**Отчет по лабораторной работе №1**

**Тема «Управление лицензиями на ПО»**

1. **Список текущих бизнес-задач**

**Регистрация лицензиатов (пользователей ПО):** Внесение данных о новых лицензиатах в таблицу Licensees, включая их имя, фамилию, организацию и контактные данные.

**Выдача лицензий:** Создание записей в таблице Licenses, чтобы выдать лицензии на ПО. Эти записи должны содержать информацию о типе лицензии, дате выдачи и сроке действия лицензии.

**Управление ключами лицензий:** Добавление ключей активации для лицензий в таблицу License\_keys. Это позволяет отслеживать использование ключей и управлять доступом к ПО.

**Создание правил использования ПО:** Заполнение таблицы License\_Rules правилами использования ПО, которые могут быть связаны с конкретными лицензиями.

**История изменений в лицензиях:** Отслеживание истории изменений в лицензиях с помощью таблицы License\_History. Это может включать в себя изменения статуса лицензии, даты изменения и информацию о том, кто совершил изменения.

**Поиск и отчетность:** Построение запросов для поиска и фильтрации данных в базе данных, а также создание отчетов о статусе лицензий, активации ключей и другой важной информации.

**Управление сроками действия лицензий:** Автоматическое уведомление о приближающихся сроках окончания действия лицензий и их обновление.

**Отслеживание нарушений правил использования:** Мониторинг соблюдения правил использования ПО и регистрация нарушений в таблице License\_History.

**Удаление или деактивация лицензий:** Управление списанием или деактивацией лицензий при их устаревании или отзыве.

**Анализ данных:** Использование данных в базе данных для анализа трендов в использовании лицензий, оптимизации расходов и принятия решений по стратегии управления лицензиями.

1. **Список пользователей проекта**

**Администраторы:** Они имеют полный доступ к базе данных и могут выполнять все операции, включая добавление, редактирование и удаление записей о лицензиях, ключах активации и других данных.

**Сотрудники поддержки:** Эти пользователи могут использовать базу данных для регистрации запросов и обращений от клиентов, связанных с лицензиями ПО, а также для поиска информации о лицензиях и ключах активации.

**Менеджеры по лицензированию:** Они ответственны за выдачу, отслеживание и управление лицензиями на ПО. Менеджеры могут добавлять новые лицензии, управлять сроками и статусами, а также генерировать ключи активации.

**Аудиторы и юристы:** Эти пользователи могут использовать базу данных для проверки соблюдения лицензионных соглашений и правил использования ПО. Они могут проверять историю изменений в лицензиях и правилах.

**Системные администраторы:** Они отвечают за обслуживание и управление базой данных, включая создание резервных копий, обновление программного обеспечения и обеспечение безопасности данных.

**Клиенты и конечные пользователи:** Если ваша база данных предоставляет функциональность для клиентов и конечных пользователей, то они могут использовать ее для регистрации и управления своими лицензиями на ПО, а также для получения информации о правилах использования.

**Разработчики ПО:** Если в вашей организации есть разработчики ПО, они могут использовать базу данных для управления лицензионными ключами и обновлениями ПО.

1. **Оценить сроки проекта и требуемые ресурсы**

Определение требований: Сначала вам нужно провести подробный анализ требований к проекту. Это включает в себя определение функциональных и нефункциональных требований, определение пользовательских ролей и функций, а также создание детального плана проекта.

Сроки: 1-2 месяца

Проектирование базы данных: Разработка схемы базы данных, определение таблиц, связей и ключей, а также определение логики хранения и доступа к данным.

Сроки: 1-2 месяца

Разработка ПО: Создание программного обеспечения для управления лицензиями, включая интерфейс пользователя, бизнес-логику и интеграцию с базой данных.

Сроки: 3-6 месяцев (или больше, в зависимости от сложности)

Тестирование: Проведение функционального тестирования, интеграционного тестирования и тестирования безопасности для обеспечения качества и надежности ПО.

Сроки: 1-2 месяца

Внедрение и обучение: Установка ПО на серверах, обучение персонала, настройка системы и запуск в реальной среде.

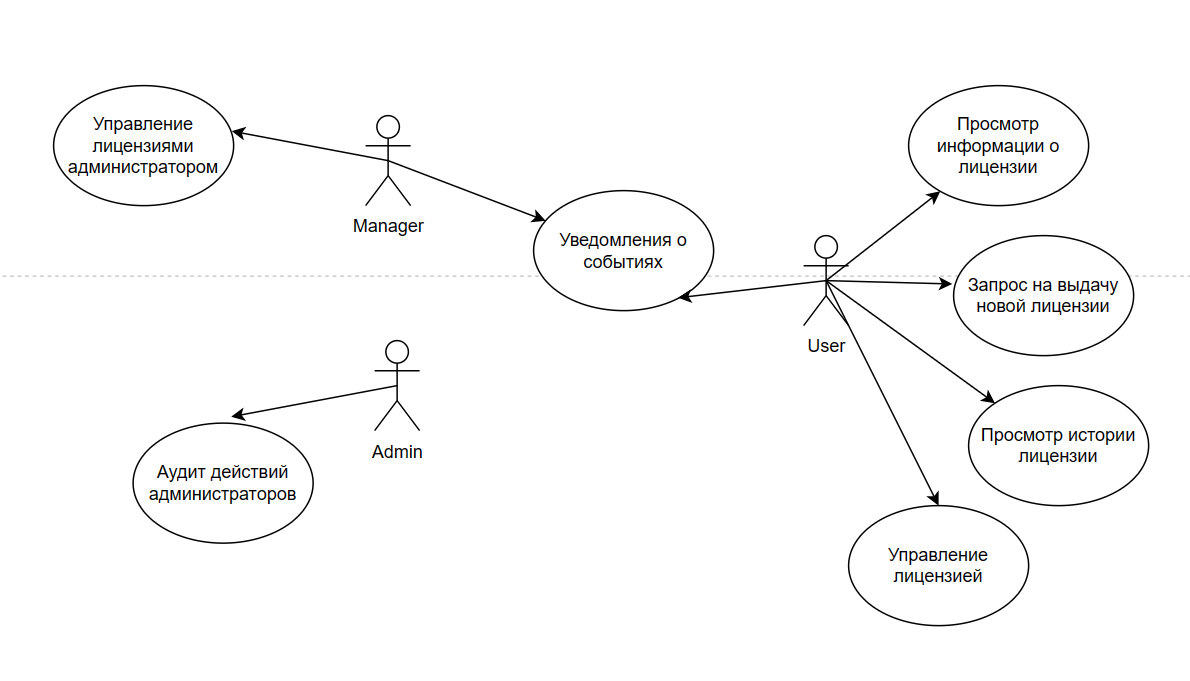
Сроки: 1-2 месяца

Поддержка и сопровождение: Обеспечение непрерывной поддержки, обновлений и регулярного обслуживания ПО.

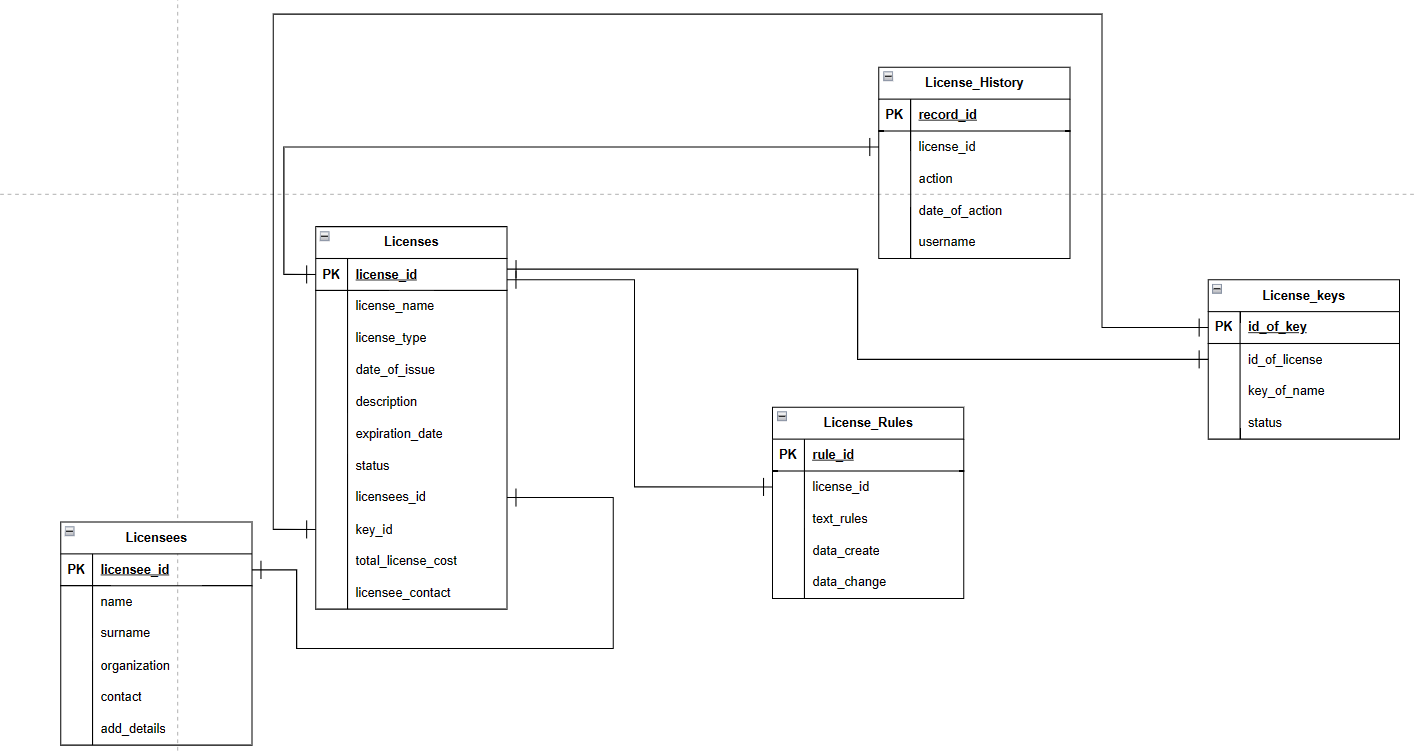
Сроки: Непрерывно

Требуемые ресурсы могут варьироваться в зависимости от размера и сложности проекта, но в общем случае вам понадобятся:

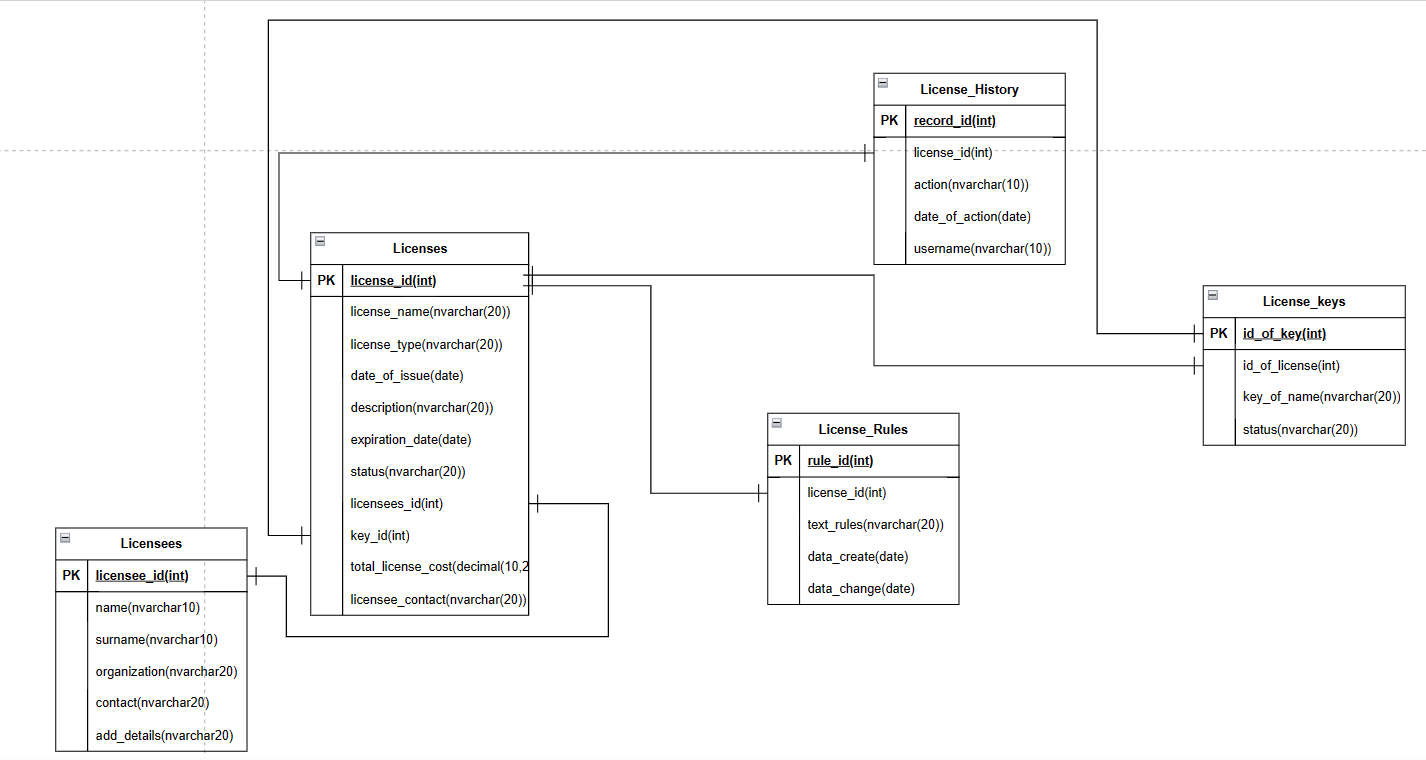
* Команда разработчиков, включая программистов, базы данных и тестировщиков.
* Бюджет для оплаты персонала и ресурсов.
* Серверное оборудование или облачные ресурсы для хранения базы данных и размещения ПО.
* Время для обучения пользователей и внедрения системы.
* Программное обеспечение для разработки и управления базой данных.

UML-диаграмма:  
  


Логическая схема БД:



Физическая схема БД:



Нормальные формы до 4НФ включительно:

Licensees:

Эта таблица содержит информацию о лицензиатах (получателях лицензии).

Нормализация: Эта таблица находится в 1-й нормальной форме (1NF) и 2-й нормальной форме (2NF), так как она имеет первичный ключ и нет частичной зависимости атрибутов от первичного ключа.

Licenses:

Эта таблица содержит информацию о лицензиях.

Нормализация: Таблица также находится в 1-й и 2-й нормальной форме, так как она имеет первичный ключ и атрибуты зависят от всего первичного ключа. Она также соответствует 3-й нормальной форме (3NF), так как нет транзитивных зависимостей атрибутов от первичного ключа. Она также соответствует 4-й нормальной форме (4NF), так как тут нет больше одной зависимости со многими значениями

License\_keys:

Эта таблица содержит информацию о ключах для лицензий.

Нормализация: Таблица также находится в 1-й и 2-й нормальной форме, так как имеет первичный ключ и атрибуты зависят от всего первичного ключа. Она также соответствует 3-й нормальной форме (3NF), так как нет транзитивных зависимостей атрибутов от первичного ключа.

License\_Rules:

Эта таблица содержит правила, связанные с лицензиями.

Нормализация: Таблица также находится в 1-й и 2-й нормальной форме, так как имеет первичный ключ и атрибуты зависят от всего первичного ключа. Она также соответствует 3-й нормальной форме (3NF), так как нет транзитивных зависимостей атрибутов от первичного ключа.

License\_History:

Эта таблица содержит историю действий с лицензиями.

Нормализация: Таблица также находится в 1-й и 2-й нормальной форме, так как имеет первичный ключ и атрибуты зависят от всего первичного ключа. Она также соответствует 3-й нормальной форме (3NF), так как нет транзитивных зависимостей атрибутов от первичного ключа.

Денормализованные атрибуты:

Total\_license\_cost – для общей стоимости лицензии

Licensee\_contract – для контактной информации лицензиата

Обеспечение безопасности:

Ограничения доступа: Удостоверьтесь, что только авторизованные пользователи имеют доступ к данным таблицам и только к тем атрибутам, к которым им разрешен доступ.

Аутентификация и авторизация: Обеспечьте аутентификацию пользователей и назначайте им соответствующие права доступа в базе данных.

Журналирование: Включите механизмы журналирования, чтобы отслеживать действия пользователей в базе данных, что помогает в обнаружении и расследовании нежелательных событий.

Шифрование: Зашифруйте конфиденциальные данные в базе данных, чтобы предотвратить их несанкционированный доступ.

Регулярное обновление: Поддерживайте систему и базу данных с помощью регулярных обновлений и патчей, чтобы устранить известные уязвимости.

Скрипт Oracle:

create table Licensees (

licensee\_id int primary key,

name nvarchar(10),

surname nvarchar(10),

organization nvarchar(20),

contact nvarchar(20),

add\_details nvarchar(20)

);

create table Licenses (

license\_id int primary key,

license\_name nvarchar(20),

license\_type nvarchar(20),

date\_of\_issue date,

expiration\_date date,

description nvarchar(20),

status nvarchar(20),

licensees\_id int,

foreign key (licensees\_id) references Licensees (licensee\_id)

);

create table License\_keys (

id\_of\_key int primary key,

id\_of\_license int,

foreign key (id\_of\_license) references Licenses (license\_id),

key\_of\_name nvarchar(20),

status nvarchar(20)

);

create table License\_Rules (

rule\_id int primary key,

license\_id int,

foreign key (license\_id) references Licenses(license\_id),

text\_rules nvarchar(20),

data\_create date,

data\_change date

);

create table License\_History (

record\_id int primary key,

license\_id int,

foreign key (license\_id) references Licenses(license\_id),

action nvarchar(10),

date\_of\_action date,

username nvarchar(10)

);

drop table License\_History;

drop table License\_Rules;

drop table License\_keys;

drop table Licensees;

drop table Licenses;

Скрипт SQL:

create table Licensees (

licensee\_id int primary key,

name nvarchar(10),

surname nvarchar(10),

organization nvarchar(20),

contact nvarchar(20),

add\_details nvarchar(20)

);

create table Licenses (

license\_id int primary key,

license\_name nvarchar(20),

license\_type nvarchar(20),

date\_of\_issue date,

expiration\_date date,

description nvarchar(20),

status nvarchar(20),

licensees\_id int,

foreign key (licensees\_id) references Licensees (licensee\_id)

);

select \* from sys.tables

create table License\_keys (

id\_of\_key int primary key,

id\_of\_license int,

foreign key (id\_of\_license) references Licenses (license\_id),

key\_of\_name nvarchar(20),

status nvarchar(20)

);

create table License\_Rules (

rule\_id int primary key,

license\_id int,

foreign key (license\_id) references Licenses(license\_id),

text\_rules nvarchar(20),

data\_create date,

data\_change date

);

create table License\_History (

record\_id int primary key,

license\_id int,

foreign key (license\_id) references Licenses(license\_id),

action nvarchar(10),

date\_of\_action date,

username nvarchar(10)

);

drop table License\_History;

drop table License\_Rules;

drop table License\_keys;

drop table Licensees;

drop table Licenses;