

Мінстерство освіти та науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет Інформатики та обчислювальної техніки

Кафедра Інформатики та програмної інженерії

Дисципліна Бази даних

Курс II Група IT-02 Семестр 3

ЗАВДАННЯ

на курсову роботу студента

Тригуб Діана Костянтинівна

1. Тема роботи Проектування та розробка реляційної бази даних «База даних готельного комплексу»

2. Термін здачі студентом закінченої роботи « 24 » грудня 2020 р.

3. Вхідні дані до роботи система управління базами даних MySQL

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

Зміст, вступ, теоретична частина (аналіз та опис предметної області, проектування бази даних), практична частина (створення бази даних, організація вибірки інформації з бази даних), висновки, список використаної літератури, додатки (заповнення таблиць реляційної бази даних, створення збережених процедур, створення тригерів, створення представлень)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання « 30 » жовтня 2020 р.

КАЛЕНДАРНИЙ ПЛАН-ГРАФІК

виконання курсової роботи

студентом Тригуб Д. К.

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання	Примітки керівника про виконання завдань
1	Видача завдання	30.10.21	
2	Вивчення літератури	15.11.21	
3	Аналіз предметного середовища	20.11.21	
4	Побудова ER-діаграми	25.11.21	
5	Створення бази даних	1.12.21	
6	Створення користувачів бази даних	5.12.21	
7	Імпорт даних з використанням засобів СУБД	10.12.21	
8	Створення запитів до розробленої БД	15.12.21	
9	Оптимізація роботи запитів (за необхідності)	20.12.21	

Дата видачі завдання «24» жовтня 2020 р.

Завдання прийняв до виконання _____

Д. К. Тригуб

Керівник _____

К.І. Ліщук

ВСТУП	5
Теоретична частина	7
1.1 Аналіз предметної області	7
1.1.1 Опис предметної області	7
1.1.2 Опис вхідних даних	8
1.1.3 Опис вихідних даних	8
1.2 Проектування бази даних	9
1.2.1 Інфологічна модель бази даних	9
1.2.1.1 Опис сутностей	9
1.2.1.2 Опис атрибутів	9
1.2.1.3 Опис зв'язків	12
1.2.1.4 Діаграма «сутність-зв'язок»	13
1.2.2 Нормалізація таблиць при проектуванні бази даних	13
1.2.3 Фізична модель бази даних	15
Практична частина	22
2.1 Створення бази даних за допомогою MySQL	22
2.1.1 Створення бази даних	22
За допомогою SQL-команди була створена база даних з назвою «Hotels». Вигляд команди представлено нижче	22
2.1.2 Створення таблиць бази даних	22
2.1.3 Створення збережених процедур	25
2.1.4 Створення збережених функцій	25
2.1.5 Створення тригерів	26
2.1.6 Створення користувачів таблиці	27
2.1.7 Створення View	28
2.2 Організація вибірки інформації	30
Висновок	39
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	41
Додаток А	42

ВСТУП

У наш час в просторі мережі інтернет зберігається купа даних, створені найрізноманітніші системи керування, які використовують для збереження даних певні бази даних. Так розвивається сфера використання баз даних, так створюються нові інструменти створення та керування ними, надаючи користувачам все більше корисних і зручних інструментів для створення, редагування та оперування базами даних.

Реляційна база даних – це цифрова база даних, заснована на реляційній моделі даних, запропонована Е. Ф. Коддом у 1970 р. [1]. Програмною системою, що використовується для ведення реляційних баз даних, є реляційна система управління базами даних (СУБД). Багато реляційних систем баз даних мають можливість використання SQL (Structured Query Language) для запитів та обслуговування бази даних [2].

Практично всі розробники сучасних додатків, які передбачають зв'язок із системами баз даних, орієнтуються на реляційні СУБД [3]. За даними аналітиків на 2010 рік, реляційні СУБД використовуються в абсолютній більшості великих проектів з розробки інформаційних систем. За результатами досліджень компанії IDC 2009 року всього близько 7% складають проекти, в яких використовуються СУБД нереляційних типу [4]. А за новими даними станом на грудень 2020 року у п'ятірці найбільш уживаних СУБД перших чотири місця належать реляційним СУБД (Oracle, MySQL, Microsoft SQL Server, PostgreSQL), на п'ятому ж місці знаходиться нереляційна СУБД MongoDB [5].

У курсовій роботі була спроектована і розроблена реляційна база даних «Готельного комплексу» за допомогою MySQL Server. Необхідно було вирішити такі завдання:

- 1) провести аналіз предметної області з описом вхідних та вихідних даних;
- 2) побудувати інфологічну модель бази даних (опис сутностей, атрибутів та зв'язків між сутностями);
- 3) під час проектування провести нормалізацію таблиць бази даних;
- 4) побудувати даталогічну модель бази даних;
- 5) за допомогою СУБД Microsoft SQL Server створити базу даних, таблиці, діаграму бази даних, збережені процедури, тригери та представлення;
- 6) заповнити таблиці даними;
- 7) виконати вибірку інформації з бази даних (просту, вибірку обчислювальних значень, з діапазону, з використанням шаблонів, при об'єднанні таблиць);
- 8) проаналізувати виконану роботу.

1. Теоретична частина

1.1 Аналіз предметної області

1.1.1 Опис предметної області

Предметна область: Готель

Основним завданням є проектування та розробка програми для автоматизації реєстрації готелю. Додаток передбачає ведення системного каталогу, що містить інформацію про гостей, про номери та оплаті номерів. У готелі працюють адміністратори, які реєструють гостей. При реєстрації гостей враховуються Повне ім'я гостя, номер паспорту, ІПН, номер телефону, Дати приїзду та від'їзду, Можливі замітки. Гості можуть вибрати номер однієї з категорій: економ, комфорт і люкс.

База даних повинна зберігати інформацію: про гостей, про номери, про оплату. При роботі з системою співробітник повинен за потребою оновлювати БД, створювати і виводити на друк вихідні документи. При оплаті проживання гостю видається чек, в якому зазначено номер квитанції, ім'я, прізвище гостя, код номера, кількість днів у номері, сума і адміністратор, що оформляє реєстрацію.

1.1.2 Опис вхідних даних

Для кожного виду вхідних даних вказується форма, у якій вони подаються, а також періодичність надходження і джерело даних. Під час розробки реляційної бази даних «Готельний комплекс» були виділені такі вхідні дані:

- Інформація про клієнтів
- Інформація про філії комплексу
- Інформація про типи кімнат в готелі
- Інформацію про додаткові послуги готелю
- Інформація про ціни на проживання в кімнатах в залежності від сезону

1.1.3 Опис вихідних даних

Вихідні дані формуються під час роботи ІС та взаємодії з базою даних. Вихідними даними у даній реляційній базі даних вихідні запити.

1.2 Проектування бази даних

Проектування реляційної бази даних «Готельного комплексу» починається зі створення концептуальної діаграми – легкої для сприйняття людьми. У концептуальній діаграмі створюється первинна архітектура бази даних, прописуються сутності, їх атрибути та зв'язки між ними. Ціль даного етапу – створити модель, що задовольнить усі необхідні потреби ІС.

1.2.1 Інфологічна модель бази даних

Далі буде описано сутності, основне призначення кожної, їх атрибути, типи зв'язків між сутностями та, як результат, представлено ER-діаграму для реляційної бази даних «Готельного комплексу».

1.2.1.1 Опис сутностей

Основні сутності ER-діаграми потрібні для базової роботи системи, допоміжні ж створюються для того, щоб ІС функціонувала правильно та ефективно. Саме для цього проводиться нормалізація бази даних.

Під час опису предметної області було виділено такі сутності:

- 1) HotelBranch
- 2) Room
- 3) Roomer
- 4) Booking
- 5) TypeOfService
- 6) AcomodationBill

Також під час нормалізації бази даних було створено додаткові таблиці: season, ProvisionOfServises, price, RoomType.

1.2.1.2 Опис атрибутів

1) HotelBranch

- hotelId
- HotelBranchName
- City
- Address
- Administrator

2) RoomType

- RoomTypeId
- NumberOfHumans
- RoomTypeName

3) Season

- SeasonId
- SeasonName

4) Price

- HotelBranch
- RoomType
- Season
- Price

5) Room

- RoomId
- RoomNumber
- Floor
- RoomType
- HotelBranch

6) Roomer

- RoomerId
- FullName

- PhoneNumber
- PassportNumber
- IPN

7) Booking

- BookingId
- DayOfBooking
- DateOfArrival
- DepartureDate
- Roomer
- Room
- Note

8) AcomodationBill

- PaymentDay
- Booking
- Total

9) TypeOfService

- TypeOfServiceId
- NameOfService
- Price

10) ServicesBill

- ServicesBillId
- PaymentDate
- Roomer
- Total

11) ProvisionServices

- TypeOfService
- ServicesBill
- ProvisionDate

1.2.1.3 Опис зв'язків

ER-діаграма чітко відображає типи зв'язків між сутностями. Під час проектування були визначені зв'язки «1: Б» – один до багатьох та «1: 1» – один до одного. Типи зв'язків, що були використані у курсовій роботі, представлені у таблиці нижче

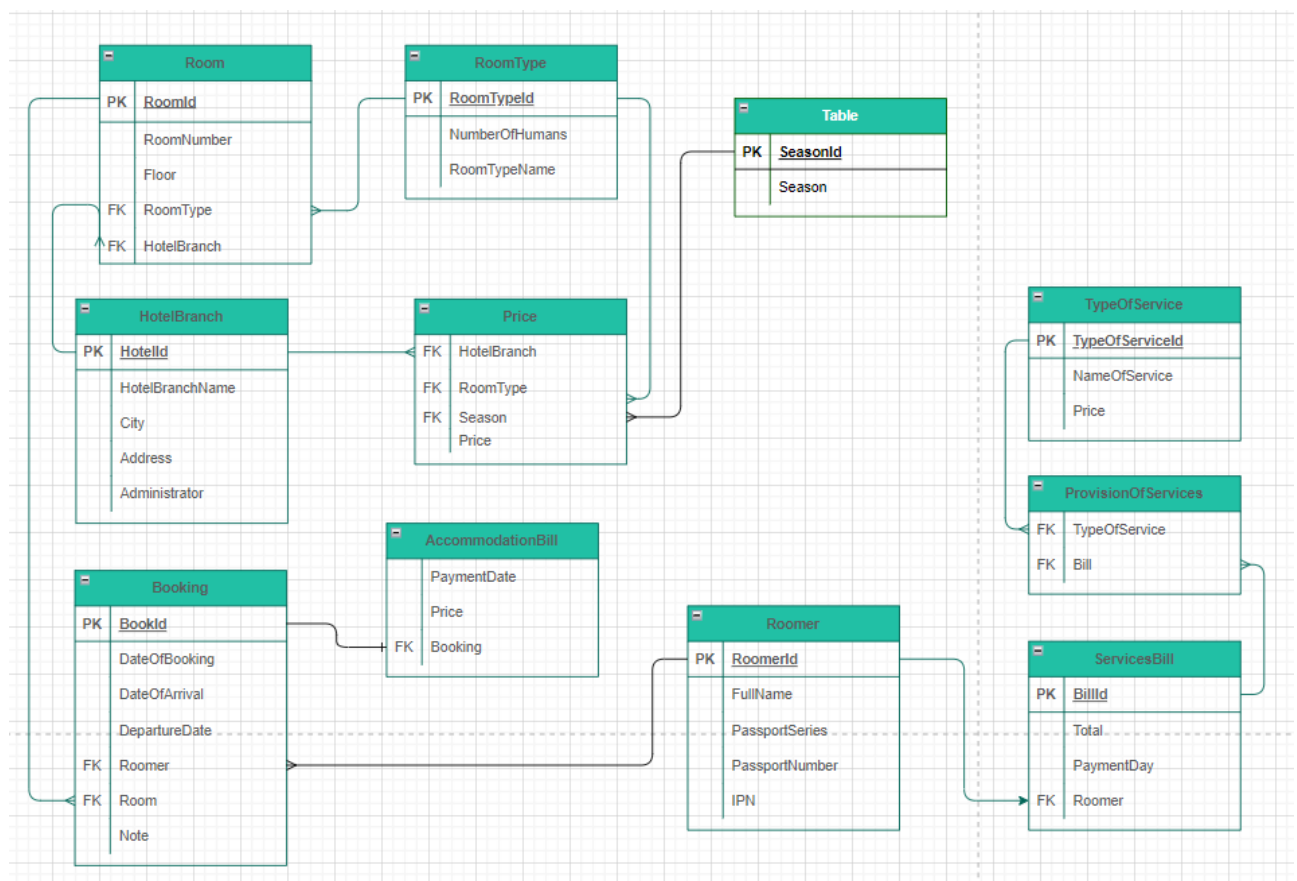
№	Головна таблиця	Дочірня таблиця	Тип зв'язку
1	HotelBranch	Room	1:Б
2	HotelBranch	Price	1:Б
3	RoomType	Room	1:Б
4	RoomType	Price	1:Б
5	Season	Price	1:Б
6	Room	Booking	1:Б
7	Booking	AcomodationBill	1:1
8	Roomer	Room	1:Б
9	Roomer	ServicesBill	1:Б

10	ServicesBill	ProvisionOfServices	1:Б
11	TypeOfService	ProvisionOfServices	1:Б

Тип «1: Б» означає, що об'єкт з головної таблиці може мати багато зв'язків із елементом із дочірньої таблиці, проте елемент із дочірньої таблиці може бути поєднаним лише з одним об'єктом із головної таблиці [7].

1.2.1.4 Діаграма «сутність-зв'язок»

Концептуальна або інфологічна (в нашому випадку ER-діаграма) має такий вигляд, що показано на рисунку нижче



1.2.2 Нормалізація таблиць при проектуванні бази даних

Модель даних необхідно приводити у нормальну форму. Для кожної ситуації визначається власний потрібний ступінь нормалізації.

Перша нормальна форма визначається такими правилами [8]:

- Кожна таблиця повинна мати основний ключ: мінімальний набір колонок, які ідентифікують запис;
- Уникнення повторень груп (категорії даних, що можуть зустрічатись різну кількість разів в різних записах) правильно визначаючи неключові атрибути;
- Атомарність: кожен атрибут повинен мати лише одне значення, а не множину значень.

Із чого виходить, що ER-діаграма повністю задовольняє заданим правилам.

Друга нормальна форма визначається так:

- Схема бази даних повинна відповідати вимогам першої нормальної форми;
- Дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці.

Перетворення моделі до другої нормальної форми виявило необхідність у створенні таких сутностей як statuses, companies, seat types, seats, positions. Значення атрибутів повторювалися б для багатьох елементів, тому було прийнято рішення про їх винесення у окремі таблиці.

Третя нормальна форма визначається так:

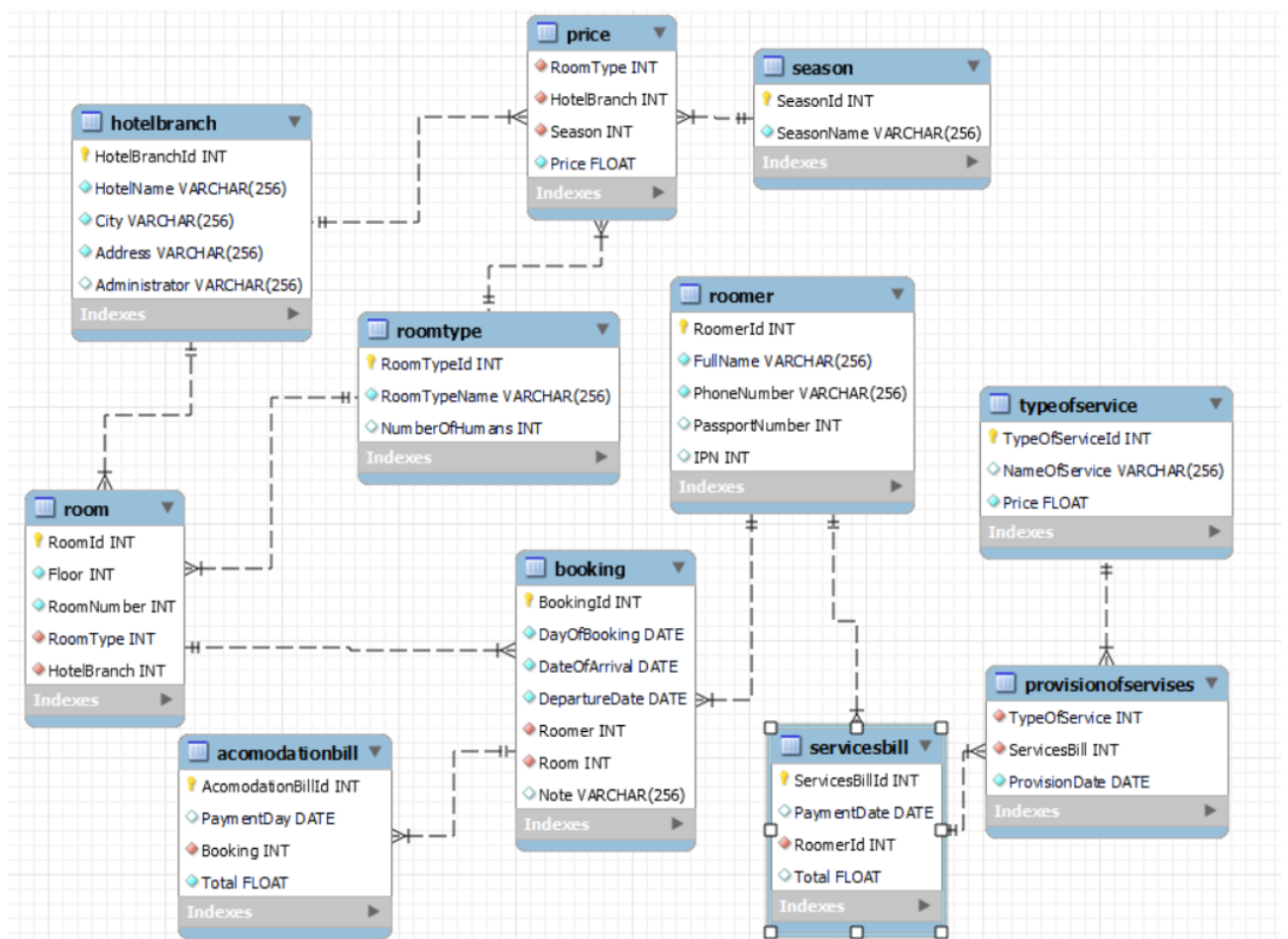
- Схема бази даних повинна відповідати всім вимогам другої нормальної форми;

– Будь-яке поле, що не залежить від основного ключа та від будь-якого іншого поля, має вноситись в окрему таблицю [8].

Створена модель повністю задовольняє правила третьої нормальної форми.

1.2.3 Фізична модель бази даних

На рисунку нижче показано фізичну модель реляційної бази даних «Продаж білетів в аеропорту», на якій вказані таблиці та зв'язки між ними, а також FK (foreign keys) – зовнішні ключі.



Для кожної таблиці та її атрибутів визначені певні властивості. Кожну таблицю описано у табл. 1.2-1.10.

Опис атрибутів таблиці HotelBranch.

Ім'я атрибута	Тип поля	Обмеження
HotelBranchId	Int	PK
HotelName	Varchar(256)	not null
City	Varchar(256)	NOT NULL
Address	Varchar(256)	NOT NULL
Administrator	Varchar(256)	

Опис атрибутів таблиці RoomType.

Ім'я атрибута	Тип поля	Обмеження
RoomTypeId	Int	PK
RoomTypeName	Varchar(256)	NOT NULL
NumberOfHumans	INT	

Опис атрибутів таблиці Season.

Ім'я атрибута	Тип поля	Обмеження
SeasonId	Int	PK
SeasonName	Varchar(256)	NOT NULL

Опис атрибутів таблиці Price.

Ім'я атрибута	Тип поля	Обмеження
HotelBranchId	Int	FK
RoomType	INT	FK, Not Null
Season	INT	FK, Not Null
Price	FLOAT	Not Null

Опис атрибутів таблиці Room

Ім'я атрибута	Тип поля	Обмеження
---------------	----------	-----------

RoomId	Int	PK
Floor	Int	Not Null
RoomNumber	Int	Not Null
RoomType	Int	Not Null
HotelBranch	Int	FK,Not Null

Опис атрибутів таблиці Roomer

Ім'я атрибута	Тип поля	Обмеження
RoomerId	Int	PK
FullName	Varchar(256)	NOT NULL
PhoneNumber	Varchar(256)	NOT NULL
PassportNumber	INT	
IPN	INT	

Опис атрибутів таблиці TypeOfService

Ім'я атрибута	Тип поля	Обмеження
TypeOfServiceId	Int	PK
NameOfService	Varchar(256)	Not Null
Price	FLOAT	Not null

Опис атрибутів таблиці Booking

Ім'я атрибута	Тип поля	Обмеження
BookingId	Int	PK
DateOfBooking	DATE	Not Null
DateOfArrival	DATE	Not Null
DepartureDate	DATE	Not Null

Roomer	INT	Not Null, FK
Room	INT	Not Null, FK
Note	Varchar(256)	

Опис атрибутів таблиці Accomodation

Ім'я атрибута	Тип поля	Обмеження
PaymentDay	Int	PK
Booking	INT	Not Null, FK
Price	FLOAT	Not Null

Опис атрибутів таблиці ProvisionOfServices

Ім'я атрибута	Тип поля	Обмеження
---------------	----------	-----------

TypeOfService	Int	Not Null, FK
Bill	INT	Not Null, FK

Опис атрибутів таблиці ServicesBill

Ім'я атрибута	Тип поля	Обмеження
BillId	Int	PK
Total	FLOAT	
PaymentDay	DATE	
Roomer		Not Null, FK

2. Практична частина

2.1 Створення бази даних за допомогою MySQL

2.1.1 Створення бази даних

За допомогою SQL-команди була створена база даних з назвою «Hotels». Вигляд команди представлено нижче

```
CREATE DATABASE IF NOT EXISTS Hotels3;
```

2.1.2 Створення таблиць бази даних

Під час роботи будуть створені таблиці, що описані у пункті 1.2.3

Код створення таблиці Hotels:

```
CREATE TABLE IF NOT EXISTS HotelBranch  
(  
    HotelBranchId INT PRIMARY KEY AUTO_INCREMENT,  
    HotelName Varchar(256) NOT NULL,  
    City Varchar(256) NOT NULL,  
    Address Varchar(256) NOT NULL,  
    Administrator Varchar(256)  
);
```

Код створення таблиці RoomType:

```
CREATE TABLE IF NOT EXISTS RoomType  
(  
    RoomTypeId INT PRIMARY KEY AUTO_INCREMENT,  
    RoomTypeName Varchar(256) NOT NULL,  
    NumberOfHumans INT  
);
```

Код створення таблиці Season:

```
CREATE TABLE IF NOT EXISTS Season
(
    SeasonId INT PRIMARY KEY AUTO_INCREMENT,
    SeasonName Varchar(256) NOT NULL
);
```

Код створення таблиці Price:

```
CREATE TABLE IF NOT EXISTS Price
(
    RoomType INT NOT NULL,
    HotelBranch INT NOT NULL,
    Season INT NOT NULL,
    Price FLOAT NOT NULL,
    FOREIGN KEY (RoomType) REFERENCES RoomType (RoomTypeId) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (HotelBranch) REFERENCES HotelBranch (HotelBranchId) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (Season) REFERENCES Season (SeasonId) ON UPDATE CASCADE ON DELETE CASCADE
);
```

Код створення таблиці Room:

```
CREATE TABLE IF NOT EXISTS Room
(
    RoomId INT PRIMARY KEY AUTO_INCREMENT,
    Floor INT NOT NULL,
    RoomNumber INT NOT NULL,
    RoomType INT NOT NULL,
    HotelBranch INT NOT NULL,
    FOREIGN KEY (RoomType) REFERENCES RoomType (RoomTypeId) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (HotelBranch) REFERENCES HotelBranch (HotelBranchId) ON UPDATE CASCADE ON DELETE CASCADE
);
```

Код створення таблиці Roomer:

```
CREATE TABLE IF NOT EXISTS Roomer
(
    RoomerId INT PRIMARY KEY AUTO_INCREMENT,
    FullName Varchar(256) NOT NULL,
    PhoneNumber Varchar(256) NOT NULL,
    PassportNumber INT,
    IPN INT
);
```

Код створення таблиці Booking:

```
CREATE TABLE IF NOT EXISTS Booking
(
    BookingId INT PRIMARY KEY AUTO_INCREMENT,
    DayOfBooking DATE NOT NULL,
    DateOfArrival DATE NOT NULL,
    DepartureDate DATE NOT NULL,
    Roomer INT NOT NULL,
    Room INT NOT NULL,
    Note Varchar(256),
    FOREIGN KEY (Roomer) REFERENCES Roomer (RoomerId) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (Room) REFERENCES Room (RoomId) ON UPDATE CASCADE ON DELETE CASCADE
);
```

Код створення таблиці TypeOfService:

```
CREATE TABLE IF NOT EXISTS TypeOfService
(
    TypeOfServiceId INT PRIMARY KEY AUTO_INCREMENT,
    NameOfService Varchar(256),
    Price FLOAT NOT NULL
);
```

Код створення таблиці ServicesBill:

```
CREATE TABLE IF NOT EXISTS ServicesBill
(
    ServicesBillId INT PRIMARY KEY AUTO_INCREMENT,
    PaymentDate DATE,
    RoomerId INT NOT NULL,
    Total FLOAT,
    FOREIGN KEY (RoomerId) REFERENCES Roomer (RoomerId) ON UPDATE CASCADE ON DELETE CASCADE
);
```

Код створення таблиці ProvisionOfServices:

```
CREATE TABLE IF NOT EXISTS ProvisionOfServices
(
    TypeOfService INT NOT NULL,
    ServicesBill INT NOT NULL,
    ProvisionDate DATE NOT NULL,
    FOREIGN KEY (TypeOfService) REFERENCES TypeOfService (TypeOfServiceId) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (ServicesBill) REFERENCES ServicesBill (ServicesBillId) ON UPDATE CASCADE ON DELETE CASCADE
);
```

Код створення таблиці AcomodationBill:


```
CREATE TABLE IF NOT EXISTS AcomodationBill
(
    AcomodationBillId INT PRIMARY KEY AUTO_INCREMENT,
    PaymentDay DATE,
    Booking INT NOT NULL,
    Total FLOAT NOT NULL,
    FOREIGN KEY (Booking) REFERENCES Booking (BookingId) ON UPDATE CASCADE ON DELETE CASCADE
);
```

2.1.3 Створення збережених процедур

Код процедури для визначення сезону в залежності від дати(Використовується в вбудованих функціях)

```
DROP PROCEDURE IF EXISTS deside_season;
DELIMITER $$
CREATE PROCEDURE deside_season(IN arrival_date INT, OUT season_ INT)
BEGIN
    DECLARE month_ TEXT;
    SET month_ = MONTH(arrival_date);
    IF month_ >= 3 AND month_ <= 5 THEN SET season_ = 2;
    ELSE IF month_ >= 6 AND month_ <= 8 THEN SET season_ = 3;
    ELSE IF month_ >= 9 AND month_ <= 11 THEN SET season_ = 4;
    ELSE SET season_ = 1;
    END IF;
    END IF;
    END IF;
END$$
```

2.1.4 Створення збережених функцій

Код створення функції, яка повертає суму, що повинен заплатити клієнт, за надані йому додаткові послуги

```

CREATE FUNCTION count_total_services(id INT)
RETURNS FLOAT
READS SQL DATA
DETERMINISTIC
) BEGIN
    DECLARE price_ FLOAT;
    SELECT SUM(Price) INTO price_
    From provisionofservises
    JOIN typeofservice ON TypeOfServiceId = TypeOfService
    WHERE ServicesBill = id;
    RETURN price_;
- END$$

```

2.1.5 Створення тригерів

Код створення тригеру, який при додаванні інформації до таблиці roomer автоматично створює записи у таблиці servicesbill

```

DROP TRIGGER IF EXISTS AddBill;
DELIMITER |
CREATE TRIGGER AddBill
AFTER INSERT ON roomer
FOR EACH ROW
INSERT INTO servicesbill(RoomerId, Total)
VALUE
    (NEW.RoomerId, 0);

```

Код створення тригера, який при додаванні інформації до provisionofservices автоматично змінює атрибут загальної суми в таблиці servicesbill для потрібних записів

```

DROP TRIGGER IF EXISTS UpdateBill;
DELIMITER |
CREATE TRIGGER UpdateBill
AFTER INSERT ON provisionofservises
)   FOR EACH ROW BEGIN
    UPDATE servicesbill
    SET total = count_total_services(NEW.ServicesBill)
    WHERE ServicesBillId = NEW.ServicesBill;
END |;

```

Код створення тригера, що додає запис в таблицю AcomodationBill після додавання запису в таблицю booking

```

DROP TRIGGER IF EXISTS AddAcomodation;
DELIMITER |
CREATE TRIGGER AddAcomodation
AFTER INSERT ON booking
    FOR EACH ROW
    INSERT INTO acomodationbill(Booking, Total)
    VALUE
        (NEW.BookingId, count_total_acomodation(NEW.Room, NEW.DepartureDate - NEW.DateOfArrival));

```

2.1.6 Створення користувачів таблиці

Код створення адміністратора готельного комплексу

```

DROP USER IF EXISTS 'Admin'@'localhost';
CREATE USER 'Admin'@'localhost' IDENTIFIED BY 'admin123';
GRANT ALL PRIVILEGES ON hotels3.* TO 'Admin'@'localhost';

```

Код створення ресепшійніста готельного комплексу

```

DROP USER IF EXISTS 'Receptionist'@'localhost';
CREATE USER 'Receptionist'@'localhost' IDENTIFIED BY 'receptionist123';
GRANT INSERT ON hotels3.booking TO 'Receptionist'@'localhost';
GRANT INSERT ON hotels3.provisionofservises TO 'Receptionist'@'localhost';
GRANT SELECT ON hotels3.roomer TO 'Receptionist'@'localhost';
GRANT SELECT ON hotels3.servicesbill TO 'Receptionist'@'localhost';

```

2.1.7 Створення View

View для всіх рахунків за проживання

```
CREATE OR REPLACE VIEW AcomodationBills As
SELECT FullName, PhoneNumber, Total FROM acomodationbill
JOIN booking ON Booking = BookingId
JOIN roomer ON RoomerId = Roomer;

SELECT * FROM acomodationbills;
```

	FullName	PhoneNumber	Total
►	Wang Reyes	1-366-366-4729	77697
	Ria Nolan	(392) 583-7759	55678
	Geoffrey Sherman	(711) 157-8410	73720
	Raya Strong	1-574-655-7486	8653
	Jeremy Matthews	(241) 573-1785	27840
	Odysseus Carney	1-823-772-3328	272440
	Brock Levine	1-978-824-4734	141868

View для всіх рахунків за додаткові послуги

```
CREATE OR REPLACE VIEW ServicesBills As
SELECT FullName, PhoneNumber, Total FROM servicesbill
JOIN roomer ON roomer.RoomerId = servicesbill.RoomerId;

SELECT * FROM ServicesBills;
```

	FullName	PhoneNumber	Total
	Quinn Flores	1-558-660-1136	0
	Fritz Frye	(202) 324-2884	0
	Amery Hill	1-457-388-1647	345
	Ocean Cantrell	(778) 405-2343	288
	Caryn Wade	(403) 756-3485	0
	Lara Fletcher	1-775-278-3652	659
	Reece Allen	(272) 562-9176	0
	Rashad Prince	(584) 874-4831	0
	Amal Snider	(171) 647-3145	0
	Coby Baldwin	(985) 732-6551	283
	Keefe Andrews	(342) 983-4862	346
	Dorothy Rutle...	(843) 351-2230	0

2.2 Організація вибірки інформації

Користувача при роботі з базою даних цікавить не весь її вміст, а деяка конкретна інформація. Знайти потрібні відомості можна послідовним переглядом записів. Однак такий спосіб пошуку незручний і малоефективний особливо при великій кількості записів.

Більшість систем управління базами даних дозволяють виконувати вибірку потрібної інформації шляхом виконання запитів. Користувач відповідно до певних правил формулює запит, указуючи, якими критеріями повинна задовольняти його цікавить інформація, а система виводить записи, що задовольняють запиту.

Використання запитів SQL є одним з найбільш ефективних і універсальних способів вибірки даних з таблиць бази даних

Нижче наведено SQL запити та результати їх виконання для бази даних готелю

- 1) В якій кімнаті живе/проживала/проживатиме конкретна людина?

```
SELECT RoomNumber, Floor, FullName
FROM room
JOIN booking ON Room = RoomId
JOIN roomer ON RoomerId = Roomer
WHERE FullName = 'Quinn Flores';
```

	RoomNumber	Floor	FullName
►	12	2	Quinn Flores

2) Всі кімнати такого типу

```
SELECT RoomNumber, Floor, RoomTypeName, HotelName
FROM room
JOIN roomtype ON RoomType = RoomTypeId
JOIN hotelbranch ON HotelBranchId = HotelBranch
WHERE RoomTypeName = 'semper';
```

	RoomNumber	Floor	RoomTypeName	HotelName
►	2	1	semper	Zeus Hotel
	3	1	semper	Zeus Hotel
	6	1	semper	Zeus Hotel

3) Ціна на кімнату конкретного типу в конкретну пору в конкретному готелі

```
SELECT RoomTypeName, HotelName, Price, SeasonName
FROM price
JOIN roomtype ON price.RoomType = RoomTypeId
JOIN hotelbranch ON HotelBranchId = price.HotelBranch
JOIN season ON SeasonId = Season
WHERE RoomTypeName = 'semper' AND HotelName = 'Zeus Hotel' AND SeasonName = 'Winter';
```

	RoomTypeName	HotelName	Price	SeasonName
►	semper	Zeus Hotel	1379	Winter

4) Найдешевша кімната в цьому готелі

```

SELECT HotelName, MIN(Price), RoomType, SeasonName
FROM price
JOIN hotelbranch ON HotelBranchId = HotelBranch
JOIN roomtype ON RoomTypeId = RoomType
JOIN season ON SeasonId = Season
WHERE HotelName = 'Zeus Hotel';

```

	HotelName	MIN(Price)	RoomType	SeasonName
▶	Zeus Hotel	538	1	Summer

5) Найдешевший тип кімнати в цей час в цьому готелі

```

SELECT HotelName, MIN(Price), RoomType, SeasonName
FROM price
JOIN hotelbranch ON HotelBranchId = HotelBranch
JOIN roomtype ON RoomTypeId = RoomType
JOIN season ON SeasonId = Season
WHERE HotelName = 'Zeus Hotel' AND SeasonName = 'Winter';

```

	HotelName	MIN(Price)	RoomType	SeasonName
▶	Zeus Hotel	764	1	Winter

6) Всі номери кімнат такого типу в цьому готелі

```

SELECT RoomNumber, Floor, HotelName, RoomTypeName
FROM room
JOIN roomtype ON RoomTypeId = RoomType
JOIN hotelbranch ON HotelBranchId = HotelBranch
WHERE HotelName = 'Padlockurity Hotel' AND RoomTypeName = 'eros'

```

	RoomNumber	Floor	HotelName	RoomTypeName
▶	1	1	Padlockurity Hotel	eros
	2	1	Padlockurity Hotel	eros

7) Всі типи кімнат у цьому готелі


```

SELECT RoomNumber, HotelName, RoomTypeName
FROM room
JOIN roomtype ON RoomTypeId = RoomType
JOIN hotelbranch ON HotelBranchId = HotelBranch
WHERE HotelName = 'Padlockurity Hotel';

```

	RoomNumber	HotelName	RoomTypeName
►	1	Padlockurity Hotel	eros
	2	Padlockurity Hotel	eros
	3	Padlockurity Hotel	consequat
	4	Padlockurity Hotel	consequat
	5	Padlockurity Hotel	consequat
	6	Padlockurity Hotel	aliquam
	7	Padlockurity Hotel	odio
	8	Padlockurity Hotel	metus

8) Середні ціни за номер у всіх готелях

```

SELECT AVG(Price) AS average_price,
(SELECT HotelName FROM hotelbranch WHERE HotelBranch = HotelBranchId) As HotelBranch
FROM price
group by HotelBranch

```

	average_price	HotelBranch
►	1218.075	Zeus Hotel
	1336.075	Hercules Hotel
	1168.35	Padlockurity Hotel

9) Всі хто живуть в готелі в конкретний час

```

SELECT FullName, RoomNumber, HotelName, DateOfArrival, DepartureDate
FROM booking
JOIN room ON Room = RoomId
JOIN roomer ON Roomer = RoomerId
JOIN hotelbranch ON HotelBranchId = HotelBranch
WHERE HotelName='Hercules Hotel' AND DepartureDate>='2021-11-01' AND DateOfArrival<='2021-11-01';

```

	FullName	RoomNumber	HotelName	DateOfArrival	DepartureDate
►	Raya Strong	11	Hercules Hotel	2021-11-01	2021-11-18
	Odysseus Carney	8	Hercules Hotel	2021-10-21	2021-12-17
	Brock Levine	7	Hercules Hotel	2021-11-01	2021-12-17
	Oscar Shaw	20	Hercules Hotel	2021-11-01	2021-11-18
	Marny Frank	4	Hercules Hotel	2021-10-28	2021-11-25
	Dara Vang	12	Hercules Hotel	2021-11-01	2021-11-16
	Lee Irwin	24	Hercules Hotel	2021-10-28	2021-11-16

10) Всі хто живуть у цьому готелі на конкретному поверсі на даний момент

```
SELECT FullName, RoomNumber, HotelName, DateOfArrival, DepartureDate, Floor
FROM booking
JOIN room ON Room = RoomId
JOIN roomer ON Roomer = RoomerId
JOIN hotelbranch ON HotelBranchId = HotelBranch
WHERE Floor=1 AND HotelName='Hercules Hotel' AND DepartureDate>='2021-11-01' AND DateOfArrival<='2021-11-01';
```

	FullName	RoomNumber	HotelName	DateOfArrival	DepartureDate	Floor
►	Marny Frank	4	Hercules Hotel	2021-10-28	2021-11-25	1
	Brock Levine	7	Hercules Hotel	2021-11-01	2021-12-17	1
	Odysseus Carney	8	Hercules Hotel	2021-10-21	2021-12-17	1

11) Хто проживає в кімнаті в конкретну дату

```
SELECT FullName, RoomNumber, Floor, HotelName, DateOfArrival, DepartureDate
FROM booking
JOIN room ON Room = RoomId
JOIN roomer ON Roomer = RoomerId
JOIN hotelbranch ON HotelBranchId = HotelBranch
WHERE RoomNumber=3 AND HotelName='Zeus Hotel' AND DepartureDate>='2021-11-01' AND DateOfArrival<='2021-11-01';
```

	FullName	RoomNumber	Floor	HotelName	DateOfArrival	DepartureDate
►	Wang Reyes	3	1	Zeus Hotel	2021-10-21	2021-11-18

12) Усі вільні кімнати на конкретну дату

```
SELECT * FROM room WHERE RoomId NOT IN
(SELECT Room FROM booking WHERE DepartureDate>='2021-11-01' AND DateOfArrival<='2021-11-01');
```

	BookingId	DayOfBooking	DateOfArrival	DepartureDate	Roomer	Room	Note
►	1	2021-10-01	2021-10-21	2021-11-18	16	3	NULL
	28	2021-10-01	2021-11-01	2021-11-25	30	6	mauris sit amet lorem se
	3	2021-10-15	2021-10-21	2021-11-18	18	12	NULL
	10	2021-10-17	2021-10-28	2021-11-25	25	20	NULL
	7	2021-10-10	2021-11-01	2021-12-17	22	23	NULL
	6	2021-10-15	2021-10-21	2021-12-17	21	24	NULL
	4	2021-10-10	2021-11-01	2021-11-18	19	27	NULL
	11	2021-10-20	2021-11-01	2021-11-16	26	28	NULL
	9	2021-10-10	2021-11-01	2021-11-18	24	36	NULL
	26	2021-10-20	2021-10-28	2021-11-16	28	40	vel pede blandit congue.
	2	2021-10-20	2021-10-21	2021-11-18	17	41	NULL
	8	2021-10-15	2021-11-01	2021-11-18	23	43	NULL

13) Усі вільні номери в конкретному готелі

```
SELECT RoomId, RoomNumber, HotelName, Floor FROM room
JOIN hotelbranch ON HotelBranchId = HotelBranch
WHERE HotelName = 'Hercules Hotel' AND RoomId NOT IN
(SELECT Room FROM booking WHERE DepartureDate>='2021-11-01' AND DateOfArrival<='2021-11-01');
```

	RoomId	RoomNumber	HotelName	Floor
►	17	1	Hercules Hotel	1
	18	2	Hercules Hotel	1
	19	3	Hercules Hotel	1
	21	5	Hercules Hotel	1
	22	6	Hercules Hotel	1
	25	9	Hercules Hotel	2
	26	10	Hercules Hotel	2
	29	13	Hercules Hotel	2
	30	14	Hercules Hotel	2
	31	15	Hercules Hotel	2
	32	16	Hercules Hotel	2
	33	17	Hercules Hotel	3

14) Усі вільні кімнати конкретного типу в цьому готелі

```
SELECT RoomId, RoomNumber, Floor, RoomTypeName, HotelName FROM room
JOIN hotelbranch ON HotelBranchId = HotelBranch
JOIN roomtype ON RoomTypeId = RoomType
WHERE HotelName = 'Hercules Hotel'
AND RoomTypeName = 'a'
AND RoomId NOT IN
(SELECT Room FROM booking WHERE DepartureDate>='2021-11-01' AND DateOfArrival<='2021-11-01');
```

	RoomId	RoomNumber	Floor	RoomTypeName	HotelName
►	25	9	2	a	Hercules Hotel

15) Усі вільні кімнати у готелях конкретного міста

```
SELECT RoomId, RoomNumber, Floor, RoomTypeName, HotelName, City FROM room
JOIN hotelbranch ON HotelBranchId = HotelBranch
JOIN roomtype ON RoomTypeId = RoomType
WHERE City = 'Odessa'
AND RoomId NOT IN
(SELECT Room FROM booking WHERE DepartureDate>='2021-11-01' AND DateOfArrival<='2021-11-01');
```

	RoomId	RoomNumber	Floor	RoomTypeName	HotelName	City
►	17	1	1	eros	Hercules Hotel	Odessa
	18	2	1	metus	Hercules Hotel	Odessa
	19	3	1	purus	Hercules Hotel	Odessa
	21	5	1	purus	Hercules Hotel	Odessa
	22	6	1	consequat	Hercules Hotel	Odessa
	25	9	2	a	Hercules Hotel	Odessa
	26	10	2	metus	Hercules Hotel	Odessa
	29	13	2	lorem	Hercules Hotel	Odessa
	30	14	2	eros	Hercules Hotel	Odessa
	31	15	2	lorem	Hercules Hotel	Odessa
	32	16	2	odio	Hercules Hotel	Odessa
	33	17	3	metus	Hercules Hotel	Odessa

16) Всі додаткові послуги, які надали людині

```
SELECT FullName, NameOfService, ProvisionDate FROM provisionofservices
JOIN servicesbill ON ServicesBillId = ServicesBill
JOIN roomer ON roomer.RoomerId = servicesbill.RoomerId
JOIN typeofservice ON TypeOfServiceId = TypeOfService
WHERE FullName = 'Lara Fletcher'
```

	FullName	NameOfService	ProvisionDate
►	Lara Fletcher	sem	2021-10-28
	Lara Fletcher	ultrices,	2021-11-09

17) Яким клієнтам була надана ця послуга

```
SELECT FullName, NameOfService, ProvisionDate FROM provisionofservices
JOIN servicesbill ON ServicesBillId = ServicesBill
JOIN roomer ON roomer.RoomerId = servicesbill.RoomerId
JOIN typeofservice ON TypeOfServiceId = TypeOfService
WHERE NameOfService = 'sem';
```

	FullName	NameOfService	ProvisionDate
►	Jeremy Matthews	sem	2021-11-01
	Lara Fletcher	sem	2021-10-28
	Tatum Hanson	sem	2021-11-15
	Jeremy Matthews	sem	2021-11-01
	Lara Fletcher	sem	2021-10-28
	Tatum Hanson	sem	2021-11-15
	Jeremy Matthews	sem	2021-11-01
	Lara Fletcher	sem	2021-10-28
	Tatum Hanson	sem	2021-11-15

18) Яка кімната заброньована на цю людину

```
SELECT FullName, PhoneNumber, DateOfArrival, DepartureDate, RoomNumber, RoomType, Floor FROM booking
JOIN roomer ON RoomerId = Roomer
JOIN room ON RoomId = Room
WHERE FullName = 'Jeremy Matthews' AND DateOfArrival = '2021-11-09'
```

	FullName	PhoneNumber	DateOfArrival	DepartureDate	RoomNumber	RoomType	Floor
►	Jeremy Matthews	(241) 573-1785	2021-11-09	2021-11-25	4	10	1

19) Скільки має заплатити конкретна людина за проживання

```
SELECT Total, FullName FROM acomodationbill
JOIN booking ON Booking = BookingId
JOIN roomer ON RoomerId = Roomer
WHERE FullName = 'Jeremy Matthews'
```

	Total	FullName
►	27840	Jeremy Matthews

20) Скільки має заплатити конкретна людина за додаткові послуги

```
SELECT Total, FullName FROM servicesbill
JOIN roomer ON roomer.RoomerId = servicesbill.RoomerId
WHERE FullName = 'Jeremy Matthews'
```

	Total	FullName
►	852	Jeremy Matthews

Висновок

В ході виконання роботи було спроектовано базу даних для готелю. Сюди входить: уся необхідна інформація про кімнати, інформація про гостя та доступний йому сервіс, бронювання а також саме проживання й відповідні ціни. Також для проектування було досліджено ринок та виділено схожі застосування.

На початку роботи було сформовано завдання та досліджено ринок. Маючи достатньо інформації про схожі роботи, було сформовано вимоги до бази. Фактично, вони дублюють загальні вимоги до проектування реляційних БД.

Після формування вимог було проведено інфологічне моделювання. Це, в першу чергу, аналіз предметної області, формування основних особливостей бази, пошук та формування основних сутностей з їх атрибутами, зв'язками та, звісно сама інфологічна модель, яка у цій роботі представлена.

Після інфологічного моделювання потрібно обрати базу даних та визначитися з вимогами до операційного середовища. Після інфологічного моделювання та вибору СУБД починається етап даталогічного моделювання – перехід від інфологічної моделі до логічної, яка підтримується вибраною СУБД.

До даталогічного моделювання входить формалізація зв'язків, приведення до нормальних форм (в цьому випадку до третьої НФ), обмеження цілісності та саме створення даталогічної моделі бази.

Наступним етапом є фізичне моделювання, тобто визначення типів даних та скорочення і відпрацювання всіх скриптів. Було створено скрипти для створення таблиць, тригерів, послідовностей, також були введені обмеження цілісності. В кінці цього етапу до бази були занесені тестові дані та були створені основні скрипти та процедури для роботи з базою.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Codd, E. F. (1970). *"A Relational Model of Data for Large Shared Data Banks"*. Communications of the ACM. **13** (6): 377 – 387. doi: 10.1145/362384.362685.
2. Ambler, Scott. "Relational Databases 101: Looking at the Whole Picture", Ambysoft Inc., 2002p.
3. Фаулер, Мартин. Архитектура корпоративных программных приложений.: Пер. с англ. — М.: Издательский дом «Вильямс», 2007. — 544с.
4. Максим Никитин, “Закончилась ли эпоха реляционных СУБД?”, 2010 г. [Электронный ресурс] – Режим доступа: <https://www.cnews.ru/reviews/free/marketBD/articles/articles2.shtml>
5. DB-Engines Ranking [Электронный ресурс] – Режим доступа: <https://db-engines.com/en/ranking>
6. Пов'язування таблиць баз даних – Інформатика [Електронний ресурс] – Режим доступу: https://kafinfo.org.ua/files/Informatyka_10_11/Glava_9_47.pdf
7. Редько М.М., Ярмуш О.В., Редько Н.С. Інформатика та комп'ютерна техніка. Навчально-методичний посібник, 2002р.
8. Lynn Beighley, Head First SQL, O'Reilly Media, Inc., 2007p.

Додаток А

Скрипти додавання інформації в таблиці

```
INSERT INTO hotelbranch(HotelName, City, Address,  
Administrator) VALUE  
('Zeus Hotel', 'Kyiv', 'Butomy, bld. 9/A, appt. 12', 'Dulce  
Mayeda'),  
('Hercules Hotel', 'Odessa', 'Dobrovolskogo Pr., bld. 80, appt.  
24', 'Nickie Cordi'),  
('Padlockurity Hotel', 'Odessa', 'Sumi / Lushpi, Vul., bld. 57,  
appt. 52', 'Sashenka Andrel');
```

```
select * from hotelbranch;
```

```
INSERT INTO roomtype(RoomTypeName, NumberOfHumans)  
VALUES  
("semper",1),  
("a",1),  
("aliquam",2),  
("metus",2),  
("Fusce",3),  
("purus",3),  
("lorem",4),  
("eros",5),  
("odio",2),  
("consequat",2);
```

```
SELECT * FROM roomtype;
```

```
INSERT INTO season(SeasonName) value  
('Summer'),
```

```
('Spring'),  
('Winter'),  
('Autumn');
```

```
SELECT * From season;
```

```
INSERT INTO price (RoomType,HotelBranch,Season,Price)  
VALUES
```

```
(1, 1, 1, 801),  
(1, 1, 2, 538),  
(1, 1, 3, 1379),  
(1, 1, 4, 1024),  
(1, 2, 1, 642),  
(1, 2, 2, 1749),  
(1, 2, 3, 1206),  
(1, 2, 4, 1661),  
(1, 3, 1, 1003),  
(1, 3, 2, 607),  
(1, 3, 3, 1007),  
(1, 3, 4, 732),  
(2, 1, 1, 1599),  
(2, 1, 2, 1045),  
(2, 1, 3, 1004),  
(2, 1, 4, 1965),  
(2, 2, 1, 1386),  
(2, 2, 2, 1213),  
(2, 2, 3, 970),  
(2, 2, 4, 565),  
(2, 3, 1, 793),  
(2, 3, 2, 525),  
(2, 3, 3, 839),  
(2, 3, 4, 1338),  
(3, 1, 1, 1331),
```

(3, 1, 2, 1638),
(3, 1, 3, 1848),
(3, 1, 4, 1190),
(3, 2, 1, 1571),
(3, 2, 2, 947),
(3, 2, 3, 1821),
(3, 2, 4, 653),
(3, 3, 1, 747),
(3, 3, 2, 1241),
(3, 3, 3, 1228),
(3, 3, 4, 813),
(4, 1, 1, 1519),
(4, 1, 2, 870),
(4, 1, 3, 1135),
(4, 1, 4, 1227),
(4, 2, 1, 509),
(4, 2, 2, 1804),
(4, 2, 3, 1146),
(4, 2, 4, 1596),
(4, 3, 1, 891),
(4, 3, 2, 987),
(4, 3, 3, 868),
(4, 3, 4, 1023),
(5, 1, 1, 662),
(5, 1, 2, 722),
(5, 1, 3, 1170),
(5, 1, 4, 838),
(5, 2, 1, 1223),
(5, 2, 2, 1367),
(5, 2, 3, 1328),
(5, 2, 4, 1270),
(5, 3, 1, 776),
(5, 3, 2, 1348),
(5, 3, 3, 1232),

(5, 3, 4, 786),
(6, 1, 1, 1983),
(6, 1, 2, 1900),
(6, 1, 3, 805),
(6, 1, 4, 1605),
(6, 2, 1, 1390),
(6, 2, 2, 1835),
(6, 2, 3, 1009),
(6, 2, 4, 1705),
(6, 3, 1, 1743),
(6, 3, 2, 1834),
(6, 3, 3, 1963),
(6, 3, 4, 505),
(7, 1, 1, 552),
(7, 1, 2, 1227),
(7, 1, 3, 764),
(7, 1, 4, 1710),
(7, 2, 1, 1197),
(7, 2, 2, 1305),
(7, 2, 3, 1277),
(7, 2, 4, 1125),
(7, 3, 1, 1376),
(7, 3, 2, 1057),
(7, 3, 3, 1512),
(7, 3, 4, 1213),
(8, 1, 1, 933),
(8, 1, 2, 1948),
(8, 1, 3, 1490),
(8, 1, 4, 781),
(8, 2, 1, 1864),
(8, 2, 2, 1809),
(8, 2, 3, 1477),
(8, 2, 4, 1419),
(8, 3, 1, 574),

(8, 3, 2, 1823),
(8, 3, 3, 1307),
(8, 3, 4, 1510),
(9, 1, 1, 760),
(9, 1, 2, 1392),
(9, 1, 3, 785),
(9, 1, 4, 1818),
(9, 2, 1, 1939),
(9, 2, 2, 1740),
(9, 2, 3, 1269),
(9, 2, 4, 1745),
(9, 3, 1, 526),
(9, 3, 2, 1681),
(9, 3, 3, 1094),
(9, 3, 4, 1954),
(10, 1, 1, 1362),
(10, 1, 2, 956),
(10, 1, 3, 830),
(10, 1, 4, 1617),
(10, 2, 1, 1445),
(10, 2, 2, 1388),
(10, 2, 3, 710),
(10, 2, 4, 1168),
(10, 3, 1, 1740),
(10, 3, 2, 1256),
(10, 3, 3, 1658),
(10, 3, 4, 1624);

SELECT * FROM price;

INSERT INTO room(Floor, RoomNumber, RoomType, HotelBranch)
VALUE

(1, 1, 10, 1),
(1, 2, 1, 1),
(1, 3, 1, 1),
(1, 4, 2, 1),
(1, 5, 3, 1),
(1, 6, 1, 1),
(1, 7, 8, 1),
(1, 8, 8, 1),
(2, 9, 4, 1),
(2, 10, 4, 1),
(2, 11, 7, 1),
(2, 12, 9, 1),
(2, 13, 4, 1),
(2, 14, 2, 1),
(2, 15, 9, 1),
(2, 16, 10, 1),
(1, 1, 8, 2),
(1, 2, 4, 2),
(1, 3, 6, 2),
(1, 4, 6, 2),
(1, 5, 6, 2),
(1, 6, 10, 2),
(1, 7, 5, 2),
(1, 8, 6, 2),
(2, 9, 2, 2),
(2, 10, 4, 2),
(2, 11, 4, 2),
(2, 12, 6, 2),
(2, 13, 7, 2),
(2, 14, 8, 2),
(2, 15, 7, 2),
(2, 16, 9, 2),
(3, 17, 4, 2),
(3, 18, 8, 2),

(3, 19, 9, 2),
(3, 20, 5, 2),
(3, 21, 8, 2),
(3, 22, 4, 2),
(3, 23, 9, 2),
(3, 24, 4, 2),
(1, 1, 8, 3),
(1, 2, 8, 3),
(1, 3, 10, 3),
(1, 4, 10, 3),
(1, 5, 10, 3),
(1, 6, 3, 3),
(1, 7, 9, 3),
(1, 8, 4, 3);

SELECT * FROM room;

INSERT INTO roomer (FullName, PhoneNumber, PassportNumber, IPN)

VALUES

("Reuben Page","(550) 224-9718",2143647857,2061271108),
("Quinn Flores","1-558-660-1136",1956564650,1686531822),
("Fritz Frye","(202) 324-2884",1786874618,1157981428),
("Amery Hill","1-457-388-1647",2074183585,1993694368),
("Ocean Cantrell","(778) 405-2343",1482208853,1662221715),
("Caryn Wade","(403) 756-3485",2101892034,1382656802),
("Lara Fletcher","1-775-278-3652",1514270733,1429423678),
("Reece Allen","(272) 562-9176",1756807699,1470432343),
("Rashad Prince","(584) 874-4831",1737064066,1722397619),
("Amal Snider","(171) 647-3145",1620857909,1867118820),

("Coby Baldwin", "(985) 732-6551", 1022307140, 1504339752),
 ("Keefe Andrews", "(342) 983-4862", 1427182074, 1264713080),
 ("Dorothy Rutledge", "(843) 351-2230", 1713962563, 1424595661),
 ("Kibo Patel", "(738) 511-5293", 1745377190, 1719091344),
 ("Eleanor Sherman", "1-398-957-8321", 1211116516, 1366376019),
 ("Wang Reyes", "1-366-366-4729", 1568934496, 2110463508),
 ("Ria Nolan", "(392) 583-7759", 1220969456, 1223560664),
 ("Geoffrey Sherman", "(711) 157-8410", 1386142404, 1103914240),
 ("Raya Strong", "1-574-655-7486", 1408928675, 1036961275),
 ("Jeremy Matthews", "(241) 573-1785", 2032900193, 1998145641),
 ("Odysseus Carney", "1-823-772-3328", 1224679130, 1326444766),
 ("Brock Levine", "1-978-824-4734", 1355720435, 1196057368),
 ("Justin Vaughan", "1-618-261-6841", 1994182081, 1856201032),
 ("Oscar Shaw", "1-684-584-3217", 1745344823, 1945914630),
 ("Marny Frank", "1-735-531-4894", 1342754854, 1222860034),
 ("Dara Vang", "1-986-768-5065", 2074272655, 1731910044),
 ("Tatum Hanson", "1-150-425-8286", 1010695549, 1142053128),
 ("Lee Irwin", "(247) 741-4215", 1935144380, 1711264609),
 ("Illiana Burton", "1-536-802-6752", 2004742369, 2023647646),
 ("Drew James", "(266) 808-7802", 1111307467, 1390749909);

SELECT * FROM roomer;

INSERT INTO booking(DayOfBooking, DateOfArrival,
 DepartureDate, Roomer, Room)
 VALUE
 ("2021-10-01", "2021-10-21", "2021-11-18", 16, 3),
 ("2021-10-20", "2021-10-21", "2021-11-18", 17, 41),
 ("2021-10-15", "2021-10-21", "2021-11-18", 18, 12),
 ("2021-10-10", "2021-11-01", "2021-11-18", 19, 27),
 ("2021-10-10", "2021-11-09", "2021-11-25", 20, 44),

("2021-10-15", "2021-10-21", "2021-12-17", 21, 24),
 ("2021-10-10", "2021-11-01", "2021-12-17", 22, 23),
 ("2021-10-15", "2021-11-01", "2021-11-18", 23, 43),
 ("2021-10-10", "2021-11-01", "2021-11-18", 24, 36),
 ("2021-10-17", "2021-10-28", "2021-11-25", 25, 20),
 ("2021-10-20", "2021-11-01", "2021-11-16", 26, 28),
 ("2021-10-20", "2021-11-15", "2021-11-23", 27, 38),
 ("2021-11-15", "2021-12-21", "2022-01-07", 3, 47),
 ("2021-12-01", "2021-12-21", "2022-01-11", 4, 11),
 ("2021-11-27", "2021-12-30", "2022-01-07", 5, 10),
 ("2021-11-27", "2021-12-22", "2022-01-11", 6, 20),
 ("2021-11-27", "2021-12-18", "2022-01-11", 7, 5),
 ("2021-11-27", "2021-12-18", "2022-01-11", 8, 25),
 ("2021-12-20", "2021-12-30", "2022-01-11", 9, 11),
 ("2021-11-27", "2021-12-22", "2022-01-11", 10, 19),
 ("2021-12-20", "2021-12-30", "2022-01-11", 11, 41),
 ("2021-11-15", "2021-12-18", "2022-01-07", 12, 9),
 ("2021-11-27", "2021-12-30", "2022-01-02", 13, 11),
 ("2021-11-15", "2021-12-30", "2022-01-02", 14, 18),
 ("2021-11-27", "2021-12-30", "2022-01-05", 15, 7);

INSERT INTO booking(DayOfBooking, DateOfArrival,
 DepartureDate, Roomer, Room, Note)
 VALUE

("2021-10-20", "2021-10-28", "2021-11-16", 28, 40, "vel pede blandit
 congue. In scelerisque"),
 ("2021-10-17", "2021-11-09", "2021-11-18", 29, 30, "quis massa.
 Mauris vestibulum, neque sed"),
 ("2021-10-01", "2021-11-01", "2021-11-25", 30, 6, "mauris sit amet
 lorem semper auctor. Mauris vel turpis. Aliquam adipiscing lobortis
 risus."),
 ("2021-12-20", "2021-12-18", "2022-01-07", 1, 5, "Duis risus odio,
 auctor vitae, aliquet nec, imperdiet nec, leo. Morbi neque"),

```
("2021-12-01", "2021-12-21", "2022-01-02", 2, 12, "Nulla dignissim.  
Maecenas ornare egestas ligula. Nullam");
```

```
SELECT * FROM booking
```

```
INSERT INTO typeofservice (NameOfService,Price)  
VALUES
```

```
("sem",284),  
("Nullam",346),  
("massa.",474),  
("sodales",283),  
("vel",413),  
("elit",369),  
("eu",483),  
("non",288),  
("posuere,",345),  
("In",228),  
("arcu",460),  
("ultrices,",375),  
("bibendum",416);
```

```
SELECT * FROM typeofservice;
```

```
INSERT INTO provisionofservises(TypeOfService, ServicesBill,  
ProvisionDate)
```

```
VALUE (1, 20, "2021-11-01"),  
(6, 29, "2021-11-15"),  
(13, 30, "2021-10-28"),  
(2, 16, "2021-11-15"),  
(4, 11, "2021-11-15"),
```

```
(2, 24, "2021-10-28"),  
(9, 4, "2021-11-15"),  
(1, 7, "2021-10-28"),  
(1, 27, "2021-11-15"),  
(3, 17, "2021-10-21"),  
(8, 26, "2021-11-15"),  
(8, 5, "2021-11-09"),  
(12, 7, "2021-11-09"),  
(4, 30, "2021-11-15"),  
(2, 12, "2021-10-28");
```

```
SELECT * FROM provisionofservises;
```