  
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«Дальневосточный федеральный университет»ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ  
Департамент программной инженерии и искусственного интеллекта**

**ОТЧЁТ**по лабораторной работе  
«Проектирование и разработка базы данных (MySQL)   
в выбранной предметной области»  
по дисциплине «Проектирование и разработка баз данных»  
Специальность 09.03.04 «Программная инженерия»

Выполнил:  
студент гр. Б9119-09.03.04прогин  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Сазонов Д.И.  
Проверил:  
старший преподаватель  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Краморенко Н.В.

г. Владивосток  
2022

**Оглавление**

[1. Описание предметной области «Гостиница» 3](#_Toc105863045)

[2. Построение концептуальной модели 4](#_Toc105863046)

[3. Проектирование и разработка БД 5](#_Toc105863047)

[3.1. Проектирование БД 5](#_Toc105863048)

[3.2. Перечень реализованных функций 6](#_Toc105863049)

[3.3. Выбор среды разработки 9](#_Toc105863050)

[4. Руководство пользователя 10](#_Toc105863051)

[4.1. Основные элементы управления 10](#_Toc105863052)

[Приложение 1. Прямое проектирование 14](#_Toc105863053)

[Приложение 2. Реализованные функции 17](#_Toc105863054)

# Описание предметной области «Гостиница»

Необходимо учесть следующие зависимости между объектами предметной области и их характеристиками:

* типы номеров имеют разные уровни сервиса, комфортности и оплаты,
* стоимость проживания в любом типе номеров устанавливается для одного человека (в сутки),
* каждый номер имеет определенное количество мест,
* гость может забронировать один или несколько номеров,
* в номере может проживать один или несколько гостей,
* от типа гостя зависит скидка на проживание в номере,
* номера обслуживаются персоналом гостиницы (горничные, электрики, сантехники),
* один сотрудник может обслуживать несколько номеров,
* гости могут оставлять жалобы о номерах, в которых они проживают.

Информация будет использоваться администрацией гостиницы. При работе с системой работник должен иметь возможность решать следующие задачи:

* добавлять и редактировать новые типы комнат и номера комнат,
* вносить в систему информацию о госте при его заселении,
* вносить информацию о сотрудниках гостиницы,
* вести ежедневный учет о состоянии комнат и учитывать жалобы гостей,
* добавлять информацию о том какой сотрудник какие номера обслуживает,
* осуществлять поиск проживающих гостей,
* обновлять информацию о номере, если в него заселились/выселились или ведется обслуживание,
* обновлять информацию о типе гостя и скидке на его проживание.

# Построение концептуальной модели

Необходимо построить такую базу данных, в которой хранится информация о гостях, проживающих в определенных типах комнат, о том сколько сотрудников работает в гостинице, а также вести учет обслуживания номеров.

Таким образом, можно выделить следующие сущности с их атрибутами:

* типы комнат (id типа, типы комнат, цена),
* сотрудники (id сотрудника, ФИО, должность, заработная плата, рабочий график),
* комнаты (номер комнаты, места, особенности номера, этаж, id типа, id сотрудника, статус комнаты),
* гости (номер паспорта, ФИО, гражданство, тип гостя, скидка),
* размещение (номер комнаты, номер паспорта, дата заезда, дата выезда),
* ежедневный учет (номер комнаты, дата обслуживания, статус комнаты, жалобы, оказанные услуги),

Результатом построения концептуальной модели является ER-диаграмма, приведенная на рис. 1.

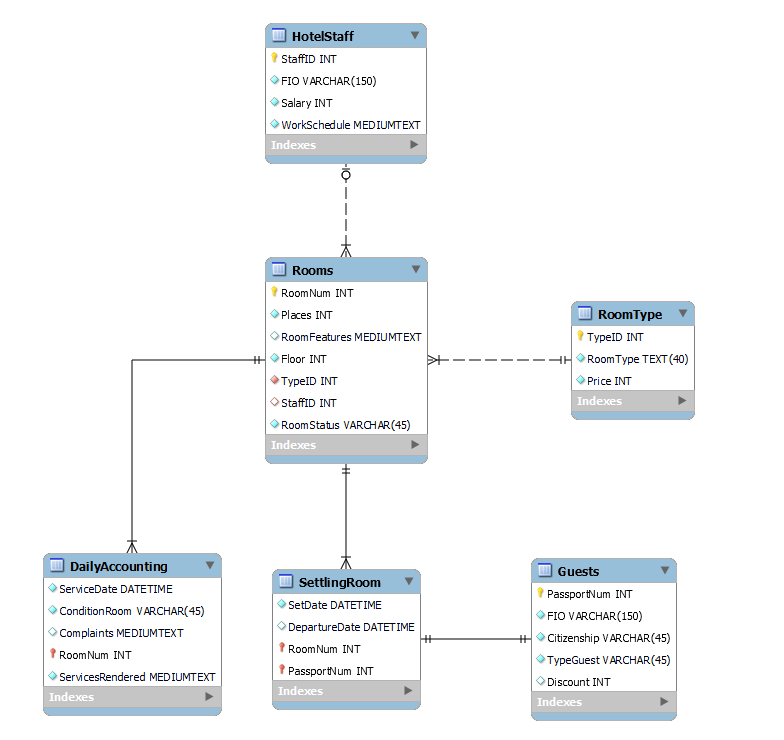


Рис. 1. ER-диаграмма

# Проектирование и разработка БД

# Проектирование БД

Для проектирования базы данных был использован MySQL Workbench и инструмент Forward Engineer, предоставляющий возможность сгенерировать скрипт создания базы данных из ER-диаграммы. Полученный скрипт, сгенерированный на основе созданной схемы (рис. 1), можно найти в приложении 1.

# Перечень реализованных функций

Все описанные далее запросы можно найти в приложении 2.

Запрос 1. Заполнение таблицы типов комнат первичными данными.

Запрос 2. Заполнение таблицы сотрудников первичными данными.

Запрос 3. Заполнение таблицы комнат первичными данными.

Запрос 4. Заполнение таблицы гостей первичными данными.

Запрос 5. Заполнение таблицы размещения гостей первичными данными.

Запрос 6. Заполнение таблицы ежедневного учета первичными данными.

Запрос 7. Триггер, срабатывающий при добавлении новых данных в таблицу ежедневного учета. Используется для изменения статуса номера в таблице комнат.

Запрос 8. Триггер, срабатывающий при обновлении данных в таблице ежедневного учета. Используется для изменения статуса номера в таблице комнат.

Запрос 9. Триггер, срабатывающий при удалении данных в таблице ежедневного учета. Используется для изменения статуса номера в таблице комнат.

Запрос 10. Выбор всех данных из таблицы типов комнат. Используется для работы с соответствующей таблицей в пользовательском интерфейсе.

Запрос 11. Выбор всех данных из таблицы сотрудников. Используется для работы с соответствующей таблицей в пользовательском интерфейсе.

Запрос 12. Выбор всех данных из таблицы комнат. Используется для работы с соответствующей таблицей в пользовательском интерфейсе.

Запрос 13. Выбор всех данных из таблицы гостей. Используется для работы с соответствующей таблицей в пользовательском интерфейсе.

Запрос 14. Выбор всех данных из таблицы размещения гостей. Используется для работы с соответствующей таблицей в пользовательском интерфейсе.

Запрос 15. Выбор всех данных из таблицы ежедневного учета. Используется для работы с соответствующей таблицей в пользовательском интерфейсе.

Запрос 16. Добавление типа комнаты в соответствующую таблицу. Значение полей задаются пользователем в форме.

Запрос 17. Удаление типа комнаты из соответствующей таблицы. Ключевое поле для удаления задается пользователем в форме.

Запрос 18. Изменение данных типа комнаты в соответствующей таблице. Значение полей задаются пользователем в форме. Изменяется заданная им запись.

Запрос 19. Выборка промежутка значений из таблицы типов комнат. Промежуток значений указывается пользователем в форме.

Запрос 20. Поиск и сортировка данных в таблице типов комнат. Значение поля и название колонки задаются пользователем в форме.

Запрос 21. Добавление сотрудника в соответствующую таблицу. Значение полей задаются пользователем в форме.

Запрос 22. Удаление сотрудника из соответствующей таблицы. Ключевое поле для удаления задается пользователем в форме.

Запрос 23. Изменение данных о сотруднике в соответствующей таблице. Значение полей задаются пользователем в форме. Изменяется заданная им запись.

Запрос 24. Выборка промежутка значений из таблицы сотрудников. Промежуток значений указывается пользователем в форме.

Запрос 25. Поиск и сортировка данных в таблице сотрудников. Значение поля и название колонки задаются пользователем в форме.

Запрос 26. Добавление комнаты в соответствующую таблицу. Значение полей задаются пользователем в форме.

Запрос 27. Удаление комнаты из соответствующей таблицы. Ключевое поле для удаления задается пользователем в форме.

Запрос 28. Изменение данных о комнате в соответствующей таблице. Значение полей задаются пользователем в форме. Изменяется заданная им запись.

Запрос 29. Выборка промежутка значений из таблицы комнат. Промежуток значений указывается пользователем в форме.

Запрос 30. Поиск и сортировка данных в таблице комнат. Значение поля и название колонки задаются пользователем в форме.

Запрос 31. Добавление гостя в соответствующую таблицу. Значение полей задаются пользователем в форме.

Запрос 32. Удаление гостя из соответствующей таблицы. Ключевое поле для удаления задается пользователем в форме.

Запрос 33. Изменение данных о госте в соответствующей таблице. Значение полей задаются пользователем в форме. Изменяется заданная им запись.

Запрос 34. Выборка промежутка значений из таблицы гостей. Промежуток значений указывается пользователем в форме.

Запрос 35. Поиск и сортировка данных в таблице гостей. Значение поля и название колонки задаются пользователем в форме.

Запрос 36. Добавление данных о размещении гостя в соответствующую таблицу. Значение полей задаются пользователем в форме.

Запрос 37. Удаление данных о размещении гостя из соответствующей таблицы. Ключевое поле для удаления задается пользователем в форме.

Запрос 38. Изменение данных о размещении гостя в соответствующей таблице. Значение полей задаются пользователем в форме. Изменяется заданная им запись.

Запрос 39. Выборка промежутка значений из таблицы размещение гостей. Промежуток значений указывается пользователем в форме.

Запрос 40. Поиск и сортировка данных в таблице размещения гостей. Значение поля и название колонки задаются пользователем в форме.

Запрос 41. Добавление данных о статуте комнаты и ее состоянии в таблицу ежедневного учета. Значение полей задаются пользователем в форме.

Запрос 42. Удаление данных о статуте комнаты и ее состоянии из таблицы ежедневного учета. Ключевое поле для удаления задается пользователем в форме.

Запрос 43. Изменение данных о статуте комнаты и ее состоянии в таблице ежедневного учета. Значение полей задаются пользователем в форме. Изменяется заданная им запись.

Запрос 44. Выборка промежутка значений из таблицы ежедневного учета. Промежуток значений указывается пользователем в форме.

Запрос 45. Поиск и сортировка данных в таблице ежедневного учета. Значение поля и название колонки задаются пользователем в форме.

# Выбор среды разработки

Интерфейс для работы с базой данных реализован на платформе Windows Forms в виде классического оконного приложения. Такой выбор обосновывается следующими факторами:

* устанавливается единожды на операционную систему для ПК,
* для работы в приложении не требуется интернет-соединение как это происходит в веб-приложениях.

Система состоит из нескольких частей.

1. База данных.
2. Интерфейс пользователя – клиентская подсистема, предоставляющая пользователю управлять данными на уровне объектов реального мира взаимодействуя с элементами в приложении и тем самым ограничивая от программного кода. Интерфейс состоит из одного окна со множеством вкладок, в которых динамически генерируются данные от взаимодействия пользователя с элементами управления.
3. Программная часть – является прослойкой между базой данных и интерфейсом пользователя. Реализация выполнена на языке программирования C#, который является высокоуровневым и объектно-ориентированным, что ускоряет процесс разработки. Все таблицы выделены в отдельные классы, что позволяет вызывать запросы для каждой таблицы индивидуально. Запрос хранится в коде в строковом значении и передается на прямую в БД. Для считывания данных из БД реализован отдельный класс заполнения таблиц, который выводит полученную информацию в пользовательский интерфейс.

# Руководство пользователя

# Основные элементы управления

На рис. 2 показан общий вид пользовательского интерфейса на вкладке с таблицей «Сотрудники».

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2. Общий вид приложения

Рассмотрим более детально основные элементы управления.

Основные вкладки – позволяют переключаться между таблицами (рис. 3). Также следующие элементы управления будут связаны с таблицей «Сотрудники» (рис. 4).



Рисунок 3. Основные вкладки

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4. Таблица «Сотрудники»

Справа от таблицы находится раздел для управления и редактирования данных в таблице «Сотрудники» (рис. 5).

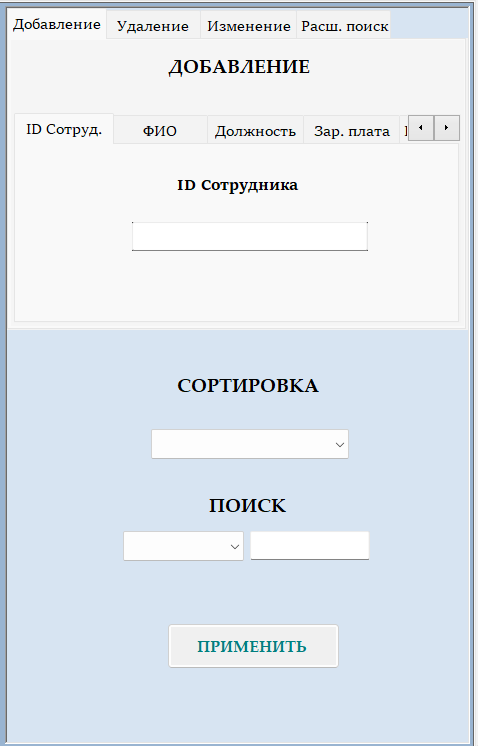


Рисунок 5. Раздел управления данными таблицы «Сотрудники»

В разделе управления данными таблицы можно отсортировать записи по определенному столбцу из выпадающего списка сортировки (рис. 6), а также найти определенную запись в таблице, выбрав из выпадающего списка столбец, по которому будет осуществляться поиск, и ввести значение этого столбца в текстовое поле рядом (рис. 7).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6. Сортировка

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7. Поиск

Для добавления, удаления, изменения и расширенного поиска значений таблицы предназначены специальные вкладки (рис. 8).



Рисунок 8. Вкладки редактирования данных

Каждая вкладка содержит текстовые полня для ввода соответствующих данных. Рассмотрим вкладку расширенного поиска, которая включается в себя поиск по диапазону значений для определенного столбца (рис. 9).

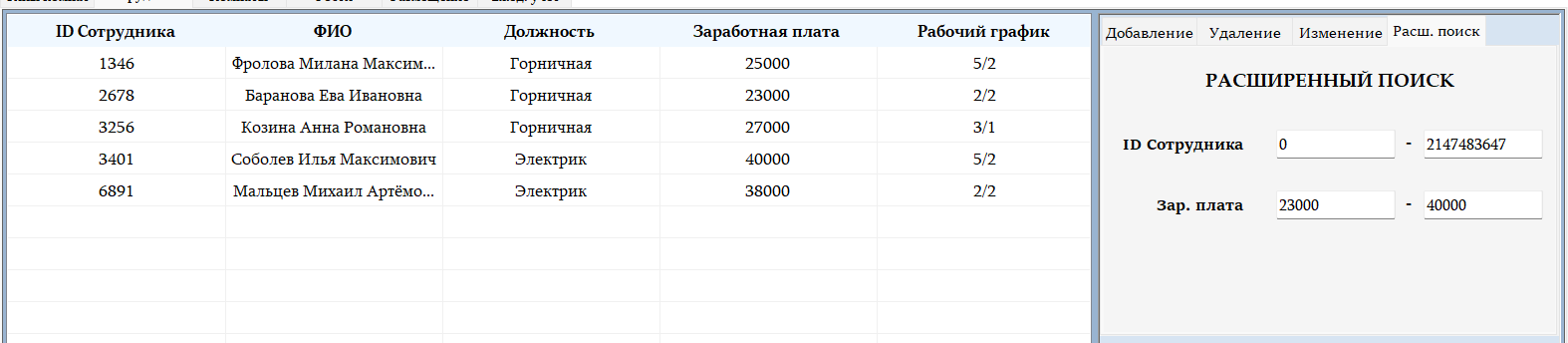


Рисунок 9. Расширенный поиск

Как видно на рисунке 9, незаполненные поля (ID Сотрудника) в расширенном поиске автоматически принимают значения по умолчанию.

Для изменения значения таблицы нужно выбрать ID Сотрудника и написать новое значения для соответствующего поля (рис 10).

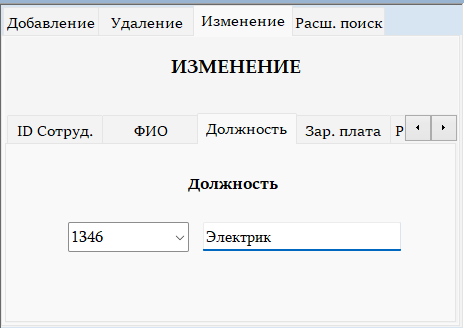


Рисунок 10. Изменение

# Приложение 1. Прямое проектирование

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

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

-- Schema Hotel

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

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

-- Schema Hotel

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

CREATE SCHEMA IF NOT EXISTS `Hotel` DEFAULT CHARACTER SET utf8 ;

USE `Hotel` ;

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

-- Table `Hotel`.`RoomType`

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

CREATE TABLE IF NOT EXISTS `Hotel`.`RoomType` (

`TypeID` INT UNSIGNED NOT NULL,

`RoomType` TEXT(40) NOT NULL,

`Price` INT NOT NULL,

PRIMARY KEY (`TypeID`))

ENGINE = InnoDB;

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

-- Table `Hotel`.`HotelStaff`

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

CREATE TABLE IF NOT EXISTS `Hotel`.`HotelStaff` (

`StaffID` INT UNSIGNED NOT NULL,

`FIO` VARCHAR(150) NOT NULL,

`Salary` INT UNSIGNED NOT NULL,

`WorkSchedule` MEDIUMTEXT NOT NULL,

PRIMARY KEY (`StaffID`))

ENGINE = InnoDB;

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

-- Table `Hotel`.`Rooms`

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

CREATE TABLE IF NOT EXISTS `Hotel`.`Rooms` (

`RoomNum` INT UNSIGNED NOT NULL,

`Places` INT UNSIGNED NOT NULL,

`RoomFeatures` MEDIUMTEXT NULL,

`Floor` INT UNSIGNED NOT NULL,

`TypeID` INT UNSIGNED NOT NULL,

`StaffID` INT UNSIGNED NULL,

`RoomStatus` VARCHAR(45) NOT NULL,

PRIMARY KEY (`RoomNum`),

INDEX `fk\_Rooms\_RoomType1\_idx` (`TypeID` ASC) VISIBLE,

INDEX `fk\_Rooms\_HotelStaff1\_idx` (`StaffID` ASC) VISIBLE,

CONSTRAINT `fk\_Rooms\_RoomType1`

FOREIGN KEY (`TypeID`)

REFERENCES `Hotel`.`RoomType` (`TypeID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Rooms\_HotelStaff1`

FOREIGN KEY (`StaffID`)

REFERENCES `Hotel`.`HotelStaff` (`StaffID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

-- Table `Hotel`.`Guests`

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

CREATE TABLE IF NOT EXISTS `Hotel`.`Guests` (

`PassportNum` INT UNSIGNED NOT NULL,

`FIO` VARCHAR(150) NOT NULL,

`Citizenship` VARCHAR(45) NOT NULL,

`TypeGuest` VARCHAR(45) NOT NULL,

`Discount` INT NULL,

PRIMARY KEY (`PassportNum`))

ENGINE = InnoDB;

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

-- Table `Hotel`.`SettlingRoom`

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

CREATE TABLE IF NOT EXISTS `Hotel`.`SettlingRoom` (

`SetDate` DATETIME NOT NULL,

`DepartureDate` DATETIME NULL,

`RoomNum` INT UNSIGNED NOT NULL,

`PassportNum` INT UNSIGNED NOT NULL,

PRIMARY KEY (`RoomNum`, `PassportNum`),

INDEX `fk\_SettlingRoom\_Rooms\_idx` (`RoomNum` ASC) VISIBLE,

INDEX `fk\_SettlingRoom\_Guests1\_idx` (`PassportNum` ASC) VISIBLE,

CONSTRAINT `fk\_SettlingRoom\_Rooms`

FOREIGN KEY (`RoomNum`)

REFERENCES `Hotel`.`Rooms` (`RoomNum`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_SettlingRoom\_Guests1`

FOREIGN KEY (`PassportNum`)

REFERENCES `Hotel`.`Guests` (`PassportNum`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

-- Table `Hotel`.`DailyAccounting`

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

CREATE TABLE IF NOT EXISTS `Hotel`.`DailyAccounting` (

`ServiceDate` DATETIME NOT NULL,

`ConditionRoom` VARCHAR(45) NOT NULL,

`Complaints` MEDIUMTEXT NULL,

`RoomNum` INT UNSIGNED NOT NULL,

`ServicesRendered` MEDIUMTEXT NOT NULL,

PRIMARY KEY (`RoomNum`),

INDEX `fk\_DailyAccounting\_Rooms1\_idx` (`RoomNum` ASC) VISIBLE,

CONSTRAINT `fk\_DailyAccounting\_Rooms1`

FOREIGN KEY (`RoomNum`)

REFERENCES `Hotel`.`Rooms` (`RoomNum`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

# Приложение 2. Реализованные функции

Примечание. В фигурных скобках указаны динамические данные, т.е. данные, получаемые в ходе использования системы пользователем.

Запрос 1.

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

-- Типы комнат --

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

Insert into RoomType (TypeID, RoomType, Price)

values

(1100, 'Однокомнатный', 4400),

(2200, 'Двухкомнатный', 5700),

(5601, 'Люкс', 6600),

(6301, 'Апартамент', 7400),

(2802, 'Семейный', 9800),

(3000, 'Сюит', 11200);

Запрос 2.

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

-- Сотрудники --

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

Insert into HotelStaff (StaffID, FIO, Post, Salary, WorkSchedule)

values

(1346, 'Фролова Милана Максимовна', 'Горничная', 25000, '5/2'),

(3401, 'Соболев Илья Максимович', 'Электрик', 40000, '5/2'),

(2678, 'Баранова Ева Ивановна', 'Горничная', 23000, '2/2'),

(7118, 'Коновалов Иван Миронович', 'Сантехник', 43000, '5/2'),

(3256, 'Козина Анна Романовна', 'Горничная', 27000, '3/1'),

(8693, 'Щербаков Тимур Иванович', 'Сантехник', 41000, '2/2'),

(5947, 'Родин Илья Александрович', 'Электрик', 45000, '3/1'),

(6891, 'Мальцев Михаил Артёмович', 'Электрик', 38000, '2/2'),

(9600, 'Артемов Александр Игоревич', 'Сантехник', 47000, '3/1');

Запрос 3.

-- ------- --

-- Комнаты --

-- ------- --

Insert into Rooms (RoomNum, Places, RoomFeatures, Floor, TypeID, StaffID, RoomStatus)

values

(145, 2, null, 1, 1100, null, 'свободно'),

(201, 3, null, 2, 2200, null, 'свободно'),

(426, 2, 'Кабинет, телевизор', 4, 5601, null, 'занято'),

(351, 4, 'Телефизор, зона для игр', 3, 2802, 5947, 'обслуживается'),

(457, 3, 'Кухня, кабинет, телевизор', 4, 6301, null, 'занято'),

(505, 6, 'Гостинная, кабинет, телевизор, проводной интернет', 5, 3000, null, 'свободно'),

(137, 4, null, 1, 2200, 2678, 'обслуживается'),

(309, 2, 'Кабинет, телевизор', 4, 5601, null, 'свободно'),

(346, 4, 'Телефизор, зона для игр', 3, 2802, null, 'свободно'),

(161, 2, null, 1, 1100, 9600, 'обслуживается');

Запрос 4.

-- ----- --

-- Гости --

-- ----- --

Insert into Guests (PassportNum, FIO, Citizenship, TypeGuest, Discount)

values

(866743, 'Максимова Агата Никитична', 'Россия', 'Обычный', null),

(491364, 'Семенов Макар Дмитриевич', 'Норвегия', 'Обычный', null),

(310846, 'Черкасов Захар Матвеевич', 'Индий', 'Постоянный', 800),

(738952, 'Лукьянова Стефания Дмитриевна', 'Греция', ' VIP', null),

(169003, 'Иванова Ева Львовна', 'Македония', 'Постоянный, VIP', 1300);

Запрос 5.

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

-- Размещение --

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

Insert into Placement (RoomNum, PassportNum, SetDate, DepartureDate)

values

(201, 866743, '2022-01-27 16:23:56', null),

(145, 491364, '2022-03-13 10:01:34', '2022-03-14 09:14:08'),

(505, 169003, '2022-02-08 13:22:11', null),

(346, 738952, '2022-03-17 17:11:29', '2022-03-20 17:00:21'),

(309, 310846, '2022-01-04 06:45:57', null);

Запрос 6.

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

-- Ежедневный учет --

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

insert DailyAccounting (RoomNum, ServiceDate, ConditionRoom, Complaints, ServicesRendered)

values

(351, '2022-03-18 18:13:41', 'обслуживается', 'Грязно, протекает труба в ванной', 'Убрано, починка'),

(137, '2022-03-21 20:43;12', 'обслуживается', 'Не работает кондиционер', 'Починка'),

(161, '2022-03-19 09:03:22', 'обслуживается', 'Грязно', 'Уборка');

Запрос 7.

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

-- Триггер, обновляющий статус комнаты при добавлении новых данных в

-- таблицу ежедневного учета.

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

DELIMITER //

CREATE TRIGGER insertStatusNumber AFTER INSERT ON DailyAccounting

FOR EACH ROW

BEGIN

if NEW.ConditionRoom = 'обслуживается' then

UPDATE Rooms Set RoomStatus = 'обслуживается' where RoomNum = NEW.RoomNum;

END IF;

END

// DELIMITER ;

Запрос 8.

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

-- Триггер, обновляющий статус комнаты при обновлении данных в

-- таблице ежедневного учета.

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

DELIMITER //

CREATE TRIGGER updateStatusNumber AFTER UPDATE ON DailyAccounting

FOR EACH ROW

BEGIN

if NEW.ConditionRoom = 'обслужено' then

UPDATE Rooms Set RoomStatus = 'свободно' where RoomNum = NEW.RoomNum;

END IF;

END

// DELIMITER ;

Запрос 9.

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

-- Триггер, обновляющий статус комнаты при удалении данных в

-- таблице ежедневного учета.

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

DELIMITER //

CREATE TRIGGER deleteStatusNumber BEFORE DELETE ON DailyAccounting

FOR EACH ROW

BEGIN

UPDATE Rooms Set RoomStatus = 'свободно' where RoomNum = OLD.RoomNum;

END

// DELIMITER ;

Запрос 10.

select\* from RoomType

Запрос 11.

select\* from HotelStaff;

Запрос 12.

select\* from Rooms;

Запрос 13.

select\* from Guests;

Запрос 14.

select\* from Placement;

Запрос 15.

select\* from DailyAccounting;

Запрос 16.

INSERT INTO RoomType values ({TypeID}, '{RoomType}', {Price})

Запрос 17.

DELETE FROM RoomType where TypeID = {TypeID}

Запрос 18.

UPDATE RoomType SET TypeID = {newTypeID} WHERE TypeID = {comboBoxes[0].Text}

Запрос 19.

SELECT \* FROM RoomType where (TypeID between {textBoxesAdv[0].Text} and {textBoxesAdv[1].Text}) and (Price between {textBoxesAdv[2].Text} and {textBoxesAdv[3].Text})

Запрос 20.

SELECT\* FROM RoomType WHERE {nameColomSearch} LIKE '{valueSearchTableRT}%' ORDER BY {nameColomSort}

Запрос 21.

INSERT INTO HotelStaff values ({textBox[0].Text}, '{textBox[1].Text}', '{textBox[2].Text}', {textBox[3].Text}, '{textBox[4].Text}')

Запрос 22.

DELETE FROM HotelStaff where StaffID = {StaffID}

Запрос 23.

UPDATE HotelStaff SET StaffID = {textBoxes[0].Text} WHERE StaffID = {comboBoxes[0].Text}

Запрос 24.

SELECT \* FROM HotelStaff where (StaffID between {textBoxesAdv[0].Text} and {textBoxesAdv[1].Text}) and (Salary between {textBoxesAdv[2].Text} and {textBoxesAdv[3].Text})

Запрос 25.

SELECT\* FROM HotelStaff WHERE {nameColomSearch} LIKE '{valueSearchTableHS}%' ORDER BY {nameColomSort}

Запрос 26.

INSERT INTO Rooms values ({textBox[0].Text}, {textBox[1].Text}, {textBox[2].Text}, {textBox[3].Text}, {comboBoxesAdd[0].Text}, {comboBoxesAdd[1].Text}, '{textBox[4].Text}')

Запрос 27.

DELETE FROM Rooms where RoomNum = {RoomNum}

Запрос 28.

UPDATE Rooms SET RoomNum = {textBoxes[0].Text} WHERE RoomNum = {comboBoxes[0].Text}

Запрос 29.

SELECT \* FROM Rooms where (RoomNum between {textBoxesAdv[0].Text} and {textBoxesAdv[1].Text}) and (Places between {textBoxesAdv[2].Text} and {textBoxesAdv[3].Text}) and (Floor between {textBoxesAdv[4].Text} and {textBoxesAdv[5].Text}) and (TypeID between {textBoxesAdv[6].Text} and {textBoxesAdv[7].Text})

Запрос 30.

SELECT\* FROM Rooms WHERE {nameColomSearch} LIKE '{valueSearchTableRooms}%' ORDER BY {nameColomSort}

Запрос 31.

INSERT INTO Guests values ({textBox[0].Text}, '{textBox[1].Text}', '{textBox[2].Text}', '{textBox[3].Text}', {textBox[4].Text})

Запрос 32.

DELETE FROM Guests where PassportNum = {PassportNum}

Запрос 33.

UPDATE Guests SET PassportNum = {textBoxes[0].Text} WHERE PassportNum = {comboBoxes[0].Text}

Запрос 34.

SELECT \* FROM Guests where PassportNum between {textBoxesAdv[0].Text} and {textBoxesAdv[1].Text}

Запрос 35.

SELECT\* FROM Guests WHERE {nameColomSearch} LIKE '{valueSearchTableGuests}%' ORDER BY {nameColomSort}

Запрос 36.

INSERT INTO Placement values ({comboBoxesReferences[0].Text}, {comboBoxesReferences[1].Text}, '{textBoxAdd[0].Text}', {textBoxAdd[1].Text})

Запрос 37.

DELETE FROM Placement where RoomNum = {textBoxesDel[0].Text} and PassportNum = {textBoxesDel[1].Text}

Запрос 38.

UPDATE Placement SET RoomNum = {comboBoxesReferences[2].Text} WHERE RoomNum = {comboBoxesUpd[0].Text}

Запрос 39.

SELECT \* FROM Placement where (RoomNum between {textBoxesAdv[0].Text} and {textBoxesAdv[1].Text}) and (PassportNum between {textBoxesAdv[2].Text} and {textBoxesAdv[3].Text}) and (SetDate between '{textBoxesAdv[4].Text}' and '{textBoxesAdv[5].Text}')

Запрос 40.

SELECT\* FROM Placement WHERE {nameColomSearch} LIKE '{valueSearchTablePlacement}%' ORDER BY {nameColomSort}

Запрос 41.

INSERT INTO DailyAccounting values ({comboBoxesReferences[0].Text}, '{textBoxAdd[0].Text}', '{textBoxAdd[1].Text}', '{textBoxAdd[2].Text}', '{textBoxAdd[3].Text}')

Запрос 42.

DELETE FROM DailyAccounting where RoomNum = {RoomNum}

Запрос 43.

UPDATE DailyAccounting SET RoomNum = {comboBoxesReferences[1].Text} WHERE RoomNum = {comboBoxesUpd[0].Text}

Запрос 44.

SELECT \* FROM DailyAccounting where (RoomNum between {textBoxesAdv[0].Text} and {textBoxesAdv[1].Text}) and (ServiceDate between '{textBoxesAdv[2].Text}' and '{textBoxesAdv[3].Text}')

Запрос 45.

SELECT\* FROM DailyAccounting WHERE {nameColomSearch} LIKE '{valueSearchTableDA}%' ORDER BY {nameColomSort}