

«Московский энергетический институт»  
Кафедра Прикладной математики и искусственного интеллекта  
(ПМиИИ).

**«Разработка БД для поддержки процессов учета программного и  
технического обеспечения компьютеров компании».**

Студент: Ростовых Александра

Москва 2021

## Оглавление

Сокращения .....	2
Введение .....	3
1. Описание предметной области .....	4
2. Отчет средствами Erwin Data Modeler .....	5
2.1. Функциональные зависимости. ....	12
2.2. Обоснование нахождения РБД в третьей нормальной форме. ....	17
3. Реализация базы данных .....	18
4. Реализация интерфейса базы данных.....	23
4.1. Описание средств реализации интерфейса .....	23
4.2. Описание разработанного приложения.....	23
Заключение .....	30
Список используемой литературы .....	31

## **Сокращения**

БД – база данных

СУБД – система управления базами данных

ПО – программное обеспечение

ПК – персональный компьютер

РБД – реляционная база данных

## Введение

Целью курсового проекта является разработка модели базы данных и интерфейсных средств для поддержки процессов учета программного и технического обеспечения компьютеров компании.

Этапы выполнения курсового проекта:

- анализ предметной области;
- проектирование БД в AllFusion ERwin Data Modeler;
- реализация БД средствами MS SQL;
- программная реализация C# интерфейса для управления БД.

БД должна поддерживать выполнение следующих функций:

- учет заявок на установку программного обеспечения и контроль их исполнения;
- учет лицензий на ПО;
- учет сведений о компьютерах (id – идентификатор, ip – адрес, mac – аппаратный адрес, name – название, cpu – модель процессора, memory – объем оперативной памяти, hdd – модель жесткого диска, videocards – тип видеокарты, category – категория установленного ПО (профиль), type – тип (моноблок, notebook, стационарный, model – модель, status – состояние, internet – подключение к интернету, investment – аудитория, Номер здания, type\_investment – тип помещения, inventory\_number – инвентарный номер, information – дополнительная информация, login – логин уч. записи администратора, password – пароль уч. записи администратора);
- учет имеющегося программного обеспечения (id – идентификатор, name – название, type\_license – тип лицензии, status\_license – статус лицензии, action\_license – действие лицензии, date\_purchase – дата покупки, expiry\_date – дата окончания лицензии, number\_licenses – количество лицензий, version – версия, description – описание, price – стоимость, category – категория (учебная, специальное и т.д.), subdivision – подразделение, update – требует обновления, inventory\_number – инвентарный номер, recommended\_hardware – рекомендуемые требования к аппаратному обеспечению, use\_license – использовано лицензий, license\_number – лицензионный номер, password – пароль, language – язык, remain\_license – осталось лицензий);
- изменение данных о компьютерах и программном обеспечении;
- учет программного обеспечения, установленного на ПК;
- составление отчетов о состоянии ПО на ПК различных категорий;
- составление отчетов о программном обеспечении (по срокам действия лицензии, по типам лицензии, по назначению).

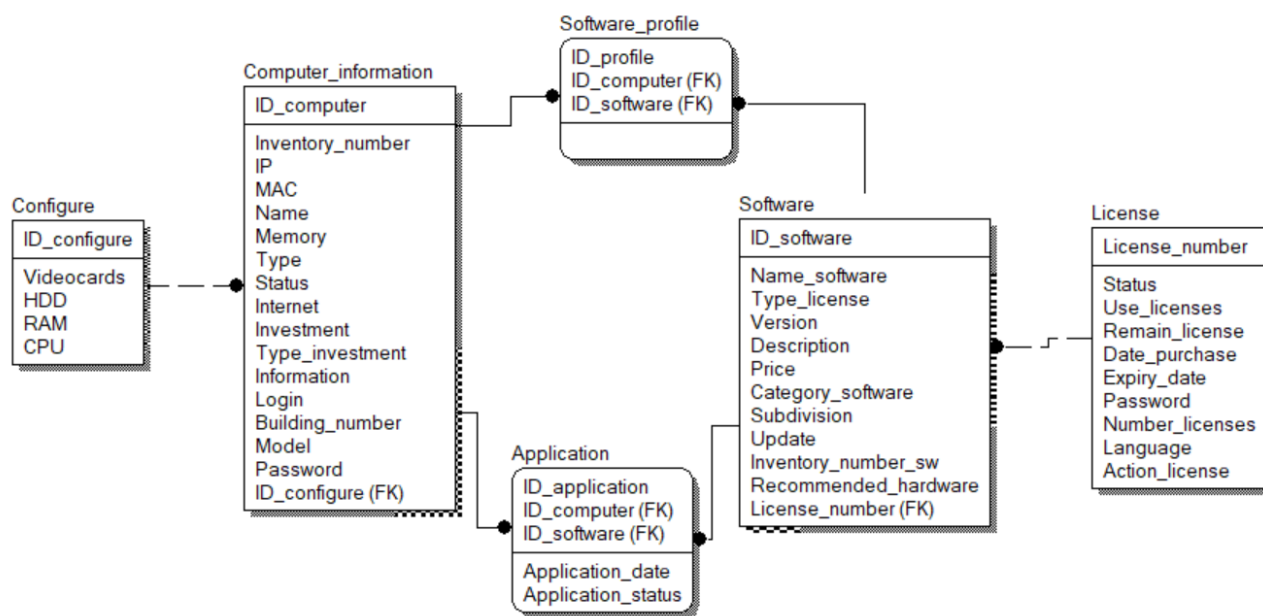
## 1. Описание предметной области

Предметная область – учёт конфигурации компьютеров, программного обеспечения (ПО), лицензий и контроль исполнения заявок отделом технической поддержки.

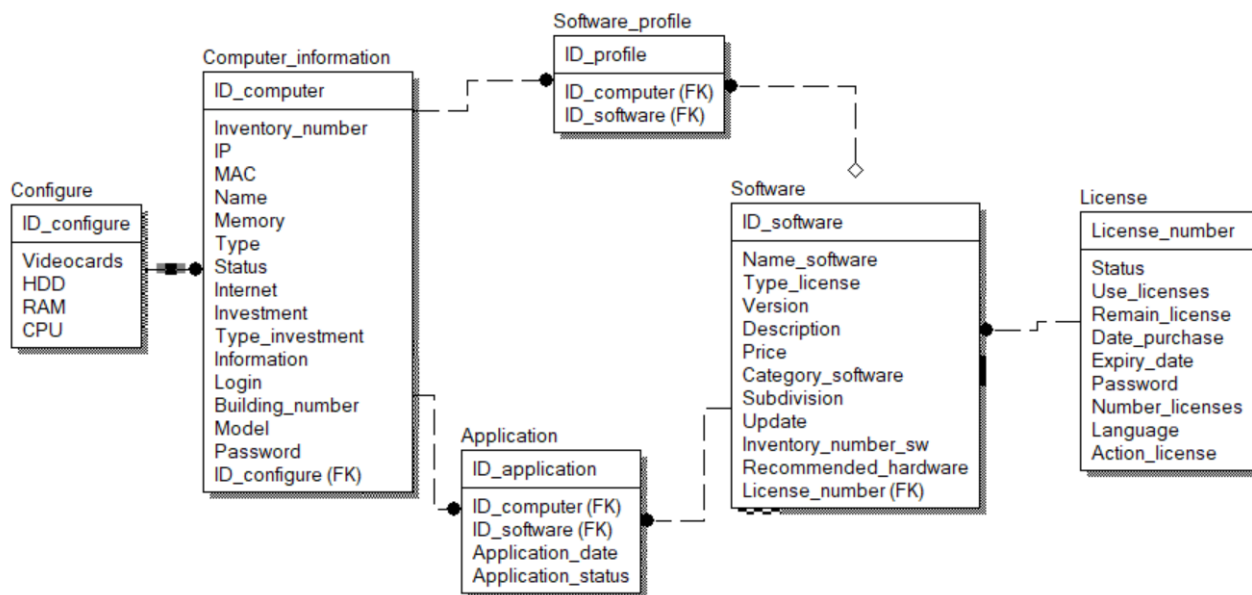
Обозначим следующие правила.

- Для каждого приложения, которое требуется установить, создаётся своя заявка.
- Одна лицензия может выдаваться на несколько ПО.
- Каждое ПО имеет только одну действующую лицензию.
- После выполнения заявки помечаются как выполненные, не удаляются
- На одном компьютере может быть установлено несколько ПО.

ER-модель:



Графическое описание РБД:



## 2. Отчет средствами Erwin Data Modeler

Entity				
Entity				
Name	Type	Definition		
Application	Independent	Заявка на установку программного обеспечения на компьютер		
Computer_information	Independent	Общие сведения о компьютере		
Configure	Independent	Определяет конфигурацию компьютера, содержит сведения о внутреннем устройстве		
License	Independent	Лицензия на ПО		
Software	Independent	Содержит сведения о программном обеспечении		
Software_profile	Independent	Профиль программного обеспечения		
Attribute(s) of "Application" Entity				
Name	Datatype	Definition	Is PK	Is FK
ID_application	INTEGER	Идентификационный номер заявки	Yes	No
ID_computer	INTEGER	Идентификационный номер компьютера	No	Yes
ID_software	INTEGER	Идентификационный номер ПО	No	Yes
Application_date	DATE	Дата подачи заявки	No	No
Application_status	CHAR(50)	Статус заявки	No	No
Validation(s) of "ID_application" Column				
Name			Rule	
Неотрицательно			>=0	
Default(s) of "Application_date" Column				
Name			Default	

Дата	Now()
------	-------

### Validation(s) of "Application\_status" Column

Name	Rule
Статус заявки	IN ('Отменен', 'Выполняется', 'Выполнен', 'В ожидании исполнения')

### Column(s) of "Application" Table

Name	Datatype	Null Option	Is PK	Is FK	Comment
ID_application	Long Integer	NOT NULL	Yes	No	Идентификационный номер заявки
ID_computer	Long Integer	NOT NULL	No	Yes	Идентификационный номер компьютера
ID_software	Long Integer	NOT NULL	No	Yes	Идентификационный номер ПО
Application_date	Date/Time	NOT NULL	No	No	Дата подачи заявки
Application_status	Text(50)	NOT NULL	No	No	Статус заявки

### Attribute(s) of "Computer\_information" Entity

Name	Datatype	Definition	Is PK	Is FK
ID_computer	INTEGER	Идентификационный номер компьютера	Yes	No
Inventory_number	INTEGER	Инвентарный номер компьютера	No	No
IP	CHAR(40)	Адрес	No	No
MAC	CHAR(40)	Аппаратный адрес	No	No
Name	CHAR(50)	Название	No	No
Memory	CHAR(50)	Объем оперативной памяти	No	No
Type	CHAR(50)	Тип (моноблок, notebook, стационарный)	No	No
Status	CHAR(50)	Состояние	No	No
Internet	CHAR(20)	Наличие подключения к интернету	No	No
Investment	CHAR(30)	Аудитория	No	No
Type_investment	CHAR(30)	Тип аудитории	No	No
Information	CHAR(500)	Дополнительные сведения о компьютере	No	No
Login	CHAR(60)	Логин учетной записи администратора	No	No
Building_number	INTEGER	Номер здания	No	No
Model	CHAR(60)	Модель	No	No
Password	CHAR(50)	Пароль учетной записи администратора	No	No
ID_configure	INTEGER	Идентификационный номер конфигурации	No	Yes

### Validation(s) of "ID\_computer" Column

Name	Rule
Неотрицательно	>=0

### Validation(s) of "Inventory\_number" Column

Name	Rule
------	------

Неотрицательно		>=0			
Validation(s) of "Internet" Column					
Name		Rule			
Да/Нет		IN ('Да', 'Нет')			
Validation(s) of "Investment" Column					
Name		Rule			
Неотрицательно		>=0			
Default(s) of "Investment" Column					
Name		Default			
Номер по умолчанию		1			
Validation(s) of "Building_number" Column					
Name		Rule			
Неотрицательно		>=0			
Default(s) of "Building_number" Column					
Name		Default			
Номер по умолчанию		1			
Column(s) of "Computer_information" Table					
Name	Datatype	Null Option	Is PK	Is FK	Comment
ID_computer	Long Integer	NOT NULL	Yes	No	Идентификационный номер компьютера
Inventory_number	Long Integer	NOT NULL	No	No	Инвентарный номер компьютера
IP	Text(40)	NOT NULL	No	No	Адрес
MAC	Text(40)	NOT NULL	No	No	Аппаратный адрес
Name	Text(50)	NOT NULL	No	No	Название
Memory	Text(50)	NULL	No	No	Объем оперативной памяти
Type	Text(50)	NULL	No	No	Тип (моноблок, notebook, стационарный)
Status	Text(50)	NOT NULL	No	No	Состояние
Internet	Text(20)	NOT NULL	No	No	Наличие подключения к интернету
Investment	Integer	NOT NULL	No	No	Аудитория
Type_investment	Text(30)	NULL	No	No	Тип аудитории
Information	Text(500)	NULL	No	No	Дополнительные сведения о компьютере
Login	Text(60)	NULL	No	No	Логин учетной записи администратора
Building_number	Long Integer	NOT NULL	No	No	Номер здания
Model	Text(60)	NOT NULL	No	No	Модель
Password	Text(50)	NULL	No	No	Пароль учетной записи администратора
ID_configure	Long Integer	NOT NULL	No	Yes	Идентификационный номер конфигурации



**Attribute(s) of "Configure" Entity**

Name	Datatype	Definition	Is PK	Is FK
ID_configure	INTEGER	Идентификационный номер конфигурации	Yes	No
Videocards	CHAR(100)	Видеокарта в компьютере	No	No
HDD	CHAR(100)	Модель жесткого диска в компьютере	No	No
RAM	CHAR(100)	Оперативная память	No	No
CPU	CHAR(100)	Центральное обрабатывающее устройство	No	No

**Validation(s) of "ID\_configure" Column**

Name	Rule
Неотрицательно	>=0

**Column(s) of "Configure" Table**

Name	Datatype	Null Option	Is PK	Is FK	Comment
ID_configure	Long Integer	NOT NULL	Yes	No	Идентификационный номер конфигурации
Videocards	Text(100)	NULL	No	No	Видеокарта в компьютере
HDD	Text(100)	NULL	No	No	Модель жесткого диска в компьютере
RAM	Text(100)	NULL	No	No	Оперативная память
CPU	Text(100)	NULL	No	No	Центральное обрабатывающее устройство

**Attribute(s) of "License" Entity**

Name	Datatype	Definition	Is PK	Is FK
License_number	INTEGER	Идентификационный номер лицензии	Yes	No
Status	CHAR(50)	Состояние	No	No
Use_licenses	INTEGER	Количество использованных лицензий	No	No
Remain_license	INTEGER	Количество оставшихся лицензий	No	No
Date_purchase	DATE	Дата покупки	No	No
Expiry_date	DATE	Дата окончания лицензии	No	No
Password	CHAR(20)	Пароль	No	No
Number_licenses	INTEGER	Количество лицензий	No	No
Language	CHAR(20)	Язык	No	No
Action_license	CHAR(20)	Действие лицензии	No	No

**Validation(s) of "License\_number" Column**

Name	Rule
Неотрицательно	>=0

**Validation(s) of "Use\_licenses" Column**

Name	Rule
Неотрицательно	>=0

**Validation(s) of "Remain\_license" Column**

Name	Rule
Неотрицательно	>=0

### Validation(s) of "Number\_licenses" Column

Name	Rule
Неотрицательно	>=0

### Validation(s) of "Action\_license" Column

Name	Rule
Да/Нет	IN ('Да', 'Нет')

### Column(s) of "License" Table

Name	Datatype	Null Option	Is PK	Is FK	Comment
License_number	Long Integer	NOT NULL	Yes	No	Идентификационный номер лицензии
Status	Text(50)	NOT NULL	No	No	Состояние
Use_licenses	Long Integer	NULL	No	No	Количество использованных лицензий
Remain_license	Long Integer	NULL	No	No	Количество оставшихся лицензий
Date_purchase	Date/Time	NOT NULL	No	No	Дата покупки
Expiry_date	Date/Time	NOT NULL	No	No	Дата окончания лицензии
Password	Text(20)	NULL	No	No	Пароль
Number_licenses	Long Integer	NULL	No	No	Количество лицензий
Language	Text(20)	NOT NULL	No	No	Язык
Action_license	Text(20)	NOT NULL	No	No	Действие лицензии

### Attribute(s) of "Software" Entity

Name	Datatype	Definition	Is PK	Is FK
ID_software	INTEGER	Идентификационный номер ПО	Yes	No
Name_software	CHAR(50)	Название ПО	No	No
Type_license	CHAR(30)	Тип лицензии на ПО	No	No
Version	CHAR(20)	Версия	No	No
Description	CHAR(1000)	Описание (не обязательно)	No	No
Price	MONEY	Стоимость	No	No
Category_software	CHAR(30)	Категория	No	No
Subdivision	CHAR(30)	Подразделение	No	No
Update	CHAR(30)	Требуется обновления (Да/Нет)	No	No
Inventory_number_sw	INTEGER	Инвентарный номер ПО	No	No
Recommended_hardware	CHAR(40)	Рекомендуемые требования к аппаратному обеспечению	No	No
License_number	INTEGER	Идентификационный номер лицензии	No	Yes

### Validation(s) of "ID\_software" Column

Name	Rule
Неотрицательно	>=0

### Validation(s) of "Price" Column

Name	Rule
Стоимость	BETWEEN 1000 AND 80000

### Validation(s) of "Category\_software" Column

Name	Rule
Категория	IN ('A', 'B', 'C', 'D')

### Default(s) of "Category\_software" Column

Name	Default
Категория по умолчанию	A

### Validation(s) of "Update" Column

Name	Rule
Да/Нет	IN ('Да', 'Нет')

### Validation(s) of "Inventory\_number\_sw" Column

Name	Rule
Неотрицательно	>=0

### Column(s) of "Software" Table

Name	Datatype	Null Option	Is PK	Is FK	Comment
ID_software	Long Integer	NOT NULL	Yes	No	Идентификационный номер ПО
Name_software	Text(50)	NOT NULL	No	No	Название ПО
Type_license	Text(30)	NOT NULL	No	No	Тип лицензии на ПО
Version	Text(20)	NOT NULL	No	No	Версия
Description	Text(1000)	NULL	No	No	Описание (не обязательно)
Price	Currency	NOT NULL	No	No	Стоимость
Category_software	Text(30)	NOT NULL	No	No	Категория
Subdivision	Text(30)	NULL	No	No	Подразделение
Update	Text(30)	NULL	No	No	Требуется обновления (Да/Нет)
Inventory_number_sw	Long Integer	NOT NULL	No	No	Инвентарный номер ПО
Recommended_hardware	Text(40)	NULL	No	No	Рекомендуемые требования к аппаратному обеспечению
License_number	Long Integer	NOT NULL	No	Yes	Идентификационный номер лицензии

### Attribute(s) of "Software\_profile" Entity

Name	Datatype	Definition	Is PK	Is FK
ID_profile	CHAR(18)	Идентификационный номер профиля	Yes	No
ID_computer	INTEGER	Идентификационный номер компьютера	No	Yes

ID_software	INTEGER	Идентификационный номер ПО	No	Yes
-------------	---------	----------------------------	----	-----

Validation(s) of "ID_profile" Column	
Name	Rule
Неотрицательно	>=0

Column(s) of "Software_profile" Table					
Name	Datatype	Null Option	Is PK	Is FK	Comment
ID_profile	Text(18)	NOT NULL	Yes	No	Идентификационный номер профиля
ID_computer	Long Integer	NOT NULL	No	Yes	Идентификационный номер компьютера
ID_software	Long Integer	NOT NULL	No	Yes	Идентификационный номер ПО

Relationship					
--------------	--	--	--	--	--

Relationship					
Name	Type	Cardinality			
Computer-Application	Non-identifying	One-to-Zero-One-or-More			
Computer-Profile	Non-identifying	One-to-Zero-One-or-More			
Configure-Computer	Non-identifying	One-to-Zero-One-or-More			
License-Profile	Non-identifying	One-to-Zero-One-or-More			
Software-Profile	Non-identifying	Zero-or-One-to-Zero-One-or-More			
Software-Application	Non-identifying	One-to-Zero-One-or-More			

Referential Integrity(s) of "Computer-Application" Relationship					
Parent Insert	Parent Update	Parent Delete	Child Insert	Child Update	Child Delete
	Cascade	Cascade	Restrict	No Action	No Action

Referential Integrity(s) of "Computer-Profile" Relationship					
Parent Insert	Parent Update	Parent Delete	Child Insert	Child Update	Child Delete
	Cascade	Cascade	Restrict	No Action	No Action

Referential Integrity(s) of "Configure-Computer" Relationship					
Parent Insert	Parent Update	Parent Delete	Child Insert	Child Update	Child Delete
	Cascade	Cascade	Restrict	No Action	No Action

Referential Integrity(s) of "License-Software" Relationship					
Parent Insert	Parent Update	Parent Delete	Child Insert	Child Update	Child Delete
	Cascade	Cascade	Restrict	No Action	No Action

Referential Integrity(s) of "Software-Profile" Relationship					
Parent Insert	Parent Update	Parent Delete	Child Insert	Child Update	Child Delete
	Cascade	Cascade	Set Null	No Action	No Action

Referential Integrity(s) of "Software-Application" Relationship					
---	--	--	--	--	--

Parent Insert	Parent Update	Parent Delete	Child Insert	Child Update	Child Delete
	Cascade	Cascade	Restrict	No Action	No Action

## 2.1. Функциональные зависимости.

### Application

ID_application
ID_computer
ID_software
Application_date
Application_status

#### **Первичный ключ:**

ID\_application

#### **Зависимости:**

ID\_application → ID\_computer

ID\_application → ID\_software

ID\_application → Application\_date

ID\_application → Application\_status

Значения атрибутов ID\_computer, ID\_software, Application\_date и Application\_status не уникальны и не позволяют однозначно идентифицировать записи в таблице. Значения любых множеств из данных атрибутов также не позволяют однозначно идентифицировать записи в таблице.

ID\_application – уникальное значение, позволяет однозначно идентифицировать любую запись в таблице.

**Ключ: ID\_application**

### Computer information

ID_computer
Inventory_number
IP
MAC
Name
Memory

Type
Status
Internet
Investment
Type_investment
Information
Login
Building_number
Model
Password
ID_configure

**Первичный ключ:**

ID\_computer

**Зависимости:**

ID\_computer → Inventory\_number

ID\_computer → IP

ID\_computer → MAC

ID\_computer → Name

ID\_computer → Memory

ID\_computer → Type

ID\_computer → Status

ID\_computer → Internet

ID\_computer → Investment

ID\_computer → Type\_investment

ID\_computer → Information

ID\_computer → Login

ID\_computer → Building\_number

ID\_computer → Model

ID\_computer → Password

ID\_computer → ID\_configure

Значения всех атрибутов кроме ID\_computer не уникальны и не позволяют однозначно идентифицировать записи в таблице. Значения любых множеств из данных атрибутов также не позволяют однозначно идентифицировать записи в таблице.

ID\_computer – уникальное значение, позволяет однозначно идентифицировать любую запись в таблице.

**Ключ: ID\_computer**

### **Configure**

ID_configure
Videocards
HDD
RAM
CPU

**Первичный ключ:**

ID\_configure

**Зависимости:**

ID\_configure → Videocards

ID\_configure → HDD

ID\_configure → RAM

ID\_configure → CPU

Значение всех атрибутов, кроме ID\_configure, не уникальны и не позволяют однозначно идентифицировать записи в таблице.

ID\_configure – уникальное значение, позволяет однозначно идентифицировать любую запись в таблице.

**Ключ: ID\_configure**

### **License**

License_number
Status
Use_licenses

Remain_license
Date_purchase
Expiry_date
Password
Number_licenses
Language
Action_license

### **Первичный ключ:**

License\_number

### **Зависимости:**

License\_number → Status

License\_number → Use\_licenses

License\_number → Remain\_license

License\_number → Date\_purchase

License\_number → Expiry\_date

License\_number → Password

License\_number →

Number\_licenses

License\_number → Language

License\_number → Action\_license

Значение атрибутов, кроме License\_number, не уникальны и не позволяют однозначно идентифицировать записи в таблице.

License\_number – уникальное значение, позволяет однозначно идентифицировать любую запись в таблице.

**Ключ: License\_number**

### **Software**

ID_software
Name_software
Type_license
Version
Description
Price
Category_software
Subdivision
Update
Inventory_number_sw



Recommended_hardware
License_number

**Первичный ключ:**

ID\_software

**Зависимости:**

ID\_software → Name\_software

ID\_software → Type\_license

ID\_software → Version

ID\_software → Description

ID\_software → Price

ID\_software → Category\_software

ID\_software → Subdivision

ID\_software → Update

ID\_software → Inventory\_number\_sw

ID\_software → Recommended\_hardware

ID\_software → License\_number

Значение всех атрибутов, кроме ID\_software , не уникальны и не позволяют однозначно идентифицировать записи в таблице.

ID\_software – уникальное значение, позволяет однозначно идентифицировать любую запись в таблице.

**Ключ: ID\_software**

**Software profile**

ID_profile
ID_computer
ID_software

**Первичный ключ:**

ID\_profile

**Зависимости:**

ID\_profile → ID\_computer

ID\_profile → ID\_software

Значения атрибутов ID\_computer, ID\_software не уникальны и не позволяют однозначно идентифицировать записи в таблице.

ID\_profile – уникальное значение, позволяет однозначно идентифицировать любую запись в таблице.

**Ключ: ID\_profile**

## **2.2. Обоснование нахождения РБД в третьей нормальной форме.**

Таблица находится в первой нормальной форме (1НФ) тогда и только тогда, когда каждое поле отношения содержит атомарное значение.

Для всех полей таблиц это условие автоматически выполняется благодаря их типам.

Например: Поле Application\_date таблицы Application имеет тип Date, а, следовательно, не может содержать неатомарное значение.

Для остальных полей атомарность значений полей предполагается по умолчанию.

Например: Поле Name таблицы Computer\_information всегда содержит только одно название конкретного компьютера. Если же название компьютера вдруг изменилось, устаревшее значение соответствующего поля заменяется новым.

Отношение находится во 2НФ, если оно находится в 1НФ и каждый неключевой атрибут функционально полно зависит от ключа.

Рассмотрим таблицу Software.

Функциональные зависимости для этого отношения:

### **Software**

ID\_software → ID\_software

ID\_software → Name\_software

ID\_software → Type\_license

ID\_software → Version

ID\_software → Description

ID\_software → Price

ID\_software → Category\_software

ID\_software → Subdivision

ID\_software → Update

ID\_software → Inventory\_number\_sw

ID\_software →

Recommended\_hardware

ID\_software → License\_number

Таким образом, показали, что каждый неключевой атрибут функционально полно зависит от ключа. Аналогично можем показать функционально полную зависимость для остальных отношений.

Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Во всех приведенных выше таблицах и их функциональных зависимостях по смыслу функциональные зависимости между неключевыми атрибутами отсутствуют, следовательно, каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

### 3. Реализация базы данных

Реализуем базу данных с помощью MS SQL Server

Для создания таблиц были созданы следующие запросы:

```
CREATE TABLE Applications
(
    ID_Application          integer NOT NULL ,
    ID_computer             integer NOT NULL FOREIGN KEY
REFERENCES Computer_information(ID_computer),
    ID_software             integer NOT NULL FOREIGN KEY
REFERENCES Software(ID_software),
    Application_date        datetime NULL,
    Application_status      varchar(50) NOT NULL CHECK
(Application_status IN('Отменен', 'Выполняется', 'Выполнен', 'В ожидании
исполнения'))
)

CREATE TABLE License
(
    License_number          integer NOT NULL,
    Status_lic              varchar(50) NOT NULL ,
    Use_licenses            integer NULL Check(Use_licenses>=0),
    Remain_license          integer NULL Check(Remain_license>=0),
    Date_purchase           datetime NULL ,
    Expiry_date             datetime NULL ,
    Password                varchar(20) NULL ,
    Number_licenses         integer Check(Number_licenses>=0),
    Language                varchar(20) NOT NULL,
    Action_license          varchar(20) NOT NULL
Check(Action_license IN('Да', 'Нет'))
)
ALTER TABLE License ADD CHECK (Status_lic IN('Свободная', 'Несвободная'))
ALTER TABLE Applications
    ADD CONSTRAINT PK_Applications PRIMARY KEY CLUSTERED
(ID_Application)

ALTER TABLE License
    ADD CONSTRAINT PK_License PRIMARY KEY CLUSTERED (License_number)
```

```

CREATE TABLE Software
(
    ID_software          integer NOT NULL,
    Name_software        varchar(50) NOT NULL ,
    Type_license         varchar(30) NOT NULL ,
    Version              varchar(20) NOT NULL ,
    Description           varchar(1000) NULL ,
    Price               integer NOT NULL ,
    Category_software    varchar(30) NOT NULL ,
    Subdivision          varchar(30) NULL,
    Updates              varchar(30) NULL ,
    Inventory_number_sw  integer NOT NULL ,
    Recommended_hardware varchar(40) NULL ,
    License_number       integer NOT NULL FOREIGN KEY REFERENCES
License(License_number),
)

ALTER TABLE Software
    ADD CONSTRAINT PK_Software PRIMARY KEY CLUSTERED (ID_software )

ALTER TABLE Software ADD CHECK (Price BETWEEN 1000 AND 80000)
ALTER TABLE Software ADD CHECK (Category_software IN('A','B','C','D'))
ALTER TABLE Software ADD CHECK (Updates IN('Да','Нет'))
ALTER TABLE Software ADD CHECK (Inventory_number_sw>=0)

CREATE TABLE Configure
(
    ID_configure          integer NOT NULL ,
    Videocards            varchar(100) NULL ,
    HDD                   varchar(100) NULL ,
    RAM                   varchar(100) NULL ,
    CPU                   varchar(100) NULL ,
)

ALTER TABLE Configure
    ADD CONSTRAINT PK_Configure PRIMARY KEY CLUSTERED (ID_configure)

CREATE TABLE Computer_information
(
    ID_computer           integer NOT NULL ,
    Inventory_number      integer NOT NULL,
    IP                    varchar(40) NOT NULL,
    MAC                   varchar(40) NOT NULL,
    Name_comp             varchar(50) NOT NULL ,
    Memory                varchar(50) NULL,
    Type_comp             varchar(50) NULL,
    Status_comp           varchar(50) NOT NULL ,
    Internet              varchar(20) NOT NULL ,
    Inverstment           integer NOT NULL ,
    Type_inverstment      varchar(30) NULL ,
    Information            varchar(500) NULL ,
    Login_pers            varchar(60) NULL ,
    Building_number       integer NOT NULL ,
    Model                 varchar(60) NOT NULL ,
    Passwd                varchar(20) NULL,

```

```

        ID_configure          integer NOT NULL FOREIGN KEY REFERENCES
Configure(ID_configure),

)
ALTER TABLE Computer_information
    ADD CONSTRAINT PK_Computer_information PRIMARY KEY CLUSTERED
(ID_computer)

CREATE TABLE Software_profile
(
    ID_profile int NOT NULL PRIMARY KEY IDENTITY,
    ID_software          integer NOT NULL FOREIGN KEY REFERENCES
Software(ID_software),
    ID_computer          integer NOT NULL FOREIGN KEY REFERENCES
Computer_information(ID_computer),

)
ALTER TABLE Computer_information ADD CHECK (Inventory_number>=0)
ALTER TABLE Computer_information ADD CHECK (Inverstment>=0)
ALTER TABLE Computer_information ADD CHECK (Internet IN('Да', 'Нет'))
ALTER TABLE Computer_information ADD CHECK (Building_number>=0)

```

С помощью пункта “Tables”->”Имя таблицы”->“Design” были проставлены правила ссылочной целостности в соответствии с предоставленным отчетом средствами Erwin Data Modeler.

Были созданы хранимые процедуры для составления отчетов в соответствии с требованиями:

Для получения сведений о ПО, установленном на компьютере:

```

CREATE PROCEDURE Software_on_comp
@PC_ID int
AS
BEGIN
    SELECT Software.Name_software, Software.Version,
Software.Type_license, License.Status_lic,
License.Action_license, License.License_number
    FROM Software
    INNER JOIN Software_profile ON Software_profile.ID_computer =
@PC_ID
    INNER JOIN License ON Software.License_number =
License.License_number
    WHERE Software.ID_software = Software_profile.ID_software
    ORDER BY Software.Name_software;
END
GO

```

Для получения сведений о лицензиях на компьютерах:

```

CREATE PROCEDURE License_info
AS
BEGIN

```

```

        SELECT Software.Name_software, Software.Category_software,
        Software.Type_license, License.Expiry_date, License.Status_lic,
        License.Action_license
        FROM Software
        INNER JOIN License ON Software.License_number =
        License.License_number
        ORDER BY Software.Type_license, License.Action_license,
        License.Status_lic, License.Expiry_date, Software.Category_software,
        Software.Name_software;
    END
GO

```

Для составления отчетов о программном обеспечении (по назначению, по срокам действия лицензии, по типам лицензии):

```

CREATE PROCEDURE FilterCateg
@cat char
AS
BEGIN
    SELECT Software.Name_software, Software.Category_software,
    Software.Type_license, License.Expiry_date, License.Status_lic,
    License.Action_license
    FROM Software
    INNER JOIN License ON Software.License_number =
    License.License_number
    WHERE Software.Category_software=@cat
    ORDER BY Software.Type_license, License.Action_license,
    License.Status_lic, License.Expiry_date, Software.Category_software,
    Software.Name_software;
END
GO

```

```

CREATE PROCEDURE FilterDate
@date DATETIME
AS
BEGIN
    SELECT Software.Name_software, Software.Category_software,
    Software.Type_license, License.Expiry_date, License.Status_lic,
    License.Action_license
    FROM Software
    INNER JOIN License ON Software.License_number =
    License.License_number
    WHERE License.Expiry_date>=@date
    ORDER BY Software.Type_license, License.Action_license,
    License.Status_lic, License.Expiry_date, Software.Category_software,
    Software.Name_software;
END
GO

```

```

CREATE PROCEDURE FilterStat
@stat varchar(20)
AS
BEGIN

```

```

        SELECT Software.Name_software, Software.Category_software,
        Software.Type_license, License.Expiry_date, License.Status_lic,
        License.Action_license
        FROM Software
        INNER JOIN License ON Software.License_number =
        License.License_number
        WHERE License.Status_lic=@stat
        ORDER BY Software.Type_license,License.Action_license,
        License.Status_lic, License.Expiry_date, Software.Category_software,
        Software.Name_software;
END
GO

```

Также были добавлены триггеры на обновление и добавление данных в таблице заявок на установку. Если заявка помечается как выполненная, значит на ПК установили ПО, что вносится в соответствующую таблицу software\_profile Если заявка наоборот помечается из выполненной в отмененную или в ожидании исполнения, соответствующая строка из таблицы удаляется.

На обновление:

```

CREATE TRIGGER updateSW3
ON Applications
AFTER Update
AS
BEGIN
IF ((SELECT Application_status FROM inserted)='Выполнен')
BEGIN
INSERT INTO Software_profile (ID_software,ID_computer) SELECT
ID_software,ID_computer from inserted
END
ELSE
Delete from Software_profile WHERE ((ID_software= (SELECT ID_software
from inserted)) AND
(ID_computer= (SELECT ID_computer from inserted)))
END

```

На добавление:

```

CREATE TRIGGER insertSW
ON Applications
AFTER INSERT
AS
BEGIN
IF ((SELECT Application_status FROM inserted)='Выполнен')
BEGIN
INSERT INTO Software_profile (ID_software,ID_computer) SELECT
ID_software,ID_computer from inserted
END
END

```

## **4. Реализация интерфейса базы данных.**

### **4.1. Описание средств реализации интерфейса**

Для создания интерфейса используется язык C# с применением технологии Windows Forms. Windows Forms — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Библиотека Windows Forms была разработана как часть .NET Framework для упрощения разработки компонентов графического интерфейса пользователя. Windows Forms построена на основе устаревающего Windows API и представляет собой, по сути, обертку низкоуровневых компонентов Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений.

В Windows Forms можно разрабатывать графически сложные приложения, которые просто развертывать, обновлять, и с которыми удобно работать как в автономном режиме, так и в сети.

В Windows Forms форма — это визуальная поверхность, на которой выводится информация для пользователя. Обычно приложение Windows Forms строится путем добавления элементов управления в формы и создания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие. Приложение реагирует на эти события, как задано в коде, и обрабатывает события при их возникновении.

### **4.2. Описание разработанного приложения**

Главная форма приложения выглядит следующим образом:



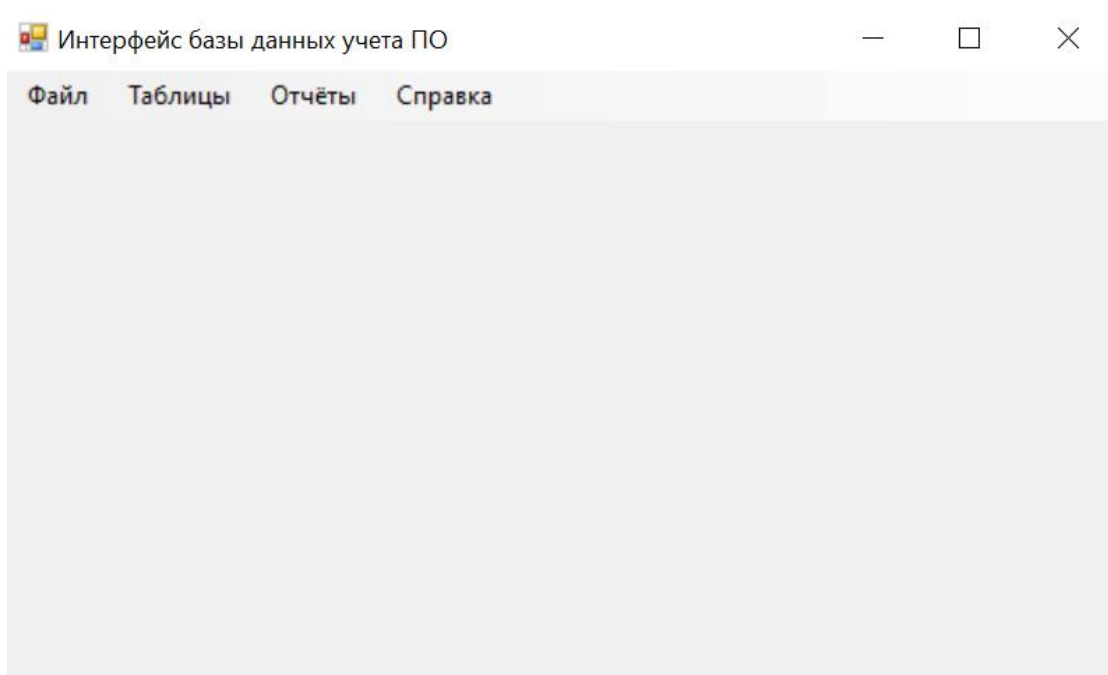


Рисунок 4.2.1. Главное окно программы

Строка меню содержит следующие элементы:

Таблицы – элемент меню со всплывающим окном для просмотра данных в таблицах базы данных.

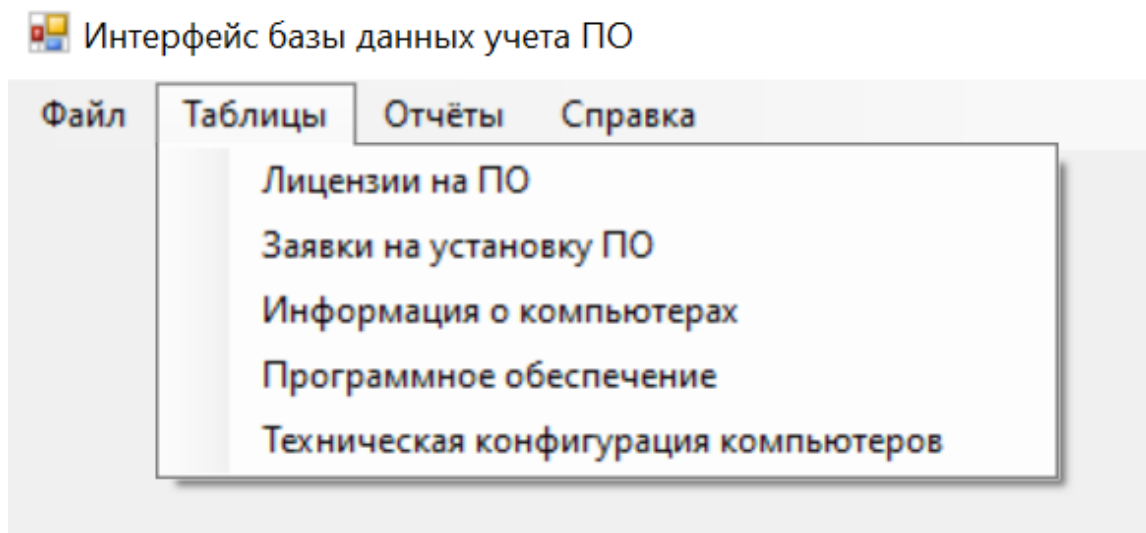


Рисунок 4.2.2. Элемент меню «Таблицы»

Отчеты – элемент меню со всплывающим окном для составления отчетов по запросам базы данных.

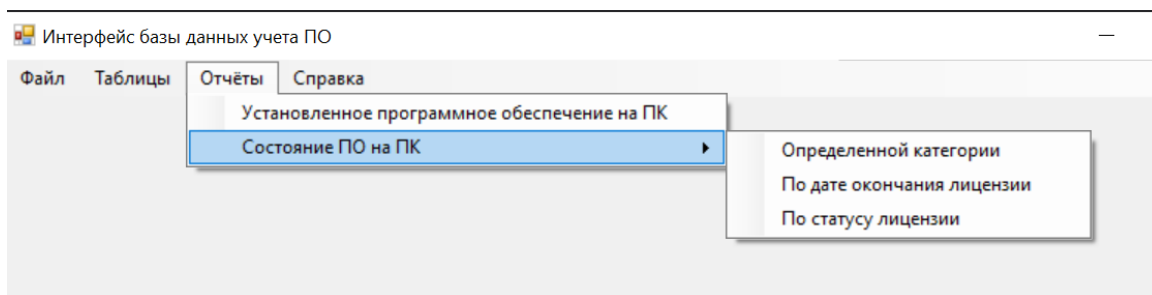


Рисунок 4.2.3. Элемент меню «Отчеты»

Справка – элемент меню со всплывающим окном для просмотра информации о программе.

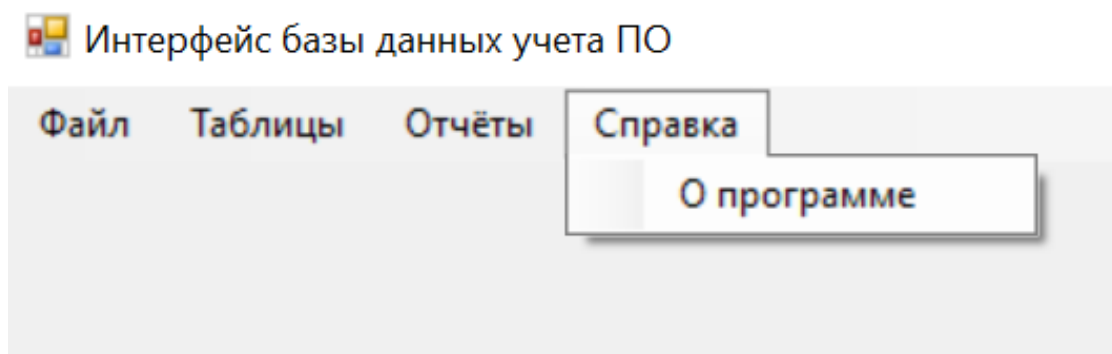


Рисунок 4.2.4. Элемент меню «Справка»

Файл – элемент меню со всплывающим окном для выхода из приложения.

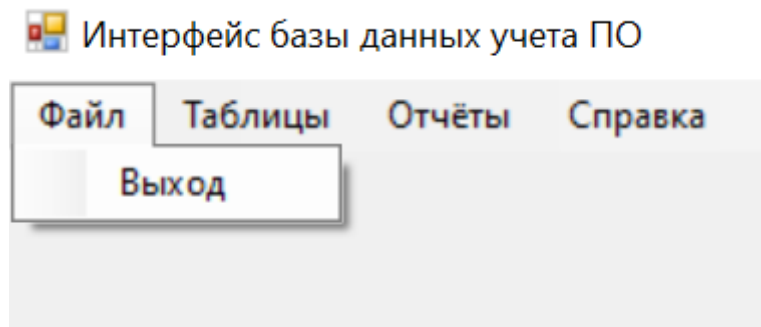
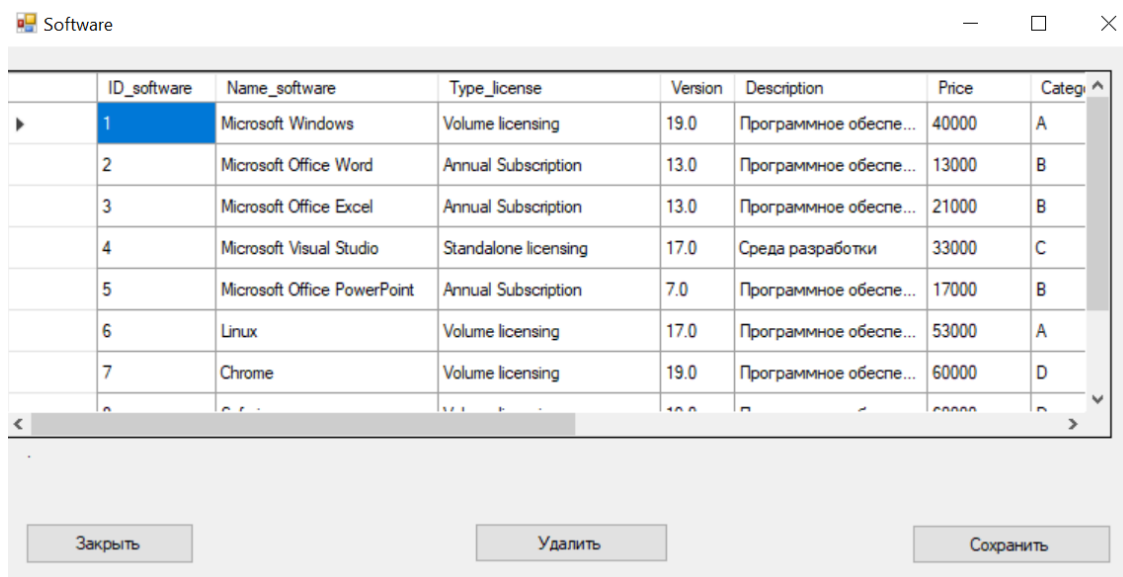


Рисунок 4.2.4. Элемент меню «Справка»

При переходе к определенной таблице, мы можем просмотреть данные в этой таблице:



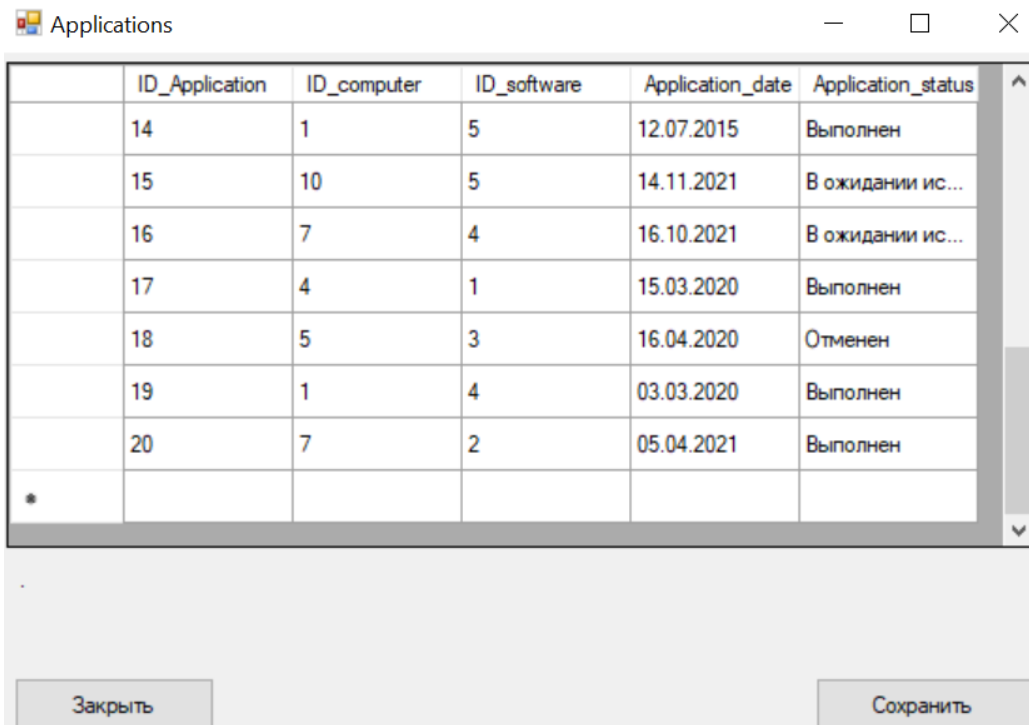
	ID_software	Name_software	Type_license	Version	Description	Price	Category
▶	1	Microsoft Windows	Volume licensing	19.0	Программное обеспе...	40000	A
	2	Microsoft Office Word	Annual Subscription	13.0	Программное обеспе...	13000	B
	3	Microsoft Office Excel	Annual Subscription	13.0	Программное обеспе...	21000	B
	4	Microsoft Visual Studio	Standalone licensing	17.0	Среда разработки	33000	C
	5	Microsoft Office PowerPoint	Annual Subscription	7.0	Программное обеспе...	17000	B
	6	Linux	Volume licensing	17.0	Программное обеспе...	53000	A
	7	Chrome	Volume licensing	19.0	Программное обеспе...	60000	D

Buttons: Заккрыть Удалить Сохранить

Рисунок 4.2.5. Окно таблицы «Программное обеспечение».

На каждой таблице имеются кнопка удаления, при нажатии на которую удаляется выбранная строка, так же в конце каждой таблицы можно добавить новую строку данных, можно обновить данные в таблице, и при нажатии кнопки «Сохранить», данные будут автоматически сохранены.

В таблице заявок удаление запрещено (так как мы помечаем заявки просто выполненными), так что кнопка удаления отсутствует.

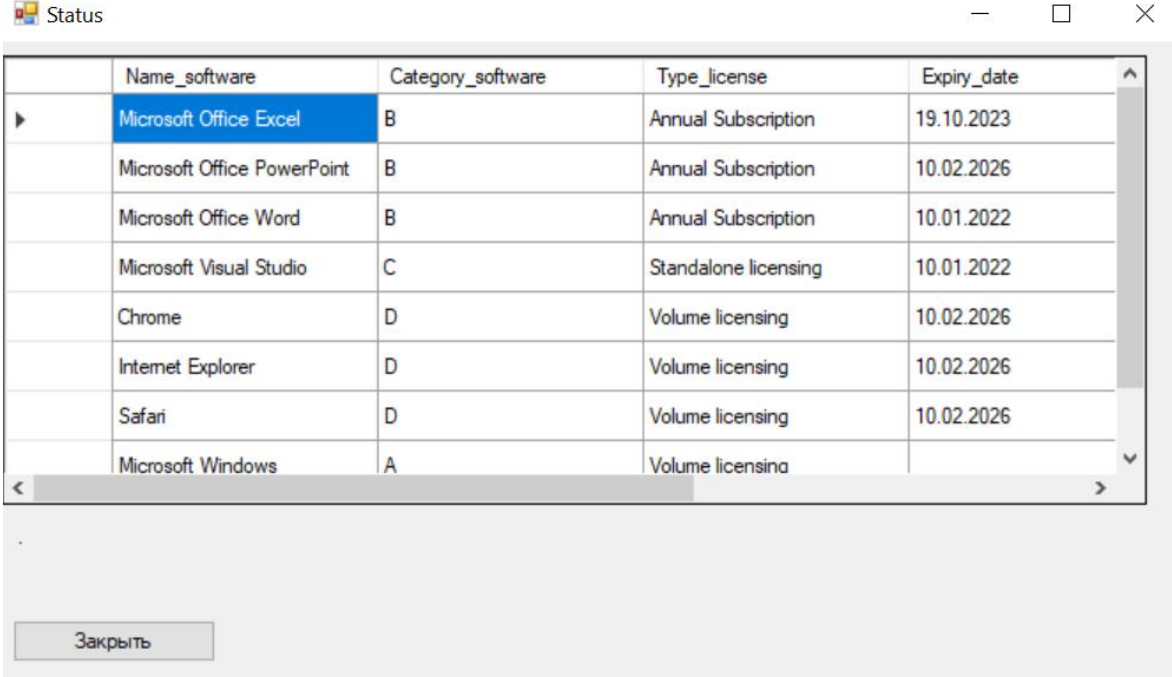


	ID_Application	ID_computer	ID_software	Application_date	Application_status
	14	1	5	12.07.2015	Выполнен
	15	10	5	14.11.2021	В ожидании ис...
	16	7	4	16.10.2021	В ожидании ис...
	17	4	1	15.03.2020	Выполнен
	18	5	3	16.04.2020	Отменен
	19	1	4	03.03.2020	Выполнен
	20	7	2	05.04.2021	Выполнен
*					

Buttons: Заккрыть Сохранить

Рисунок 4.2.6. Окно таблицы «Заявки на установку ПО».

При открытии таблицы отчетов, в отчетах, который не требуется ввод дополнительных данных, сразу открывается таблица. Редактирование в таблицах отчетов запрещено.

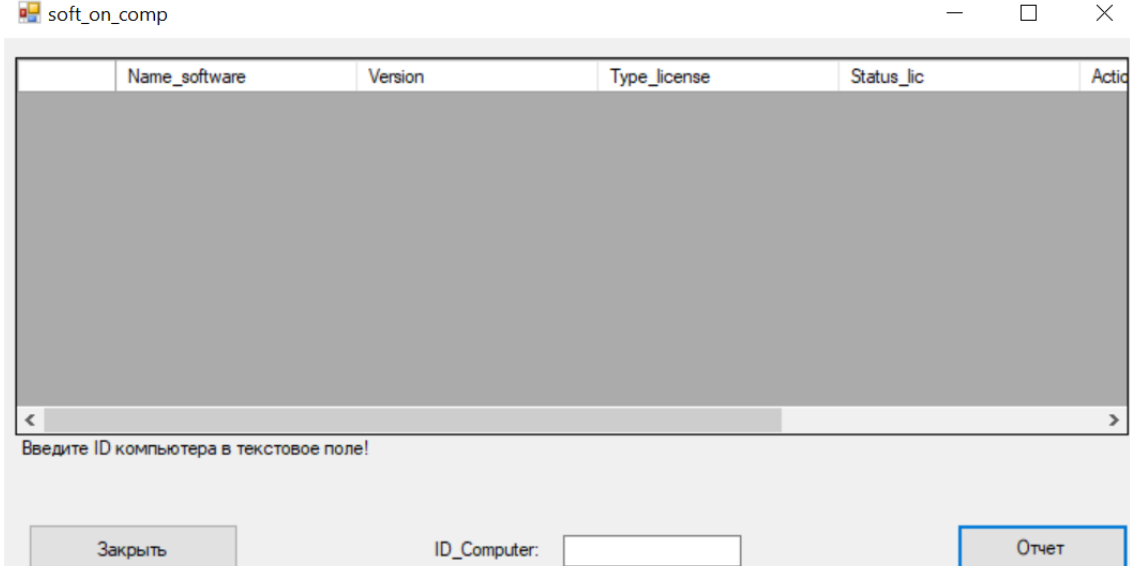


	Name_software	Category_software	Type_license	Expiry_date
▶	Microsoft Office Excel	B	Annual Subscription	19.10.2023
	Microsoft Office PowerPoint	B	Annual Subscription	10.02.2026
	Microsoft Office Word	B	Annual Subscription	10.01.2022
	Microsoft Visual Studio	C	Standalone licensing	10.01.2022
	Chrome	D	Volume licensing	10.02.2026
	Internet Explorer	D	Volume licensing	10.02.2026
	Safari	D	Volume licensing	10.02.2026
	Microsoft Windows	A	Volume licensing	

Заккрыть

Рисунок 4.2.7. Окно таблицы «Состояние лицензий на ПО».

В отчетах, где нужно ввести некоторые данные для поиска, есть специальное текстовое поле для ввода данных, если пользователь забыл ввести данные и хочет создать отчет, появится сообщение о том, что нужно ввести данные в текстовое поле.



Name_software	Version	Type_license	Status_lic	Action
---------------	---------	--------------	------------	--------

Введите ID компьютера в текстовое поле!

Заккрыть ID\_Computer:  Отчет

Рисунок 4.2.8. Окно таблицы «Установленное ПО на ПК».

При вводе данных в текстовое поле, появляется таблица ПО заданной категории с лицензиями на это ПО.

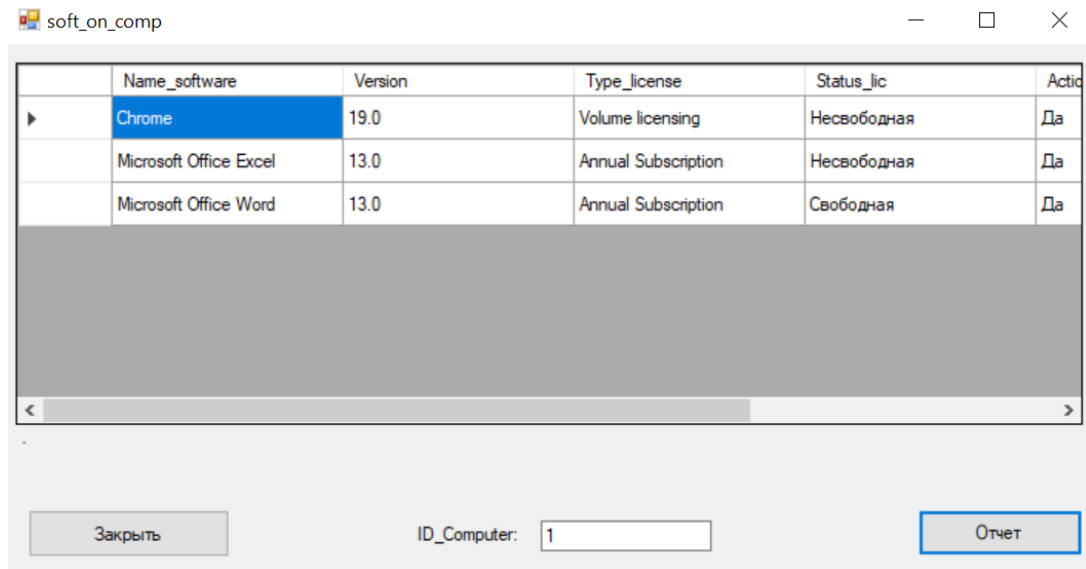


Рисунок 4.2.9. Окно таблицы «Установленное ПО на ПК».

Аналогично для отчетов о состоянии лицензий ПО:

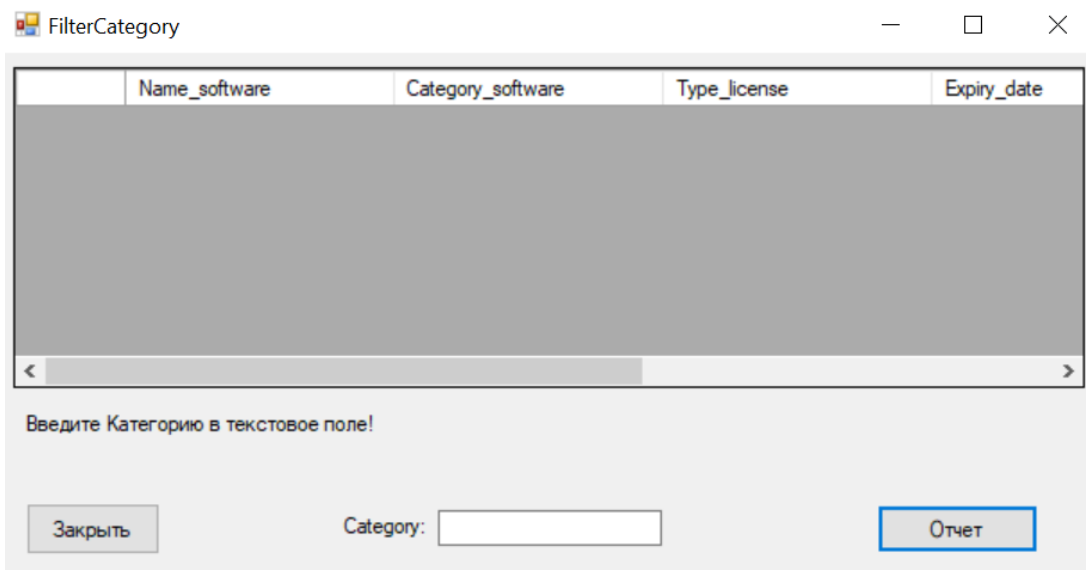


Рисунок 4.2.10. Окно таблицы «Состояние лицензии на ПО заданной категории».

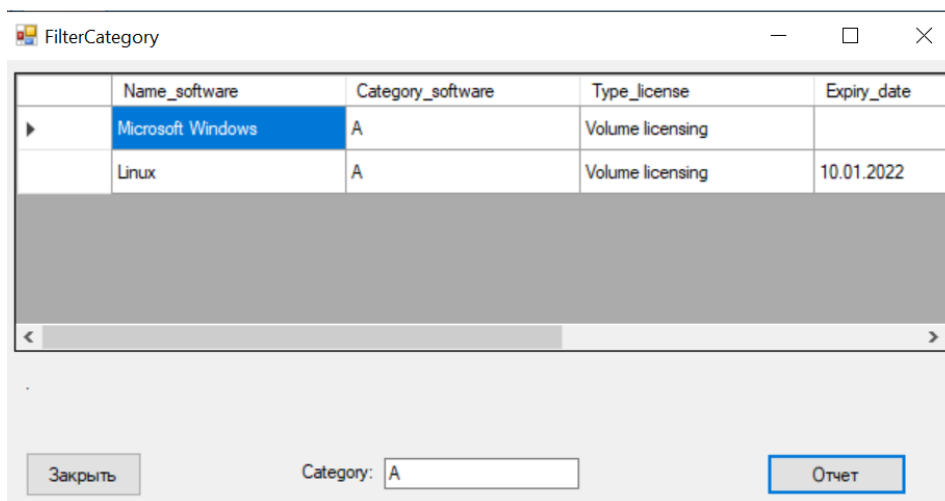


Рисунок 4.2.11. Окно таблицы «Состояние лицензии на ПО заданной категории».

Для отчета о лицензиях по сроку действия лицензии пользователь вводит в текстовое поле дату, и при составлении отчета выводятся все ПО и лицензии на них, срок действия у которых заканчивается позже заданной даты.

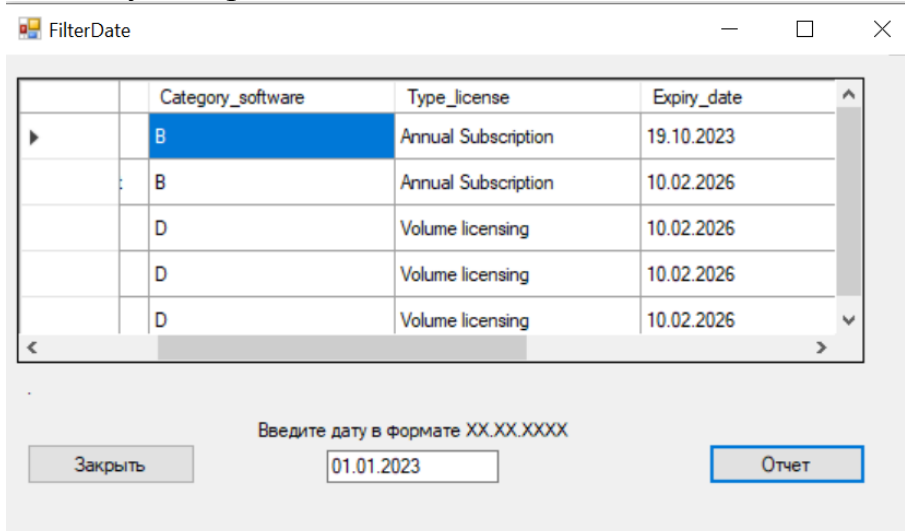


Рисунок 4.2.12. Окно таблицы «Состояние лицензии на ПО заданной даты».

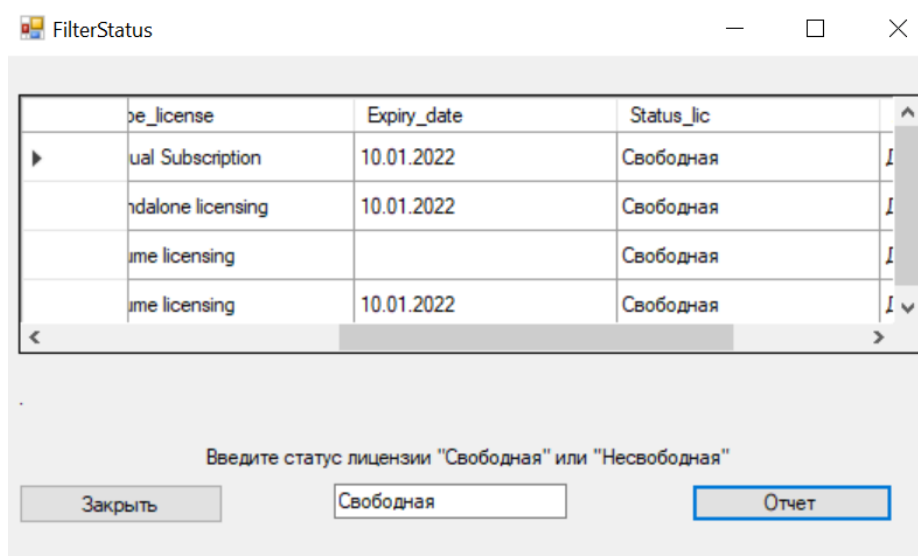


Рисунок 4.2.13. Окно таблицы «Состояние лицензии на ПО по статусу лицензии».

## Заключение

В результате выполнения курсового проекта:

- Был проведен анализ предметной области для БД и выделены основные информационные объекты;
- Разработана ER-модель БД, в которой выделены 6 сущностей и определены связи между ними;
- Разработана реляционная модель БД, включающая 6 таблиц. Модель БД доведена до 3НФ.
- Разработана БД в среде MS SQL Server, хранимые процедуры и триггеры, реализующие заданные функции работы с БД;
- Разработано приложение с БД на языке C# с применением Windows Forms.

## Список используемой литературы

1. Сидорова Н.П. Основы T-SQL. Практикум по курсу "Базы данных" учебное пособие / Н.П. Сидорова, Г.Н. Исаева. - Королев М.О. : МГОТУ, 2017. - 68 с. - ISBN 978-5-91730-784-8.
2. Сайт со справочными материалами по программному обеспечению Microsoft и технологиям с ним связанным [сайт] URL:  
<http://msdn.microsoft.com>
3. Пример программы демонстрации основных операций над данными таблицы базы данных MS SQL Server [сайт] URL:  
<https://www.bestprog.net/ru/2016/09/07/014-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B-%D0%B4%D0%B5%D0%BC%D0%BE%D0%BD%D1%81%D1%82%D1%80%D0%B0%D1%86%D0%B8%D0%B8-%D0%BE%D1%81%D0%BD%D0%BE/>