Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**ИВАНОВСКИЙ ГОСУДАРСТВЕННЫЙ ХИМИКО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет техники, управления и цифровой инфраструктуры**

**Кафедра информационных технологий и цифровой экономики**

**КУРСОВОЙ ПРОЕКТ**

**ПО ДИСЦИПЛИНЕ**

**УПРАВЛЕНИЕ ДАННЫМИ**

**Тема:** Разработка базы данных MS SQL

и приложения средствами Visual Studio.

Вариант №2

**Выполнил студент: Родина Виктория Сергеевна курс/гр. 3/147**

ФИО

**Проверил: Куваев Александр Евгеньевич**

ФИО

Иваново 2020

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**ИВАНОВСКИЙ ГОСУДАРСТВЕННЫЙ ХИМИКО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет** ТУиЦИ

**Кафедра** ИТиЦЭ

**Направление** 09.03.02 Информационные системы и технологии

**Профиль** Информационные системы и технологии

**Дисциплина** Управление данными

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

**Студенту** Родиной Виктории Сергеевны курс/гр. **3/147**

(фамилия, имя, отчество - полностью)

**Тема:** Разработка базы данных MS SQL и приложения средствами Visual Studio. Вариант №2

Исходные данные: *законодательные и подзаконные нормативные акты; энциклопедическая и справочная литература; статистические и аналитические материалы; монографии; данные профессиональных периодических изданий; интернет-ресурсы; самостоятельно собранные первичные материалы.*

Содержание работы:

Введение: *цель и задачи работы; предмет и объект исследования; обоснование структуры работы.*

Теоретическая часть должна содержать: *системный анализ предметной области, инфологическую модель БД, список запросов, которые должны быть реализованы в БД, список требований к приложению*

В практической части необходимо: *сгенерировать БД, реализовать SQL-запросы, разработать программное приложение к БД.*

Заключение*: основные выводы по проекту, обобщение основных предложений и рекомендаций.*

Руководитель работы:  *Куваев А.Е.*

**Календарный план-график**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование этапов выполнения курсовой работы | Срок выполнения этапов | Примечание |
| 1 | Исследование предметной области и составление логической модели БД | 17.02.20 – 17.03.20 |  |
| 2 | Создание физической модели БД | 18.03.20 – 31.03.20 |  |
| 3 | Генерация БД средствами MS SQL и реализация SQL-запросов | 01.04.20 – 14.04.20 |  |
| 4 | Разработка приложения средствами Visual Studio | 15.04.20 – 12.05.20 |  |
| 5 | Оформление пояснительной записки | 13.05.20 – 17.05.20 |  |
| 6 | Защита курсового проекта | 18.05.20 - 23.05.20 |  |

Дата выдачи задания 17.02.2020

Дата представления законченной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись /Ф.И.О.)

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Родина В.С.\_\_\_\_\_\_

(Подпись /Ф.И.О.)

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc50670350)

[1 СИСТЕМНЫЙ АНАЛИЗ 6](#_Toc50670351)

[1.1Описание предметной области 6](#_Toc50670352)

[1.2 Логическая модель «Как есть» 7](#_Toc50670353)

[1.3 Логическая модель «Как должно быть» 8](#_Toc50670354)

[1.4 Список запросов, которые должны быть реализованы в БД 9](#_Toc50670355)

[2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 11](#_Toc50670356)

[3. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ 23](#_Toc50670357)

[3.1 Таблица «Клиенты» 25](#_Toc50670358)

[3.2 Таблица «Скидки» 25](#_Toc50670359)

[3.3 Таблица «Комната» 26](#_Toc50670360)

[3.4 Таблица «Заселения» 27](#_Toc50670361)

[3.5 Таблица «Скидки и Заселения» 27](#_Toc50670362)

[3.6 Поиск записей 28](#_Toc50670363)

[3.7 Экспорт таблицы Excel 31](#_Toc50670364)

[ЗАКЛЮЧЕНИЕ 32](#_Toc50670365)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 33](#_Toc50670366)

[ПРИЛОЖЕНИЕ А. СКРИПТ БАЗЫ ДАННЫХ 34](#_Toc50670367)

[ПРИЛОЖЕНИЕ Б. ПЕРЕЧЕНЬ ФАЙЛОВ 41](#_Toc50670368)

# ВВЕДЕНИЕ

С развитием информационных технологий и предпринимательства, актуальность использования баз данных значительно увеличилось. Успешные и крупные компании не могут представить свой бизнес без четко построенной информационной системы. Базы данных отвечают высоким требованиям производительности и безопасности.

Данная пояснительная записка описывает процесс создание базы данных, организации логики базы данных, создания приложения, работающего с этой базой данных.

Для достижения необходимо решить следующие задачи:

– изучить и проанализировать предметную область;

– определить основные функции системы;

– описать интерфейс системы, создать макет интерфейса;

– описать запросы, используемые в системе.

В первый этап разработки входит ознакомление с заданием, построение инфологических моделей «как есть» и «как должно быть». Составление модели данных в нотации IDEF1x, составления вопросов и требований того, что должна делать система.

Второй этап разработки - описание процесса разработки физической модели базы данных SQL, создание представлений, хранимых процедур, наполнения базы данных и выполнения запросов.

Третий этап разработки - описание работы разработанного приложения для работы с базой данных, - работа с таблицами, добавление, удаление и редактирование записей таблиц, поиск записей в базе данных и отчетность.

# 1 СИСТЕМНЫЙ АНАЛИЗ

ЗАДАНИЕ «ГОСТИНИЦА»

## 1.1Описание предметной области

Вы работаете в гостинице. Вашей задачей является отслеживание финансовой стороны ее работы. Ваша деятельность организована следующим образом: гостиница предоставляет номера клиентам на определенный срок. Каждый номер характеризуется вместимостью, комфортностью (люкс, полулюкс, обычный) и ценой. Вашими клиентами являются различные лица, о которых вы собираете определенную информацию (фамилия, имя, отчество и некоторый комментарий). Сдача номера клиенту производится при наличии свободных мест в номерах, подходящих клиенту по указанным выше параметрам. При поселении фиксируется дата поселения. При выезде из гостиницы для каждого места запоминается дата освобождения.

Таблицы:

Клиенты (Код клиента, Фамилия, Имя, Отчество, Паспортные данные,  
Комментарий).  
 Номера (Код номера, Номер, Количество человек, Комфортность, Цена).  
 Поселение (Код поселения, Код клиента, Код номера, Дата поселения, Дата освобождения, Примечание).

Развитие постановки задачи:

Необходимо не только хранить информацию по факту сдачи номера  
клиенту, но и осуществлять бронирование номеров. Кроме того, для  
постоянных клиентов, а также для определенных категорий клиентов  
предусмотрена система скидок. Скидки могут суммироваться.  
Внести в структуру таблиц изменения, учитывающие этот факт, и  
изменить существующие запросы. Добавить новые запросы.

В этапе системного анализа проводится осмысление поставленной задачи. Далее необходимо составить инфологические модели

## 1.2 Логическая модель «Как есть»

Клиенты (Код клиента, Фамилия, Имя, Отчество, Паспортные данные,  
Комментарий).  
Номера (Код номера, Номер, Количество человек, Комфортность, Цена).  
Поселение (Код поселения, Код клиента, Код номера, Дата поселения, Дата  
освобождения, Примечание).

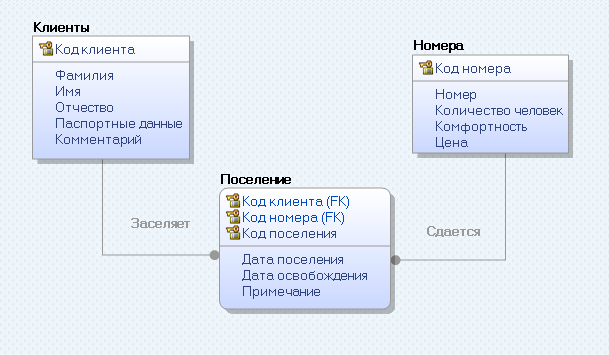


Рисунок 1.1 – Построение модели «как есть» в нотации IDEF1x

Исходя из анализа развития постановки задачи можно сделать вывод, что:

* Появляются новые сущности «Бронирование», «Скидки» и «Услуги»;
* Бронирование (Код брони, Наименование брони, Дата заезда, Дата выезда);
* Скидки (Код скидки, Наименование Скидки, Размер скидки);
* Услуги (Код услуги, Название Скидки, Цена).

## 1.3 Логическая модель «Как должно быть»

* Клиенты (Код клиента, ФИО клиента, Паспортные данные, Комментарий);
* Номера (Код номера, Наименование номера, Количество человек, Комфортность, Цена);
* Поселение (Код поселения, Клиент, Номер, Дата поселения, Дата освобождения, Примечание);
* Бронирование (Код брони, Наименование брони, Дата заезда, Дата выезда);
* Скидки (Код скидки, Наименование скидки, Размер скидки);
* Услуги (Код услуги, Название скидки, Цена).
* Построение модели в нотации IDEF1x выполнено в программе Erwin Data Modeler 2018 R1

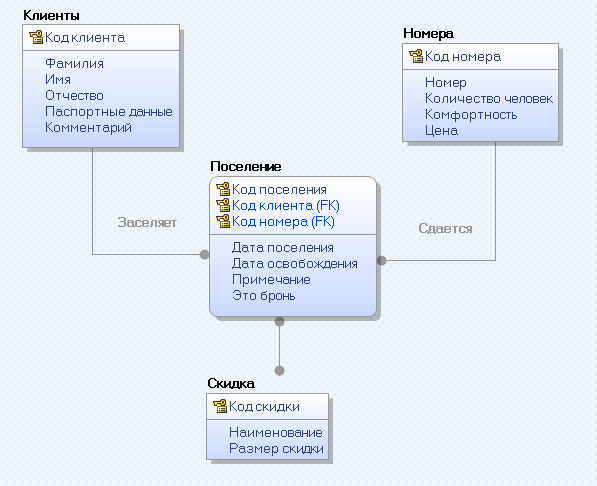


Рисунок 1.2 – Построение модели в нотации IDEF1x

## Список запросов, которые должны быть реализованы в БД

1. Вывести полные имена клиентов, чей id номер совпадает с номером комнаты.
2. Вывести количество клиентов со скидками, использовавших скидки.
3. Вывести количество постоянных клиентов.
4. Вывести список номеров комнат на 3 и более человек и комфортностью люкс.
5. Вывести номер и цену самого дорогого люкса.
6. Вывести паспортные данные и имена клиентов, которые бронировали номера.
7. Вывести паспортные данные и имена клиентов, которые не сдали свои номера.
8. Вывести количество бронированных номеров за 2020 год.
9. Вывести среднюю стоимость не люкс номеров.
10. Вывести имена и фамилии клиентов, которые снимали номера на Новый Год.
11. Вывести топ 3 самых популярных имен клиентов.
12. Вывести названия скидок больше 10%.
13. Вывести топ 3 клиента по количеству одновременно использованных скидок.
14. Вывести во сколько раз самый дорогой обычный номер для 3 и более человек дороже самого дешевого.
15. Вывести названия скидок, которые использовали при выборе люкс номеров.

1.5 Набор требований, которым должно отвечать разрабатываемое приложение

- Обеспечение ведения списка клиентов:

- Обеспечение ведения списка услуг:

- Обеспечение ведения списка номеров:

- Обеспечение ведения списка скидок:

- Обеспечить возможность сортировки по всем имеющимся полям:

- Обеспечить возможность поиска записей в базе данных:

- Обеспечить выгрузку данных в таблицы Excel;

# 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Основной задачей данного этапа является преобразование инфологической модели в физическую модель базы данных, преобразование ее в схему базы данных для MS SQL Server 2008, используя Transact-SQL, проверки корректности сделанных преобразований и соответствию исходных требований, а также проектированию приложения к базам данных.

Физическая модель представлена на рисунке 2.1.

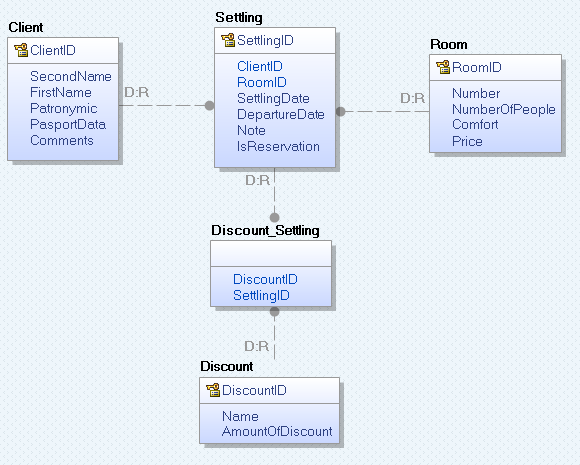


Рисунок 2.1 – Физическая модель

На рисунке 2.2 представлена физическая модель со всеми типами данных каждой таблицы.

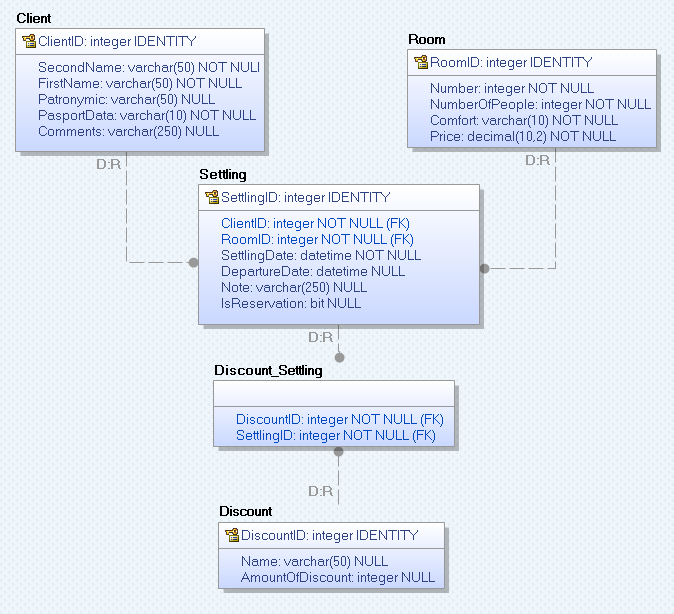


Рисунок 2.2 – Физическая модель со всеми типами данных

Представление базовой информации о соискателе показано на рисунке 2.3.

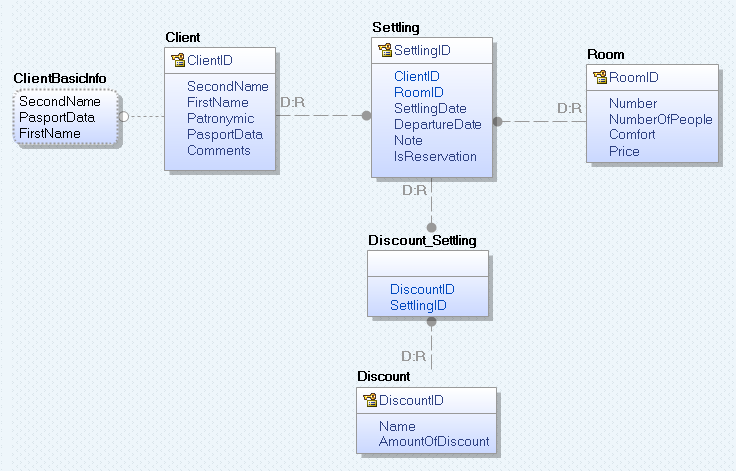


Рисунок 2.3 – Представление ClientBasicInfo

После создания физической модели переносим ее в базу данных SQL с помощью Forward Engineering и заполним ее.

Далее представлены таблицы базы данных и запросы по заполнению таблиц.

Запрос SQL на добавление записи в таблицу Client

INSERT INTO Client (FirstName, SecondName, Patronymic, PasportData, Comments)

VALUES('ИМЯ','ФАМИЛИЯ','ОТЧЕСТВО','ПАСПОРТ', 'КОММЕНТАРИЙ')

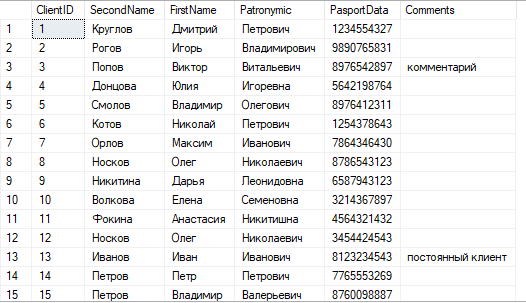


Рисунок 2.4 – Таблица Клиентов(Client)

Запрос SQL на добавление записи в таблицу Discount:

INSERT INTO Discount (Name,AmountOfDiscount)

VALUES ('ИМЯ', 'РАЗМЕР СКИДКИ')

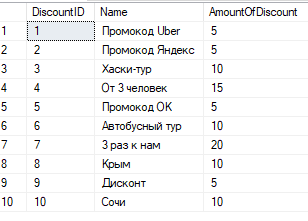


Рисунок 2.5 – Таблица скидок (Discount)

Запрос SQL на добавление записи в таблицу Room

INSERT INTO Room (Number,Comfort,Price,NumberOfPeople)

values ('НОМЕР КОМНАТЫ' , 'КОМФОРТНОСТЬ' , 'ЦЕНА' , 'КОЛИЧЕСТВО ЧЕЛОВЕК')

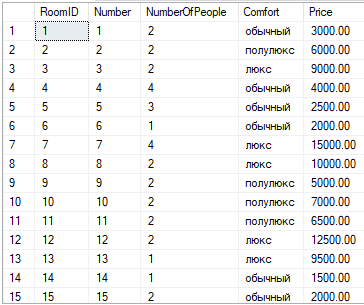


Рисунок 2.6 – Таблица комнат(Room)

Запрос SQL на добавление записи в таблицу Discount\_Settling

INSERT INTO Discount\_Settling (DiscountID, SettlingID)

VALUES ('ID СКИДКИ', 'ID ЗАСЕЛЕНИЯ')

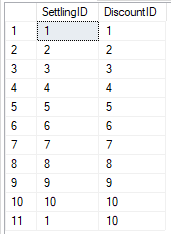


Рисунок 2.7 – Таблица ID скидок и заселений (Discount\_Settling)

Запрос SQL на добавление записи в таблицу Settling

INSERT INTO Settling( Commission, PositionID, SeekerID, DateOfDeal )

VALUES ('КОММИССИОННЫЕ', 'НОМЕР ВАКАНСИИ', 'НОМЕР СОИСКАТЕЛЯ', 'ДАТА СДЕЛКИ')

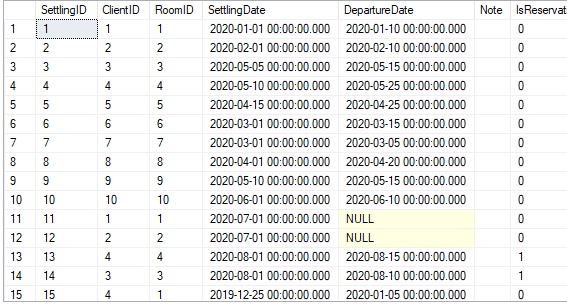


Рисунок 2.8 – Таблица заселений(Settling)

Далее проверяем работу представлений (Рисунок 11-12).

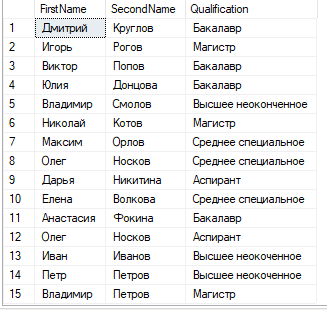


Рисунок 2.9 – Вывод представления JobSeeker\_BasicInfo

Создадим ограничение для таблицы Settling. Нам нужно чтобы нельзя было добавить запись у которой дата заселения позже даты выезда.

Создание ограничения:

ALTER TABLE Settling

ADD CONSTRAINT check\_date CHECK (DepartureDate > SettlingDate)

Проверка ограничения:

Конфликт инструкции INSERT с ограничением CHECK "check\_date". Конфликт произошел в базе данных "hosteldb", таблица "dbo.Settling".

Выполнение данной инструкции было прервано.

Далее создаем хранимую процедуру CreateSettlingWithDiscount, которая создает новое заселение и соединяет его со скидкой, запрос на создание процедуры представлен ниже.

CREATE PROCEDURE CreateSettlingWithDiscount

(

@SETTLING\_ClientID int,

@SETTLING\_RoomID int,

@SETTLING\_SettlingDate datetime,

@SETTLING\_DepartureDate datetime,

@SETTLING\_Note varchar(50),

@SETTLING\_IsReservation bit,

@DiscountID int

)

AS

BEGIN

INSERT INTO Settling(ClientID, RoomID, SettlingDate, DepartureDate, Note, IsReservation)

VALUES (@SETTLING\_ClientID, @SETTLING\_RoomID, @SETTLING\_SettlingDate, @SETTLING\_DepartureDate, @SETTLING\_Note, @SETTLING\_IsReservation)

INSERT INTO Discount\_Settling(SettlingID, DiscountID)

VALUES (SCOPE\_IDENTITY(), @DiscountID)

END

RETURN

Запрос на инициализацию процедуры:

exec CreateSettlingWithDiscount 1,1, '10.09.2020', '20.09.2020', '', 1, 1

После выполнения процедуры посмотрим добавленные записи в таблице (Рисунок 2.10 и Рисунок 2.11).



Рисунок 2.10 – Новая запись добавлена в таблицу Settling



Рисунок 2.11 – Новая запись добавлена в таблицу Discount\_Settling

Далее создадим триггер SetDefaultReservation для таблицы Settling, который будет выставлять значение 0(не зарезервированная), если при добавлении новой записи мы не указали значение в поле IsReservation.

Запрос на создание триггера:

CREATE TRIGGER SetDefaultReservation ON Settling

INSTEAD OF INSERT

AS

DECLARE

@ClientID int,

@RoomID int,

@SettlingDate datetime,

@DepartureDate datetime,

@Note varchar(50),

@IsReservation bit

SELECT @ClientID=ClientID, @RoomID=RoomID, @SettlingDate=SettlingDate, @DepartureDate=DepartureDate,@Note = Note, @IsReservation = IsReservation

FROM inserted

IF @IsReservation is NULL

BEGIN

INSERT INTO Settling(IsReservation, ClientID, RoomID, SettlingDate, DepartureDate, Note)

VALUES(0, @ClientID, @RoomID, @SettlingDate, @DepartureDate, @Note)

END

IF @IsReservation is not Null

BEGIN

INSERT INTO Settling(IsReservation, ClientID, RoomID, SettlingDate, DepartureDate, Note)

VALUES(@IsReservation, @ClientID, @RoomID, @SettlingDate, @DepartureDate, @Note)

END

Тестовый запрос для проверки триггера:

INSERT INTO Settling(SettlingDate, DepartureDate, Note, ClientID, RoomID) values('01.05.2020', '05.05.2020', '', 7, 7)

Проверяем таблицу Settling

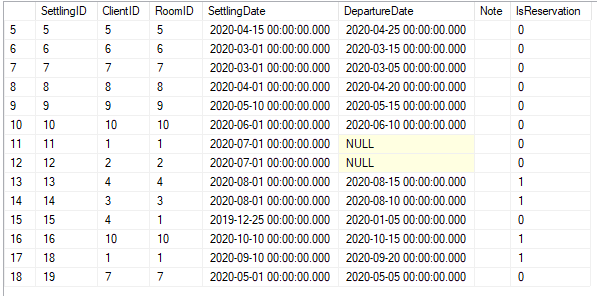


Рисунок 2.12 – Таблица Settling с новой записью

После проверки триггера выполним запросы, которые были даны ранее (1 этап: Системный анализ):

1. Вывести полные имена клиентов, чей id номер совпадает с номером комнаты.

select c.FirstName,c.SecondName,c.Patronymic from Client c

join Settling s on c.ClientID = s.ClientID

Where c.ClientID = s.RoomID



Рисунок 2.13 – Вывод запроса

1. Вывести количество клиентов со скидками, использовавших скидки

select COUNT(c.ClientID) as discount from Client c

join Settling s on c.ClientID = s.ClientID

join Discount\_Settling ds on ds.SettlingID = s.SettlingID



Рисунок 2.14 – Вывод запроса

1. Вывести количество постоянных клиентов

Select COUNT(c.ClientID) as pc from Client c

Where c.Comments = 'постоянный клиент'



Рисунок 2.15 – Вывод запроса

1. Вывести список номеров комнат на 3 и более человек и комфортностью люкс

select r.Number from Room r

where r.NumberOfPeople > 2 and r.Comfort = 'люкс'

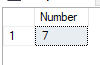


Рисунок 2.16 – Вывод запроса

1. Вывести номер и цену самого дорогого люкса

select ro.Number,ro.Price from Room ro

Where ro.Price = (

select Max(r.Price) as maxPrice from Room r

where r.Comfort = 'люкс')



Рисунок 2.17– Вывод запроса

1. Вывести паспортные данные и имена клиентов, которые бронировали номера

select c.FirstName,c.PasportData from Client c

join Settling s on c.ClientID = s.ClientID

Where s.IsReservation = 1

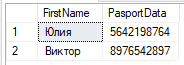


Рисунок 2.18 – Вывод запроса

1. Вывести паспортные данные и имена клиентов, которые не сдали свои номера

select c.FirstName,c.PasportData from Client c

join Settling s on c.ClientID = s.ClientID

where s.DepartureDate is null

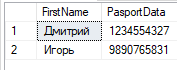


Рисунок 2.19 – Вывод запроса

1. Вывести количество бронированных номеров за 2020 год

select COUNT(s.SettlingID) as reservedRooms from Settling s

where s.IsReservation = 1 and YEAR(s.DepartureDate) = 2020



Рисунок 2.20 – Вывод запроса

1. Вывести среднюю стоимость не люкс номеров

select AVG(r.Price) as avgPrice from Room r

Where r.Comfort <> 'люкс'



Рисунок 2.21 – Вывод запроса

1. Вывести имена и фамилии клиентов, которые снимали номера на Новый Год

select c.FirstName, c.SecondName from Client c

join Settling s on c.ClientID = s.ClientID

where YEAR(s.DepartureDate)-YEAR(s.SettlingDate) = 1



Рисунок 2.22 – Вывод запроса

1. Вывести топ 3 самых популярных имен клиентов

select top 3 c.FirstName, Count(c.FirstName) as rate from Client c

group by c.FirstName

order by rate desc

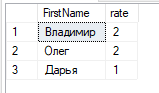


Рисунок 2.23 – Вывод запроса

1. Вывести названия скидок больше 10%

select d.Name from Discount d

where d.AmountOfDiscount > 10

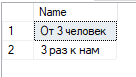


Рисунок 2.24 – Вывод запроса

1. Вывести топ 3 клиента по количеству одновременно использованных скидок

select top 3 c.FirstName, Count(c.ClientID) as kolvo from Client c

join Settling s on c.ClientID = s.ClientID

join Discount\_Settling ds on ds.SettlingID = s.SettlingID

group by c.FirstName

order by count(c.ClientID) desc

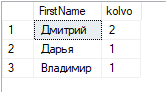


Рисунок 2.25 – Вывод запроса

1. Во сколько раз самый дорогой обычный номер для 3 и более человек дороже самого дешевого.

select (MAX(Room.Price)/MIN(Room.Price)) as voSkolko from Room

where Room.Comfort = 'обычный' and Room.NumberOfPeople > 2



Рисунок 2.26 – Вывод запроса

1. Вывести названия скидок, которые использовали при выборе люкс номеров

select Discount.Name from Settling

join Discount\_Settling on Discount\_Settling.SettlingID = Settling.SettlingID

join Room on Room.RoomID = Settling.RoomID

join Discount on Discount\_Settling.DiscountID = Discount.DiscountID

Where Room.Comfort = 'люкс'

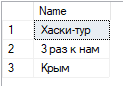


Рисунок 2.27 – Вывод запроса

# 3. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

На данном этапе, необходимо спроектировать приложение-ассистент для быстрой работы с базой данных. Проектирование приложения основывалось по следующим требованиям:

Обеспечение ведения списка клиентов:

- Обеспечение ведения списка услуг:

- Обеспечение ведения списка номеров:

- Обеспечение ведения списка скидок:

- Обеспечить возможность сортировки по всем имеющимся полям:

- Обеспечить возможность поиска записей в базе данных:

- Обеспечить выгрузку данных в таблицы Excel;

Главное окно приложения представлено на рисунке 3.1.



Рисунок 3.1 – Главное окно приложения

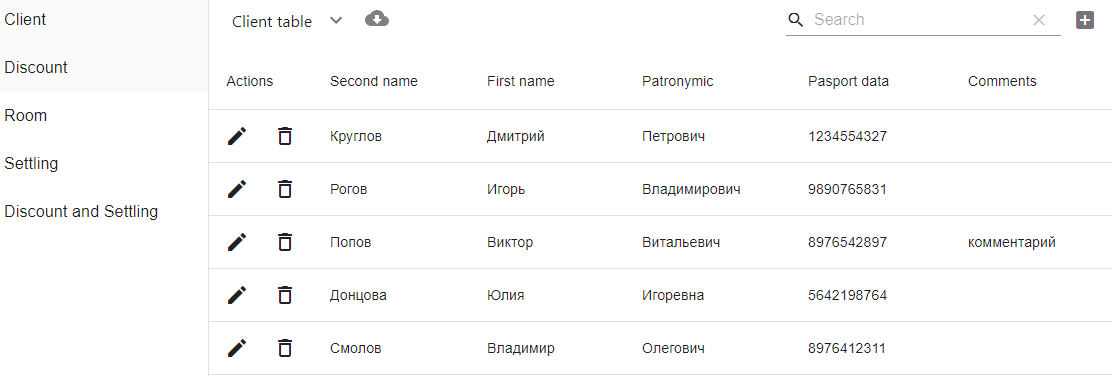
Нажимая кнопки в левой части приложения, можно переключаться между таблицами (Рисунки 3.2-3.6). 

Рисунок 3.2 – «Client table»

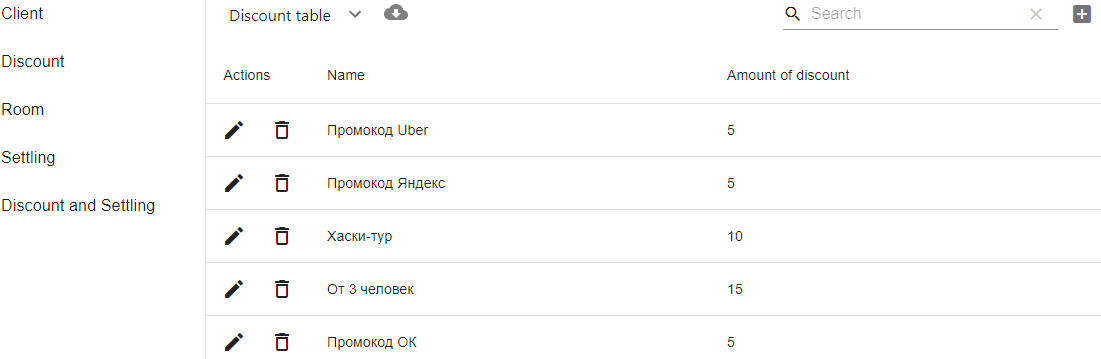


Рисунок 3.3 – «Discount table»



Рисунок 3.4 – «Room table»

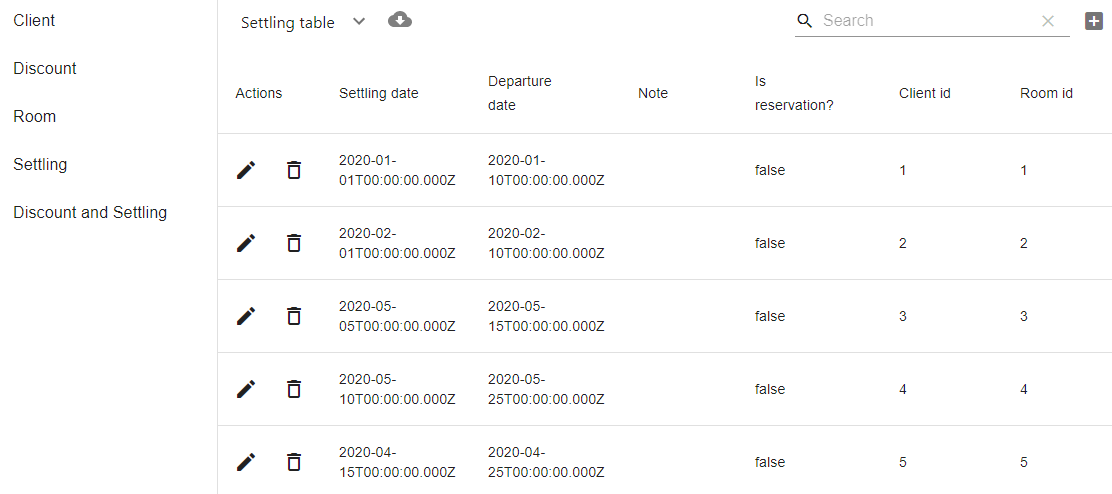


Рисунок 3.5 – «Settling table»



Рисунок 3.6 – «Discount and Settling table»

Далее рассмотрим подробнее вид окон и работу с разными таблицами:

## 3.1 Таблица «Клиенты»

Кнопка «+» в правом верхнем углу добавляет новую запись в таблицу (Рисунок 3.7).



Рисунок 3.7 – Добавление клиента

Кнопка в виде «карандаша» загружает в форму «Клиента» данные строки, чтобы ее изменить (Рисунок 3.8).

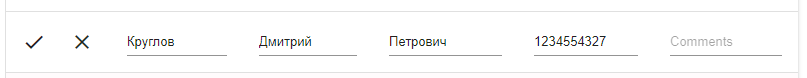


Рисунок 3.8 – Изменение клиента

Кнопка в виде «Корзины» удаляет выбранную запись из таблицы (Рисунок 3.9).



Рисунок 3.9 – Удаление клиента

## 3.2 Таблица «Скидки»

Кнопка «+» в правом верхнем углу добавляет новую запись в таблицу (Рисунок 3.10).



Рисунок 3.10 – Добавление скидки

Кнопка в виде «Карандаша» загружает в форму «Скидки» данные строки, чтобы ее изменить (Рисунок 3.11).

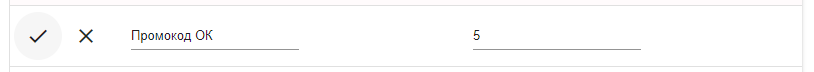


Рисунок 3.11 – Изменение скидки

Кнопка в виде «Корзины» удаляет выбранную запись из таблицы (Рисунок 3.12).



Рисунок 3.12 – Удаление скидки

## 3.3 Таблица «Комната»

Кнопка «+» в правом верхнем углу добавляет новую запись в таблицу (Рисунок 3.13).

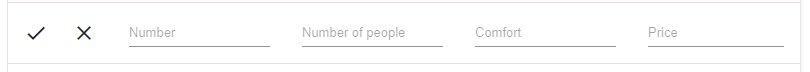


Рисунок 3.13 – Добавление комнаты

Кнопка в виде «Карандаша» загружает в форму «Комнаты» данные строки, чтобы ее изменить (Рисунок 3.14).



Рисунок 3.14 – Изменение Комнаты

Кнопка в виде «Корзины» удаляет выбранную запись из таблицы (Рисунок 3.15).



Рисунок 3.15 – Удаление комнаты

## 3.4 Таблица «Заселения»

Кнопка «+» в правом верхнем углу добавляет новую запись в таблицу (Рисунок 3.16).



Рисунок 3.16 – Добавление заселения

Кнопка в виде «Карандаша» загружает в форму «Заселения» данные строки, чтобы ее изменить (Рисунок 3.17).

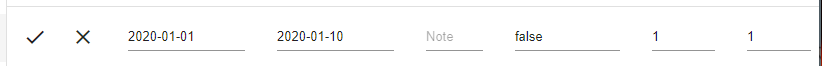


Рисунок 3.17 – Изменение заселения

Кнопка в виде «Корзины» удаляет выбранную запись из таблицы (Рисунок 3.18).



Рисунок 3.18 – Удаление заселения

## 3.5 Таблица «Скидки и Заселения»

Кнопка «+» в правом верхнем углу добавляет новую запись в таблицу (Рисунок 3.19).

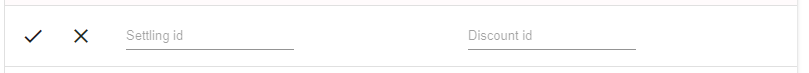


Рисунок 3.19 – Добавление скидки и заселения

Кнопка в виде «Карандаша» загружает в форму «Скидки и Заселения» данные строки, чтобы ее изменить (Рисунок 3.20).

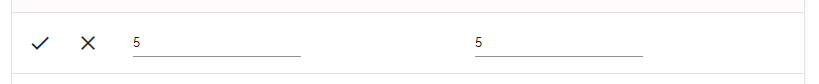


Рисунок 3.20 – Изменение скидки и заселения

Кнопка в виде «Корзины» удаляет выбранную запись из таблицы (Рисунок 3.21).



Рисунок 3.21 – Удаление скидки и заселения

## 3.6 Поиск записей

Для поиска записей в базе данных, необходимо выбрать нужную таблицу, затем при заполнении строки поиска записи, будут отображаться результаты поиска. Строка поиска представлено на рисунке 3.22.



Рисунок 3.22 – Окно поиска

Для поиска записей в базе данных, также можно использовать фильтрацию, для этого необходимо сначала выбрать нужную таблицу, затем нажать на кнопку в виде “стрелки вниз”. Окна фильтров для разных таблиц представлено на рисунках 3.23-3.27.

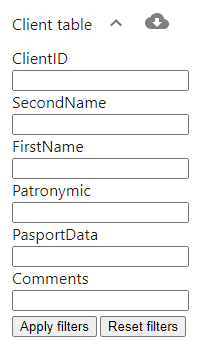


Рисунок 3.23 – Фильтры для клиентов

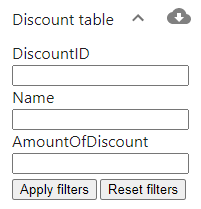


Рисунок 3.24 – Фильтры для скидок

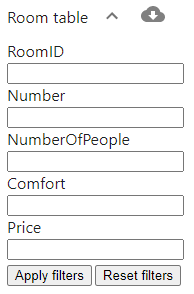


Рисунок 3.25 – Фильтры для комнат

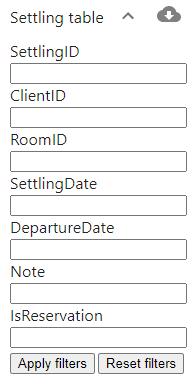


Рисунок 3.26 – Фильтры для заселения

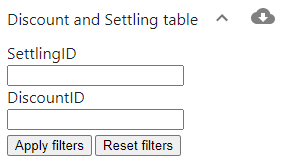


Рисунок 3.27 – Фильтры для скидок и заселения

Рассмотрим пример использования фильтра. Найдем все комнаты, где количество мест 2 и комфортностью ‘полулюкс’ (Рисунок 3.28 и Рисунок 3.29).

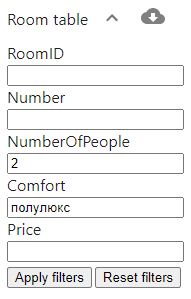


Рисунок 3.28 – Поиск полулюкса с количеством человек 2



Рисунок 3.29 – Результат фильтрации

## 3.7 Экспорт таблицы Excel

Для выгрузки таблицы Excel необходимо сначала выбрать таблицу для сохранения, затем нажать на кнопку в виде ‘облака’, после этого сразу загрузиться, и откроется файл excel с нужной таблицей (Рисунок 3.30).

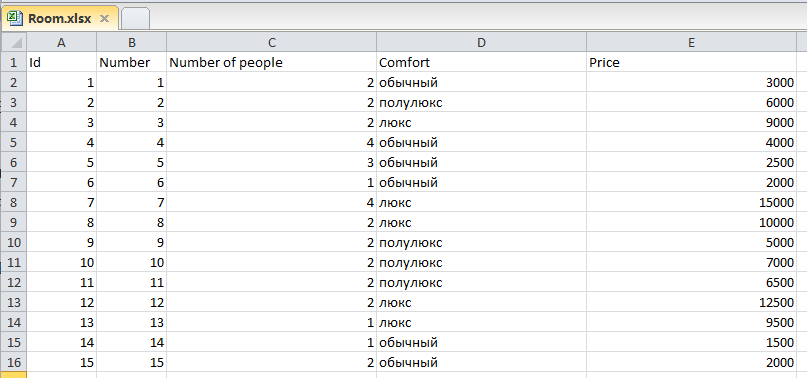


Рисунок 3.30 – Результат экспорта в Microsoft Excel

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта по заданию «Гостиница» были выполнены задачи по анализу поставленной задачи, создания физической модели.

На основе физической модели была создана реляционная база данных, в которой дополнительно были созданы триггеры, ограничения, хранимые процедуры для корректной работы базы данных.

Было спроектировано приложение-ассистент для работы с данной базой данных. В этом приложении были реализованы функции просмотра, добавления, изменения и удаления записей базы данных, поиска записей, фильтрация для записей и экспорт.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Требования к оформлению квалификационных работ: метод. указания для студентов по направлению 230200 «Информационные системы» / Сост.: А.П.Власов, Н.А. Марчук: Иван. гос. хим.-технол. ун-т. – Иваново, 2010, 35 с.
2. Проектирование баз данных: лабораторный практикум / Э.Г. Галиаскаров, А.Ю. Крылов; Иван. гос. хим.-технол. ун-т.- Иваново, 2012.- 96 с.
3. Разработка приложений баз данных: лабораторный практикум / Э.Г. Галиаскаров и др.; Иван. гос. хим.-технол. ун-т. – Иваново, 2015. – 112 с.
4. Сайт обмена знаниями и решения проблем StackOverflow [Электронный ресурс] URL <https://stackoverflow.com/> (дата обращения 15.04.2020)
5. Упражнения по SQL [Электронный ресурс] URL <http://sql-ex.ru/> (дата обращения 15.04.2020)
6. Татьяна Карпова, курс «Базы данных: модели, разработка, реализация» Сайт «НОУ ИНТУИТ» [Электронный ресурс] URL <https://www.intuit.ru/studies/courses/1001/297/info> (дата обращения 01.04.2020)
7. Сайт с документацией по react, React [Электронный ресурс] URL <https://ru.reactjs.org/> (дата обращения 15.04.20)

# ПРИЛОЖЕНИЕ А. СКРИПТ БАЗЫ ДАННЫХ

CREATE TABLE [Client]

(

[ClientID] integer NOT NULL IDENTITY ( 1,1 ) ,

[SecondName] varchar(50) NOT NULL ,

[FirstName] varchar(50) NOT NULL ,

[Patronymic] varchar(50) NULL ,

[PasportData] varchar(10) NOT NULL ,

[Comments] varchar(250) NULL

)

ALTER TABLE [Client]

ADD CONSTRAINT [XPKClient] PRIMARY KEY CLUSTERED ([ClientID] ASC)

CREATE TABLE [Discount]

(

[DiscountID] integer NOT NULL IDENTITY ( 1,1 ) ,

[Name] varchar(50) NULL ,

[AmountOfDiscount] integer NULL

)

ALTER TABLE [Discount]

ADD CONSTRAINT [XPKDiscount] PRIMARY KEY CLUSTERED ([DiscountID] ASC)

CREATE TABLE [Discount\_Settling]

(

[SettlingID] integer NOT NULL ,

[DiscountID] integer NOT NULL

)

CREATE TABLE [Room]

(

[RoomID] integer NOT NULL IDENTITY ( 1,1 ) ,

[Number] integer NOT NULL ,

[NumberOfPeople] integer NOT NULL ,

[Comfort] varchar(10) NOT NULL ,

[Price] decimal(10,2) NOT NULL

)

ALTER TABLE [Room]

ADD CONSTRAINT [XPKRoom] PRIMARY KEY CLUSTERED ([RoomID] ASC)

CREATE TABLE [Settling]

(

[SettlingID] integer NOT NULL IDENTITY ( 1,1 ) ,

[ClientID] integer NOT NULL ,

[RoomID] integer NOT NULL ,

[SettlingDate] datetime NOT NULL ,

[DepartureDate] datetime NULL ,

[Note] varchar(250) NULL ,

[IsReservation] bit NULL

)

ALTER TABLE [Settling]

ADD CONSTRAINT [XPKSettling] PRIMARY KEY CLUSTERED ([SettlingID] ASC)

CREATE VIEW [ClientBasicInfo]([SecondName],[PasportData],[FirstName])

AS

SELECT [SecondName],[PasportData],[FirstName]

FROM [Client]

ALTER TABLE [Discount\_Settling]

ADD CONSTRAINT [R\_9] FOREIGN KEY ([SettlingID]) REFERENCES [Settling]([SettlingID])

ALTER TABLE [Discount\_Settling]

ADD CONSTRAINT [R\_8] FOREIGN KEY ([DiscountID]) REFERENCES [Discount]([DiscountID])

ALTER TABLE [Settling]

ADD CONSTRAINT [R\_2] FOREIGN KEY ([ClientID]) REFERENCES [Client]([ClientID])

ALTER TABLE [Settling]

ADD CONSTRAINT [R\_6] FOREIGN KEY ([RoomID]) REFERENCES [Room]([RoomID])

CREATE TRIGGER tD\_Client ON Client FOR DELETE AS

/\* erwin Builtin Trigger \*/

/\* DELETE trigger on Client \*/

BEGIN

DECLARE @errno int,

@severity int,

@state int,

@errmsg varchar(255)

/\* erwin Builtin Trigger \*/

/\* Client Заселяет Settling on parent delete restrict \*/

/\* ERWIN\_RELATION:CHECKSUM="00010dae", PARENT\_OWNER="", PARENT\_TABLE="Client"

CHILD\_OWNER="", CHILD\_TABLE="Settling"

P2C\_VERB\_PHRASE="Заселяет", C2P\_VERB\_PHRASE="",

FK\_CONSTRAINT="R\_2", FK\_COLUMNS="ClientID" \*/

IF EXISTS (

SELECT \* FROM deleted,Settling

WHERE

/\* %JoinFKPK(Settling,deleted," = "," AND") \*/

Settling.ClientID = deleted.ClientID

)

BEGIN

SELECT @errno = 30001,

@errmsg = 'Cannot delete Client because Settling exists.'

GOTO error

END

CREATE TRIGGER tD\_Discount ON Discount FOR DELETE AS

/\* erwin Builtin Trigger \*/

/\* DELETE trigger on Discount \*/

BEGIN

DECLARE @errno int,

@severity int,

@state int,

@errmsg varchar(255)

/\* erwin Builtin Trigger \*/

/\* Discount Discount\_Settling on parent delete restrict \*/

/\* ERWIN\_RELATION:CHECKSUM="00012d89", PARENT\_OWNER="", PARENT\_TABLE="Discount"

CHILD\_OWNER="", CHILD\_TABLE="Discount\_Settling"

P2C\_VERB\_PHRASE="", C2P\_VERB\_PHRASE="",

FK\_CONSTRAINT="R\_8", FK\_COLUMNS="DiscountID" \*/

IF EXISTS (

SELECT \* FROM deleted,Discount\_Settling

WHERE

/\* %JoinFKPK(Discount\_Settling,deleted," = "," AND") \*/

Discount\_Settling.DiscountID = deleted.DiscountID

)

BEGIN

SELECT @errno = 30001,

@errmsg = 'Cannot delete Discount because Discount\_Settling exists.'

GOTO error

END

CREATE TRIGGER tD\_Room ON Room FOR DELETE AS

/\* erwin Builtin Trigger \*/

/\* DELETE trigger on Room \*/

BEGIN

DECLARE @errno int,

@severity int,

@state int,

@errmsg varchar(255)

/\* erwin Builtin Trigger \*/

/\* Room Сдается Settling on parent delete restrict \*/

/\* ERWIN\_RELATION:CHECKSUM="00010693", PARENT\_OWNER="", PARENT\_TABLE="Room"

CHILD\_OWNER="", CHILD\_TABLE="Settling"

P2C\_VERB\_PHRASE="Сдается", C2P\_VERB\_PHRASE="",

FK\_CONSTRAINT="R\_6", FK\_COLUMNS="RoomID" \*/

IF EXISTS (

SELECT \* FROM deleted,Settling

WHERE

/\* %JoinFKPK(Settling,deleted," = "," AND") \*/

Settling.RoomID = deleted.RoomID

)

BEGIN

SELECT @errno = 30001,

@errmsg = 'Cannot delete Room because Settling exists.'

GOTO error

END

CREATE TRIGGER tD\_Settling ON Settling FOR DELETE AS

/\* erwin Builtin Trigger \*/

/\* DELETE trigger on Settling \*/

BEGIN

DECLARE @errno int,

@severity int,

@state int,

@errmsg varchar(255)

/\* erwin Builtin Trigger \*/

/\* Settling Discount\_Settling on parent delete restrict \*/

/\* ERWIN\_RELATION:CHECKSUM="0001223b", PARENT\_OWNER="", PARENT\_TABLE="Settling"

CHILD\_OWNER="", CHILD\_TABLE="Discount\_Settling"

P2C\_VERB\_PHRASE="", C2P\_VERB\_PHRASE="",

FK\_CONSTRAINT="R\_9", FK\_COLUMNS="SettlingID" \*/

IF EXISTS (

SELECT \* FROM deleted,Discount\_Settling

WHERE

/\* %JoinFKPK(Discount\_Settling,deleted," = "," AND") \*/

Discount\_Settling.SettlingID = deleted.SettlingID

)

BEGIN

SELECT @errno = 30001,

@errmsg = 'Cannot delete Settling because Discount\_Settling exists.'

GOTO error

END

# ПРИЛОЖЕНИЕ Б. ПЕРЕЧЕНЬ ФАЙЛОВ

1. Пояснительная записка к курсовому проекту в папке doc Родина пояснительная записка.docx
2. Инфологическая модель в папке erwin logicalModel.erwin
3. Физическая модель базы данных в папке erwin physicalModel.erwin
4. Файлы базы данных в папке db hosteldb.mdf
5. Папки с исходным кодом приложения Server и Client