**Dokument: Beskrivning av 3NF “SUN FLOWER HOTEL” Projektet**

**1. Vad är 3NF?**

Tredje normalformen (3NF) är en databasdesignprincip som syftar till att:

* Eliminera redundanta data.
* Säkerställa att varje kolumn är **beroende av den primära nyckeln** och ingenting annat.

En tabell är i **3NF** om den uppfyller följande:

1. Tabellen är i **1NF**: Alla kolumner innehåller atomära värden (inga upprepningar).
2. Tabellen är i **2NF**: Alla icke-nyckelkolumner är **fullt funktionellt beroende av hela primärnyckeln**.
3. Tabellen är i **3NF**: Inga icke-nyckelkolumner är transitivt beroende av andra icke-nyckelkolumner.

**2. Implementation av 3NF**

**Tabell 1: Guests**

CREATE TABLE Guests (

GuestId INT PRIMARY KEY,

Name NVARCHAR(100) NOT NULL,

ContactNumber NVARCHAR(15),

Email NVARCHAR(100)

);

* **1NF**: Varje kolumn innehåller ett unikt och atomärt värde.
* **2NF**: GuestId är primärnyckeln, och alla övriga kolumner är fullt beroende av GuestId.
* **3NF**: Inga kolumner är beroende av varandra; Name, ContactNumber, och Email är endast beroende av GuestId.

**Tabell 2: Rooms**

CREATE TABLE Rooms (

RoomNumber INT PRIMARY KEY,

Type NVARCHAR(50),

Price DECIMAL(10,2),

IsAvailable BIT,

ExtraBeds INT,

Size INT,

MaxGuests INT

);

* **1NF**: Alla kolumner innehåller enkla, atomära värden.
* **2NF**: Alla kolumner är fullt beroende av RoomNumber som är primärnyckeln.
* **3NF**: Ingen kolumn är transitivt beroende av någon annan kolumn.

**Tabell 3: Bookings**

CREATE TABLE Bookings (

BookingId INT PRIMARY KEY,

RoomNumber INT FOREIGN KEY REFERENCES Rooms(RoomNumber),

GuestId INT FOREIGN KEY REFERENCES Guests(GuestId),

CheckInDate DATE NOT NULL,

CheckOutDate DATE NOT NULL,

EmployeeId INT FOREIGN KEY REFERENCES Employees(EmployeeId)

);

* **1NF**: Alla kolumner innehåller atomära värden.
* **2NF**: Alla kolumner är fullt funktionellt beroende av BookingId.
* **3NF**: RoomNumber, GuestId, och EmployeeId är **nycklar** som refererar till andra tabeller; inga redundanta data finns.

**Tabell 4: PaymentInvoices**

CREATE TABLE PaymentInvoices (

PaymentInvoiceId INT PRIMARY KEY,

BookingId INT FOREIGN KEY REFERENCES Bookings(BookingId),

Amount DECIMAL(10,2),

PaymentDate DATE,

PaymentMethod NVARCHAR(50),

PaymentStatus NVARCHAR(50),

InvoiceDate DATE,

InvoiceNotes NVARCHAR(255)

);

* **1NF**: Alla kolumner innehåller enkla, atomära värden.
* **2NF**: Alla kolumner är fullt beroende av PaymentInvoiceId.
* **3NF**: BookingId är en referens till Bookings, vilket eliminerar redundans; inga andra kolumner är beroende av varandra.

**Tabell 5: BookingServices**

CREATE TABLE BookingServices (

BookingServiceId INT PRIMARY KEY,

BookingId INT FOREIGN KEY REFERENCES Bookings(BookingId),

ServiceId INT FOREIGN KEY REFERENCES Services(ServiceId),

Quantity INT,

ServiceName NVARCHAR(100)

);

* **1NF**: Alla värden är atomära.
* **2NF**: Alla kolumner är fullt beroende av BookingServiceId.
* **3NF**: BookingId och ServiceId är referenser, vilket förhindrar redundans.

**Tabell 6: Services**

CREATE TABLE Services (

ServiceId INT PRIMARY KEY,

Name NVARCHAR(100),

Price DECIMAL(10,2),

Description NVARCHAR(255)

);

* **1NF**: Alla kolumner innehåller atomära värden.
* **2NF**: Alla kolumner är fullt beroende av ServiceId.
* **3NF**: Ingen kolumn är beroende av någon annan icke-nyckelkolumn.