ПРИЛОЖЕНИЕ А

Текст программы

АННОТАЦИЯ

Данный документ предназначен для описания всех элементов программы учета и распределения лицензионного программного обеспечения, а именно элементы базы данных, а также формы, классы программного продукта.

СОДЕРЖАНИЕ

[1. СТРУКТУРА БАЗЫ ДАННЫХ 3](#_Toc10494782)

[1.1. Описание объектов источника данных 3](#_Toc10494783)

[1.2. Скрипт объектов источника данных 4](#_Toc10494784)

[2. ПРОГРАММНЫЙ ПРОДУКТ 9](#_Toc10494785)

[2.1. Описание модулей программного продукта или программного изделия 9](#_Toc10494786)

[2.2. Текст программного продукта или программного изделия 9](#_Toc10494787)

1. СТРУКТУРА БАЗЫ ДАННЫХ
   1. Описание объектов источника данных

Таблица 1 – Описание объектов источника данных

| Название объекта | Описание |
| --- | --- |
| 1 | 2 |
| Таблицы | |
| Dolj | Таблица, содержащая информацию о должностях |
| Error | Таблица, содержащая информацию об ошибках |
| Lickluch | Таблица, содержащая информацию о лицензионных ключах |
| PO | Таблица, содержащая информацию о программном обеспечении |
| Polz | Таблица, содержащая информацию о пользователях |
| Role | Таблица, содержащая информацию о ролях |
| Sovm | Таблица, содержащая информацию о совместимости пользователя с должностью |
| Sovmosh | Таблица, содержащая информацию о совместимости пользователя с ошибками |
| Zakaz | Таблица, содержащая информацию о заказах |
| Zayavka | Таблица, содержащая информацию о заявках |
| Процедуры | |
| Polz\_add | Обеспечивает добавление элементов в таблицу Polz |
| Polz\_edit | Обеспечение изменения элементов в таблице Polz |
| Fullpolz\_edit | Обеспечение изменения элементов в таблице Polz |
| Polzpass\_edit | Обеспечение изменения элементов в таблице Polz |
| Role\_add | Обеспечивает добавление элементов в таблицу Role |
| Dolj\_add | Обеспечивает добавление элементов в таблицу Dolj |
| Sovm\_add | Обеспечивает добавление элементов в таблицу Sovm |
| Zayavka\_add | Обеспечивает добавление элементов в таблицу Zayavka |
| Zayavkast\_edit | Обеспечение изменения элементов в таблице Zayavka |
| Po\_add | Обеспечение изменения элементов в таблице Po |
| Po\_update | Обеспечение изменения элементов в таблице Po |
| Pokol\_update | Обеспечение изменения элементов в таблице Po |
| Po\_delete | Обеспечение удаления элементов в таблице Po |
| Kluch\_add | Обеспечивает добавление элементов в таблицу Lickluch |
| Kluch\_edit | Обеспечение изменения элементов в таблице Lickluch |
| Lickluchtd\_edit | Обеспечение изменения элементов в таблице Lickluch |
| Представления | |
| Polzv | Выводит информацию из таблиц Polz, Sovm, Dolj, Role |
| Zahazi | Выводит информацию из таблиц Zayavka, Polz, PO |
| Rols | Выводит информацию из таблиц Dolj, Role |
| Statistika | Выводит информацию из таблиц Zakaz, Lickluch, PO, Zayavka, Polz |
| Izmlickluch | Выводит информацию из таблиц Lickluch, PO |

* 1. Скрипт объектов источника данных

PO

CREATE TABLE [DBO].[PO]

(

[id\_PO] INT NOT NULL IDENTITY (1,1),

[naim\_po] varchar(max) NOT NULL,

[kol\_po] int NOT NULL,

[vers\_po] varchar(max) NOT NULL,

constraint [PK\_id\_po] PRIMARY KEY CLUSTERED

([id\_po] ASC) on [PRIMARY],

)

Role

CREATE TABLE [DBO].[role]

(

[id\_role] INT NOT NULL IDENTITY (1,1),

[naim\_role] varchar(max) NOT NULL,

[polz\_role] bit NOT NULL,

[zayavka\_role] bit NOT NULL,

[po\_role] bit NOT NULL,

[zakaz\_role] bit NOT NULL,

constraint [PK\_id\_role] PRIMARY KEY CLUSTERED

([id\_role] ASC) on [PRIMARY],

)

Dolj

CREATE TABLE [DBO].[dolj]

(

[id\_dolj] INT NOT NULL IDENTITY (1,1),

[naim\_dolj] varchar(max) NOT NULL,

[role\_id] int null,

constraint [PK\_id\_dolj] PRIMARY KEY CLUSTERED

([id\_dolj] ASC) on [PRIMARY],

CONSTRAINT [FK\_role\_id] FOREIGN KEY ([role\_id])

REFERENCES [DBO].[role]([id\_role]),

)

Polz

CREATE TABLE [DBO].[polz]

(

[id\_polz] INT NOT NULL IDENTITY (1,1),

[F\_P] varchar(max) NOT NULL,

[I\_P] varchar(max) NOT NULL,

[O\_P] varchar(max) NULL,

[email] varchar(max) NOT NULL,

[login] varchar(max) NOT NULL,

[password] varchar(max) NOT NULL,

[dostup] bit NOT NULL,

constraint [PK\_id\_polz] PRIMARY KEY CLUSTERED

([id\_polz] ASC) on [PRIMARY],

)

Sovm

CREATE TABLE [DBO].[sovm]

(

[id\_sovm] INT NOT NULL IDENTITY (1,1),

[polzsovm\_id] int Null,

[dolj\_id] int null,

constraint [PK\_id\_sovm] PRIMARY KEY CLUSTERED

([id\_sovm] ASC) on [PRIMARY],

CONSTRAINT [FK\_polzsovm\_id] FOREIGN KEY ([polzsovm\_id])

REFERENCES [DBO].[polz]([id\_polz]),

CONSTRAINT [FK\_dolj\_id] FOREIGN KEY ([dolj\_id])

REFERENCES [DBO].[dolj]([id\_dolj]),

)

Zayavka

CREATE TABLE [DBO].[zayavka]

(

[id\_zayavka] INT NOT NULL IDENTITY (1,1),

[status] varchar(max) NOT NULL,

[polz\_id] int NOT NULL,

[poz\_id] int NOT NULL,

constraint [PK\_id\_zayavka] PRIMARY KEY CLUSTERED

([id\_zayavka] ASC) on [PRIMARY],

CONSTRAINT [FK\_polz\_id] FOREIGN KEY ([polz\_id])

REFERENCES [DBO].[polz]([id\_polz]),

CONSTRAINT [FK\_poz\_id] FOREIGN KEY ([poz\_id])

REFERENCES [DBO].[po]([id\_po]),

)

Lickluch

CREATE TABLE [DBO].[lickluch]

(

[id\_lickluch] INT NOT NULL IDENTITY (1,1),

[kod] varchar(max) NOT NULL,

[statuskluch] bit NOT NULL,

[time] varchar(5) NULL,

[date] varchar(10) NULL,

[pol\_id] int NOT NULL,

constraint [PK\_id\_lickluch] PRIMARY KEY CLUSTERED

([id\_lickluch] ASC) on [PRIMARY],

CONSTRAINT [FK\_pol\_id] FOREIGN KEY ([pol\_id])

REFERENCES [DBO].[po]([id\_po]),

)

Error

CREATE TABLE [DBO].[error]

(

[id\_error] INT NOT NULL IDENTITY (1,1),

[naim\_error] varchar(max) NOT NULL,

[opisanie] varchar(max) NOT NULL,

[statusError] bit NOT NULL,

[sposobYstranenia] varchar(max) NULL,

constraint [PK\_id\_error] PRIMARY KEY CLUSTERED

([id\_error] ASC) on [PRIMARY],

)

Sovmosh

CREATE TABLE [DBO].[sovmosh]

(

[id\_sovmosh] INT NOT NULL IDENTITY (1,1),

[polzsovmosh\_id] int Null,

[error\_id] int null,

constraint [PK\_id\_sovmosh] PRIMARY KEY CLUSTERED

([id\_sovmosh] ASC) on [PRIMARY],

CONSTRAINT [FK\_polzsovmosh\_id] FOREIGN KEY ([polzsovmosh\_id])

REFERENCES [DBO].[polz]([id\_polz]),

CONSTRAINT [FK\_error\_id] FOREIGN KEY ([error\_id])

REFERENCES [DBO].[error]([id\_error]),

)

go

Polzv

create view [dbo].[polzv]

as select id\_polz as 'Номер пользователя',F\_P as 'Фамилия пользователя', I\_P as 'Имя пользователя',O\_P as 'Очество пользователя',email as 'Email',login as'Логин', password as'Пароль',id\_dolj as 'Номер должности',naim\_dolj as 'Должность',id\_role as 'Номер роли',naim\_role as 'роль', polz\_role as 'Доступ к пользователям',zayavka\_role as 'Доступ к заявкам',po\_role as'Доступ к ПО', zakaz\_role as 'Доступ к заказам'

from

polz inner join

sovm on sovm.polzsovm\_id=polz.id\_polz inner join

dolj on dolj.id\_dolj=sovm.dolj\_id inner join

role on role.id\_role=dolj.role\_id

go

Zahazi

create view [dbo].[zahazi]

as select id\_zayavka as 'Номер заявки',naim\_po as 'Название ПО', vers\_po as 'Версия ПО',status as 'Статус',login as 'Логин'

from

zayavka inner join

polz on zayavka.polz\_id=polz.id\_polz inner join

PO on id\_PO=poz\_id

go

Rols

create view [dbo].[rols]

as select id\_dolj as 'Номер должности',naim\_dolj as 'Должность',id\_role as 'Номер роли',naim\_role as 'роль', polz\_role as 'Доступ к пользователям',zayavka\_role as 'Доступ к заявкам',po\_role as'Доступ к ПО', zakaz\_role as 'Доступ к заказам'

from

dolj inner join

role on role.id\_role=dolj.role\_id

go

Statistika

create view [dbo].[statistika]

as select id\_lickluch as 'Номер',naim\_po as 'Название ПО', vers\_po as 'Версия ПО',F\_P as'Фамилия',I\_P as 'Имя',O\_P as 'Отчество',time as 'Время заказа',date as 'Дата заказа' from zayavka

join polz on polz.id\_polz = zayavka.polz\_id

join PO on Po.id\_PO = zayavka.poz\_id

join lickluch on lickluch.pol\_id = PO.id\_PO

where lickluch.statuskluch = 1 and zayavka.status = 'Готово'

go

Izmlickluch

create view [dbo].[izmlickluch]

as select id\_lickluch as 'Номер',naim\_po as 'Название ПО', kod as 'Код',statuskluch as 'Выдан' from lickluch

join PO on PO.id\_PO = lickluch.pol\_id

go

Polz\_add

CREATE PROCEDURE [DBO].[polz\_add]

(

@F\_P varchar(max),

@I\_P varchar(max),

@O\_P varchar(max),

@email varchar(max),

@login varchar(max),

@password varchar(max),

@dostup bit

)

AS

insert into [dbo].[polz]([F\_P],[I\_P],[O\_P],[email],[login],[password],[dostup]) values((@F\_P),(@I\_P),(@O\_P),(@email),(@login),(@password),(@dostup));

go

Polz\_edit

CREATE PROCEDURE [DBO].[polz\_edit]

@id\_polz int,

@F\_P varchar(max),

@I\_P varchar(max),

@O\_P varchar(max),

@email varchar(max),

@login varchar(max)

AS

update [dbo].polz

set

F\_P=@F\_P,

I\_P=@I\_P,

O\_P=@O\_P,

email=@email,

login=@login

where id\_polz=@id\_polz;

go

Fullpolz\_edit

CREATE PROCEDURE [DBO].[fullpolz\_edit]

@id\_polz int,

@F\_P varchar(max),

@I\_P varchar(max),

@O\_P varchar(max),

@email varchar(max),

@login varchar(max),

@password varchar(max)

AS

update [dbo].polz

set

F\_P=@F\_P,

I\_P=@I\_P,

O\_P=@O\_P,

email=@email,

login=@login,

password=@password

where id\_polz=@id\_polz;

go

Polzpass\_edit

CREATE PROCEDURE [DBO].[polzpass\_edit]

@id\_polz int,

@password varchar(max)

AS

update [dbo].polz

set

password=@password

where id\_polz=@id\_polz;

go

Role\_add

CREATE PROCEDURE [DBO].[role\_add]

(

@naim\_role varchar(max),

@polz\_role bit,

@zayavka\_role bit,

@po\_role bit,

@zakaz\_role bit

)

AS

insert into [dbo].[role]([naim\_role],[polz\_role],[zayavka\_role],[po\_role],[zakaz\_role]) values((@naim\_role),(@polz\_role),(@zayavka\_role),(@po\_role),(@zakaz\_role));

go

Dolj\_add

CREATE PROCEDURE [DBO].[dolj\_add]

(

@naim\_dolj varchar(max),

@role\_id int

)

AS

insert into [dbo].[dolj]([naim\_dolj],[role\_id]) values((@naim\_dolj),(@role\_id));

go

Sovm\_add

CREATE PROCEDURE [DBO].[sovm\_add]

(

@polzsovm\_id int,

@dolj\_id int

)

AS

insert into [dbo].[sovm]([polzsovm\_id],[dolj\_id]) values((@polzsovm\_id),(@dolj\_id));

go

Zayavka\_add

CREATE PROCEDURE [DBO].[zayavka\_add]

(

@status varchar(max),

@poz\_id int,

@polz\_id int

)

AS

insert into [dbo].[zayavka]([poz\_id],[status],[polz\_id]) values((@poz\_id),(@status),(@polz\_id));

go

Zayavka\_edit

CREATE PROCEDURE [DBO].[zayavkast\_edit]

@id\_zayavkast int,

@status varchar(max)

AS

update [dbo].zayavka

set

status=@status

where id\_zayavka=@id\_zayavkast;

go

Lickluchtd\_edit

CREATE PROCEDURE [DBO].[lickluchtd\_edit]

(

@id\_lickluch int,

@time varchar(5),

@date varchar(10),

@statuskluch bit

)

AS

update [dbo].lickluch

set

time=@time,

date=@date,

statuskluch=@statuskluch

where id\_lickluch=@id\_lickluch;

go

PO\_add

CREATE PROCEDURE [DBO].[po\_add]

(

@naim\_po varchar(max),

@kol\_po int,

@vers\_po varchar(max)

)

AS

insert into [dbo].[po]([naim\_po],[kol\_po],[vers\_po]) values((@naim\_po),(@kol\_po),(@vers\_po));

go

PO\_update

CREATE PROCEDURE [DBO].[po\_update]

(

@id\_PO int,

@naim\_po varchar(max),

@kol\_po int,

@vers\_po varchar(max)

)

AS

update [dbo].po

set

naim\_po=@naim\_po,

kol\_po=@kol\_po,

vers\_po=@vers\_po

where @id\_PO = id\_PO

go

Pokol\_update

CREATE PROCEDURE [DBO].[pokol\_update]

(

@id\_PO int,

@kol\_po int

)

AS

update [dbo].po

set

kol\_po=@kol\_po

where @id\_PO = id\_PO

go

PO\_delete

CREATE PROCEDURE [DBO].[po\_delete]

@id\_po int

AS

DELETE from dbo.po

where id\_po=@id\_po;

go

Kluch\_add

CREATE PROCEDURE [DBO].[kluch\_add]

(

@kod varchar(max),

@statuskluch bit,

@pol\_id int

)

AS

insert into [dbo].[lickluch]([kod],[statuskluch],[pol\_id]) values((@kod),(@statuskluch),(@pol\_id));

go

Error\_add

CREATE PROCEDURE [DBO].[error\_add]

(

@naim\_error varchar(max),

@opisanie varchar(max),

@statusError bit,

@sposobYstranenia varchar(max)

)

AS

insert into [dbo].[error]([naim\_error],[opisanie],[statusError],[sposobYstranenia]) values((@naim\_error),(@opisanie),(@statusError),(@sposobYstranenia));

go

Error\_update

CREATE PROCEDURE [DBO].[error\_update]

(

@id\_error int,

@naim\_error varchar(max),

@opisanie varchar(max),

@statusError bit,

@sposobYstranenia varchar(max)

)

AS

update [dbo].error

set

naim\_error=@naim\_error,

opisanie=@opisanie,

statusError=@statusError,

sposobYstranenia=@sposobYstranenia

where @id\_error = id\_error

go

Error\_delete

CREATE PROCEDURE [DBO].[error\_delete]

@id\_error int

AS

DELETE from dbo.error

where

id\_error=@id\_error;

go

Kluch\_edit

CREATE PROCEDURE [DBO].[kluch\_edit]

@id\_lickluch int,

@kod varchar(max)

AS

update [dbo].lickluch

set

kod=@kod

where id\_lickluch=@id\_lickluch;

go

1. ПРОГРАММНЫЙ ПРОДУКТ
   1. Описание модулей программного продукта или программного изделия

Таблица 2 – Описание модулей программного продукта или программного изделия

|  |  |
| --- | --- |
| Название модуля | Описание |
| Autoriz.cs | Форма авторизации пользователей |
| Glavnaya.cs | Главная форма приложения |
| Registr.cs | Форма регистрации пользователей |
| Srttings.cs | Форма настройки интерфейса |
| Zastavka.cs | Форма заставки приложения |
| BDconnect.cs | Класс для подключения к БД |
| ProgressReport.cs | Класс для выведения данных загрузки |
| Shifrovka.cs | Класс для шифрования данных |
| DeShifrovka.cs | Класс для расшифровки данных |

* 1. Текст программного продукта или программного издели