Приложение Д. Скрипт базы данных

АННОТАЦИЯ

В данном программном документа приведен скрипт базы данных для информационной системы управления пациентами в стоматологии «Healthy smile».

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 – «dentistry.sql»

* 1. Область применения объекта

Скрипт предназначен для веб-приложения и API информационной системы управления пациентами в стоматологии «Healthy smile».

* 1. Скрипт

set ansi\_nulls on

go

set ansi\_padding on

go

set quoted\_identifier on

go

create database [Dentistry]

go

use [Dentistry]

go

create table [dbo].[Clinic]

(

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

[Addres\_Clinic] [varchar] (100) not null,

constraint [PK\_Clinic] primary key clustered

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

)

go

create table [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]

)

go

create table [dbo].[Employee]

(

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

[Surname\_Employee] [varchar] (50) not null,

[First\_Name\_Employee] [varchar] (50) not null,

[Middle\_Employee] [varchar] (50) not null,

[Email\_Employee] [varchar] (50) not null,

[Password\_Employee] [varchar] (10) not null,

[Experience] [int] not null,

[Role\_ID] [int] not null,

[Clinic\_ID] [int] not null,

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

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

constraint [FK\_Clinic] foreign key ([Clinic\_ID])

references [dbo].[Clinic] ([ID\_Clinic]),

constraint [PK\_Employee] primary key clustered

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

)

go

create table [dbo].[Service]

(

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

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

[Price\_Service] [int] not null,

constraint [PK\_Service] primary key clustered

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

)

go

create table [dbo].[Patient]

(

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

[Surname\_Patient] [varchar] (50) not null,

[First\_Name\_Patient] [varchar] (50) not null,

[Middle\_Patient] [varchar] (50) not null,

[Email\_Patient] [varchar] (50) not null,

[Password\_Patient] [varchar] (10) not null,

[Sex] [varchar] (50) not null,

[Role\_ID] [int] not null,

[Age] [varchar] (50) not null,

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

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

constraint [PK\_Patient] primary key clustered

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

)

go

create table [dbo].[Date\_Appointment]

(

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

[Date] [date] not null,

[Time] [time] not null,

[Free] [bit] not null,

constraint [PK\_Date\_Appointment] primary key clustered

([ID\_Date\_Appointment] ASC) on [PRIMARY]

)

go

create table [dbo].[Appointment]

(

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

[Number] [varchar] (5) not null,

[Complaint] [varchar] (200) not null,

[Treatment] [varchar] (200) not null,

[Employee\_ID] [int] not null,

[Patient\_ID] [int] not null,

[Service\_ID] [int] not null,

[Date\_Appointment\_ID] [int] not null,

constraint [FK\_Employee] foreign key ([Employee\_ID])

references [dbo].[Employee] ([ID\_Employee]),

constraint [FK\_Patient] foreign key ([Patient\_ID])

references [dbo].[Patient] ([ID\_Patient]),

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

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

constraint [FK\_Date\_Appointment] foreign key ([Date\_Appointment\_ID])

references [dbo].[Date\_Appointment] ([ID\_Date\_Appointment]),

constraint [PK\_Appointment] primary key clustered

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

)

go

use [Dentistry]

go

insert into [Clinic] ( [Addres\_Clinic])

values ('г. Москва, ул. Родниковая, д. 6, к.1')

go

select \* from [Clinic]

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

values ('Пациент'),('Врач'),('Менеджер по услугам')

go

select \* from [Role]

insert into [Employee] ([Surname\_Employee], [First\_Name\_Employee], [Middle\_Employee], [Email\_Employee], [Password\_Employee], [Experience], [Role\_ID], [Clinic\_ID])

values ('Олегов','Игнат','Адреевич','olegov@mail.ru','oleg1', 10, 2,1 ),

('Сухова','Екатерина','Алексеевна','syxv@mail.ru','syx1', 20, 2,1 ),

('Санпасова','Дарья','Макаровна','kis@mail.ru','kis1', 2, 2,1 ),

('Фадеева','Анастасия','Александровна','fad@mail.ru','fad1', 3, 3,1 )

go

select \* from [Employee]

insert into [Service] ([Name\_Service], [Price\_Service])

values ('Осмотр', 1000),

('Удаление кариеса', 7000),

('Чистка зубов', 4000)

go

select \* from [Service]

insert into [Patient] ([Surname\_Patient], [First\_Name\_Patient], [Middle\_Patient], [Email\_Patient], [Password\_Patient], [Sex],[Age], [Role\_ID])

values ('Юров','Юрий','Юрьевич','yur@mail.ru','yur1', 'Мужчина', 22, 1 ),

('Максимов','Максим','Максимович','max@mail.ru','max1', 'Мужчина', 32, 1 ),

('Евгеньев','Евгений','Евгеньевич','evg@mail.ru','evg1', 'Другое', 18, 1 )

go

select \* from [Patient]

insert into [Date\_Appointment] ([Date], [Time], [Free])

values ('25-09-04', '15:00', 1),

('25-09-04', '16:00', 1),

('25-09-04', '19:00', 1),

('25-12-04', '19:00', 1)

go

select \* from [Date\_Appointment]

insert into [Appointment] ([Number], [Complaint], [Treatment], [Employee\_ID],[Patient\_ID], [Service\_ID], [Date\_Appointment\_ID])

values ('1111', 'Боль в деснах','Хороший уход и обезболивающий гель', 1, 1, 1, 1),

('2111', 'Боль в зубе','Паста с кальцием и новая щетка', 2, 2, 2, 2 ),

('3111','Налёт на зубах','Регулярная чистка раз в полгода', 3, 3, 3, 3 )

go

select \* from [Appointment]

--Триггеры

CREATE OR ALTER TRIGGER Update\_Free\_Status

ON [dbo].[Appointment]

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

UPDATE da

SET da.[Free] = 0

FROM [dbo].[Date\_Appointment] da

INNER JOIN inserted i ON da.[ID\_Date\_Appointment] = i.[Date\_Appointment\_ID];

END;

CREATE OR ALTER TRIGGER Delete\_Appointment

ON [dbo].[Appointment]

AFTER DELETE

AS

BEGIN

SET NOCOUNT ON;

UPDATE da

SET da.[Free] = 1

FROM [dbo].[Date\_Appointment] da

INNER JOIN inserted i ON da.[ID\_Date\_Appointment] = i.[Date\_Appointment\_ID];

END;

--Виртуальные таблицы

CREATE OR ALTER VIEW [History] AS

SELECT

Appointment.Complaint AS 'Жалоба',

Appointment.Treatment AS 'Лечение',

Date\_Appointment.Date AS 'Дата',

Patient.Surname\_Patient+' '+Patient.First\_Name\_Patient+' '+Patient.Middle\_Patient AS 'Пациент'

FROM Appointment INNER JOIN Patient ON Appointment.Patient\_ID = Patient.ID\_Patient

INNER JOIN Date\_Appointment ON Appointment.Date\_Appointment\_ID = Date\_Appointment.ID\_Date\_Appointment

select \* from History

CREATE OR ALTER VIEW Price\_Appointment AS

SELECT Date\_Appointment.Date AS 'Дата',

Appointment.Number AS 'Номер записи',

Service.Price\_Service AS 'Цена'

FROM Appointment INNER JOIN Service ON Appointment.Service\_ID = Service.ID\_Service

INNER JOIN Date\_Appointment ON Appointment.Date\_Appointment\_ID = Date\_Appointment.ID\_Date\_Appointment

select \* from Price\_Appointment

--Транзакции

BEGIN TRY

BEGIN TRANSACTION

UPDATE Service SET Price\_Service = 2000

WHERE ID\_Service = 1;

UPDATE Service SET Price\_Service = 9000

WHERE ID\_Service = 2;

END TRY

BEGIN CATCH

--Откат транзакции

ROLLBACK TRANSACTION

SELECT ERROR\_NUMBER() AS [Номер ошибки],

ERROR\_MESSAGE() AS [Описание ошибки]

RETURN

END CATCH

COMMIT TRANSACTION

GO

SELECT ID\_Service, Name\_Service, Price\_Service

FROM Service;

--

BEGIN TRY

BEGIN TRANSACTION

UPDATE Appointment SET Number = 1212

WHERE Patient\_ID = 1;

UPDATE Appointment SET Number = 1133

WHERE Patient\_ID=2;

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION

SELECT ERROR\_NUMBER() AS [Номер ошибки],

ERROR\_MESSAGE() AS [Описание ошибки]

RETURN

END CATCH

COMMIT TRANSACTION

GO

SELECT Number AS 'Номер записи', [Surname\_Patient]+' '+[First\_Name\_Patient]+' '+[Middle\_Patient] AS 'Пациент'

FROM Appointment INNER JOIN Patient ON Appointment.Patient\_ID = Patient.ID\_Patient;

--Процедуры

CREATE OR ALTER PROCEDURE Date\_App AS

BEGIN

SELECT [Date] AS 'Дата', [Time] AS 'Время', [Surname\_Patient]+' '+[First\_Name\_Patient]+' '+[Middle\_Patient] AS 'Пациент'

FROM Appointment INNER JOIN Patient ON Appointment.Patient\_ID = Patient.ID\_Patient

INNER JOIN Date\_Appointment ON Appointment.Date\_Appointment\_ID = Date\_Appointment.ID\_Date\_Appointment

END;

EXEC Date\_App

CREATE OR ALTER PROCEDURE Experience\_Employee AS

BEGIN

SELECT [Experience] AS 'Опыт работы, г', [Addres\_Clinic] AS 'Адрес клиники', [Surname\_Employee]+' '+[First\_Name\_Employee]+' '+[Middle\_Employee] AS 'Сотрудник'

FROM Employee INNER JOIN Clinic ON Employee.Clinic\_ID = Clinic.ID\_Clinic

END;

EXEC Experience\_Employee

--Функции

CREATE OR ALTER FUNCTION Info\_Appointment()

RETURNS TABLE

AS

RETURN

(

SELECT Name\_Service AS 'Название услуги', Price\_Service AS 'Цена', [Surname\_Employee]+' '+[First\_Name\_Employee]+' '+[Middle\_Employee] AS 'Врач', [Date] AS 'Дата', [Time] AS 'Время', [Addres\_Clinic] AS 'Адрес клиники'

FROM Appointment INNER JOIN Service ON Appointment.Service\_ID = Service.ID\_Service

INNER JOIN Employee ON Appointment.Employee\_ID = Employee.ID\_Employee

INNER JOIN Clinic ON Employee.Clinic\_ID = Clinic.ID\_Clinic

INNER JOIN Date\_Appointment ON Appointment.Date\_Appointment\_ID = Date\_Appointment.ID\_Date\_Appointment

)

SELECT \* FROM Info\_Appointment()

CREATE OR ALTER FUNCTION NDFL\_Price

(@name varchar(50))

RETURNS decimal(10,2)

BEGIN

DECLARE @Summ decimal(10,2)

SELECT @Summ = Price\_Service-(Price\_Service\*0.13)

FROM Service

WHERE Name\_Service=@name

RETURN @Summ

END

SELECT dbo.NDFL\_Price('Осмотр')