

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 2 з дисципліни
«Бази даних»

«Створення бази даних»

Варіант 15

Виконала студентка ІП-12 Кушнір Ганна Вікторівна

Перевірила Марченко Олена Іванівна

Київ 2022

Лабораторна робота №2

Мета:

- Створення бази даних шляхом визначення схеми БД та заповнення її тестовими даними
- Навчитися проектувати бази даних, вводити і редагувати структуру таблиць та дані в таблицях
- Вивчити DDL-команди SQL для роботи з таблицями (створення, модифікації та видалення таблиць)
- Вивчити використовувані в SQL засоби для підтримки цілісності даних та їх практичне застосування

Теоретичні основи

Викладені в лекційному матеріалі

1. Постановка задачі лабораторної роботи № 2

При виконанні лабораторної роботи необхідно виконати наступні дії:

- 1) Створити схему БД згідно з розробленою в роботі №1 ER-моделлю
- 2) Розробити SQL-скрипти для:
 - a. створення таблиць в БД засобами мови SQL (для 5-6 таблиць).
Передбачити необхідність наявності обмежень для підтримки цілісності та коректності даних;
 - b. зміни структури таблиць та даних в таблицях засобами мови SQL (до 20 різних за суттю запитів для 5-6 таблиць);
 - c. видалення окремих елементів або самих таблиць засобами мови SQL (до 10 різних за суттю команд);
 - d. встановлення зв'язків між таблицями засобами мови SQL.
- 3) Згенерувати схему даних засобами СУБД
- 4) Імпортувати дані в створену БД з використанням засобів СУБД

2. Завдання варіанту

Програмне забезпечення готелю.

Основна задача програмного забезпечення, котре проєктується - є відстеження фінансової сторони роботи готелю. Діяльність організована наступним чином: готель надає номери клієнтам на певний термін. Кожен номер характеризується місткістю, комфортністю (люкс, напівлюкс, звичайний, тощо) і ціною. Клієнтами є різні особи, по яким збирається певна інформація (прізвище, ім'я, по-батькові, номер та серія паспорту або іншого документу, що посвідчує особу, додаткова інформація). Здача номера клієнтові проводиться за наявності вільних місць в номерах, за параметрами, котрі вказав клієнт. При поселенні фіксується дата поселення, при виїзді – дата звільнення. номеру. Необхідно не лише зберігати інформацію за фактом здачі номера клієнтові, але і здійснювати бронювання номерів. Крім того, для постійних клієнтів, а також для певних категорій клієнтів передбачена система знижок. Знижки можуть підсумовуватися.

3. Виконання

3.1. Створення схеми бази даних

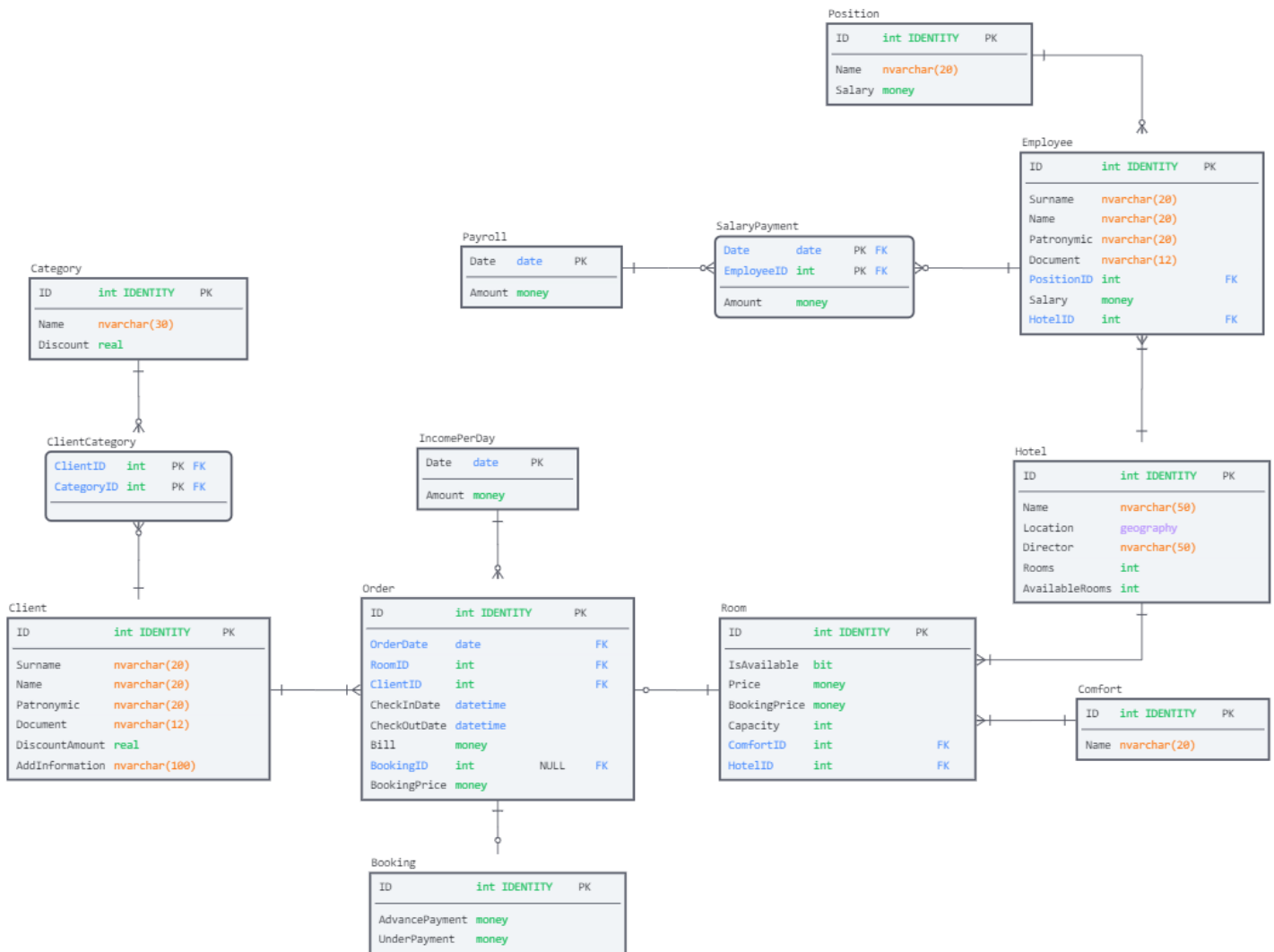


Рисунок 3.1 – Схема бази даних

3.2. Розробка SQL-скриптів

3.2.1. SQL-скрипти для створення таблиць в БД

```
-- =====  
-- Створення таблиць.  
-- =====
```

USE MyDB;

```
CREATE TABLE Clients (  
    ID INT IDENTITY,  
    Surname NVARCHAR(20) NOT NULL,  
    Name NVARCHAR(20) NOT NULL,  
    Patronymic NVARCHAR(20),  
    Document NVARCHAR(12),
```

```

DiscountAmount REAL,
CONSTRAINT PK_Clients PRIMARY KEY(ID),
CONSTRAINT CK_Clients_Surname CHECK(Surname != ''),
CONSTRAINT CK_Clients_Name CHECK(Name != ''),
CONSTRAINT CK_Clients_Patronymic CHECK(Patronymic != ''),
CONSTRAINT UQ_Clients_Document UNIQUE (Document)
);

CREATE TABLE Categories (
    ID INT IDENTITY,
    Name NVARCHAR(30) NOT NULL,
    Discount REAL,
    CONSTRAINT PK_Categories PRIMARY KEY(ID),
    CONSTRAINT CK_Categories_Discount CHECK (Discount BETWEEN 0 AND 1)
);

CREATE TABLE ClientCategory (
    ClientID INT NOT NULL,
    CategoryID INT NOT NULL,
);

CREATE TABLE Hotels (
    ID INT IDENTITY,
    Name NVARCHAR(20) NOT NULL,
    Location NVARCHAR(100),
    Director NVARCHAR(50),
    Rooms INT,
    AvailableRooms INT,
    CONSTRAINT PK_Hotels PRIMARY KEY(ID)
);

CREATE TABLE Positions (
    ID INT IDENTITY,
    Name NVARCHAR(20) NOT NULL,
    Salary MONEY NOT NULL,
    CONSTRAINT PK_Positions PRIMARY KEY(ID),
    CONSTRAINT CK_Positions_Name CHECK (Name != '')
);

CREATE TABLE Employees (
    ID INT IDENTITY,
    Surname NVARCHAR(20) NOT NULL,
    Name NVARCHAR(20) NOT NULL,
    Patronymic NVARCHAR(20),
    Document NVARCHAR(12),
    PositionID INT,
    Salary MONEY,
    HotelID INT,
    CONSTRAINT PK_Employees PRIMARY KEY(ID),
    CONSTRAINT CK_Employees_Surname CHECK(Surname != ''),
    CONSTRAINT CK_Employees_Name CHECK(Name != ''),
    CONSTRAINT CK_Employees_Patronymic CHECK(Patronymic != '')
);

```

```
CREATE TABLE SalaryPayment (  
    ID INT IDENTITY,  
    EmployeeID INT NOT NULL,  
    PayrollID INT NOT NULL,  
    CONSTRAINT PK_SalaryPayment PRIMARY KEY(ID)  
);
```

```
CREATE TABLE Payroll (  
    ID INT IDENTITY,  
    Date DATE NOT NULL,  
    Amount MONEY,  
    CONSTRAINT PK_Payroll PRIMARY KEY(ID)  
);
```

```
CREATE TABLE Comfort (  
    ID INT IDENTITY,  
    Name NVARCHAR(20) NOT NULL,  
    CONSTRAINT PK_Comfort PRIMARY KEY(ID)  
);
```

```
CREATE TABLE Rooms (  
    ID INT IDENTITY,  
    IsAvailable BIT NOT NULL,  
    Price MONEY NOT NULL,  
    BookingPrice MONEY,  
    Capacity INT NOT NULL,  
    ComfortID INT NOT NULL,  
    HotelID INT NOT NULL,  
    CONSTRAINT PK_Rooms PRIMARY KEY(ID)  
);
```

```
CREATE TABLE Booking (  
    ID INT IDENTITY,  
    OrderID INT,  
    AdvancePayment MONEY,  
    UnderPayment MONEY,  
    CONSTRAINT PK_Booking PRIMARY KEY(ID)  
);
```

```
CREATE TABLE Orders (  
    ID INT IDENTITY,  
    OrderDate DATE NOT NULL,  
    RoomID INT NOT NULL,  
    ClientID INT NOT NULL,  
    CheckInDate DATETIME NOT NULL,  
    CheckOutDate DATETIME NOT NULL,  
    Bill MONEY NOT NULL,  
    BookingPrice MONEY,  
    CONSTRAINT PK_Orders PRIMARY KEY(ID)  
);
```

```

/* Таблиця, створена для подальшого видалення. */
CREATE TABLE OddTable (
    ID INT,
    OddColumn NVARCHAR(20)
        CONSTRAINT DF_OddTable_OddColumn DEFAULT 7,
    CONSTRAINT PK_OddTable PRIMARY KEY(ID),
    CONSTRAINT UQ_OddTable_OddColumn UNIQUE (OddColumn)
);

CREATE TABLE IncomePerDay (
    Date DATE NOT NULL,
    Amount MONEY
        CONSTRAINT DF_IncomePerDay_Amount DEFAULT 0,
    OddID INT,
    CONSTRAINT PK_IncomePerDay PRIMARY KEY(Date),
    CONSTRAINT FK_IncomePerDay_OddID FOREIGN KEY (OddID) REFERENCES
OddTable (ID)
);

```

3.2.2. SQL-скрипти для зміни структури таблиць

```

-- =====
-- Додавання та редагування полів, обмежень та таблиць.
-- =====

USE MyDB;

/* Додавання первинного ключа таблиці. */
ALTER TABLE ClientCategory
    ADD CONSTRAINT PK_ClientCategory
    PRIMARY KEY(ClientID, CategoryID);

/* Додавання поля до таблиці. */
ALTER TABLE Clients
    ADD AddInformation NVARCHAR(100);

/* Додавання перевірки значення полів. */
ALTER TABLE Clients
    ADD CONSTRAINT CK_Clients_DiscountAmount
    CHECK (DiscountAmount BETWEEN 0 AND 1);

ALTER TABLE Hotels
    ADD CONSTRAINT CK_Hotels_Rooms
    CHECK (Rooms >= 0 AND AvailableRooms >= 0);

/* Редагування типу поля таблиці. */
ALTER TABLE Hotels
    ALTER COLUMN Name NVARCHAR(50) NOT NULL;

/* Редагування атрибутів поля таблиці. */
ALTER TABLE Hotels
    ALTER COLUMN Rooms INT NOT NULL;

```

```

ALTER TABLE Hotels
    ALTER COLUMN AvailableRooms INT NOT NULL;

ALTER TABLE IncomePerDay
    ALTER COLUMN Amount MONEY NOT NULL;

/* Додавання значення за замовчуванням для поля. */
ALTER TABLE Clients
    ADD CONSTRAINT DF_Clients_DiscountAmount
    DEFAULT 0 FOR DiscountAmount;

ALTER TABLE Hotels
    ADD CONSTRAINT DF_Hotels_Rooms
    DEFAULT 0 FOR Rooms;

ALTER TABLE Hotels
    ADD CONSTRAINT DF_Hotels_AvailableRooms
    DEFAULT 0 FOR AvailableRooms;

/* Додавання обмеження на унікальність значення поля. */
ALTER TABLE Employees
    ADD CONSTRAINT UQ_Employees_Document
    UNIQUE (Document);

ALTER TABLE Booking
    ADD CONSTRAINT UQ_Booking_OrderID
    UNIQUE (OrderID);

/* Відключення перевірки значення поля. */
ALTER TABLE Positions
    NOCHECK CONSTRAINT CK_Positions_Name;

```

3.2.3. SQL-скрипти для видалення окремих елементів або самих таблиць

```

-- =====
-- Видалення полів, обмежень та таблиць.
-- =====

USE MyDB;

/* Видалення перевірки значення поля. */
ALTER TABLE Clients
    DROP CONSTRAINT CK_Clients_Patronymic;

ALTER TABLE Employees
    DROP CONSTRAINT CK_Employees_Patronymic;

/* Видалення зовнішнього ключа. */
ALTER TABLE IncomePerDay
    DROP CONSTRAINT FK_IncomePerDay_OddID;

```



```

/* Видалення поля. */
ALTER TABLE IncomePerDay
    DROP COLUMN OddID;

/* Видалення значення поля за замовчуванням. */
ALTER TABLE OddTable
    DROP CONSTRAINT DF_OddTable_OddColumn;

/* Видалення первинного ключа. */
ALTER TABLE OddTable
    DROP CONSTRAINT PK_OddTable;

/* Видалення обмеження на унікальність значення поля. */
ALTER TABLE OddTable
    DROP CONSTRAINT UQ_OddTable_OddColumn;

/* Видалення таблиці. */
DROP TABLE IF EXISTS OddTable;

```

3.2.4. SQL-скрипти для встановлення зв'язків між таблицями

```

-- =====
-- Створення зв'язків між таблицями, додавання зовнішніх ключів.
-- =====

USE MyDB;

ALTER TABLE ClientCategory
    ADD CONSTRAINT FK_ClientCategory_ClientID
    FOREIGN KEY(ClientID)
    REFERENCES Clients (ID)
    ON DELETE CASCADE;

ALTER TABLE ClientCategory
    ADD CONSTRAINT FK_ClientCategory_CategoryID
    FOREIGN KEY(CategoryID)
    REFERENCES Categories (ID)
    ON DELETE CASCADE;

ALTER TABLE Employees
    ADD CONSTRAINT FK_Employees_PositionID
    FOREIGN KEY(PositionID)
    REFERENCES Positions (ID)
    ON DELETE SET NULL;

ALTER TABLE Employees
    ADD CONSTRAINT FK_Employees_HotelID
    FOREIGN KEY(HotelID)
    REFERENCES Hotels (ID)
    ON DELETE CASCADE;

```

```
ALTER TABLE Rooms
  ADD CONSTRAINT FK_Rooms_ComfortID
  FOREIGN KEY(ComfortID)
  REFERENCES Comfort (ID)
  ON DELETE CASCADE;
```

```
ALTER TABLE Rooms
  ADD CONSTRAINT FK_Rooms_HotelID
  FOREIGN KEY(HotelID)
  REFERENCES Hotels (ID)
  ON DELETE CASCADE;
```

```
ALTER TABLE Orders
  ADD CONSTRAINT FK_Orders_OrderDate
  FOREIGN KEY(OrderDate)
  REFERENCES IncomePerDay (Date)
  ON DELETE NO ACTION;
```

```
ALTER TABLE Orders
  ADD CONSTRAINT FK_Orders_RoomID
  FOREIGN KEY(RoomID)
  REFERENCES Rooms (ID)
  ON DELETE CASCADE;
```

```
ALTER TABLE Orders
  ADD CONSTRAINT FK_Orders_ClientID
  FOREIGN KEY(ClientID)
  REFERENCES Clients (ID)
  ON DELETE CASCADE;
```

```
ALTER TABLE Booking
  ADD CONSTRAINT FK_Booking_OrderID
  FOREIGN KEY(OrderID)
  REFERENCES Orders (ID)
  ON DELETE CASCADE;
```

```
ALTER TABLE SalaryPayment
  ADD CONSTRAINT FK_SalaryPayment_EmployeeID
  FOREIGN KEY(EmployeeID)
  REFERENCES Employees (ID)
  ON DELETE CASCADE;
```

```
ALTER TABLE SalaryPayment
  ADD CONSTRAINT FK_SalaryPayment_PayrollID
  FOREIGN KEY(PayrollID)
  REFERENCES Payroll (ID)
  ON DELETE CASCADE;
```

3.2.5. SQL-скрипти для додавання даних до таблиць

```
-- =====
-- Додавання даних до таблиць.
-- =====

USE MyDB;

INSERT INTO Clients (
    Surname, Name, Patronymic, Document, AddInformation
)
VALUES
    ('Tate', 'Tabita', null, '001956782426', null),
    ('Tate', 'Bob', null, '002046183263', 'Works in a diner'),
    ('Blossom', 'Sheril', null, '003791936729', 'Has her own company'),
    ('Diesel', 'Vin', null, '003527193001', 'An actor'),
    ('Ukrainka', 'Lesya', null, 'CX271835', 'A writer'),
    ('Shevchenko', 'Taras', 'Shryhorovych', '57913729', 'The most famous
ukrainian writer'),
    ('Noname', 'Sombody', null, '000000000', null),
    ('Stetsenko', 'Alisa', 'Volodymyrivna', '001729372', null),
    ('Pogoreltseva', 'Tetiana', 'Mykhaylivna', '002738171', null),
    ('Reinhart', 'Lily', null, '081936273', 'An actress'),
    ('Zaltsman', 'Lizzy', null, '001983936', null),
    ('Cuper', 'Polly', null, '009826381', null),
    ('Salvatore', 'Stephan', null, '010101010', 'Knows everything about
everyone'),
    ('Salvatore', 'Damon', null, '111111111', null),
    ('Shevchenko', 'Kateryna', 'Shryhorivna', 'EE829128', null),
    ('Shostak', 'Artem', 'Serhiyovych', '001729362', 'Student')
;

INSERT INTO Categories (
    Name, Discount
)
VALUES
    ('Pensioner', 0.15),
    ('Disabled', 0.2),
    ('Single mother', 0.1),
    ('Single father', 0.1),
    ('Large family', 0.07),
    ('Unemployed', 0.05),
    ('Orphan', 0.15),
    ('Student', 0.1),
    ('Regular client', 0.1)
;

INSERT INTO ClientCategory (
    ClientID, CategoryID
)
VALUES
    (3, 7),
    (6, 7),
```

```
(2, 2),  
(8, 8),  
(9, 3),  
(9, 5),  
(16, 8)  
;
```

```
INSERT INTO Hotels (  
    Name, Director  
)  
VALUES  
( 'Kyiv Hotel', 'Ribalka Mykola Mykolayovych'),  
( 'Odessa Hotel', 'Rishko Oleksiy Victorovych'),  
( 'Kharkiv Hotel', 'Kozak Serhiy Volodymyrovych')  
;
```

```
INSERT INTO Comfort (  
    Name  
)  
VALUES  
( 'Presidential'),  
( 'Suite'),  
( 'Junior suite'),  
( 'Standard'),  
( 'Economy')  
;
```

```
INSERT INTO Rooms (  
    isAvailable, Price, BookingPrice, Capacity, ComfortID, HotelID  
)  
VALUES  
(1, 3000, 1500, 3, 1, 1),  
(1, 2000, 1000, 2, 1, 1),  
(1, 2000, 700, 3, 2, 1),  
(1, 1500, 500, 2, 2, 1),  
(1, 1000, 250, 3, 3, 1),  
(1, 700, 175, 2, 3, 1),  
(1, 500, 100, 3, 4, 1),  
(1, 400, 80, 2, 4, 1),  
(1, 300, 50, 2, 5, 1),  
(1, 200, 35, 1, 5, 1),  
(1, 1000, 300, 2, 1, 2),  
(1, 700, 200, 2, 2, 2),  
(1, 500, 100, 2, 4, 2),  
(1, 250, 50, 2, 5, 2),  
(1, 1000, 350, 2, 2, 3),  
(1, 500, 100, 3, 4, 3)  
;
```

```
INSERT INTO Positions (  
    Name, Salary  
)  
VALUES
```

```

('Administrator', 15000),
('Manager', 12000),
('Maid', 5000),
('Cook', 7000),
('Waiter', 5000),
('Security guard', 7000),
('Sommelier', 6000)
;

```

```

INSERT INTO Employees (
    Surname, Name, Patronymic, Document, PositionID, HotelID
)
VALUES
('Grande', 'Ariana', null, '001927381', 1, 1),
('Rishko', 'Tetiana', 'Mykolaivna', '001825728', 2, 1),
('Karpiuk', 'Oles', null, '001728192', 4, 1),
('Larin', 'Mykola', 'Valeriyovych', '001927628', 6, 1),
('Kropiv', 'Victor', 'Victorovych', '002818283', 7, 1),
('Rurik', 'Stanislav', null, '002828162', 1, 2),
('Rahiv', 'Karina', 'Andriivna', '001562738', 1, 3),
('Shkred', 'Iryna', 'Pavlovna', 'CB728193', 6, 2),
('West', 'Hanna', null, '00172829182', 2, 2),
('Mykulynych', 'Ivan', 'Ivanovych', 'CB627183', 3, 2),
('Klymenko', 'Nadiya', 'Stepanivna', 'EB637183', 4, 2),
('Freyman', 'Nick', null, '01283947281', 6, 2),
('Leibnits', 'Martin', null, '26173829193', 2, 3),
('Wirna', 'Ivanna', 'Artemivna', '001828361', 4, 3),
('Stepanenko', 'Stephania', 'Stepanivna', 'EE617293', 7, 3)
;

```

```

INSERT INTO Payroll (
    Date
)
VALUES
('2022-08-28'),
('2022-08-30'),
('2022-08-31'),
('2022-09-28'),
('2022-09-30'),
('2022-10-28'),
('2022-10-30'),
('2022-10-31')
;

```

```

INSERT INTO SalaryPayment (
    PayrollID, EmployeeID
)
VALUES
(1, 6),
(1, 8),
(2, 1),
(2, 2),
(2, 3),

```

```
(2, 4),
(2, 5),
(3, 7),
(4, 6),
(4, 8),
(5, 1),
(5, 2),
(5, 3),
(5, 4),
(5, 5),
(5, 7),
(6, 6),
(6, 8),
(7, 1),
(7, 2),
(7, 3),
(7, 4),
(7, 5),
(8, 7)
;
```

```
INSERT INTO IncomePerDay (
    Date
)
VALUES
    ('2022-09-01'),
    ('2022-09-02'),
    ...
    ('2022-09-30'),
    ('2022-10-01'),
    ...
    ('2022-10-31'),
    ('2022-11-01'),
    ...
    ('2022-11-30')
;
```

```
INSERT INTO Orders (
    OrderDate, RoomID, ClientID, CheckInDate, CheckOutDate
)
VALUES
    ('2022-09-05', 2, 1, '2022-09-05', '2022-09-07'),
    ('2022-09-07', 5, 3, '2022-09-07', '2022-09-15'),
    ('2022-09-12', 7, 8, '2022-09-12', '2022-09-16'),
    ('2022-11-12', 7, 2, '2022-11-12', '2022-11-16'),
    ('2022-11-13', 5, 3, '2022-11-13', '2022-11-18'),
    ('2022-11-15', 2, 6, '2022-11-15', '2022-11-16'),
    ('2022-11-16', 3, 1, '2022-11-17', '2022-11-19'),
    ('2022-11-16', 4, 2, '2022-11-17', '2022-11-19'),
    ('2022-11-14', 6, 4, '2022-11-16', '2022-11-18'),
    ('2022-11-11', 8, 5, '2022-11-15', '2022-11-19'),
    ('2022-11-13', 9, 9, '2022-11-13', '2022-11-15'),
```

```
( '2022-11-14', 10, 7, '2022-11-14', '2022-11-15'),
( '2022-11-17', 12, 11, '2022-11-19', '2022-11-25')
;
```

```
INSERT INTO Booking (
    OrderID, AdvancePayment, UnderPayment
)
SELECT
    Orders.ID, Orders.BookingPrice, Orders.Bill - Orders.BookingPrice
FROM Orders
WHERE Orders.BookingPrice IS NOT NULL
;
```

3.2.6. SQL-скрипти для редагування даних у таблицях

```
-- =====
-- Редагування даних у таблицях.
-- =====
```

```
USE MyDB;
```

```
UPDATE Categories
SET Discount = 0.1
WHERE Discount < 0.1
```

```
UPDATE Employees
SET Patronymic = 'Pavlivna'
WHERE Patronymic = 'Pavlovna'
```

```
UPDATE Employees
SET HotelID = 2
WHERE Surname = 'Rishko'
```

```
UPDATE Positions
SET Salary = 5500
WHERE Name = 'Sommelier'
```

```
UPDATE Employees
SET Salary = Positions.Salary
FROM Positions
WHERE Employees.PositionID = Positions.ID;
```

```
UPDATE Orders
SET Bill = (SELECT Price FROM Rooms WHERE Rooms.ID = Orders.RoomID) *
DATEDIFF(day, CheckInDate, CheckOutDate);
```

```
UPDATE Orders
SET Bill = Bill - Bill * Clients.DiscountAmount
FROM Clients
WHERE Clients.ID = Orders.ClientID;
```

```
UPDATE Orders
SET BookingPrice = Rooms.BookingPrice * DATEDIFF(day, CheckInDate,
CheckOutDate)
FROM Rooms
WHERE Rooms.ID = Orders.RoomID
AND DATEDIFF(day, Orders.OrderDate, Orders.CheckInDate) > 0;
```

```
UPDATE Rooms
SET IsAvailable = 0
FROM Orders
WHERE Orders.RoomID = Rooms.ID
AND CURRENT_TIMESTAMP BETWEEN Orders.CheckInDate
AND Orders.CheckOutDate;
```

```
UPDATE IncomePerDay
SET Amount = Orders.Bill
FROM Orders
WHERE Orders.OrderDate = IncomePerDay.Date;
```

```
UPDATE Clients
SET DiscountAmount = (SELECT SUM(Categories.Discount)
FROM ClientCategory, Categories
WHERE ClientCategory.ClientID = Clients.ID
AND Categories.ID = ClientCategory.CategoryID)
WHERE EXISTS (SELECT *
FROM ClientCategory, Categories
WHERE ClientCategory.ClientID = Clients.ID
AND Categories.ID = ClientCategory.CategoryID);
```

```
UPDATE Clients
SET DiscountAmount = 0
WHERE DiscountAmount IS NULL;
```

```
UPDATE Payroll
SET Amount = (SELECT SUM(Employees.Salary)
FROM SalaryPayment, Employees
WHERE SalaryPayment.PayrollID = Payroll.ID
AND SalaryPayment.EmployeeID = Employees.ID);
```


3.3. Генерування схеми даних засобами СУБД

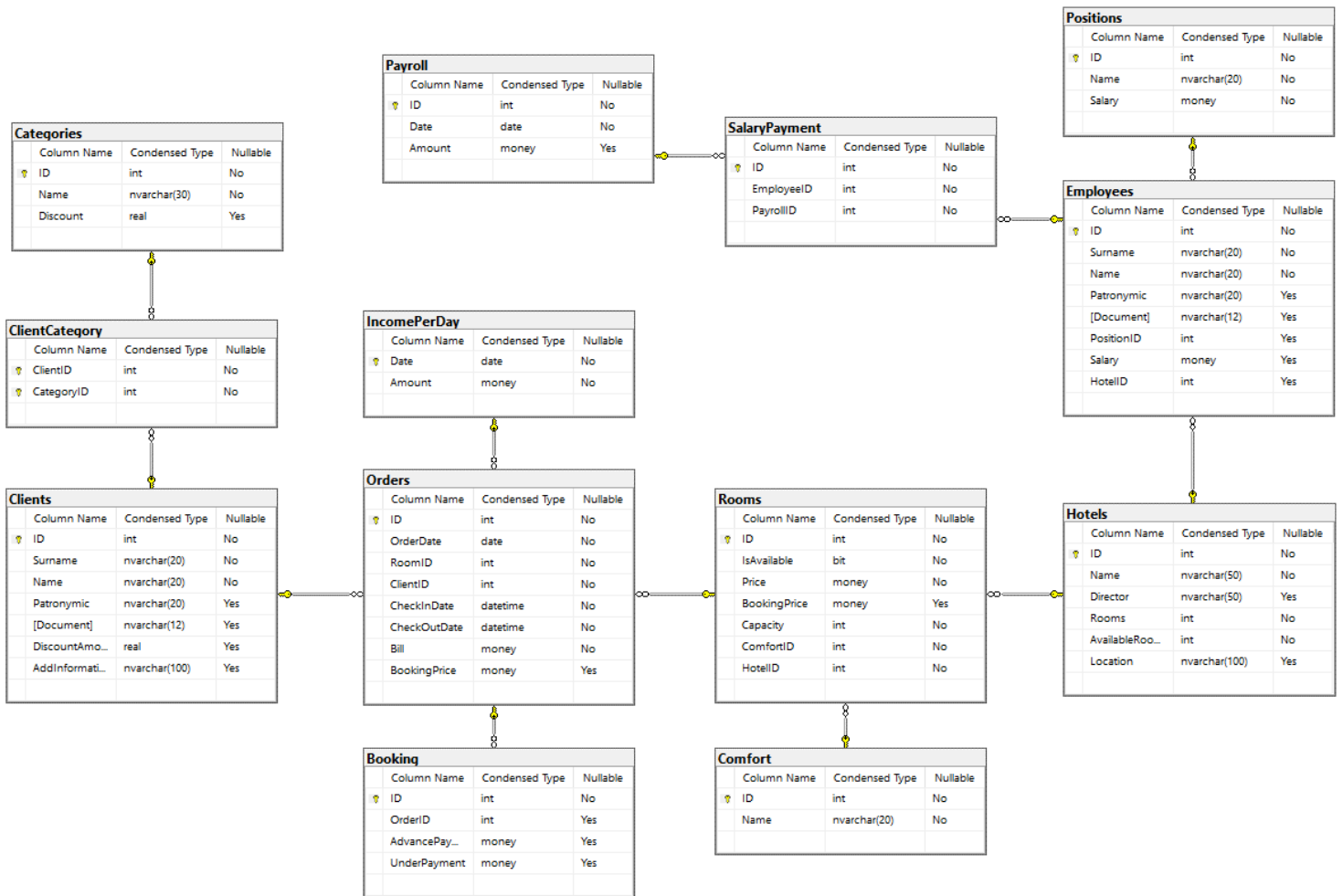


Рисунок 3.2 – Схема бази даних, згенерована засобами СУБД

4. Висновок

Під час виконання лабораторної роботи з теми «Створення бази даних» було отримано навички проектування бази даних, введення і редагування структури таблиць та даних в таблицях. Було вивчено DDL-команди SQL для роботи з таблицями (створення, модифікація та видалення таблиць) та засоби SQL для підтримки цілісності даних. Отримані знання було практично закріплено при створенні бази даних шляхом визначення схеми БД та заповненні її тестовими даними.