МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по учебной практике**

**ПМ 02. Разработка и администрирование баз данных**

**Тема: «Разработка базы данных «****ХранительПРО»**

Студента

Платунов Павел Андреевич

Группа 21П-1

Специальность

09.02.07 Информационные системы и программирование

Руководитель практики от колледжа

Калинин Арсений Олегович

\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись расшифровка

2024 год

Оглавление

[Выполнение анализа и предварительной обработки информации 3](#_Toc164444411)

[Выделение объектов и атрибутов в соответствии с заданием 4](#_Toc164444412)

[Проектирование и нормализация БД в полном соответствии с поставленной задачей и применением CASE-средств 6](#_Toc164444413)

[Выполнение построения БД в предложенной СУБД и заполнение всех таблиц с помощью соответствующих средств 7](#_Toc164444414)

[Реализация уровней доступа для различных категорий пользователей 11](#_Toc164444415)

[Создание запросов и отчетов в соответствии с заданием 12](#_Toc164444416)

[Создание и обоснование группы пользователей, принципов регистрации и системы паролей 17](#_Toc164444417)

[Выполнение резервного копирования БД и восстановление состояния БД на заданную дату. 19](#_Toc164444418)

[Заключение 21](#_Toc164444419)

# Выполнение анализа и предварительной обработки информации

При выполнение учебной практики необходимо разработать базу данных и модули системы защиты объекта критической информационной инфраструктуры (КИИ).

На всех объектах критической информационной инфраструктуры (КИИ) действует обязательная система пропускного режима, применяемая как внутри периметра контролируемых зон, так и в опасных зонах. Модуль ПО "ХранительПРО" предназначен для организации пропускного режима и включает в себя следующие компоненты:

* веб-сервис для заказа пропуска на предприятие для гостей и экскурсионных групп;
* терминал сотрудника общего отдела для формальной проверки поданной заявки;
* терминал сотрудника охраны для реализации пропускного режима;
* терминал сотрудника подразделения для учета посещений.

В программе MS Visio 2019 мною была построена UML диаграмма вариантов использования (Рис. 1).



Рисунок - UML Диаграмма вариантов использования

# Выделение объектов и атрибутов в соответствии с заданием

Для базы данных «ХранительПРО» выделены следующие объекты:

* Авторизация (содержит информацию для авторизации и регистрации)
* Группа (содержит информацию о Группах)
* Заявка (содержит информацию о заявках на групповое и личное посвящение)
* Организация (содержит информацию о организация посетителей)
* Отдел (содержит информацию об отделах в которых работают сотрудники)
* Подразделения (содержит информацию об подразделениях в которых работают сотрудники)
* Посетитель (содержит информацию о посетителях)
* Посещающие (содержит информацию о посетителях в заявке)
* Пропуск (содержит информацию о пропуске)
* Сотрудник (содержит информацию о сотрудниках)
* Статус (содержит статусы заявок)
* Тип заявки (содержит типы заявок)
* Черный список (содержит информацию о посетителях добавленных в черный список)

Авторизация

* Код авторизации
* Код посетителя
* Логин
* Пароль

Группа

* Код группы
* Дата создания
* Название

Заявка

* Код заявки
* Код типа заявки
* Код пропуска
* Код подразделения
* Код сотрудника
* Дата создания
* Дата и время посещения
* Время входа
* Время выхода
* Доступ на территорию
* Код статуса
* Причина отказа

Организация

* Код организации
* Название

Отдел

* Код отдела
* Название

Подразделения

* Код подразделения
* Название

Посетитель

* Код посетителя
* Фамилия
* Имя
* Отчество
* Номер телефона
* Email
* Дата рождения
* Код организации
* Серия паспорта
* Номер паспорта
* Фотография
* Скан паспорта
* Примечания

Посещающие

* Код посещающего
* Код заявки
* Код посетителя
* Код группы

Пропуск

* Код пропуска
* Срок начала действия
* Срок окончания действия
* Цель посещения

Сотрудник

* Код сотрудника
* Код авторизации
* Фамилия
* Имя
* Отчество
* Дата рождения
* Номер телефона
* Номер телефона
* Email
* Код подразделения
* Код отдела

Статус

* Код статуса
* Название

Тип заявки

* Код типа заявки
* Название

Черный список

* Код черного списка
* Код посетителя
* Причина добавления

# Проектирование и нормализация БД в полном соответствии с поставленной задачей и применением CASE-средств

Построение в программы MS Visio 2019 ERD-диаграммы (Рис. 2).



Рисунок – ERD-диаграмма

# Выполнение построения БД в предложенной СУБД и заполнение всех таблиц с помощью соответствующих средств

Спроектирована база данных в SQL Server Management Studio (SSMS) (Рис. 3)

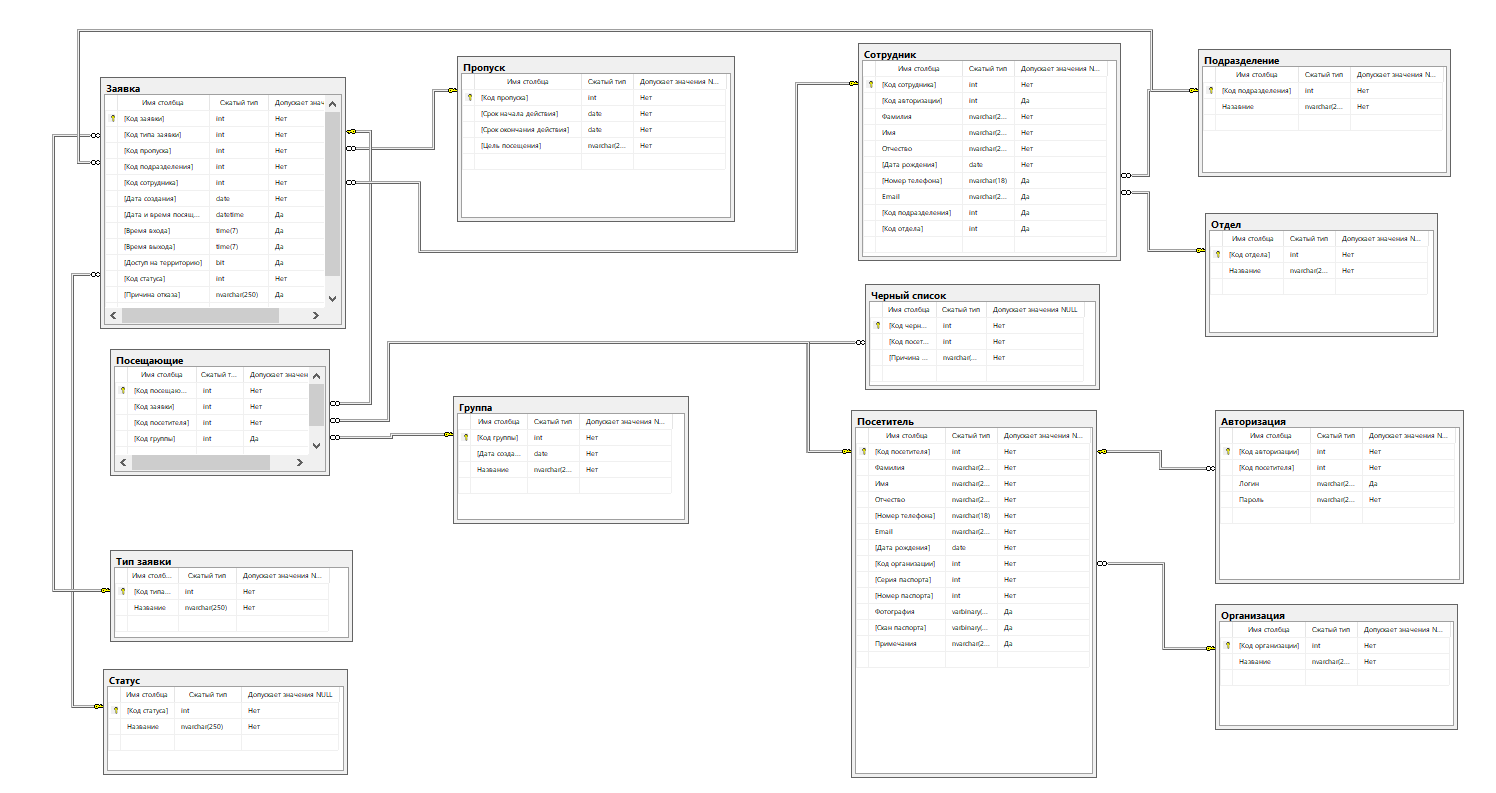


Рисунок - Диаграмма базы данных

Таблица 1 - Словарь данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Подразделение | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код подразделения | int | Yes |  |
|  | Название | Nvarchar(250) | Yes |  |
| Отдел | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код отдела | Int | Yes |  |
|  | Название | Nvarchar(250) | Yes |  |
| Авторизация | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код авторизации | Int | Yes |  |
| FK | Код посетителя | Int | Yes | FK to Посетитель |
|  | Логин | Nvarchar(250) | Yes |  |
|  | Пароль | Nvarchar(250) | Yes |  |
| Организация | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код организации | Int | Yes |  |
|  | Название | Nvarchar(250) | Yes |  |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сотрудник | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код сотрудника | Int | Yes |  |
|  | Код авторизации | Int | No |  |
|  | Фамилия | Nvarcahr(250) | Yes |  |
|  | Имя | Nvarcahr(250) | yes |  |
|  | Отчество | Nvarcahr(250) | Yes |  |
|  | Дата рождения | Date | Yes |  |
|  | Номер телефона | Nvarcahr(18) | No |  |
|  | Email | Nvarcahr(250) | No |  |
| FK | Код подразделения | Int | No | FK to Подразделения |
| FK | Код отдела | Int | No | FK to Отдел |
| Посетитель | | | | |
| KEY | FILED NAME | DATA TYPE | REQUIED? | NOTES |
| PK | Код посетителя | Int | Yes |  |
|  | Фамилия | Nvarchar(250) | Yes |  |
|  | Имя | Nvarchar(250) | Yes |  |
|  | Отчество | Nvarchar(250) | Yes |  |
|  | Номер телефона | Nvarchar(18) | Yes |  |
|  | Email | Nvarchar(250) | Yes |  |
|  | Дата рождения | Date | Yes |  |
| FK | Код организации | Int | Yes | FK to Организация |
|  | Серия паспорта | Int | Yes |  |
|  | Номер паспорта | Int | Yes |  |
|  | Фотография | Image | No |  |
|  | Скан паспорта | Image | No |  |
|  | Примечания | Nvarchar(250) | No |  |
| Черный список | | | | |
| KEY | FILED NAME | DATA TYPE | REQUIED? | NOTES |
| PK | Код черного списка | Int | Yes |  |
| FK | Код посетителя | Int | Yes | FK to Посетитель |
|  | Причина добавления | Nvarchar(250) | Yes |  |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заявка | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код заявки | Int | Yes |  |
| FK | Код типа заявки | Int | Yes | FK to Тип заявки |
| FK | Код пропуска | Int | Yes | FK to Пропуск |
| FK | Код подразделения | Int | Yes | FK to Подразделение |
| FK | Код сотрудника | int | Yes | FK to Сотрудник |
|  | Дата создания | Date | Yes |  |
|  | Дата и время посещения | DateTime | Yes |  |
|  | Время входа | Time(7) | No |  |
|  | Время ухода | Time(7) | No |  |
|  | Доступ на территорию | Bit | No |  |
| FK | Код статуса | Int | Yes | FK to Статус |
|  | Причина отказа | Nvarchar(250) | No |  |
| Группа | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код группы | Int | Yes |  |
|  | Дата создания | Date | Yes |  |
|  | Название | Nvarchar(250) | Yes |  |
| Посещающие | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код посещающих | Int | Yes |  |
| FK | Код заявки | Int | Yes | FK to Заявка |
| FK | Код посетителя | Int | Yes | FK to Посетитель |
| FK | Код группы | Int | No | FK to Группа |
| Пропуск | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код пропуска | Int | Yes |  |
|  | Срок начала действия | Date | Yes |  |
|  | Срок окончания действия | Date | Yes |  |
|  | Цель посещения | Nvarchar(250) | Yes |  |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип заявки | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код типа заявки | Int | Yes |  |
|  | Название | Nvarchar(250) | Yes |  |
| Статус | | | | |
| KEY | FIELD NAME | DATA TYPE | REQUIRED? | NOTES |
| PK | Код статуса | Int | Yes |  |
|  | Название | Nvarchar(250) | Yes |  |

# Реализация уровней доступа для различных категорий пользователей

В системе реализовано 2 основных уровня доступа: посетитель и сотрудник

Посетитель может просматривать список заявок оформленных на него. А также может оформить личную или групповую заявку.

Сотрудник имеет 3 разных уровня доступа: сотрудник общего отдела, сотрудник охраны, сотрудник подразделения.

Сотрудник общего отдела может работать с заявками, изменяя дату и время посещения, а также меняя статус заявки. Также он может формировать отчеты.

Сотрудник охраны может работать только с одобренными заявками менять время выхода посетителя и доступ его на территорию

Сотрудник подразделения может работать с одобренными заявками, изменяя время входа и время выхода посетителей с территории. А также добавлять посетителя в черный список.

# Создание запросов и отчетов в соответствии с заданием

Хранимые процедуры:

Код авторизация

|  |
| --- |
| --Авторизация пользователя  CREATE PROCEDURE Autorisation (@Login VARCHAR(250), @Password NVARCHAR(250))  AS  BEGIN  --Выбор подходящих посетителей  SELECT \* FROM Авторизация WHERE Логин = @Login AND Пароль = @Password  END |

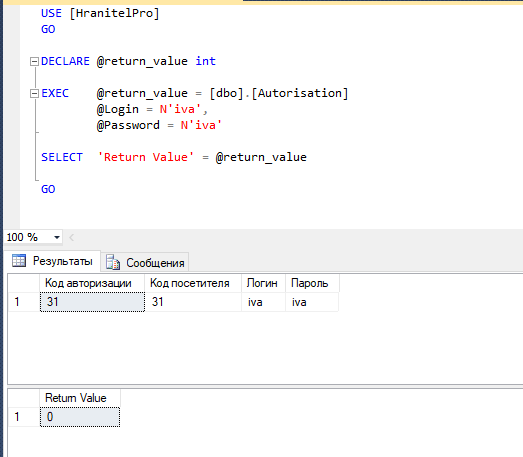


Рисунок - Хранимая процедура на авторизацию

Код регистрация

|  |
| --- |
| --Регистрация посетителей  CREATE PROCEDURE Registration (@Family VARCHAR(250), @Name NVARCHAR(250), @Otchestvo NVARCHAR(250), @Phone NVARCHAR(18), @Email NVARCHAR(250), @Organisation NVARCHAR(250), @Birthday NVARCHAR(250), @Series INT, @Number INT, @Login NVARCHAR(250), @Password NVARCHAR(250))  AS  BEGIN  --Переменая  DECLARE @Kode INT;  --Проверка существует ли организация  IF EXISTS (SELECT \* FROM Организация WHERE Название = @Organisation)  --Если существует выборка кода организации  SELECT @Kode = [Код организации] FROM Организация WHERE Название = @Organisation;  ELSE  --Если не существует создание и выборка кода организации  BEGIN  INSERT INTO Организация VALUES (@Organisation)  SELECT TOP 1 @Kode = [Код организации] FROM Организация ORDER BY [Код организации] DESC;  END  --Добавление посетителя  INSERT INTO Посетитель (Фамилия, Имя, Отчество, [Номер телефона], Email, [Дата рождения], [Код организации], [Серия паспорта], [Номер паспорта]) VALUES (@Family, @Name, @Otchestvo, @Phone, @Email, @Birthday, @Kode, @Series, @Number)  --Выборка кода добавленного посетителя  SELECT TOP 1 @Kode = [Код посетителя] FROM Посетитель ORDER BY [Код организации] DESC;  --Добавление посетителя в авторизацию  INSERT INTO Авторизация VALUES (@Kode, @Login, @Password)  END |

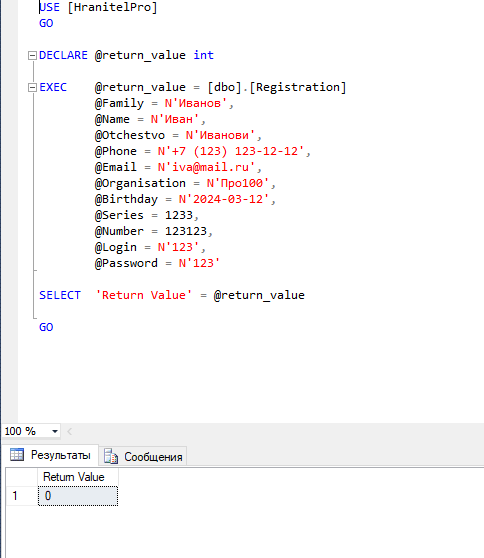


Рисунок - Хранимая процедура на регистрацию

Код фильтрации заявок

|  |
| --- |
| --Фильтрация заявок  CREATE PROCEDURE FilteringRequests (@Type VARCHAR(250), @Podrazdelenie VARCHAR(250), @Status VARCHAR(250))  AS  BEGIN  --Выборка нужных данных  SELECT Заявка.[Код заявки], [Тип заявки].Название AS [Тип заявки], Статус.Название AS Статус, Подразделение.Назавние AS Подразделение, Сотрудник.[Код сотрудника], Пропуск.[Код пропуска],  Пропуск.[Срок начала действия], Пропуск.[Срок окончания действия], Пропуск.[Цель посещения], Посетитель.Фамилия, dbo.Посетитель.Имя, Посетитель.Отчество,  Посетитель.[Номер телефона], Посетитель.Email, Посетитель.[Серия паспорта], Посетитель.[Номер паспорта], Организация.Название AS организация  FROM Заявка INNER JOIN  Пропуск ON Заявка.[Код пропуска] = Пропуск.[Код пропуска] INNER JOIN  Статус ON Заявка.[Код статуса] = Статус.[Код cтатуса] INNER JOIN  [Тип заявки] ON Заявка.[Код типа заявки] = [Тип заявки].[Код типа заявки] INNER JOIN  Подразделение ON Заявка.[Код подразделения] = Подразделение.[Код подразделения] INNER JOIN  Сотрудник ON Заявка.[Код сотрудника] = Сотрудник.[Код сотрудника] INNER JOIN  Посещающие ON Заявка.[Код заявки] = Посещающие.[Код заявки] INNER JOIN  Посетитель ON Посещающие.[Код посетителя] = Посетитель.[Код посетителя] INNER JOIN  Организация ON Посетитель.[Код организации] = Организация.[Код организации]  --Проверка записей по данным которые нужно искать  WHERE (@Type IS NULL OR [Тип заявки].Название = @Type)  AND (@Podrazdelenie IS NULL OR Подразделение.Назавние = @Podrazdelenie)  AND (@Status IS NULL OR Статус.Название = @Status)  END |

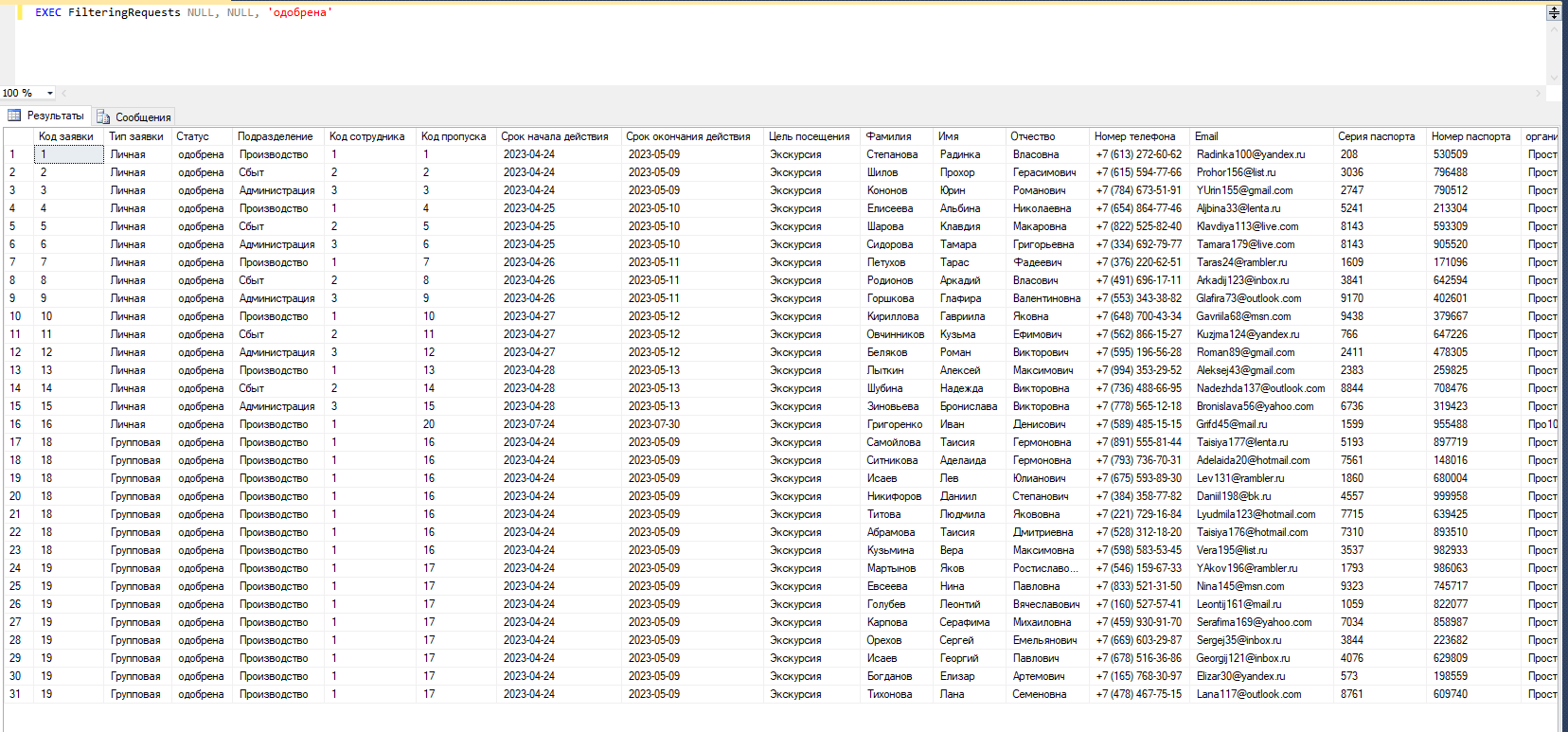


Рисунок - Хранимая процедура для фильтрации

Представления

Код выбора заявок

|  |
| --- |
| SELECT dbo.Заявка.[Код заявки], dbo.[Тип заявки].Название AS [Тип заявки], dbo.Статус.Название AS Статус, dbo.Подразделение.Назавние AS Подразделение, dbo.Сотрудник.[Код авторизации], dbo.Пропуск.[Код пропуска],  dbo.Пропуск.[Срок начала действия], dbo.Пропуск.[Срок окончания действия], dbo.Пропуск.[Цель посещения], dbo.Посетитель.Фамилия, dbo.Посетитель.Имя, dbo.Посетитель.Отчество,  dbo.Посетитель.[Номер телефона], dbo.Посетитель.Email, dbo.Посетитель.[Серия паспорта], dbo.Посетитель.[Номер паспорта], dbo.Организация.Название AS организация  FROM dbo.Заявка INNER JOIN  dbo.Пропуск ON dbo.Заявка.[Код пропуска] = dbo.Пропуск.[Код пропуска] INNER JOIN  dbo.Статус ON dbo.Заявка.[Код статуса] = dbo.Статус.[Код cтатуса] INNER JOIN  dbo.[Тип заявки] ON dbo.Заявка.[Код типа заявки] = dbo.[Тип заявки].[Код типа заявки] INNER JOIN  dbo.Подразделение ON dbo.Заявка.[Код подразделения] = dbo.Подразделение.[Код подразделения] INNER JOIN  dbo.Сотрудник ON dbo.Заявка.[Код сотрудника] = dbo.Сотрудник.[Код сотрудника] INNER JOIN  dbo.Посещающие ON dbo.Заявка.[Код заявки] = dbo.Посещающие.[Код заявки] INNER JOIN  dbo.Посетитель ON dbo.Посещающие.[Код посетителя] = dbo.Посетитель.[Код посетителя] INNER JOIN  dbo.Организация ON dbo.Посетитель.[Код организации] = dbo.Организация.[Код организации] |

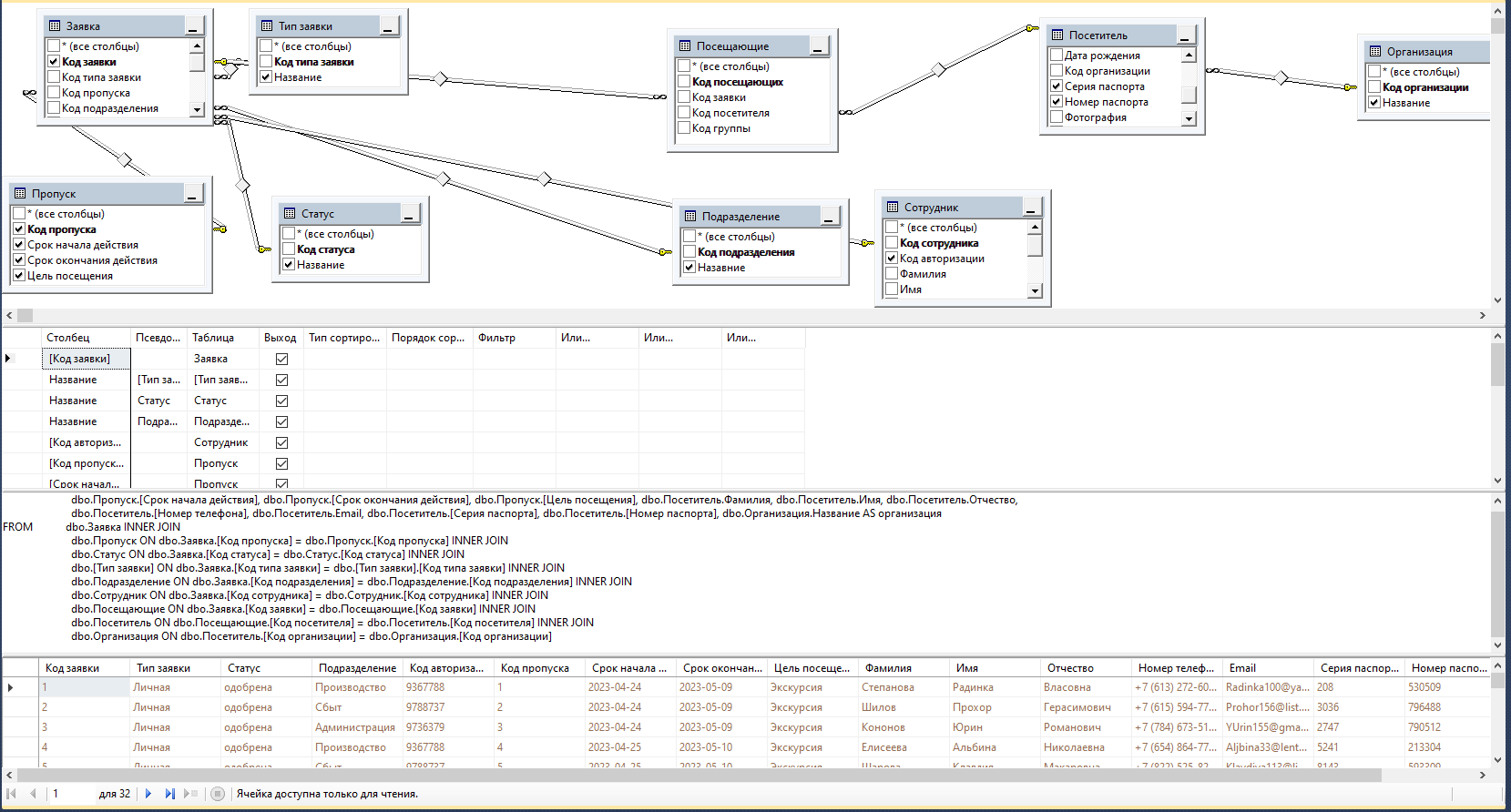


Рисунок - Представление о заявках

Триггер

Код создания триггера для автоматической генерации уникального логина

|  |
| --- |
| CREATE TRIGGER Visitor\_INSERT  ON Авторизация  --Событие после добавления записи  AFTER INSERT  AS  BEGIN  --Переменные  DECLARE @a int, @ind int, @emal nvarchar(max)  --Выборка кода посетителя  SET @a = (select [Код посетителя] from Посетитель WHERE [Код посетителя] = (SELECT [Код посетителя] FROM inserted))  --Выборка почты посетителя  SET @emal = (select Email from Посетитель WHERE [Код посетителя] = (SELECT [Код посетителя] FROM inserted))  --Поиск позиции заданного символа  SET @ind = CHARINDEX('@', @emal)  --Обнавление записи в таблице авторизации  UPDATE Авторизация  --Генерация уникального логина  SET Логин = CONCAT (LEFT(@emal, @ind-1), @a)  WHERE [Код авторизации] = (SELECT [Код авторизации] FROM inserted)  END |

# Создание и обоснование группы пользователей, принципов регистрации и системы паролей

Регистрация посетителя проходит на форме регистрации после заполнения всех полей (Рис. 8).

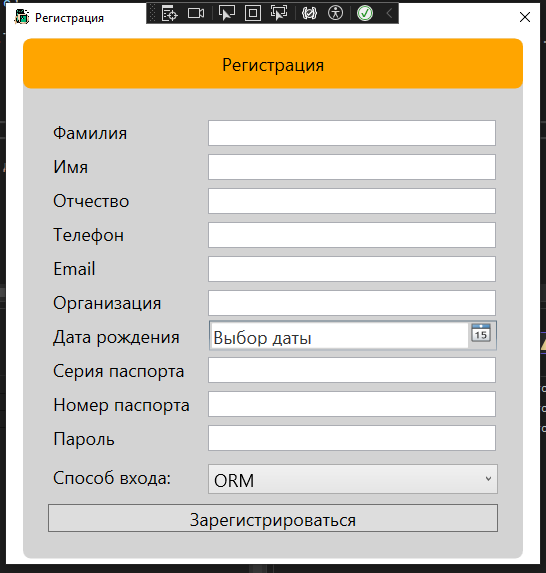


Рисунок - Окно регистрации

Авторизация в приложении происходит с помощью логина и пароля который автоматически генерируется при регистрации (Рис. 9). Логин генерируется по средством метода в приложении и триггера на сервере.

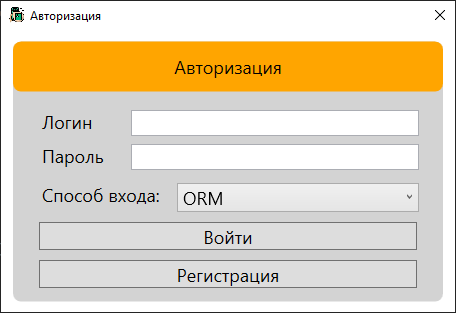


Рисунок – Окно авторизации посетителя

Авторизация для сотрудников производиться по коду авторизации (Рис. 10).

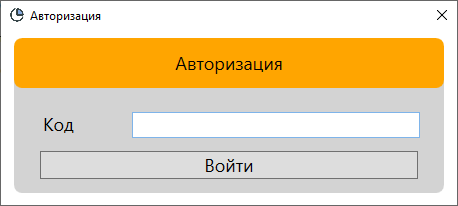


Рисунок – Окно авторизации сотрудника

# Выполнение резервного копирования БД и восстановление состояния БД на заданную дату.

Выполнение резервного копирования через встроенную службу Management Studio (Рис. 11)

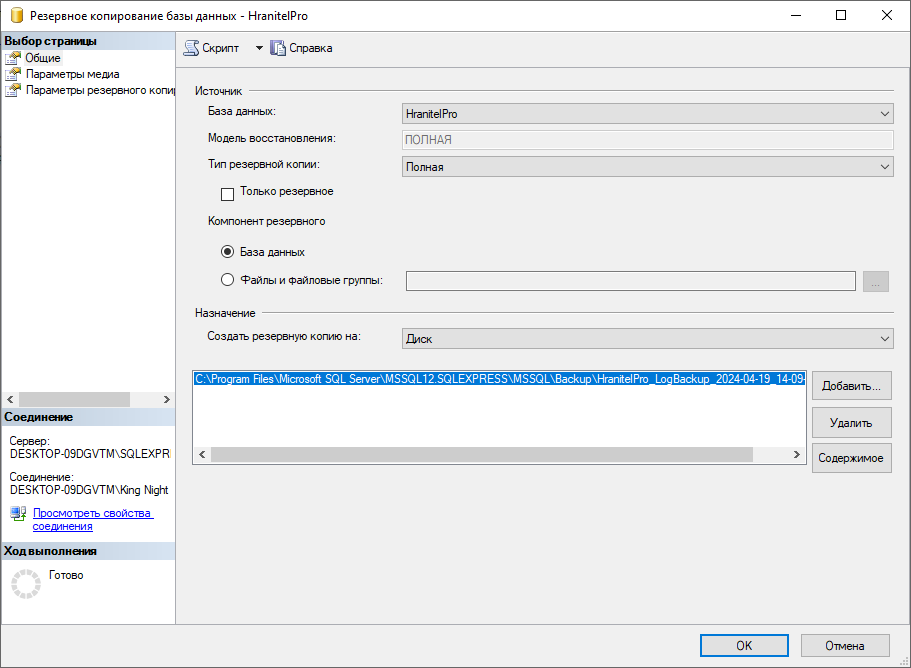


Рисунок - Выполнение резервного копирования

При создание резервных копий использовалась как полная модель так и разностная (Рис. 12,13).

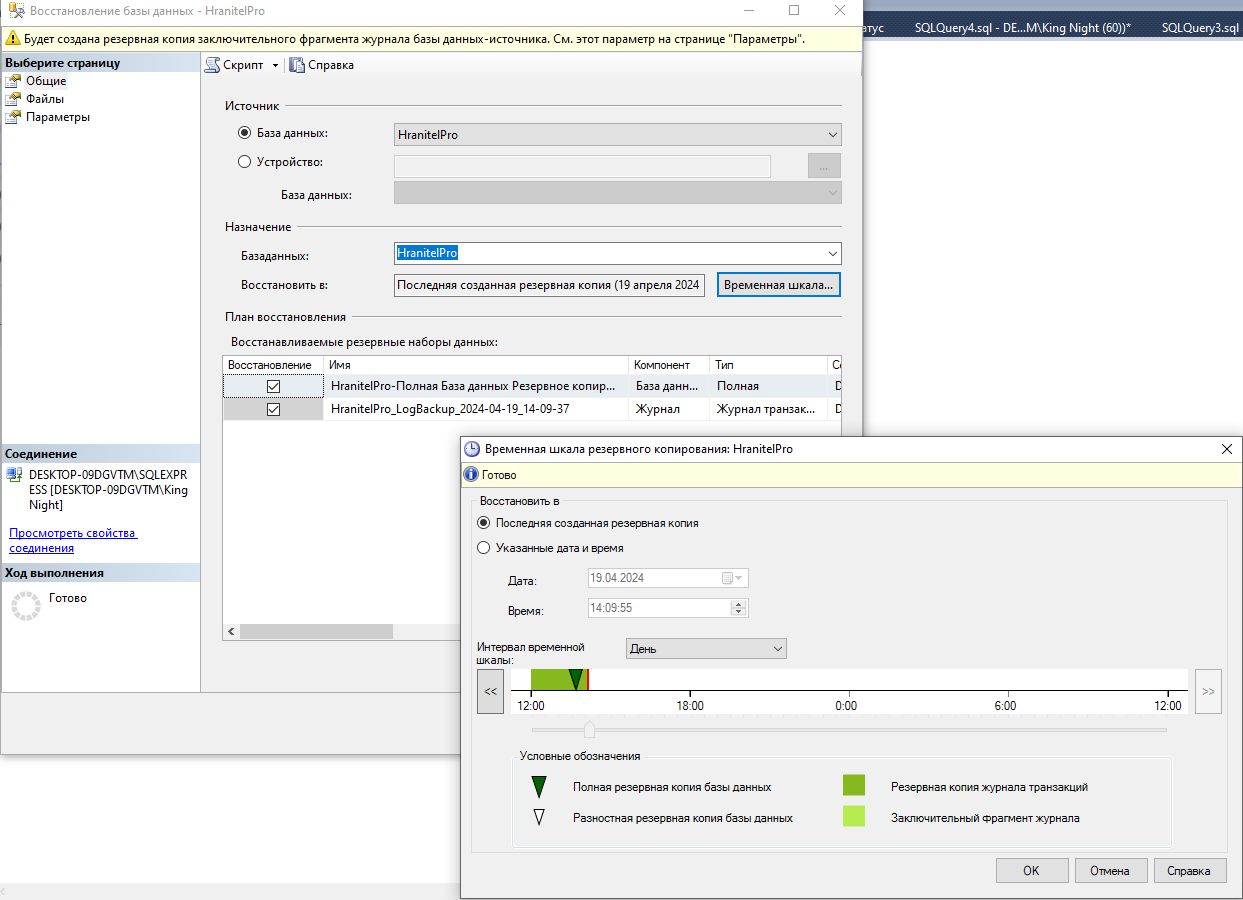


Рисунок - Выбор времени, до которого нужно откатиться

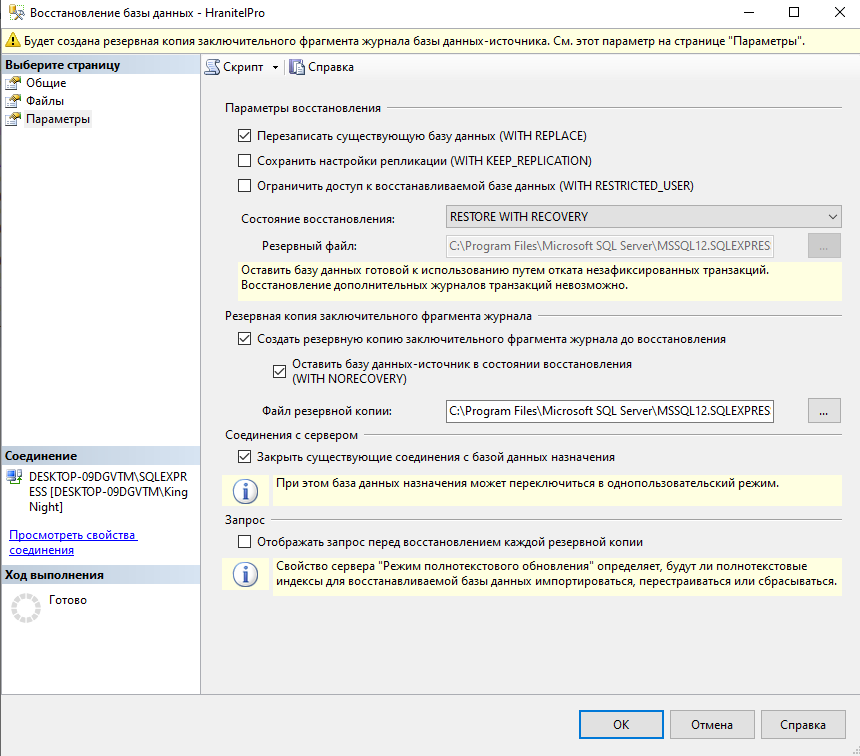


Рисунок - Параметры восстановления

# Заключение

В ходе выполнения практики по модулю ПМ.11. Разработка, администрирование и защита баз данных. Мною была создана база данных «ХранительПРО», которая хранит информацию о посетителях, сотрудниках и заявках. При проектировании базы данных учитывалась 3НФ.

В базе данных были созданные триггеры, хранимые процедуры и представления для удобной работы.

Были написаны 4 приложения: для посетителя, сотрудника общего отдела, сотрудника охраны, сотрудника подразделения.

Для достижения поставленных задач были использованы знания полученные при обучение.