE OR REPLACE VIEW ROLE\_LIST\_ENTRY ("USER\_ROLE", "LIST\_ENTRY\_ID") AS

select distinct p.user\_role, l.list\_entry\_id

from apex\_application\_list\_entries l

join role\_page p on l.entry\_target like 'f?p=&APP\_ID.:'||p.page||':&%SESSION%' or entry\_target is null

where l.list\_name = 'Desktop Navigation Menu'

and l.application\_id = 100;

**Приложение В**

Примеры отчетов (в печатных формах)

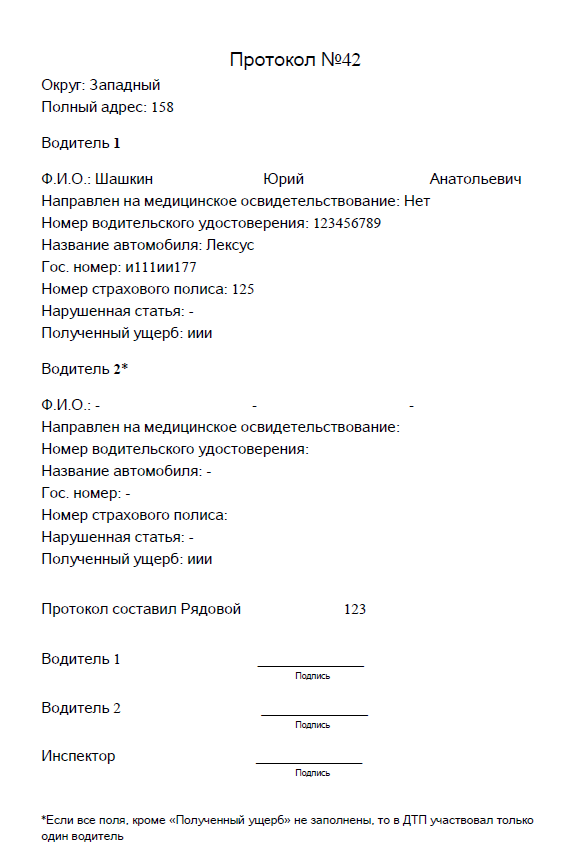


Рис. В.1 Протокол ДТП

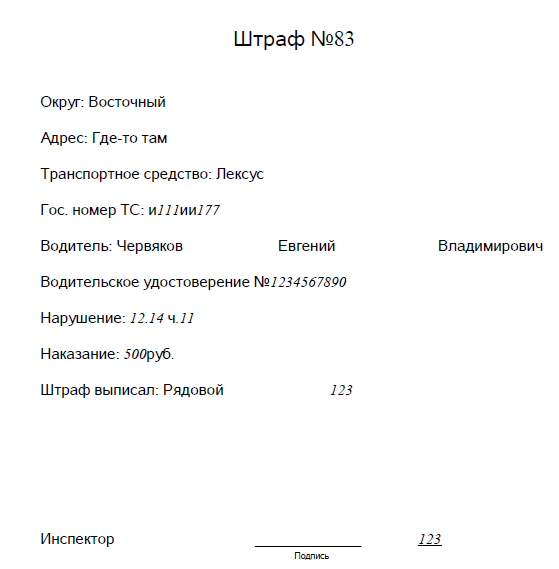


Рис. В.2 Штраф

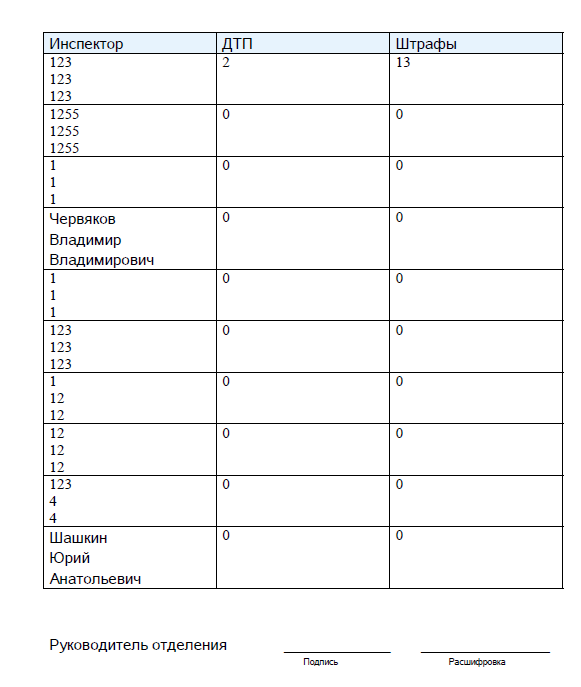


Рис. В.3 Статистика по инспекторам

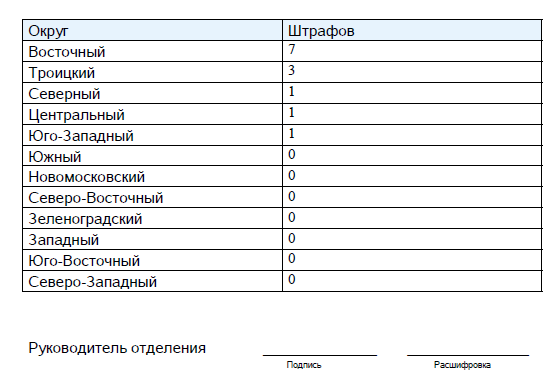


Рис. В.4 Статистика штрафов по округам

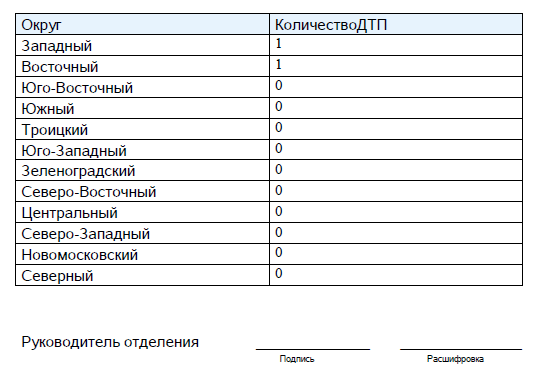


Рис. В.5 Статистика ДТП по округам