**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ**

**(МАИ)**

**ИНСТИТУТ №7**

**«РОБОТОТЕХНИЧЕСКИЕ И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ»**

**Кафедра 703**

**«Системное проектирование авиакомплексов»**

**Курс лекций по предмету**

**«Методы обработки информации в АСУ»**

**Курсовая работа на тему**

**«Разработка единой автоматизированной системы проектной организации»**

Выполнил:

ст. гр. М7О-506С-18

Сарнавский А.О.

Принял:

Заведующий каф. 703

Неретин Е.С.

Оглавление

[1 Введение 4](#_Toc135821486)

[2 Теоретическая часть 5](#_Toc135821487)

[2.1 Знакомство с SQL 5](#_Toc135821488)

[2.1.1 Создание и удаление базы данных и таблиц 6](#_Toc135821489)

[2.1.2 Типы данных 7](#_Toc135821490)

[2.1.3 Атрибуты столбцов 7](#_Toc135821491)

[2.1.4 Атрибуты уровня таблицы 8](#_Toc135821492)

[2.1.5 Изменение таблиц и столбцов 8](#_Toc135821493)

[2.1.6 Добавление, выборка и удаление данных 9](#_Toc135821494)

[2.1.7 Фильтрация и обновление данных 9](#_Toc135821495)

[2.1.8 Запросы 10](#_Toc135821496)

[2.1.9 Операторы фильтрации 10](#_Toc135821497)

[2.1.10 Агрегатные функции 11](#_Toc135821498)

[2.1.11 Соединения таблиц 11](#_Toc135821499)

[2.1.12 Встроенные функции 12](#_Toc135821500)

[2.1.13 Работа с файловой системой 12](#_Toc135821501)

[2.1.14 Индексация 13](#_Toc135821502)

[2.2 Знакомство с MySQL Workbench 13](#_Toc135821503)

[2.3 Знакомство с разработкой ТЗ для АИС согласно ГОСТ 14](#_Toc135821504)

[2.4 Общие сведения 16](#_Toc135821505)

[3 Практическая часть 17](#_Toc135821506)

[3.1 Файл воспроизведения БД 17](#_Toc135821507)

[3.2 Файл заполнения тестовыми данными 21](#_Toc135821508)

[3.3 Файл тестирования ввода данных в БД 24](#_Toc135821509)

[3.4 Файл тестирования запросов выборки данных 25](#_Toc135821510)

[4 Техническое задание 27](#_Toc135821511)

[4.1 Общие сведения 27](#_Toc135821512)

[4.2 Цели и назначение создания АС 27](#_Toc135821513)

[4.2.1 Назначение АС 27](#_Toc135821514)

[4.2.2 Цели создания АС 27](#_Toc135821515)

[4.3 Характеристика объекта автоматизации 27](#_Toc135821516)

[4.4 Требования к АС 28](#_Toc135821517)

[4.4.1 Функциональные требования 28](#_Toc135821518)

[4.4.2 Нефункциональные требования 39](#_Toc135821519)

[4.4.3 Требования к безопасности 43](#_Toc135821520)

[4.5 Состав и содержание работ по созданию АС 44](#_Toc135821521)

[4.6 Порядок разработки АС 44](#_Toc135821522)

[4.7 Порядок контроля и приемки АС 45](#_Toc135821523)

[4.8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу АС в действие 45](#_Toc135821524)

[4.9 Требования к документированию 45](#_Toc135821525)

[4.10 Источники разработки 46](#_Toc135821526)

[5 Список использованных источников 47](#_Toc135821527)

# Введение

Курсовая работа по курсу «Методы обработки информации в АСУ» преследует следующие цели:

1. Ознакомиться с языком запросов SQL, в частности с синтаксисом, используемым в СУБД MySQL;
2. Ознакомиться с СУБД MySQL;
3. Ознакомиться с разработкой автоматизированных информационных систем согласно ГОСТ 34.602-2020;
4. Ознакомиться с предметной областью темы курсовой работы и разработать прототип автоматизированной информационной системы.

Разработка автоматизированной информационной системы производится на ПК под управлением операционной системы Windows 10, с помощью установленной на нем среды разработки MySQL Workbench версии 8.0.32.

Разработка всех данных ЖЦ курсовой работы располагается в репозитории GitHub последующей ссылке:

<https://github.com/Sarnavskiy-Alexey/MOIASU_MAI_course_work>

# Теоретическая часть

## Знакомство с SQL

В современном мире базы данных в той или иной форме применяются практически повсеместно: в банках, в интернете, в мобильных устройствах, на телевидении и пр. Они позволяют хранить большое количество систематизированных данных и предоставляют простой доступ к этим данным с помощью систем управления базами данных.

Система управления базой данных – средство манипулирования данными, находящимися в БД, а также самой структурой БД. База данных (БД) – это совокупность хранимых в соответствии с определенной структурой данных. Таблица – сущность БД, хранящая структуру представляемых с помощью нее данных. Под данными понимается часть информации, пригодная для хранения в соответствии с определенной структурой.

Базы данных различаются на реляционные (предоставляющие доступ к взаимосвязанным элементам информации) и нереляционные (без использования системы строк и столбцов).

SQL (Structed Query Language) – язык программирования, применяемый для следующих манипуляций в реляционной БД:

* создание и удаление БД;
* создание, удаление и корректировка таблиц БД;
* добавление, удаление и корректировка полей таблиц БД;
* добавление, удаление и корректировка данных таблиц БД;
* выборка записей из одной или нескольких таблиц в соответствии с задаваемыми условиями;
* создание и управление индексами, триггерами и прочими объектами.

Многие системы управления базами данных (СУБД) предоставляют пользователю средства для визуализации создаваемой структуры БД. Пример визуализации структуры БД с помощью СУБД MySQL представлен на Рисунке Рисунок 1.

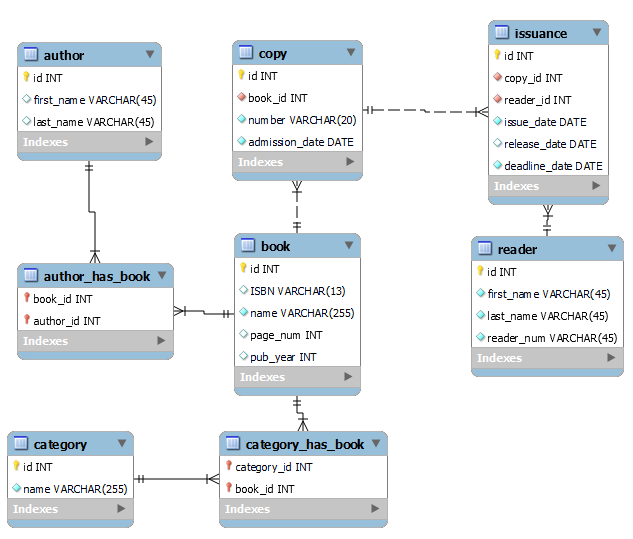


Рисунок – Пример визуализации структуры БД с помощью MySQL

В ниже следующих подразделах представлены наиболее часто употребимые в ЖЦ БД команды SQL. В квадратных скобках указываются части синтаксиса команды, необязательные при ее использовании, в фигурных – обязательные, но с выбором одного из нескольких вариантов.

### Создание и удаление базы данных и таблиц

Для создания схемы БД используется команда CREATE DATABASE:

CREATE DATABASE [IF NOT EXISTS] <название БД>;

Для удаления схемы БД используется команда DROP DATABASE:

DROP DATABASE [IF EXISTS] <название БД>;

В случае если необходима проверка на отсутствие или существование схемы БД с указанным названием, используется модификация IF NOT EXISTS или IF EXISTS соответственно.

Для создания таблиц используется команда CREATE TABLE:

CREATE TABLE <название таблицы>

(<название столбца 1> <тип данных> [<атрибуты столбца 1>][,

<название столбца 2> <тип данных> [<атрибуты столбца 2>],

.....................................................

<название столбца N> <тип данных> [<атрибуты столбца N>]][,

атрибуты уровня таблицы]

);

В качестве названия столбца желательно, но необязательно указывать идентификаторы в соответствии с их назначением (например, идентификатор компьютера – «pc\_id», фамилия – «last\_name»).

В качестве типа данных используется любой доступный в выбранной СУБД тип данных. Наиболее часто встречаемые типы данных перечислены в подразделе 2.1.2.

Атрибуты рассмотрены в подразделе 2.1.3.

### Типы данных

Наиболее часто употребимыми типами данных являются следующие:

* INT – знаковое целое число (4 байт);
* CHAR(N) – строка фиксированной длины в N символов, в случае если при инициализации поля такого типа не задействовано некоторое количество символов, эти символы заполняются пробелами;
* VARCHAR(N) – строка изменяемой длины. В отличие от CHAR занимает столько места в памяти, сколько это необходимо, но не больше N;
* DECIMAL(P, S) – число с фиксированной точностью. P – общее количество знаков в числе, S – количество символов после запятой (P>S);
* TEXT – текст длиной до 65 килобайт.
* DOUBLE, FLOAT – вещественные числа двойной и одинарной точности.
* DATE, TIME, DATETIME – дата, время и дата со временем.

### Атрибуты столбцов

В SQL существуют следующие атрибуты столбцов:

* PRIMARY KEY – первичный ключ, уникально идентифицирующий строку в таблице.
* AUTO\_INCREMENT – атрибут, указывающий на то, что поле указанного столбца автоматически увеличивает значение для каждой новой добавляемой строки.
* UNIQUE – атрибут, указывающий на то, что поле указанного столбца в пределах всей таблицы хранит только уникальные (не повторяющиеся) значения.
* NULL и NOT NULL – атрибуты, указывающие на то, может ли поле указанного столбца иметь пустое значение. По умолчанию любое поле, кроме первичного ключа, может иметь пустое значение.
* DEFAULT X – атрибут, указывающий на то, что поле указанного столбца по умолчанию принимает значение X.
* CHECK (<условие>) – атрибут, назначающий проверку поля указанного столбца на выполнение некоторого условия.

### Атрибуты уровня таблицы

В качестве атрибутов уровня таблицы используются следующие:

* определения внешних ключей FOREIGN KEY:

FOREIGN KEY (<столбец 1>[, <столбец 2>, ..., <столбец N>])

REFERENCES <главная таблица> (<столбец главной таблицы 1>, <столбец главной таблицы 2>, ..., <столбец главной таблицы N>)

* действия при удалении и обновлении строки данных ON DELETE и ON UPDATE:
  + CASCADE – автоматически удаляет или обновляет строки при удалении или обновлении связанных строк главной таблицы;
  + SET NULL – при удалении или обновлении связанных строк главной таблицы внешнему ключу присваивается пустое значение (NULL);
  + RESTRICT, NO ACTION – отклоняет любое удаление или обновление связанных строк главной таблицы при наличии взаимосвязей;
  + SET DEFAULT X – при удалении или обновлении связанных строк главной таблицы устанавливает значение по умолчанию X.

### Изменение таблиц и столбцов

Для изменения таблицы и столбца используется команда ALTER TABLE:

ALTER TABLE <название таблицы>

Она доступна со следующими модификациями:

* добавление нового столбца:

ADD <название столбца> <тип данных> [<атрибуты столбца>];

* удаление столбца:

DROP COLUMN <название столбца>;

* изменение значения по умолчанию:

ALTER COLUMN <название столбца> SET DEFAULT <значение>;

* изменение типа столбца:

MODIFY COLUMN <название столбца> <тип данных> [<атрибуты столбца>];

* добавление внешнего ключа:

ADD FOREIGN KEY (<название столбца>) REGERENCES <название главной таблицы> (<название столбца главной таблицы>);

* удаление внешнего ключа:

DROP FOREIGN KEY <название столбца>;

* добавление первичного ключа:

ADD PRIMARY KEY (<название столбца>);

* удаление первичного ключа:

DROP PRIMARY KEY;

### Добавление, выборка и удаление данных

#### Добавление данных

Добавление данных в таблицу осуществляется с помощью команды INSERT.

Для добавления заранее определенных значений:

INSERT [INTO] <название таблицы>[(<список столбцов>)]

VALUES (<значение 1>[, <значение 2>, …, <значение N>])[,… (<значение 1>[,<значение 2>, …, <значение N>])];

При перечислении столбцов и соответствующих им значений допустимо (при необходимости) опущение тех столбцов, которым назначены атрибуты DEFAULT или NULL.

#### Выборка данных

Выборка данных из таблицы осуществляется с помощью команды SELECT:

SELECT <список столбцов>

FROM <название таблицы>;

[WHERE <условие>]

[GROUP BY <список столбцов>

[HAVING <условие фильтрации>]]

[ORDER BY <список столбцов>]

[LIMIT [<отступ>,] <количество строк>];

Допустимо вместо списка столбцов указывать символ «\*» при необходимости выборки всех полей. В качестве названия таблицы допустимо использовать как одну таблицу, так и соединения таблиц (см. подраздел X).

#### Удаление данных

Удаление данных из таблицы осуществляется с помощью команды DELETE:

DELETE FROM <название таблицы> [

WHERE <условие>];

При отсутствии условия запрос удалит все данные из таблицы.

### Фильтрация и обновление данных

#### Фильтрация данных

Для фильтрации данных используется команда WHERE с условием:

WHERE <условие>

#### Обновление данных

При обновлении данных используется команда UPDATE в комбинации с SET:

UPDATE <название таблицы>

SET <столбец 1>=<значение 1>[, …, <столбец N>=<значение N>] [

WHERE <условие>];

### Запросы

В запросах при выборке могут использоваться следующие команды:

* DISTINCT <список столбцов> – позволяет сделать выборку уникальных значений по одному или нескольким столбцам;
* ORDER BY <список столбцов> [ASC | DESC] – позволяет отсортировать данные в порядке возрастания или убыванию по одному или нескольким столбцам;
* GROUP BY <список столбцов> – позволяет группировать строки запросы;
* HAVING <условие фильтрации> – позволяет выполнить фильтрацию групп;
* LIMIT [<отступ>,] <количество строк> – позволяет получить определенный диапазон строк. Отступ задает количество пропускаемых в выборке строк.

### Операторы фильтрации

В условиях после названия столбца могут быть применены следующие операторы фильтрации:

* [NOT] IN (<список значений>) – определяет набор значений, в соответствии с которым проверяются (при использовании NOT – не проверяются) значения столбца;
* [NOT] BETWEEN <начальное значение> AND <конечное значение> – определяет диапазон значений, в соответствии с которым проверяются (при использовании NOT – не проверяются) значения столбца. Границы диапазонов входят в диапазон;
* [NOT] LIKE <шаблон строки> – определяет шаблон строки, в соответствии с которым проверяются (при использовании NOT – не проверяются) значения столбца. Столбец должен быть только символьного типа (CHAR, VARCHAR, TEXT). В шаблоне используется следующие сущность:
  + символы – буквы любого алфавита, числа, пробелы и пр.
  + «%» – для обозначения того, что в данном месте шаблона может быть ноль, один или несколько символов;
  + «\_» – для обозначения того, что в данном месте шаблона должен находится один символ;
* IS NULL – осуществляет проверку на то, что значение столбца является пустым.

### Агрегатные функции

В запросах могут использоваться агрегатные функции, т.е. функции, которые возвращают некоторые значения, полученные в ходе вычисления из набора строк:

* AVG (<название столбца>) – вычисляет среднее значение;
* SUM (<название столбца>) – вычисляет сумму всех значений;
* MIN (<название столбца>) – вычисляет минимальное из всех значений;
* MAX (<название столбца>) – вычисляет максимальное из всех значений;
* COUNT (<название столбца>) – вычисляет количество строк в запросе.

### Соединения таблиц

Для соединения таблиц можно воспользоваться различными способами: неявное соединение, внутреннее соединение, внешнее соединение и объединение.

Неявное соединение является самым простым способом соединения таблиц. Такое соединение объединяет все строки первой таблицы со всеми строками второй таблицы, а строки получившейся соединенной таблицы со строками третьей и т.д. Синтаксис:

FROM <таблица 1>, <таблица 2>[, …, <таблица N>]

[WHERE <условие>];

Внутреннее соединение позволяет присоединить все строки первой таблицы ко всем строкам второй таблицы на основе какого-либо обязательного условия. Для внутреннего соединения применяется оператор INNER JOIN:

FROM <таблица 1>

[INNER] JOIN <таблица 2> ON <условие 1>

[…[INNER] JOIN <таблица N> ON <условие N-1>];

Внешнее соединение позволяет сохранить одну из таблиц в неизменном виде и добавить к ней при совпадении определенного условия данные из второй. Для внешнего соединения применяется оператор OUTER JOIN:

FROM <таблица 1>

{LEFT | RIGHT} [OUTER] JOIN <таблица 2> ON <условие 1>

[…{LEFT | RIGHT} [OUTER] JOIN <таблица N> ON <условие N-1>];

Объединение позволяет объединить две выборки в одной, при этом при полном совпадении столбцов две таблицы будут «склеены». Для объединения применяется оператор UNION:

SELECT <выражение 1>

UNION SELECT <выражение 2>

[…UNION SELECT <выражение N>];

### Встроенные функции

Для работы с числовыми данными применяются следующие функции:

* ROUND(<значение>, <точность>) – округление числа с указанной точностью. Если точность указывается отрицательным числом, то используется округление до целой части числа («-1» – округление до десятков, «-2» – округление до сотен и т.д.).
* TRUNCATE(<значение>, <точность>) – оставляет в дробной части числа определенное точностью количество знаков.
* ABS(<значение>) – взятие модуля от числа.
* CEILING(<значение>) – округление до большего.
* FLOOR(<значение>) – округление до меньшего.
* POWER(<значение>, <степень>) – возведение числа в степень.
* SQRT(<значение>) – взятие квадратного корня от числа.
* SIGN(<значение>) – взятие знака от числа («-1» для отрицательного, «1» для положительного, «0» для 0).
* RAND() – генерирует случайное число в диапазоне от 0 до 1 (не включительно).

Для работы с датами и временем применяются следующие функции:

* NOW() – возвращает текущие дату и время;
* CURRENT\_DATE() – возвращает текущую дату;
* CURRENT\_TIME() – возвращает текущее время;
* DATE\_ADD(<дата или дата со временем>, INTERVAL <временной интервал>) – возвращает дату или дату со временем со смещением на указанное количество дней вперед;
* DATE\_SUB(<дата или дата со временем>, INTERVAL <временной интервал>) – возвращает дату или дату со временем со смещением на указанное количество дней назад;
* DATEDIFF(<дата 1>, <дата 2>) – возвращает разницу в количестве дней между двумя датами.

### Работа с файловой системой

Для выгрузки данных из файлов, находящихся на ПК или сервере используется следующий синтаксис:

LOAD DATA

INFILE <название файла>

[REPLACE | IGNORE] INTO TABLE <название таблицы>

[{FIELDS | COLUMNS} TERMINATED BY <символ разделения>]

[LINES TERMINATED BY <символ разделения строк>]

[IGNORE <количество> {LINES | ROWS}];

Оператор INFILE определяет путь к файлу, из которого должна быть осуществлена выгрузка.

Оператор INTO TABLE определяет, в какую таблицу указанные данные необходимо передать. Если использовать модификатор REPLACE, то имеющиеся в таблице данные при наличии первичных ключей или уникальных значений будут заменены в соответствии с первичным ключом или уникальным значением, а при отсутствии таковых будут добавлены, если использовать модификатор IGNORE, то имеющиеся данные сохранятся, а после них будут добавлены данные из файла, в случае если первичные ключи или уникальные не имеются в текущей таблице.

Оператор FIELDS TERMINATED BY или COLUMNS TERMINATED BY определяет символ разделения полей или столбцов в файле.

Оператор LINES TERMINATED BY определяет символ разделения строк в файле.

Оператор IGNORE определяет количество строк, которые необходимо пропустить при чтении файла.

### Индексация

Индексация применяется для облегчения поиска данных, при этом используется какой-либо алгоритм хранения данных с использованием методов хеширования. Для индексации можно использовать любое поле, но лучше всего использовать те, которые имеют размер больший, чем остальные.

По умолчанию в MySQL 8.0 для полей, создаваемых с модификатором PRIMARY KEY, автоматически создаются индексы, благодаря чему соединения таблиц по первичным и соответствующим им внешним ключам ведутся быстрее.

Для того чтобы самостоятельно создать индекс для поля, можно воспользоваться следующим синтаксисом:

CREATE INDEX <название индекса>

ON <название таблицы> (<название поля 1>[, <другие поля>])[

USING <метод хеширования>];

Отличие составного индекса от нескольких простых: несколько простых не ускоряют, а даже понижают скорость поиска информации за счет того, что только по первому простому индексу ведется хеширование и поиск в хеш-таблице, а по остальным ведется обычный линейный поиск, в то время как по составному индексу ведется полное хеширование и поиск осуществляется с помощью хеш-таблицы составных склеенных данных. Таким образом, поиск по одному составному индексу осуществляется быстрее, чем по нескольким простым.

## Знакомство с MySQL Workbench

Установка MySQL Workbench произведена согласно инструкции преподавателя.

MySQL Workbench предоставляет доступ ко всем схемам БД, подтянутым к созданному аккаунту с помощью скрипта, написанного на языке SQL, к созданию собственных запросов, а также к созданию моделей, отображающих структуру БД, на основе схем БД.

## Знакомство с разработкой ТЗ для АИС согласно ГОСТ

Разработка технического задания (ТЗ) для любой автоматизированной информационной системы (АИС) должна быть проведена согласно ГОСТ 34.602-2020 [1].

Структура ТЗ представлена в Таблице Таблица 1.

Таблица – Структура ТЗ

| **Раздел** | **Подразделы** | **Описание** |
| --- | --- | --- |
| Общие сведения | – | Содержит полное наименование и условное обозначение АС, а также наименование организации-заказчика АС, перечень документов, на основании которого разрабатывается АС и плановые сроки начала и окончания ведения работ по разработке АС |
| Цели и назначение создания АС | Назначение АС | Описывает назначение АС |
| Цели создания АС | Описывает цели создания АС |
| Характеристика объекта автоматизации | – | Описывает объект автоматизации, определяет смежные проекты, а также документы для разработки АС |
| Требования к АС | Функциональные требования | Определяет требования, непосредственно влияющие на реализацию АС, включая: коэффициент полезной эксплуатации (КПЭ), бизнес-требования, функциональные требования, функциональную модель бизнес-решения, требования к интеграции со смежными системами, диаграмму или таблицу информационных потоков, логическую модель (концептуальное, логическое и физическое проектирование), требования к функциональным ролям, подразделения пользователей бизнес-систем, требования к сценариям работы бизнес систем |
| Нефункциональные требования | Определяет требования, непосредственно НЕ влияющие на реализацию АС, включая: требования к инфраструктуре и каналам связи, требования к целевой технологической архитектуре проектного решения, требования к характеристикам каналов связи, требования к отказоустойчивости, требования к диагностированию системы, требования по модернизации и развитию системы, требования к численности и квалификации персонала, требования к надежности, требования к резервному копированию, требования по доступности и производительности, требования к масштабированию, требованию к режиму функционирования, требования к внешней среде, требования к конфиденциальности, требования к эргономике и технической эстетике, требования к технической поддержке, требования к лицензированию |
| Требования к безопасности | Определяет требования, влияющие непосредственно на безопасность пользования АС, включая: решения по идентификации и аутентификации, решения по управлению доступом, регистрация и учет событий безопасности, контроль целостности, межсетевое экранирование, резервное копирование, антивирусная защита, контроль защищенности |
| Состав и содержание работ по созданию АС | – | Определяет необходимые работы по разработке АС и сроки их выполнения |
| Порядок разработки АС | – | Определяет порядок организации разработки АС |
| Порядок контроля и приемки АС | – | Определяет порядок проверки разработанной АС на соответствие всем определенным в ТЗ требованиям |
| Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу АС в действие | – | Определяет требования, согласно которым объект автоматизации подготавливается к введению разработанной АС в эксплуатацию |
| Требования к документированию | – | Определяет перечень документов, необходимых для передачи заказчику |
| Источники разработки | – | Перечень документов, согласно которым разработано ТЗ и АС. |

## Общие сведения

Объектом автоматизации в данной курсовой работе является деятельность некоторой проектной организации. Проектные организации занимаются разработкой планов зданий и сооружений, а также их строительством.

Для создания АИС используется множество сведений:

* перечень проектов организации,
* перечень направлений в пределах проектов, определяющий этапы разработки проекта,
* перечень процессов в пределах направлений, определяющий задачи направления,
* перечень сотрудников, организаций и самозанятых, задействованных в реализации проектов,
* календарь выполняемых работ сотрудников, организаций и самозанятых,
* перечень финансовых перечислений проектам с указанием цели финансирования,
* перечень документов, согласно которым разрабатываются проекты.

С помощью разрабатываемой АИС существует возможность автоматизировать следующие задачи:

* выборка из графика работ всех субъектов разработки;
* уточнение данных по финансированию проектов;
* уточнение данных о состоянии того или иного процесса, направления или проекта.

# Практическая часть

В данном разделе представлен исходный код для воспроизведения, заполнения и тестирования разрабатываемой БД. Все файлы проверены на отсутствие синтаксических ошибок, т.е. ошибок запуска скриптов, а также на отсутствие семантических ошибок, т.е. ошибок, приводящих к некорректной реализации скриптов.

## Файл воспроизведения БД

Воспроизведение БД доступно SQL-скриптом в файле Initialization.sql директории Scripts. Исходный код программы:

-- DROP DATABASE project\_organization;

CREATE DATABASE project\_organization;

use project\_organization;

-- reference depth 0

-- Типы документов

CREATE TABLE doc\_types (

id\_doc\_type INT PRIMARY KEY NOT NULL CHECK (id\_doc\_type > 0),

doc\_type\_name TEXT NOT NULL

);

-- Статусы готовности процессов / направлений / проектов

CREATE TABLE statuses (

id\_status INT PRIMARY KEY NOT NULL CHECK (id\_status > 0),

status\_name TEXT NOT NULL

);

-- Цели финансирования

CREATE TABLE fin\_targets (

id\_fin\_target INT PRIMARY KEY NOT NULL CHECK (id\_fin\_target > 0),

fin\_target TEXT NOT NULL

);

-- Типы сотрудников / субподрядчиков

CREATE TABLE employee\_types (

id\_employee\_type INT PRIMARY KEY,

employee\_type\_name TEXT

);

-- reference depth 1

-- Документы

CREATE TABLE docs (

id\_doc INT PRIMARY KEY NOT NULL CHECK (id\_doc > 0),

doc\_name TEXT NOT NULL,

doc\_revision INT DEFAULT 0,

doc\_approval\_date DATE NOT NULL CHECK (doc\_approval\_date > "1900-01-01"),

doc\_type\_id INT,

doc\_reference TEXT,

FOREIGN KEY (doc\_type\_id) REFERENCES doc\_types (id\_doc\_type) ON DELETE SET NULL ON UPDATE CASCADE

);

-- Проекты

CREATE TABLE projects (

id\_project INT PRIMARY KEY NOT NULL CHECK (id\_project > 0),

project\_name TEXT NOT NULL,

status\_proj\_id INT,

plan\_start\_proj\_date DATE NOT NULL CHECK (plan\_start\_proj\_date > "1900-01-01"),

plan\_end\_proj\_date DATE NOT NULL CHECK (plan\_end\_proj\_date > "1900-01-01"),

commentary TEXT,

FOREIGN KEY (status\_proj\_id) REFERENCES statuses (id\_status) ON DELETE SET NULL ON UPDATE CASCADE

);

-- Финансы

CREATE TABLE finances (

fin\_sum DECIMAL(15, 3) NOT NULL,

project\_id INT,

fin\_target\_id INT,

commentary TEXT,

FOREIGN KEY (project\_id) REFERENCES projects (id\_project) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY (fin\_target\_id) REFERENCES fin\_targets (id\_fin\_target) ON DELETE SET NULL ON UPDATE CASCADE

);

-- Сотрудники / субподрядчики

CREATE TABLE employees (

id\_employee INT PRIMARY KEY,

employee\_name TEXT NOT NULL,

employee\_type\_id INT,

FOREIGN KEY (employee\_type\_id) REFERENCES employee\_types (id\_employee\_type) ON DELETE SET NULL ON UPDATE CASCADE

);

-- reference depth 2

-- Связь "Проекты-Документы"

CREATE TABLE projects\_docs (

project\_id INT,

doc\_id INT,

FOREIGN KEY (project\_id) REFERENCES projects (id\_project) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY (doc\_id) REFERENCES docs (id\_doc) ON DELETE SET NULL ON UPDATE CASCADE

);

-- Направления

CREATE TABLE works (

id\_work INT PRIMARY KEY NOT NULL CHECK (id\_work > 0),

work\_name TEXT NOT NULL,

project\_id INT,

status\_work\_id INT,

plan\_start\_work\_date DATE NOT NULL CHECK (plan\_start\_work\_date > "1900-01-01"),

plan\_end\_work\_date DATE NOT NULL CHECK (plan\_end\_work\_date > "1900-01-01"),

commentary TEXT,

FOREIGN KEY (project\_id) REFERENCES projects (id\_project) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY (status\_work\_id) REFERENCES statuses (id\_status) ON DELETE SET NULL ON UPDATE CASCADE

);

-- Процессы

CREATE TABLE processes (

id\_process INT PRIMARY KEY NOT NULL CHECK (id\_process > 0),

process\_name TEXT NOT NULL,

status\_proc\_id INT,

plan\_start\_proc\_datetime DATETIME NOT NULL CHECK (plan\_start\_proc\_datetime > "1900-01-01 00:00:00"),

plan\_end\_proc\_datetime DATETIME NOT NULL CHECK (plan\_end\_proc\_datetime > "1900-01-01 00:00:00"),

commentary TEXT,

FOREIGN KEY (status\_proc\_id) REFERENCES statuses (id\_status) ON DELETE SET NULL ON UPDATE CASCADE

);

-- reference depth 3

-- График частей процессов по каждому направлению

CREATE TABLE proc\_scheduler (

id\_proc\_schedule INT PRIMARY KEY NOT NULL CHECK (id\_proc\_schedule > 0),

work\_id INT,

process\_id INT,

part\_proc\_date DATETIME NOT NULL CHECK (part\_proc\_date > "1900-01-01 00:00:00"),

FOREIGN KEY (work\_id) REFERENCES works (id\_work) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY (process\_id) REFERENCES processes (id\_process) ON DELETE SET NULL ON UPDATE CASCADE

);

-- reference depth 4

-- График работы сотрудников / субподрядчиков

CREATE TABLE employee\_scheduler (

proc\_schedule\_id INT,

employee\_id INT,

FOREIGN KEY (proc\_schedule\_id) REFERENCES proc\_scheduler (id\_proc\_schedule) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY (employee\_id) REFERENCES employees (id\_employee) ON DELETE SET NULL ON UPDATE CASCADE

);

## Файл заполнения тестовыми данными

Заполнение тестовыми данными происходит с помощью запросов в файле Initialization\_Filling.sql директории Scripts. Для работы скрипта необходимо скопировать в директорию целевого ПК

c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/

файлы из директории Examples. Данное требование обусловлено нестабильной работой MySQL Workbench версии 8.0 в части работы с файловой системой Windows 10. Исходный код программы:

-- Удаление всех данных из всех таблиц

DELETE FROM employee\_scheduler;

DELETE FROM proc\_scheduler;

DELETE FROM processes;

DELETE FROM works;

DELETE FROM projects\_docs;

DELETE FROM employees;

DELETE FROM finances;

DELETE FROM projects;

DELETE FROM docs;

DELETE FROM employee\_types;

DELETE FROM fin\_targets;

DELETE FROM statuses;

DELETE FROM doc\_types;

-- Заполнение таблиц 0 depth

-- Заполнение таблицы doc\_types

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/doc\_types\_table.txt"

REPLACE INTO TABLE doc\_types

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблицы statuses

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/statuses\_table.txt"

REPLACE INTO TABLE statuses

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблицы fin\_targets

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/fin\_targets\_table.txt"

REPLACE INTO TABLE fin\_targets

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблицы employee\_types

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/employee\_types\_table.txt"

REPLACE INTO TABLE employee\_types

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблиц 1 depth

-- Заполнение таблицы docs

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/docs\_table.txt"

REPLACE INTO TABLE docs

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблицы projects

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/projects\_table.txt"

REPLACE INTO TABLE projects

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблицы finances

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/finances\_table.txt"

REPLACE INTO TABLE finances

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблицы employees

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/employees\_table.txt"

REPLACE INTO TABLE employees

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблиц 2 depth

-- Заполнение таблицы projects\_docs

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/projects\_docs\_table.txt"

REPLACE INTO TABLE projects\_docs

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблицы works

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/works\_table.txt"

REPLACE INTO TABLE works

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблицы processes

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/processes\_table.txt"

REPLACE INTO TABLE processes

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблиц 3 depth

-- Заполнение таблицы proc\_scheduler

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/proc\_scheduler\_table.txt"

REPLACE INTO TABLE proc\_scheduler

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

-- Заполнение таблиц 4 depth

-- Заполнение таблицы employee\_scheduler

LOAD DATA INFILE "c:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EASPO/employee\_scheduler\_table.txt"

REPLACE INTO TABLE employee\_scheduler

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r\n';

## Файл тестирования ввода данных в БД

Для тестирования ввода данных в БД создан файл Adding\_Requests.sql, находящийся в директории Scripts. Для тестирования выбраны два случая добавления данных:

* Добавление данных в таблицу без зависимости от внешних ключей;
* Добавление данных в таблицу с зависимостью от новых данных внешних ключей.

Исходный код программы:

-- добавление данных

-- 1) в таблицу выполненных процессов (processes)

INSERT INTO processes

SELECT count(\*) + 1, 'Соединение блоков', 4, '2022-12-21', '2023-02-07', NULL

FROM processes;

SELECT \* FROM processes;

-- 2) в таблицу проектов с новым статусом "in planning"

INSERT INTO statuses

SELECT count(\*) + 1, 'in planning'

FROM statuses;

SELECT \* FROM statuses;

INSERT INTO projects

SELECT count(\*) + 1, 'Ферма 1', 5, '2023-07-03', '2023-09-10', NULL

FROM projects;

SELECT \* FROM projects;

SELECT \*

FROM projects

LEFT JOIN statuses ON projects.status\_proj\_id = statuses.id\_status;

-- DELETE FROM processes

-- WHERE id\_process=10;

-- DELETE FROM projects

-- WHERE id\_project=4;

-- DELETE FROM statuses

-- WHERE id\_status=5;

## Файл тестирования запросов выборки данных

Для тестирования запросов выборки данных из БД создан файл Data\_Requests.sql, находящийся в директории Scripts. Для тестирования выбраны два случая:

* сводка данных о сотрудниках с указанием общего количества смен,
* сводка данных о соответствии документов и проектов,
* сводка данных о плановых и фактических сроках выполнения работ.

Исходный код программы:

-- запросы, которые могут быть выполнены

-- 1) Сводка данных о сотрудниках: общее количество смен по каждому работавшему

SELECT employees.employee\_name as 'Сотрудник / организация',

count(proc\_schedule\_id) as 'Количество смен'

FROM employee\_scheduler

LEFT JOIN employees ON employees.id\_employee = employee\_scheduler.employee\_id

GROUP BY employee\_id

ORDER BY 1 ASC;

-- 2) Сводка данных о соответствии документов проектам

SELECT project\_name as 'Проект',

doc\_name as 'Документ'

FROM docs

RIGHT JOIN projects\_docs ON docs.id\_doc = projects\_docs.doc\_id

LEFT JOIN projects ON projects\_docs.project\_id = projects.id\_project

ORDER BY 1 ASC;

-- 3) Сводка данных о плановых и фактических сроках выполнения работ

-- 3.a) с перечислением процессов

SELECT work\_name as 'Направление',

plan\_start\_work\_date as 'Дата начала (по плану)',

plan\_end\_work\_date as 'Дата окончания (по плану)',

max(part\_proc\_date) as 'Дата выполнения',

process\_name as 'Процесс'

FROM works

LEFT JOIN statuses ON works.status\_work\_id = statuses.id\_status

RIGHT JOIN proc\_scheduler ON works.id\_work = proc\_scheduler.work\_id

LEFT JOIN processes ON proc\_scheduler.process\_id = processes.id\_process

WHERE statuses.status\_name = 'verified' or statuses.status\_name = 'finished'

GROUP BY work\_name, plan\_start\_work\_date, plan\_end\_work\_date, process\_name

ORDER BY 4 ASC;

-- 3.b) без перечисления процессов, только выполненные работы

SELECT work\_name as 'Направление',

plan\_start\_work\_date as 'Дата начала (по плану)',

plan\_end\_work\_date as 'Дата окончания (по плану)',

max(part\_proc\_date) as 'Дата выполнения'

FROM works

LEFT JOIN statuses ON works.status\_work\_id = statuses.id\_status

RIGHT JOIN proc\_scheduler ON works.id\_work = proc\_scheduler.work\_id

LEFT JOIN processes ON proc\_scheduler.process\_id = processes.id\_process

WHERE statuses.status\_name = 'verified' or statuses.status\_name = 'finished'

GROUP BY work\_name, plan\_start\_work\_date, plan\_end\_work\_date

ORDER BY 4 ASC;

# Техническое задание

## Общие сведения

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| Полное наименование | Единая автоматизированная система проектной организации |
| Условное обозначение | ЕАСПО |
| Наименование организации-заказчика АС | МАИ, Институт №7, кафедра 703 |
| Перечень документов, на основании которых разрабатывается АС | Учебный план кафедры 703 курса «Методы обработки информации в АСУ» |
| Плановые сроки начала и окончания работ | С 09.02.2023 по 20.06.2023 |

## Цели и назначение создания АС

### Назначение АС

ЕАСПО предназначена для повышения надежности хранения и обработки информации. Основным назначением АС является:

1. Хранение и обработка информации для отчетов;
2. Анализ деятельности предприятия и заказчика.

### Цели создания АС

Информационная система создается с целью:

1. Автоматизации контроля строительства объектов и выпуска документации;
2. Сбора данных для разработки отчетности;
3. Повышения качества (полноты, точности, достоверности, своевременности, согласованности) информации;
4. Облегчения работы с информацией;
5. Опытно-промышленной эксплуатации.

## Характеристика объекта автоматизации

Объектом автоматизации является проектная организация, занимающаяся проектированием и строительством объектов недвижимости.

ЕАСПО разрабатывается согласно следующим документам:

* стандарт на кодирование инструментов проектной организации.

Взаимодействие со смежными проектами не предусматривается.

## Требования к АС

### Функциональные требования

#### КПЭ

|  |  |  |  |
| --- | --- | --- | --- |
| **Цель** | **Наименование КПЭ** | **Значение КПЭ**  **(текущее)** | **Критерии достижения** |
| Доступ заинтересованных лиц к корректировке календаря выполняемых работ | * Развертывание сервиса * Организация доступа | Сервис не развернут | Сервис развернут |
| Доступ заинтересованных лиц к документации | Организация доступа | Доступ не организован | Доступ организован |
| Доступ заинтересованных лиц к выявленным отклонениям в проекте | Организация доступа | Доступ не организован | Доступ организован |
| Составление графика работ рабочих | * Развертывание сервиса * Организация доступа | Сервис не развернут | Сервис развернут |
| Доступ заинтересованных лиц к просмотру календаря выполняемых работ | Импорт данных | Процесс не автоматизирован | Процесс автоматизирован |

#### Бизнес-требования

| **№ БТ** | **Описание БТ** |
| --- | --- |
| БТ-1 | ЕАСПО должна предоставлять данные для генерации отчета по выполненным работам: состав и вид работ, плановые и фактические сроки выполнения работ, сводку данных о сотрудниках, задействованных в процессах. |
| БТ-2 | ЕАСПО должна обеспечивать ввод данных выполненных по календарю работ. |
| БТ-3 | ЕАСПО должна обеспечивать ввод данных о документации. |
| БТ-4 | ЕАСПО должна позволять хранить данные в течение как минимум 5 лет с момента их внесения. |
| БТ-5 | ЕАСПО должна хранить историю изменений планов: добавление / изменение / удаление процессов, дату внесения изменения в планы. |
| БТ-6 | ЕАСПО должна хранить данные о календаре выполненных и запланированных работ и графике сотрудников. |
| БТ-7 | ЕАСПО должна предоставлять GUI-приложение с возможностью авторизации пользователей и распределения их прав доступа к базе. |
| БТ-8 | ЕАСПО должна хранить следующие данные о пользователях: логин и пароль для входа, фамилия и имя пользователя, корпоративный e-mail пользователя, дата последнего входа в систему. |
| БТ-9 | ЕАСПО должна предоставлять возможность авторизации пользователей и распределения их прав доступа к базе его с помощью браузера Microsoft Edge. |

#### Функциональные требования

|  |  |  |  |
| --- | --- | --- | --- |
| **№ ФТ** | **Наименование функции** | **Описание** | **Бизнес-требование** |
| ФТ-1 | Требования к функциям ввода информации | Ввод данных об изменениях в графиках работ осуществляется оператором в ручном режиме. | БТ-2, БТ-3 |
| ФТ-2 | Требования к функциям хранения информации | * Состав хранимой информации * Время хранения информации | БТ-4, БТ-5, БТ-6, БТ-8 |
| ФТ-3 | Требования к функциям обработки информации | Методы обработки данных | БТ-1 |
| ФТ-4 | Требования к визуализации | Определить качественные и количественные требования | БТ-7, БТ-9 |

#### Функциональная модель бизнес-решения

Функциональная модель бизнес-решения представляется следующим образом:

* на вход оператором в ручном режиме подаются данные об изменениях в графиках работ: добавление / изменение / удаление процессов, изменение данных о составе, количестве и графике рабочих, задействованных в том или ином процессе;
* вносимые в базу данные могут храниться в течение как минимум 5 лет с момента их внесения;
* ЕАСПО обеспечивает генерацию отчетов по выполненным проектной организацией работам, включая сравнение сроков плановых и фактических работ, сводку данных сотрудниках, задействованных в процессах, общее количество часов наработки каждого из них;

#### Требования к интеграции со смежными системами

Информация о смежных системах отсутствует.

#### Диаграмма (таблица) информационных потоков

Информационные потоки со смежными системами отсутствуют.

#### Логическая модель

1. **Концептуальное проектирование**
   1. **Определение сущностей и их документирование**

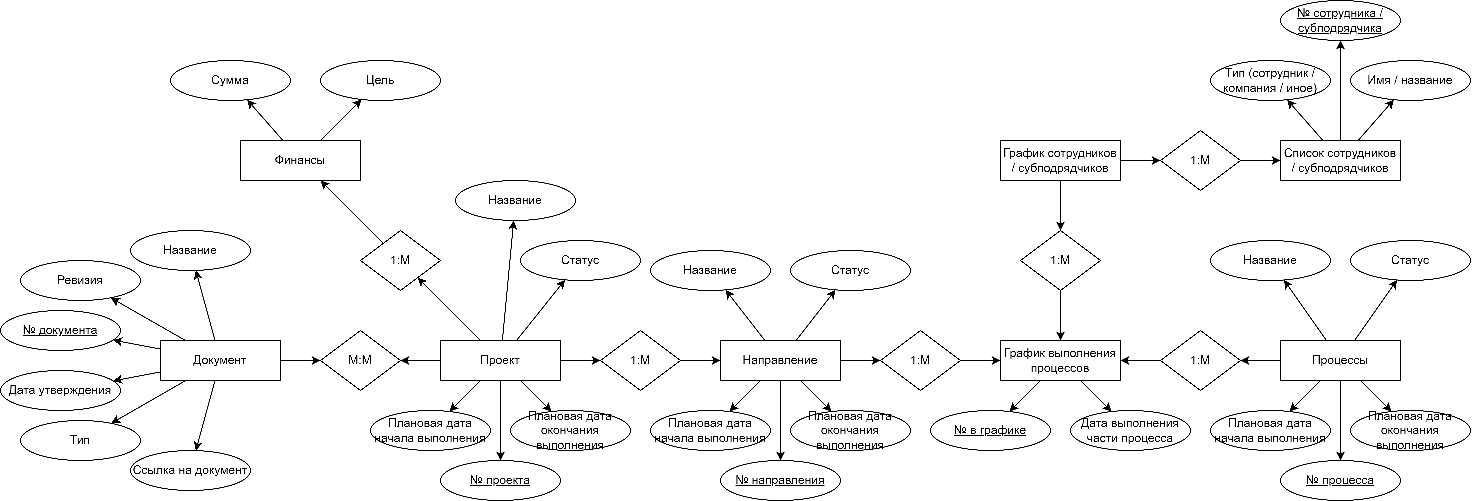
В ЕАСПО должны присутствовать следующие сущности:

* График выполнения процессов – календарь выполнения процесса, включающий в себя разделение процесса на смены.
* График сотрудников / субподрядчиков – календарь выхода сотрудников / субподрядчиков для исполнения рабочих обязанностей по какой-либо части процесса. Реализует связь «Многие ко многим» для сущностей «Список сотрудников / субподрядчиков» и «График выполнения процессов».
* Документ – стандарт (ГОСТ, ОСТ, СТО и пр.); файл документации, относящийся к конкретному проекту; файл документации, являющийся вспомогательным при выполнении проекта, но не являющийся проектным.
* Направление – обобщенное название комплекса процессов, определяющее цель, которую необходимо достичь по окончанию выполнения всех процессов.
* Проект – обобщенное название комплекса направлений работ, определяющее цель, которую необходимо достичь по окончанию выполнения всех направлений работ.
* Процесс – вид деятельности, направленный на достижение некоторой малой цели направления.
* Список сотрудников / субподрядчиков – список субъектов процессов, представляемых либо именем физического лица, являющегося сотрудником проектной организации, либо названием юридического лица, являющегося субподрядчиком проектной организации.
* Финансы – все денежные средства, направленные на компенсацию затрат на выполнение проекта или его части.
  1. **Определение связей между сущностями и их документирование**

Связь между сущностями определена в таблице ниже. Условные обозначения: «1:М» – связь «Один ко многим», «М:М» – связь «Многие ко многим».

|  |  |  |
| --- | --- | --- |
| **Сущность 1** | **Тип связи** | **Сущность 2** |
| График выполнения процессов | 1:М | График сотрудников / субподрядчиков |
| Процесс | 1:М | График выполнения процессов |
| Направление | 1:М | График выполнения процессов |
| Проект | 1:М | Направление |
| Документ | М:М | Проект |
| Проект | 1:М | Финансы |
| Список сотрудников / субподрядчиков | 1:М | График сотрудников / субподрядчиков |

* 1. **Создание ER-модели предметной области**



* 1. **Определение атрибутов и их документирование**

| **Сущность** | **Атрибут** | **Описание** | **Тип** | **NN** | **Сост.** | **Расч.** |
| --- | --- | --- | --- | --- | --- | --- |
| Документ | № документа | Идентификатор документа | Целое число | Да | Нет | Нет |
| Название | Название документа | Текст | Да | Нет | Нет |
| Ревизия | Ревизия документа | Целое число | Нет | Нет | Нет |
| Дата утверждения | Дата утверждения документа | Дата | Да | Нет | Нет |
| Тип | Тип документа | Целое число | Нет | Нет | Нет |
| Ссылка на документ | Ссылка на документ в систему контроля версий | Текст | Нет | Нет | Нет |
| Проект | № проекта | Идентификатор проекта | Целое число | Да | Нет | Нет |
| Название | Название проекта | Текст | Да | Нет | Нет |
| Статус | Статус выполнения проекта | Целое число | Нет | Нет | Нет |
| Плановая дата начала выполнения | Дата начала выполнения проекта (по плану) | Дата | Да | Нет | Нет |
| Плановая дата окончания выполнения | Дата окончания выполнения проекта (по плану) | Дата | Да | Нет | Нет |
| Направление | № направления | Идентификатор направления | Целое число | Да | Нет | Нет |
| Название | Название направления работ, проводимых в рамках проекта | Текст | Да | Нет | Нет |
| Статус | Статус выполнения работ в направлении | Целое число | Нет | Нет | Нет |
| Плановая дата начала выполнения | Дата начала выполнения работ в направлении (по плану) | Дата | Да | Нет | Нет |
| Плановая дата окончания выполнения | Дата окончания выполнения работ в направлении (по плану) | Дата | Да | Нет | Нет |
| График выполнения процессов | № в графике | Идентификатор записи в графике выполнения процесса | Целое число | Да | Нет | Нет |
| Дата выполнения части процесса | Дата, когда осуществляется выполнение процесса или его части (в случае, если процесс рассчитан на выполнение в течение не одного дня) | Дата со временем | Да | Нет | Нет |
| График сотрудников / субподрядчиков | – | – | – | – | – | – |
| Список сотрудников / субподрядчиков | № сотрудника / субподрядчика | Идентификатор субъекта процесса | Целое число | Да | Нет | Нет |
| Тип (сотрудник / компания / иное) | Тип субъекта процесса | Целое число | Нет | Нет | Нет |
| Имя / название | Имя (физическое лицо) или название (юридическое лицо) субъекта процесса | Текст | Да | Нет | Нет |
| Процесс | № процесса | Идентификатор процесса | Целое число | Да | Нет | Нет |
| Название | Название процесса | Текст | Да | Нет | Нет |
| Статус | Статус выполнения процесса | Целое число | Нет | Нет | Нет |
| Плановая дата начала выполнения | Дата начала выполнения процесса (по плану) | Дата со временем | Да | Нет | Нет |
| Плановая дата окончания выполнения | Дата окончания выполнения процесса (по плану) | Дата со временем | Да | Нет | Нет |
| Финансы | Сумма | Сумма переведенного транша | Вещественное число | Да | Нет | Нет |
| Цель | Цель финансирования | Целое число | Да | Нет | Нет |

* 1. **Определение допустимых значений атрибутов и их документирование**

Условные обозначения:

* «**NN**» – not null, непустое значение;
* «**> X**» – больше значения X.

| **Сущность** | **Атрибут** | **Допустимые значения** |
| --- | --- | --- |
| Документ | № документа | NN |
| Название | NN |
| Ревизия | По умолчанию 0 |
| Дата утверждения | NN, > “1900-01-01” |
| Тип | NN |
| Ссылка на документ | – |
| Проект | № проекта | NN |
| Название | NN |
| Статус | – |
| Плановая дата начала выполнения | NN, > “1900-01-01” |
| Плановая дата окончания выполнения | NN, > “1900-01-01” |
| Направление | № направления | NN |
| Название | NN |
| Статус | – |
| Плановая дата начала выполнения | NN, > “1900-01-01” |
| Плановая дата окончания выполнения | NN, > “1900-01-01” |
| График выполнения процессов | № в графике | NN |
| Дата выполнения части процесса | NN, > “1900-01-01 00:00:00” |
| График сотрудников / субподрядчиков | – | – |
| Список сотрудников / субподрядчиков | № сотрудника / субподрядчика | NN |
| Тип (сотрудник / компания / иное) | – |
| Имя / название | NN |
| Процесс | № процесса | NN |
| Название | NN |
| Статус | – |
| Плановая дата начала выполнения | NN, > “1900-01-01 00:00:00” |
| Плановая дата окончания выполнения | NN, > “1900-01-01 00:00:00” |
| Финансы | Сумма | NN |
| Цель | NN |

* 1. **Определение первичных ключей для сущностей и их документирование**

| **Сущность** | **Атрибут** | **Тип** | **Ключ** |
| --- | --- | --- | --- |
| Документ | № документа | INT | PK |
| Тип | INT | FK |
| Проект | № проекта | INT | PK |
| Статус | INT | FK |
| Направление | № направления | INT | PK |
| Статус | INT | FK |
| График выполнения процессов | № в графике | INT | PK |
| График сотрудников / субподрядчиков | – | – | – |
| Список сотрудников / субподрядчиков | № сотрудника / субподрядчика | INT | PK |
| Тип (сотрудник / компания / иное) | INT | FK |
| Процесс | № процесса | INT | PK |
| Статус | INT | FK |
| Финансы | Цель | INT | FK |

1. **Логическое проектирование**
   1. **Определение набора таблиц исходя из ER-модели и их документирование**

| **Сущность** | **Таблица** | **Атрибут** | **Поле** |
| --- | --- | --- | --- |
| Документ | Docs | № документа | Id\_doc |
| Название | Doc\_name |
| Ревизия | Doc\_revision |
| Дата утверждения | Doc\_approval\_date |
| Тип | Doc\_type\_id |
| Ссылка на документ | Doc\_reference |
| Проект | Projects | № проекта | Id\_project |
| Название | Project\_name |
| Статус | Status\_proj\_id |
| Плановая дата начала выполнения | Plan\_start\_proj\_date |
| Плановая дата окончания выполнения | Plan\_end\_proj\_date |
| Комментарий | Commentary |
| Направление | Works | № направления | Id\_work |
| Название | Work\_name |
| Идентификатор проекта | Project\_id |
| Статус | Status\_work\_id |
| Плановая дата начала выполнения | Plan\_start\_work\_date |
| Плановая дата окончания выполнения | Plan\_end\_work\_date |
| График выполнения процессов | Proc\_scheduler | № в графике | Id\_proc\_schedule |
| Идентификатор направления | Work\_id |
| Идентификатор процесса | Process\_id |
| Дата выполнения части процесса | Part\_proc\_date |
| График сотрудников / субподрядчиков | Employee\_scheduler | Идентификатор в графике выполнения процессов | Proc\_schedule\_id |
| Идентификатор сотрудника | Employee\_id |
| Список сотрудников / субподрядчиков | Employees | № сотрудника / субподрядчика | Id\_employee |
| Имя / название | Employee\_name |
| Тип (сотрудник / компания / иное) | Employee\_type\_id |
| Процесс | Processes | № процесса | Id\_process |
| Название | Process\_name |
| Статус | Status\_proc\_id |
| Плановая дата начала выполнения | Plan\_start\_proc\_datetime |
| Плановая дата окончания выполнения | Plan\_end\_proc\_datetime |
| Комментарий | Commentary |
| Финансы | Finances | Сумма | Fin\_sum |
| Идентификатор проекта | Project\_id |
| Цель | Fin\_target\_id |
| Комментарий | Commentary |
| – | Doc\_types | Идентификатор типа документа | Id\_doc\_type |
| Название типа документа | Doc\_type\_name |
| – | Employee\_types | Идентификатор типа субъекта | Id\_employee\_type |
| Название типа субъекта | Employee\_type\_name |
| – | Fin\_targets | Идентификатор цели финансирования | Id\_fin\_target |
| Название цели финансирования | Fin\_target |
| – | Projects\_docs | Идентификатор проекта | Project\_id |
| Идентификатор документа | Doc\_id |
| – | Statuses | Идентификатор статуса | Id\_status |
| Название статуса | Status\_name |

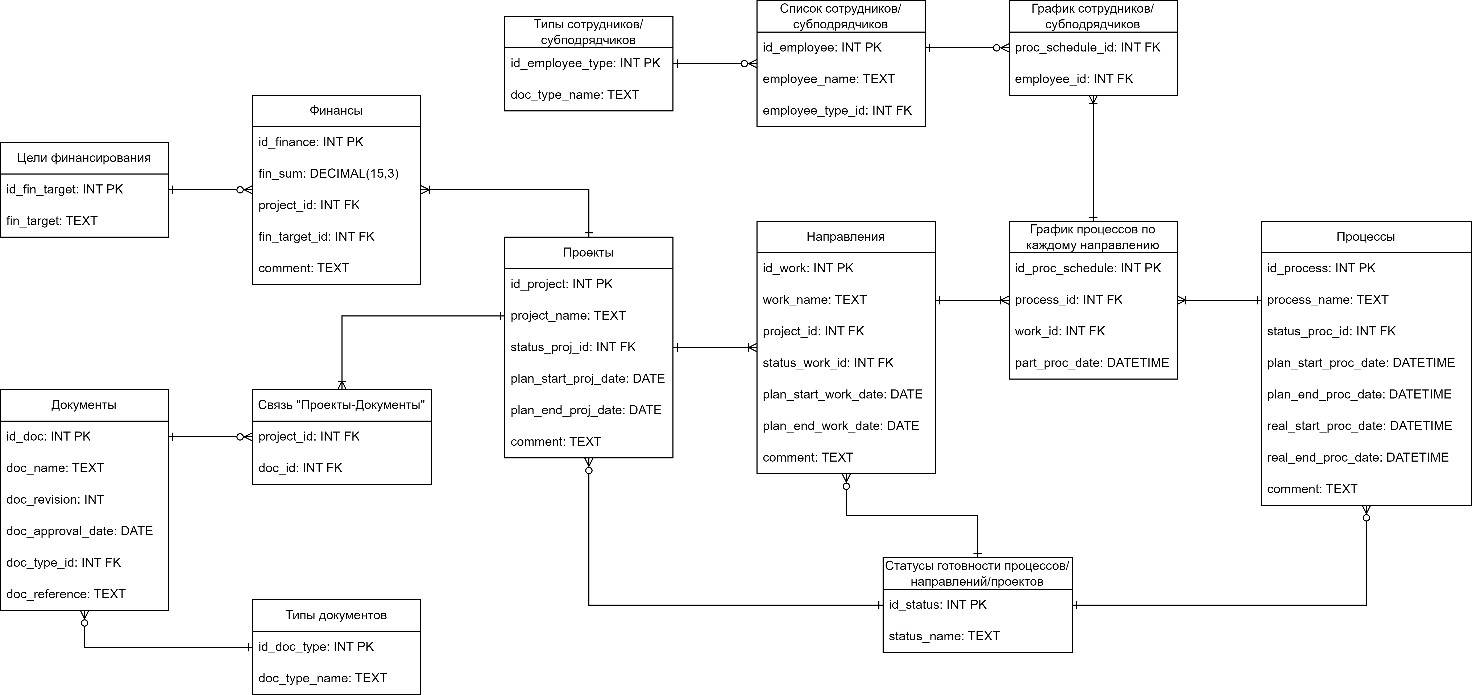
* 1. **Проверка логической модели данных на предмет возможности выполнения всех транзакций, предусмотренных пользователями**

Транзакция – это набор действий, выполняемый отдельным пользователем или прикладной программой с целью изменения содержимого базы данных. В рассматриваемой БД добавление данных должно происходить по следующей схеме (приоритет внесения новых данных выше у первого пункта):

1. Если необходимо, добавить данные о типах документов (doc\_types), статусах (statuses), целях финансирования (fin\_targets) или типах субъектов процессов (employee\_types).
2. Если необходимо, добавить данные о документах (docs), проектах (projects), финансировании (finances) или субъектах процессов (employees).
3. Если необходимо, добавить данные о направлениях (works), процессах (processes) или связях между проектами и документами (projects\_docs).
4. Если необходимо, добавить данные о графике выполнения процессов (proc\_scheduler).
5. Если необходимо, добавить данные о графике сотрудников / субподрядчиков (employee\_scheduler).
   1. **Определение требований поддержки целостности данных и их документирование**

Поля с префиксом «id» (первичные ключи) должны быть связаны с полями с постфиксами «id» (внешние ключи). Соответствия первичных и внешних ключей можно проследить в таблицах выше.

* 1. **Создание окончательного варианта логической модели данных**



1. **Физическое проектирование**
   1. **Проектирование таблиц базы данных средствами выбранной СУБД**

Скрипт на языке SQL для создания описанной БД представлен в файле Scripts/Initialization.sql.

* 1. **Реализация бизнес-требований в среде выбранной СУБД**

Запросы в спроектированную БД необходимо разрабатывать с учетом особенностей СУБД MySQL.

#### Требования к функциональным ролям

|  |  |
| --- | --- |
| **Функциональные роли** | **Основные функции роли** |
| Директор проектной организации | Контролирует выполнение и статус всех проектов, назначает на каждый проект руководителя и его команду. |
| Руководитель проекта | Корректирует график работ сотрудников, отмечает статус выполнения процессов, работ и проекта, координирует несколько процессов в течение одного направления и несколько направлений в течение проекта, добавляет документацию по ее выпуску, обновляет документацию при необходимости. |
| Руководитель направления | Корректирует график работ сотрудников, отмечает статус выполнения процессов и своего направления. |
| Руководитель процесса | Корректирует график работ сотрудников, отмечает статус выполнения своего процесса. |
| Сотрудник/субподрядчик | Просматривает график на наличие смен и вовлечение в процессы, просматривает документацию для выполнения процессов согласно планам/стандартам и прочим документам. |
| Заказчик | Следит за статусом всех проектов, которые финансирует, а также за статусами их направлений и процессов. |
| Администратор ИС | Имеет доступ к управлению и администрированию АС, определяет роли для каждого из участников АС. |
| Администратор ИБ | Имеет доступ к управлению и администрированию ИС, контроль правами доступа на объекты в системе, мониторинг доступа пользователей объекта, проведение периодического тестирования. |

#### Подразделения пользователей бизнес-систем

| **Подразделение** | **Функциональные роли** | **Количество** |
| --- | --- | --- |
| 1 | Директор проектной организации | 1 |
| 2 | Руководитель проекта | 1 на проект |
| 3 | Руководитель направления | 3+ на проект |
| 4 | Руководитель процесса | 5+ на направление |
| 5 | Сотрудник | 10+ на процесс |
| 6 | Администратор АС | 2 |
| Администратор ИБ | 3 |

#### Требования к сценариям работы бизнес-систем

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Функциональная роль** | **Бизнес-требование** | **Операция** | **Применяемый функционал** | **Перечень выполняемых действий** |
| 1 | Директор проектной организации | БТ-1 | Анализ информации | ФТ-3 | Просмотр |
| 2 | Руководитель проекта | БТ-1, БТ-2, БТ-3 | Анализ и корректировка данных | ФТ-1 | Просмотр, внесение данных |
| 3 | Руководитель направления | БТ-1, БТ-2, БТ-3 | Анализ и корректировка данных | ФТ-1 | Просмотр, внесение данных |
| 4 | Руководитель процесса | БТ-1, БТ-2, БТ-3 | Анализ и корректировка данных | ФТ-1 | Просмотр, внесение данных |
| 5 | Сотрудник | БТ-7, БТ-9 | Анализ информации | ФТ-4 | Просмотр |
| 6 | Администратор АС | БТ-9, БТ-8 | Регистрация пользователей в ЕАСПО, хранение информации | ФТ-4, ФТ-2 | Регистрация, хранение |
| 7 | Администратор ИБ | БТ-9, БТ-4, БТ-5, БТ-6 | Анализ пользовательских сессий | ФТ-4, ФТ-2 | Просмотр |

### Нефункциональные требования

#### Требования к инфраструктуре и каналам связи

Требования к инфраструктуре и каналам связи будут определены на этапе разработки технического проекта.

#### Требования к целевой технологической архитектуре проектного решения

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тип ландшафта** | **ОС** | **Назначение** | **CPU, ГБ** | **RAM, ГБ** | **Дисковая подсистема** | |
| **Назначение** | **Объем, ГБ** |
| Виртуальный | Windows 10 | Среда разработки  Тестовая среда  Продуктовая среда | 8, 16, 32, 64 | 16, 32, 64 | Системные данные | 1 |
| Данные базы | 1 |

Требования могут быть доработаны на этапе технического проектирования.

#### Требования к характеристикам каналов связи

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Адрес площадки** | **Количество пользователей** | **Требуемая пропускная способность (Кбит/с)** |
| 1 | Москва | 250 | 1024 |
| 2 | Казань | 50 | 1024 |
| 3 | Новосибирск | 5 | Канал модемной связи |

Требования могут быть доработаны на этапе технического проектирования.

#### Требования к отказоустойчивости

* Высокая доступность – да.
* Катастрофоустойчивость – нет.
* RCO – нет.

#### Требования по диагностированию системы

* События безопасности:
  + Вход пользователя в систему;
  + Выход пользователя из системы;
  + Создание/изменение/удаление репозитория, объекта;
  + Ввод вручную/загрузка данных;
  + Выгрузка отчетов/данных.
* Разрезы регистрации события:
  + Объект доступа;
  + Субъект доступа;
  + Дата и время;
  + Результат (отказ/успех).

#### Требования по модернизации и развитию системы:

Должна быть реализована возможность горизонтального масштабирования путем увеличения количества вычислительных узлов и предусмотрено вертикальное масштабирование путем увеличения ресурсов отдельных серверов.

#### Требования к численности и квалификации персонала

|  |  |  |
| --- | --- | --- |
| **Функциональная роль** | **Количество** | **Квалификация** |
| Пользователь | До 100 | Уверенный пользователь ПК, уверенный пользователь браузера с ПК |
| Администратор | До 5 | Квалифицированный специалист в администрировании платформенных компонентов ОС |

Пользователь системы должен иметь опыт работы с ПК в браузере на уровне квалифицированного пользователя и свободно осуществлять базовые операции. Администратор системы должен быть квалифицированным специалистом в администрировании платформенных компонентов ИС. Допускаются исключения в случае проведения сервиса по платформенному компоненту специализированной сервисной группой в соответствии с соглашением.

#### Требования к надежности

* Среднее время на восстановление работоспособности после технических сбоев: 2 часа.
* Система должна функционировать в режиме 24/7 за исключением периодов технического обслуживания

#### Требования к резервному копированию

* Максимально допустимое время восстановления сервисов 5 часов;
* Максимально допустимый интервал потери данных 3 месяца;
* Срок хранения резервных копий:
  + копии ежегодного копирования – 5 лет;
  + копии ежемесячного копирования – 1 год;
  + копии ежедневного копирования – 1 месяц.

#### Требования по доступности и производительности

* Открытие страницы портала не должно превышать 4 минуты;
* Выполнение операций добавления/изменения/копирования/удаления не должно превышать 10 секунд;
* Выполнение операций по администрированию не должно превышать 1 минуту.

#### Требования к масштабированию

ЕАСПО должна сохранять свою работоспособность и доступность для 10000 пользователей одновременно. Для увеличения максимального количества пользователей необходима модификация ЕАСПО.

#### Требования к режиму функционирования

* Эксплуатация:

Доступ к изменениям данных пользователей ЕАСПО должен быть предоставлен только администраторам ЕАСПО по запросу пользователей с помощью средств связи на портале.

Пользователям запрещено вручную обращаться к базе данных и вручную менять данные в ней – для внесения данных необходимо пользоваться формой на портале и/или в GUI-приложении.

* Регламентное обслуживание:

Регламентное обслуживание ЕАСПО должно производиться 1 раз в 6 месяцев. Технические специалисты исполнителя во время регламентного обслуживания должны проверить работоспособность ЕАСПО, удостовериться в отсутствии системных ошибок, а также при необходимости обновить программное обеспечение ЕАСПО. Все обнаруженные системные ошибки должны быть задокументированы обнаружившим лицом и переданы техническим специалистам исполнителя.

* Конфигурирование:

Конфигурация базы данных подлежит изменению только согласно дополнительным соглашениям, заключаемым в письменном виде заказчиком и исполнителем, и при согласовании работ.

* Восстановление:

Восстановление системы может быть выполнено администратором системы согласно дополнительному соглашению либо техническим специалистом со стороны исполнителя по письменному запросу заказчика.

* Обновление

Обновление данных в базе данных выполняют пользователи согласно их правам доступа. Обновление базы данных может быть произведено заказчиком, в случае если это не нарушает целостности базы данных. Обновление портала и/или GUI-приложения может быть выполнено только техническим специалистом исполнителя по письменному запросу заказчика.

#### Требования к внешней среде

Сервера, на которых хранятся компоненты ЕАСПО, не должны быть подвержены облучению солнцем, обливанию водой и нанесению другого физического вреда.

#### Требования к конфиденциальности

ЕАСПО должна обеспечивать конфиденциальность данных пользователей, включая логины и пароли для доступа в систему. Для этого должна быть применена технология шифрования данных согласно дополнительному соглашению.

#### Требования к эргономике и технической эстетике

Компоненты ЕАСПО должны разрабатываться согласно стандарту на кодирование инструментов проектной организации.

#### Требования к технической поддержке

Техническая поддержка осуществляется исполнителем в рабочие дни с 7:00 до 23:00. Работа технической поддержки исполнителя осуществляется в режиме «до последнего клиента». Взаимодействие с технической поддержкой исполнителя производится администраторами и руководителями заказчика.

#### Требования к лицензированию

Программное обеспечение ЕАСПО должно поставляться бесплатно в режиме чтения данных без возможности их прямого копирования из базы данных, портала или GUI-приложения. Для предоставления доступа к изменению данных каждому пользователю отдельно должен быть предоставлен ключ доступа, регистрируемый с помощью технического специалиста исполнителя в режиме «один ключ – один пользователь». Ключ доступа должен быть сформирован согласно дополнительному соглашению.

### Требования к безопасности

#### Решения по идентификации и аутентификации

Идентификация пользователя в сети производится по IP-адресу компьютера в локальной сети проектной организации. Аутентификация пользователя производится с помощью ввода логина и пароля пользователя. Требования к создаваемому логину и задаваемому паролю описаны в дополнительном соглашении.

#### Решение по управлению доступом

Выдавать и изменять права доступа в любую часть системы может только администратор ЕАСПО.

#### Регистрация и учет событий безопасности

ЕАСПО фиксирует и логирует всякое изменение прав доступа к системе, обновления данных в базе данных.

#### Контроль целостности

Контроль целостности ЕАСПО обеспечивается с помощью внедрения прав доступа к определенным данным определенных пользователей. Права доступа пользователям выдает администратор ЕАСПО.

#### Межсетевое экранирование

Межсетевое экранирование должно быть обеспечено заказчиком в целях предоставления доступа пользователей к данным с требуемой скоростью.

#### Резервное копирование

Резервное копирование может выполняться вручную. Администратором может быть задан график и порядок выполнения резервного копирования в автоматическом режиме.

#### Антивирусная защита

Антивирусное ПО не должно блокировать возможность изменения данных в базе данных пользователям с предоставленным к тому доступом.

#### Контроль защищенности

В случае угрозы или подозрения на угрозу безопасности и конфиденциальности данных ЕАСПО и/или пользователей администраторы специалиста должны незамедлительно направить сообщение об этом технической поддержке исполнителя для анализа и совместного противодействия возникшей угрозе.

## Состав и содержание работ по созданию АС

При создании АС выделяются следующие работы:

* разработка концептуального проекта системы,
* разработка логического проекта системы,
* разработка физического проекта системы.

|  |  |  |
| --- | --- | --- |
| **№** | **Процесс** | **Сроки** |
| 1 | Передача данных заказчиком исполнителю | 2 недели |
| 2 | Разработка исполнителем концептуального проекта и согласование его с заказчиком | 6 недель |
| 3 | Разработка исполнителем логического проекта и согласование его с заказчиком | 6 недель |
| 4 | Разработка исполнителем физического проекта и согласование его с заказчиком | 6 недель |
| 5 | Прием заказчиком выполненной работы | 2 недели |

## Порядок разработки АС

Порядок организации разработки АС:

1. Заказчик предоставляет всю необходимую для создания АС информацию: количество и виды пользователей ЕАСПО (включая администраторов), требования к функционированию, визуализации и пр.
2. Исполнитель принимает данные от заказчика и предоставляет концептуальный проект системы на согласование с заказчиком.
3. После согласования концептуального проекта исполнитель предоставляет логический проект системы на согласование с заказчиком.
4. После согласования логического проекта исполнитель предоставляет физический проект системы на согласование с заказчиком.
5. После согласования физического проекта исполнитель передает все необходимые для развертывания данные заказчику, включая программное обеспечение ЕАСПО, лицензионные ключи пользователей.

При необходимости с каждого из процессов может быть переход на один или несколько процессов назад для обеспечения ускорения разработки АС.

При разработке ЕАСПО исходный код программного обеспечения должен быть написан в соответствии со стандартом на кодирование инструментов проектной организации.

По окончании разработки ЕАСПО исполнитель предоставляет акт о завершении работы с перечнем файлов исходного кода инструмента и ссылкой на информационный ресурс контроля версий и обмена данными, определенный в дополнительном соглашении, а также руководство пользователя ЕАСПО и руководство администратора ЕАСПО.

## Порядок контроля и приемки АС

При приеме системы заказчик должен проверить:

* реализованы ли требования по визуализации ЕАСПО;
* не возникают ли системные ошибки при запуске системы и модификации (добавлении / изменении / удалении) данных в базе данных;
* корректно ли работает шифрование данных пользователей;
* корректно ли работает лицензирование пользователей.

При возникновении какой-либо ошибки (в реализации или системной) заказчик должен оформить сообщение о проблеме, передаваемое исполнителю. Исполнитель в свою очередь должен в разумные сроки выполнить все необходимые корректировки и передать обновленное программное обеспечение ЕАСПО заказчику.

## Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу АС в действие

Заказчик должен быть обеспечен необходимым оборудованием для интеграции программного обеспечения ЕАСПО.

Для интеграции ЕАСПО среди пользователей и администраторов заказчику передаются руководства, согласно которым заказчик должен ознакомить пользователей и администраторов с аспектами работы с системой, и только после этого допускать пользователя к использованию ЕАСПО.

## Требования к документированию

В процессе разработки ЕАСПО должны быть разработаны и переданы заказчику следующие документы:

* концептуальный проект ЕАСПО – 1 документ;
* логический проект ЕАСПО – 1 документ;
* руководство пользователя – 1 документ;
* руководство администратора – 1 документ;
* отчеты о верификации ЕАСПО – 1 документ.

## Источники разработки

Документ разработан на основе ГОСТ 34.602-2020.

# Список использованных источников

1. ГОСТ 34.602-2020 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ. КОМПЛЕКС СТАНДАРТОВ НА АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА СОЗДАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ
2. Шилдс Уолтер SQL: быстрое погружение. – СПб.: Питер, 2022. – 224 с.: ил. – (Серия «Библиотека программиста»).
3. Маркин, А.В. Программирование на SQL. В 2 ч. Часть 1: учебник и практикум для бакалавриата и магистратуры / А.В. Маркин. – М.: Издательство Юрайт, 2017. – 362 с. – Серия : Бакалавр и магистр. Академический курс.
4. Маркин, А.В. Программирование на SQL. В 2 ч. Часть 2: учебник и практикум для бакалавриата и магистратуры / А.В. Маркин. – М.: Издательство Юрайт, 2017. – 292 с. – Серия : Бакалавр и магистр. Академический курс.
5. Святослав Куликов. Работа с MySQL, MS SQL Server и Oracle в примерах. Практическое пособие для программистов и тестировщиков. Версия книги 1.0.3 от 28.12.2019.