**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Разработка, сопровождение и обеспечение безопасности информационных систем |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Лабораторный практикум № 9**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **по дисциплине** |  | Администрирование информационных систем | | |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | **Клепиков Сергей Григорьевич** |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | **Вбио - 304 рсоб** |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г**

Для выполнения лабораторного практикума необходимо спроектировать базу данных для гостиницы, а затем реализовать защиту этой базы данных через создание пользовательских ролей и управление правами доступа. Давайте пошагово рассмотрим выполнение задания.

Шаг 1: Создание схемы базы данных

Предположим, у нас есть база данных гостиницы, которая включает следующие сущности:

1. Guest (Клиент) – информация о клиентах.

2. Room (Номер) – номера отеля.

3. Reservation (Бронирование) – бронирование номеров клиентами.

4. Payment (Оплата) – платежи клиентов за проживание.

Диаграмма сущностей и связей (ER-диаграмма):

erDiagram

Guest ||--o{ Reservation : " "

Room ||--o{ Reservation : " "

Reservation ||--|{ Payment : " "

Таблицы:

CREATE TABLE Guest (

GuestID INT PRIMARY KEY IDENTITY,

FirstName NVARCHAR(50),

LastName NVARCHAR(50),

PhoneNumber NVARCHAR(20),

Email NVARCHAR(100)

);

CREATE TABLE Room (

RoomID INT PRIMARY KEY IDENTITY,

RoomType NVARCHAR(30), -- Например, 'Стандарт', 'Люкс'

Price DECIMAL(10, 2),

IsAvailable BIT DEFAULT 1

);

CREATE TABLE Reservation (

ReservationID INT PRIMARY KEY IDENTITY,

GuestID INT FOREIGN KEY REFERENCES Guest(GuestID),

RoomID INT FOREIGN KEY REFERENCES Room(RoomID),

CheckInDate DATE,

CheckOutDate DATE,

TotalPrice DECIMAL(10, 2)

);

CREATE TABLE Payment (

PaymentID INT PRIMARY KEY IDENTITY,

ReservationID INT FOREIGN KEY REFERENCES Reservation(ReservationID),

Amount DECIMAL(10, 2),

PaymentMethod NVARCHAR(50), -- Например, 'Кредитная карта', 'Наличные'

PaymentDate DATETIME

);

Заполнение таблиц данными:

INSERT INTO Guest (FirstName, LastName, PhoneNumber, Email)

VALUES ('Иван', 'Иванов', '+7 999 1234567', 'ivanov@example.com'),

('Петр', 'Петров', '+7 888 7654321', 'petrov@example.com');

INSERT INTO Room (RoomType, Price, IsAvailable)

VALUES ('Стандарт', 5000.00, 1),

('Люкс', 8000.00, 1);

INSERT INTO Reservation (GuestID, RoomID, CheckInDate, CheckOutDate, TotalPrice)

VALUES (1, 1, '2023-05-01', '2023-05-03', 15000.00),

(2, 2, '2023-06-15', '2023-06-18', 32000.00);

INSERT INTO Payment (ReservationID, Amount, PaymentMethod, PaymentDate)

VALUES (1, 15000.00, 'Кредитная карта', '2023-04-25T12:34:56'),

(2, 16000.00, 'Наличные', '2023-06-14T09:45:23');

Шаг 2: Определение объектов базы данных только для чтения (SELECT)

Объекты, доступные только для чтения, могут включать информацию о клиентах, номерах и бронированиях. Это может понадобиться администраторам для просмотра статистики и анализа данных без возможности внесения изменений.

Шаг 3: Определение изменяемых объектов базы данных (INSERT, UPDATE, DELETE) и частота их корректировки

Изменяемыми объектами будут те же таблицы, но с разными уровнями прав доступа:

- Guest: добавление новых клиентов (INSERT), обновление информации о существующих клиентах (UPDATE), удаление клиентов (DELETE). Частота корректировок: высокая при регистрации новых гостей, низкая при удалении.

- Room: добавление новых номеров (INSERT), изменение статуса доступности номера (UPDATE), редкое удаление номеров (DELETE). Частота корректировок: средняя.

- Reservation: добавление новых бронирований (INSERT), изменение дат проживания (UPDATE), отмена бронирований (DELETE). Частота корректировок: высокая.

- Payment: добавление платежей (INSERT), редко обновление метода оплаты или суммы (UPDATE), крайне редко удаление платежа (DELETE). Частота корректировок: высокая при поступлении платежей, низкая при изменениях.

Шаг 4: Создание пользовательских ролей и назначение прав доступа

Создадим две роли: `ReadOnly` для сотрудников, которым нужен доступ только для чтения, и `Admin` для администраторов, имеющих полный доступ.

Роль ReadOnly:

CREATE ROLE ReadOnly;

GRANT SELECT ON Guest TO ReadOnly;

GRANT SELECT ON Room TO ReadOnly;

GRANT SELECT ON Reservation TO ReadOnly;

GRANT SELECT ON Payment TO ReadOnly;

Роль Admin:

CREATE ROLE Admin;

GRANT ALL ON Guest TO Admin;

GRANT ALL ON Room TO Admin;

GRANT ALL ON Reservation TO Admin;

GRANT ALL ON Payment TO Admin;

Шаг 5: Создание пользователей и назначение их ролям

Создадим двух пользователей: одного с ролью `ReadOnly`, другого с ролью `Admin`.

CREATE LOGIN Reader WITH PASSWORD = 'P@ssw0rd';

CREATE USER Reader FOR LOGIN Reader;

ALTER ROLE ReadOnly ADD MEMBER Reader;

CREATE LOGIN Administrator WITH PASSWORD = 'A@dminP@ssw0rd';

CREATE USER Administrator FOR LOGIN Administrator;

ALTER ROLE Admin ADD MEMBER Administrator;

CREATE LOGIN Reader WITH PASSWORD = 'P@ssw0rd';

CREATE USER Reader FOR LOGIN Reader;

ALTER ROLE ReadOnly ADD MEMBER Reader;

CREATE LOGIN Administrator WITH PASSWORD = 'A@dminP@ssw0rd';

CREATE USER Administrator FOR LOGIN Administrator;

ALTER ROLE Admin ADD MEMBER Administrator;

Теперь пользователи `Reader` и `Administrator` имеют соответствующие права доступа к базе данных согласно своим ролям.

Итоговая структура базы данных

USE master;

GO

IF DB\_ID('HotelDB') IS NOT NULL DROP DATABASE HotelDB;

GO

CREATE DATABASE HotelDB;

GO

USE HotelDB;

GO

CREATE TABLE Guest (

GuestID INT PRIMARY KEY IDENTITY,

FirstName NVARCHAR(50),

LastName NVARCHAR(50),

PhoneNumber NVARCHAR(20),

Email NVARCHAR(100)

);

CREATE TABLE Room (

RoomID INT PRIMARY KEY IDENTITY,

RoomType NVARCHAR(30),

Price DECIMAL(10, 2),

IsAvailable BIT DEFAULT 1

);

CREATE TABLE Reservation (

ReservationID INT PRIMARY KEY IDENTITY,

GuestID INT FOREIGN KEY REFERENCES Guest(GuestID),

RoomID INT FOREIGN KEY REFERENCES Room(RoomID),

CheckInDate DATE,

CheckOutDate DATE,

TotalPrice DECIMAL(10, 2)

);

CREATE TABLE Payment (

PaymentID INT PRIMARY KEY IDENTITY,

ReservationID INT FOREIGN KEY REFERENCES Reservation(ReservationID),

Amount DECIMAL(10, 2),

PaymentMethod NVARCHAR(50),

PaymentDate DATETIME

);

INSERT INTO Guest (FirstName, LastName, PhoneNumber, Email)

VALUES ('Иван', 'Иванов', '+7 999 1234567', 'ivanov@example.com'),

('Петр', 'Петров', '+7 888 7654321', 'petrov@example.com');

INSERT INTO Room (RoomType, Price, IsAvailable)

VALUES ('Стандарт', 5000.00, 1),

('Люкс', 8000.00, 1);

INSERT INTO Reservation (GuestID, RoomID, CheckInDate, CheckOutDate, TotalPrice)

VALUES (1, 1, '2023-05-01', '2023-05-03', 15000.00),

(2, 2, '2023-06-15', '2023-06-18', 32000.00);

INSERT INTO Payment (ReservationID, Amount, PaymentMethod, PaymentDate)

VALUES (1, 15000.00, 'Кредитная карта', '2023-04-25T12:34:56'),

(2, 16000.00, 'Наличные', '2023-06-14T09:45:23');

CREATE ROLE ReadOnly;

GRANT SELECT ON Guest TO ReadOnly;

GRANT SELECT ON Room TO ReadOnly;

GRANT SELECT ON Reservation TO ReadOnly;

GRANT SELECT ON Payment TO ReadOnly;

CREATE ROLE Admin;

GRANT ALL ON Guest TO Admin;

GRANT ALL ON Room TO Admin;

GRANT ALL ON Reservation TO Admin;

GRANT ALL ON Payment TO Admin;

CREATE LOGIN Reader WITH PASSWORD = 'P@ssw0rd';

CREATE USER Reader FOR LOGIN Reader;

ALTER ROLE ReadOnly ADD MEMBER Reader;

CREATE LOGIN Administrator WITH PASSWORD = 'A@dminP@ssw0rd';

CREATE USER Administrator FOR LOGIN Administrator;

ALTER ROLE Admin ADD MEMBER Administrator;

Таким образом, мы создали базу данных для гостиницы, определили права доступа для различных ролей и назначили этих пользователей соответствующим ролям.