Приложение Д. Скрипт базы данных

АННОТАЦИЯ

В данном программном документа приведен скрипт базы данных для информационной системы управления гостиницей «Star Maze Hotel».

В данном программном документе, в разделе «Скрипт базы данных» указано наименование программы для которой разрабатывалась база данных, область применения скрипта, а также сам скрипт базы данных.

СОДЕРЖАНИЕ

[1. СКРИПТ БАЗЫ ДАННЫХ 3](#_Toc135151765)

## СКРИПТ БАЗЫ ДАННЫХ

* 1. Наименование скрипта

Наименование – «Hotel.sql»

* 1. Область применения объекта

Скрипт предназначен для приложения по управлению гостиницей «Star Maze Hotel»

* 1. Скрипт

create database Hotel

go

use Hotel

go

CREATE TABLE Tokens (

token\_id INT IDENTITY(1,1) PRIMARY KEY,

token VARCHAR(200) NOT NULL,

token\_datetime DATETIME2 NOT NULL DEFAULT(SYSDATETIME())

);

create table [dbo].[Role]

(

[ID\_Role] [int] not null identity(1,1),

[Name\_Role] [varchar] (50) not null,

constraint [PK\_Role] primary key clustered

([ID\_Role] ASC) on [PRIMARY],

constraint [UQ\_Name\_Role] unique ([Name\_Role])

)

go

insert into [dbo].[Role] ([Name\_Role])

values ('Администратор БД'),

('Гость'),

('Администратор'),

('Управляющий')

go

create table [dbo].[User]

(

[ID\_User] [int] not null identity(1,1),

[First\_Name\_User] [varchar] (30) not null,

[Last\_Name\_User] [varchar] (30) not null,

[Middle\_name\_User] [varchar] (30) not null,

[Email\_User] [varchar] (60) null,

[Number\_Phone] [varchar] (30) null,

[Pasport\_series] [varchar] (4) null,

[Pasport\_number] [varchar] (6) null,

[Login\_User] [varchar] (50) not null,

[Password\_User] [varchar] (100) not null,

[Salt] [varchar] (256) null,

[Role\_ID] [int] not null,

constraint [PK\_User] primary key clustered

([ID\_User] ASC) on [PRIMARY],

constraint [FK\_Role\_User] foreign key ([Role\_ID])

references [dbo].[Role] ([ID\_Role])

)

go

insert into [dbo].[User] ([First\_Name\_User], [Last\_Name\_User], [Middle\_name\_User], [Login\_User], [Password\_User], [Role\_ID])

values('First', 'Second', 'Middle', 'admin', 'password', 1)

select \* from [dbo].[User]

create table Status(

[ID\_Status] [int] not null identity(1,1),

[Availability] [varchar] (30) not null,

constraint [PK\_Status] primary key clustered

([ID\_Status] ASC) on [PRIMARY],

constraint [UQ\_Name\_Status] unique ([Availability])

)

insert into Status([Availability])

values ('Свободен'),

('Занят')

create table Hotels(

[ID\_Hotels] [int] not null identity(1,1),

[Adress\_hotel] [varchar] (max) not null,

[Rate\_hotel] [int] not null,

[Number\_Phone\_hotel] [varchar] (20) not null,

[Email\_hotel] [varchar] (60) not null,

constraint [PK\_Hotels] primary key clustered

([ID\_Hotels] ASC) on [PRIMARY]

)

insert into Hotels([Adress\_hotel], [Rate\_hotel], [Number\_Phone\_hotel], [Email\_hotel])

values ('адрес', 5, '84953688752', 'hotelemail@gmail.com')

go

create table Room(

[ID\_Room] [int] not null identity(1,1),

[Count\_room] [int] not null,

[Price] [int] not null,

[Number\_room] [int] not null,

[Type\_room] [varchar] (60) not null,

Status\_ID int not null,

Hotels\_ID int not null,

constraint [FK\_Status\_Room] foreign key (Status\_ID)

references [dbo].[Status] ([ID\_Status]),

constraint [FK\_Hotels\_Room] foreign key (Hotels\_ID)

references [dbo].[Hotels] ([ID\_Hotels]),

constraint [PK\_Room] primary key clustered

([ID\_Room] ASC) on [PRIMARY]

)

insert into Room([Count\_room], [Price], [Number\_room], [Type\_room], Status\_ID, Hotels\_ID)

values (100, 2500, 4, 'Эконом', 1, 1)

go

create table Service(

[ID\_Service] [int] not null identity(1,1),

[Name\_Services] [varchar] (50) not null,

[Description\_Services] [varchar] (max) not null,

[Price\_Services] [int] not null,

constraint [PK\_Service] primary key clustered

([ID\_Service] ASC) on [PRIMARY]

)

insert into Service([Name\_Services], [Description\_Services], [Price\_Services])

values('Услуга', 'Описание', 1000)

go

create table Booking(

[ID\_Booking] [int] not null identity(1,1),

Arrival\_date date not null,

Departure\_date date not null,

User\_ID int not null,

Service\_ID int not null,

Room\_ID int not null,

[Is\_Booking] BIT NOT NULL DEFAULT 0,

constraint [FK\_User\_Booking] foreign key (User\_ID)

references [dbo].[User] ([ID\_User]),

constraint [FK\_Service\_Booking] foreign key (Service\_ID)

references [dbo].[Service] ([ID\_Service]),

constraint [FK\_Room\_Booking] foreign key (Room\_ID)

references [dbo].[Room] ([ID\_Room]),

constraint [PK\_Booking] primary key clustered

([ID\_Booking] ASC) on [PRIMARY]

)

insert into Booking(Arrival\_date, Departure\_date, User\_ID, Service\_ID, Room\_ID)

values('01.09.2023', '10.09.2023', 1, 1, 1)

go

insert into Booking(Arrival\_date, Departure\_date, User\_ID, Service\_ID, Room\_ID)

values('01.09.2023', '10.09.2023', 2, 1, 1)

select \* from Booking

create table Check\_in(

[ID\_Check\_in] [int] not null identity(1,1),

Status\_Check\_in varchar (20) not null,

User\_ID int not null,

Booking\_ID int not null,

constraint [FK\_User\_Check\_in] foreign key (User\_ID)

references [dbo].[User] ([ID\_User]),

constraint [FK\_Booking\_Check\_in] foreign key (Booking\_ID)

references [dbo].[Booking] ([ID\_Booking]),

constraint [PK\_Check\_in] primary key clustered

([ID\_Check\_in] ASC) on [PRIMARY]

)

insert into Check\_in(Status\_Check\_in, User\_ID, Booking\_ID)

values('Заселено', 1, 1)

go

create table Check\_out(

[ID\_Check\_out] [int] not null identity(1,1),

Payment\_date date not null,

Total\_cost int not null,

Check\_in\_ID int not null,

User\_ID int not null,

constraint [FK\_User\_Check\_out] foreign key (User\_ID)

references [dbo].[User] ([ID\_User]),

constraint [FK\_Check\_in\_Check\_out] foreign key (Check\_in\_ID)

references [dbo].[Check\_in] ([ID\_Check\_in]),

constraint [PK\_Check\_out] primary key clustered

([ID\_Check\_out] ASC) on [PRIMARY]

)

insert into Check\_out(Payment\_date, Total\_cost, Check\_in\_ID, User\_ID)

values ('01.09.2023', 3500, 1, 2)

go

select \* from Check\_out

BEGIN TRANSACTION;

DECLARE @NewBookingID INT;

-- Вставка нового бронирования

INSERT INTO Booking (Arrival\_date, Departure\_date, User\_ID, Service\_ID, Room\_ID)

VALUES ('2023-09-01', '2023-09-10', 3, 1, 2);

-- Получение ID нового бронирования

SET @NewBookingID = SCOPE\_IDENTITY();

-- Вставка записи о заселении

INSERT INTO Check\_in (Status\_Check\_in, User\_ID, Booking\_ID)

VALUES ('Заселено', 3, @NewBookingID);

-- Если дошли до этой точки, фиксируем изменения

COMMIT;

CREATE TRIGGER UpdateBookingStatusOnCheckIn

ON Check\_in

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @BookingID INT;

SELECT @BookingID = i.Booking\_ID

FROM inserted i;

UPDATE Booking

SET Is\_Booking = 1

WHERE ID\_Booking = @BookingID;

END;

CREATE FUNCTION GetBookingInfo(@bookingID INT)

RETURNS TABLE

AS

RETURN

(

SELECT

B.ID\_Booking,

B.Arrival\_date,

B.Departure\_date,

U.First\_Name\_User + ' ' + U.Last\_Name\_User AS Guest\_Name,

R.Number\_room,

S.Name\_Services AS Booked\_Service,

CI.Status\_Check\_in AS CheckIn\_Status,

CO.Payment\_date,

CO.Total\_cost

FROM Booking B

JOIN [User] U ON B.User\_ID = U.ID\_User

JOIN Room R ON B.Room\_ID = R.ID\_Room

LEFT JOIN Service S ON B.Service\_ID = S.ID\_Service

LEFT JOIN Check\_in CI ON B.ID\_Booking = CI.Booking\_ID

LEFT JOIN Check\_out CO ON CI.ID\_Check\_in = CO.Check\_in\_ID

WHERE B.ID\_Booking = @bookingID

);

SELECT \* FROM GetBookingInfo(1);