СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 2](#__RefHeading___Toc4112_1312630076)

[1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ 3](#__RefHeading___Toc4114_1312630076)

[Общая информация. 3](#__RefHeading___Toc4116_1312630076)

[Основные используемые технологии 3](#__RefHeading___Toc4120_1312630076)

[2. ОСНОВНЫЕ НАПРАВЛЕНИЯ АДМИНИСТРИРОВАНИЯ БАЗ ДАННЫХ И СЕРВЕРОВ 5](#__RefHeading___Toc8566_1312630076)

[2.1. Технические проблемы, возникающие в процессе эксплуатации баз данных 5](#__RefHeading___Toc6043_903355023)

[2.2. Администрирование отдельных компонент серверов 5](#__RefHeading___Toc6045_903355023)

[2.3. Требования, предъявляемые к конфигурации локальных компьютерных сетей и серверного оборудования 6](#__RefHeading___Toc6047_903355023)

[2.4. Аудит систем безопасности БД и серверов 6](#__RefHeading___Toc6049_903355023)

[2.5. Регламенты по защите информации баз данных 6](#__RefHeading___Toc6051_903355023)

[3. ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ 8](#__RefHeading___Toc6053_903355023)

[ЗАКЛЮЧЕНИЕ 17](#__RefHeading___Toc4132_1312630076)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#__RefHeading___Toc4134_1312630076)

[ПРИЛОЖЕНИЯ 19](#__RefHeading___Toc4136_1312630076)

# ВВЕДЕНИЕ

Цель практики заключается в получении практического опыта и знаний в сфере проектирования и разработки информационных систем с обработкой информации.

Задачи практики заключались в управлении процессами разработки приложений с использованием инструментальных средств; обеспечении сбора данных для анализа использования и функционирования информационной системы; применении методики тестирования разрабатываемых приложений; определении состава оборудования и программных средств разработки информационной системы с использованием алгоритмов обработки информации.

ООО "Малленом Системс" – ведущий российский разработчик программно-аппаратных решений в области **промышленного машинного зрения и интеллектуальной видеоаналитики**. Компания специализируется на создании высокотехнологичных систем, использующих алгоритмы компьютерного зрения (CV), машинного обучения (ML) и глубокого обучения (Deep Learning) для автоматизации критически важных задач контроля качества, безопасности, логистики и учета на промышленных предприятиях, транспортных узлах и объектах инфраструктуры. Портфель продуктов компании охватывает решения для идентификации объектов (продукция, вагоны, автотранспорт), контроля технологических процессов, ситуационной безопасности и анализа транспортных потоков.

Сроки практики с 06.07.2025 по 11.07.2025, место прохождения практики ООО “Малленом Системс” - Металлургов 21б.

# 1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

## Общая информация.

Основная среда деятельности: Разработка, производство (интеграция аппаратной части), внедрение и сопровождение программно-аппаратных комплексов (ПАК) на основе компьютерного зрения и искусственного интелекта.

Ключевые технологические направления:

* Машинное зрение (Machine Vision) для промышленной автоматизации.
* Ситуационная видеоаналитика (Video Analytics, Video Content Analysis – VCA).
* Распознавание образов (Pattern Recognition), включая распознавание объектов, текста (OCR), лиц (при необходимости в рамках продуктов).
* Алгоритмы машинного обучения и глубокого обучения (нейронные сети) для анализа изображений и видео.

Роль: ИС (как продуктовые, так и внутренние) являются **абсолютно критической инфраструктурой** для создания, поставки и поддержки инновационных решений компании, обеспечивая ее технологическое лидерство.

## Основные используемые технологии

Языки программирования:

* Основные: Python. SQLite
* Вспомогательные: JavaScript, CSS, HTML (веб-интерфейсы).

Фреймворки и платформы общего назначения:

* Бэкенд: FastAPI.
* Микросервисы: Docker - для масштабируемости компонентов аналитики.

Системы контроля версий: GitHub.

Инструменты проектирования:

* Диаграммы: UML, блок-схемы алгоритмов.
* Моделирование данных: ER-диаграммы.

Методологии разработки:

* Гибкие методологии (Agile): Scrum (для ПО), Kanban (для задач поддержки, оперативной разработки).

Системы управления задачами: GitHab Projects.

Инструменты CI/CD: GitHab CI/CD.

Другие ключевые компоненты: IDE: Visual Studio Code.

# 2. ОСНОВНЫЕ НАПРАВЛЕНИЯ АДМИНИСТРИРОВАНИЯ БАЗ ДАННЫХ И СЕРВЕРОВ

### **2.1. Технические проблемы, возникающие в процессе эксплуатации баз данных**

В процессе эксплуатации баз данных в ООО «Малленом Системс» могут возникать типичные технические проблемы, связанные как с программной, так и с аппаратной частью. К ним относятся снижение производительности при большом объёме записей, нарушение ограничений целостности данных, блокировки при одновременном выполнении транзакций, а также ошибки при резервном копировании или восстановлении. Кроме того, периодически возникают проблемы с синхронизацией между различными модулями системы, особенно при использовании нескольких СУБД в связке.

### **2.2. Администрирование отдельных компонент серверов**

Администрирование серверной инфраструктуры охватывает работу с виртуальными машинами, системами контейнеризации (в частности, Docker), настройку служб хранения, журналирования, мониторинга и автоматического развертывания приложений. Системные администраторы также обеспечивают бесперебойную работу серверов баз данных, контролируют использование ресурсов и управляют обновлениями программного обеспечения. Особое внимание уделяется управлению доступом, настройке ролей и прав пользователей, а также резервному копированию и проверке целостности хранилищ.

### **2.3. Требования, предъявляемые к конфигурации локальных компьютерных сетей и серверного оборудования**

В компании применяются высокие требования к сетевой и серверной инфраструктуре. Основу сети составляет гигабитное соединение с логическим разделением по VLAN для ограничения доступа. Серверы должны обладать высокой производительностью, иметь процессоры серверного уровня (например, Intel Xeon), достаточный объём ОЗУ (от 32 ГБ и выше), SSD-диски в конфигурации RAID и поддержку ECC-памяти. Обязательно наличие резервных серверов и средств автоматического переключения в случае отказа. Также используются VPN и системы межсетевого экранирования для защиты внутреннего трафика.

### **2.4. Аудит систем безопасности БД и серверов**

Проверка состояния информационной безопасности проводится регулярно. В ходе аудита анализируются журналы событий, проверяются права доступа, контролируется соблюдение политик безопасности на уровне ОС и СУБД. Используются как встроенные механизмы безопасности самих СУБД, так и сторонние инструменты для обнаружения уязвимостей, а также системы мониторинга сетевой активности. При выявлении отклонений немедленно предпринимаются меры по устранению уязвимостей и усилению политики доступа.

### **2.5. Регламенты по защите информации баз данных**

Внутренние регламенты по защите данных в компании включают обязательное резервное копирование всех критичных данных, использование шифрования при передаче и хранении информации, разграничение доступа на основе ролей, а также журналирование действий пользователей с последующим анализом. Все процедуры соответствуют требованиям ГОСТ и внутренним политикам информационной безопасности, а сама инфраструктура проектируется с учётом принципов отказоустойчивости, масштабируемости и контроля доступа.

# 3. ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

Задание 1:

В ходе анализа существующих СУБД были рассмотрены наиболее популярные системы: PostgreSQL, MySQL, MariaDB, Microsoft SQL Server, Oracle и SQLite. На основе оценки их функциональности, сложности настройки и соответствия задачам проекта, в качестве СУБД была выбрана **SQLite**. Это встроенная база данных, не требующая отдельного сервера и обеспечивающая полноценную поддержку SQL, индексов, ограничений целостности и представлений. Выбор обоснован её простотой, лёгкостью внедрения и достаточностью возможностей для реализации учебного проекта.

Задание 2:

Выбранная предметная область — **оптический контроль качества керамики**. На производстве каждое изделие проверяется визуально (автоматически или оператором). Цель — выявление дефектов: трещин, сколов, пузырей, деформаций. Проверка сопровождается записью результатов и, при необходимости, фиксацией фото дефектов. База данных предназначена для учёта всех проверок, хранения информации об изделиях, результатах и выявленных дефектах.

Структура предметной области проста: изделие → проверка → дефекты. Дополнительно может храниться информация об операторах и времени проверок. Такой подход обеспечивает прослеживаемость и статистический анализ качества продукции.

Задание 3:  
3.1. Техническое задание (ТЗ)

#### a. **Описание целей и задач БД**

Цель — создание простой, надёжной и автономной базы данных для хранения информации о результатах оптического контроля керамики.

Задачи:

* учёт проверяемых изделий;
* сохранение информации о каждой проверке;
* фиксация обнаруженных дефектов;
* обеспечение целостности и удобства доступа к данным.

#### b. **Требования к функциональности**

БД должна уметь:

* регистрировать изделие;
* сохранять дату и результат каждой проверки;
* связывать проверку с найденными дефектами;
* хранить краткую информацию о дефекте (тип, зона, фото, комментарий);
* обеспечивать быстрый поиск по изделиям, датам, дефектам.

#### c. **Требования к данным**

Хранимые данные:

* идентификаторы;
* текстовая и числовая информация;
* дата/время;
* ссылки на изображения.

Объёмы:

* ~100 изделий в день;
* ~1–2 проверки на изделие;
* ~0–2 дефекта на проверку.

Источники:

* операторы контроля;
* автоматизированные системы контроля.

#### d. **Ограничения**

* Простота — использование SQLite.
* Производительность — до 5000 записей в месяц.
* Безопасность — файл БД хранится на защищённом сервере.
* Масштабируемость — возможно обновление под PostgreSQL при росте.

#### e. **Сроки и этапы разработки**

1. Проектирование модели — 1 день
2. Реализация БД (SQLite) — 1 день
3. Тестирование — 1 день
4. Подготовка документации — 1 день

### 3.2. Концептуальная модель (описание сущностей и связей)

#### a. **Сущности**:

1. **Изделие** (Product)

* product\_id — уникальный ID
* serial\_number — серийный номер
* date\_created — дата создания изделия (опционально)

1. **Проверка** (Inspection)

* inspection\_id — уникальный ID
* product\_id — внешний ключ на Product
* inspection\_date — дата/время проверки
* inspector\_name — имя оператора
* result — результат ("годен" / "не годен")

1. **Дефект** (Defect)

* defect\_id — уникальный ID
* inspection\_id — внешний ключ на Inspection
* defect\_type — тип (трещина, скол и т.д.)
* location — зона (лево, центр, низ и т.д.)
* photo\_path — путь к изображению дефекта
* comment — комментарий

#### b. **Связи:**

* Один Product связан с несколькими Inspection → связь 1 ко многим.
* Один Inspection связан с несколькими Defect → связь 1 ко многим.

### 3.3. Логическая модель

#### a. Поля и типы данных:

#### 1. Product: product\_id INTEGER PRIMARY KEY AUTOINCREMENT

serial\_number TEXT NOT NULL UNIQUE

date\_created DATE

2. Inspection:

inspection\_id INTEGER PRIMARY KEY AUTOINCREMENT

product\_id INTEGER NOT NULL

inspection\_date DATETIME NOT NULL

inspector\_name TEXT

result TEXT CHECK(result IN ('годен', 'не годен')) NOT NULL

FOREIGN KEY(product\_id) REFERENCES Product(product\_id) ON DELETE CASCADE

3. Defect

defect\_id INTEGER PRIMARY KEY AUTOINCREMENT

inspection\_id INTEGER NOT NULL

defect\_type TEXT NOT NULL

location TEXT

photo\_path TEXT

comment TEXT

FOREIGN KEY(inspection\_id) REFERENCES Inspection(inspection\_id) ON DELETE CASCADE

#### b. Нормализация:

Таблицы соответствуют **3НФ**:

* Нет повторяющихся групп,
* Все поля зависят от первичного ключа,
* Нет транзитивных зависимостей.

#### c. Ключи:

* PRIMARY KEY — во всех таблицах (autoincrement).
* FOREIGN KEY — inspection.product\_id, defect.inspection\_id.
* UNIQUE — serial\_number.

#### d. Индексы:

* SQLite создаёт индексы на PRIMARY KEY и UNIQUE по умолчанию.
* Можно явно добавить:

CREATE INDEX idx\_inspection\_date ON Inspection(inspection\_date);

CREATE INDEX idx\_defect\_type ON Defect(defect\_type);  
ER-диаграмма представленна на Рисунке 1

3.4. Физическая модель (реализация в СУБД SQLite)

**SQL-скрипт создания таблиц:**

-- Операторы контроля

CREATE TABLE operators (

operator\_id INTEGER PRIMARY KEY,

name TEXT NOT NULL,

position TEXT,

cert\_number TEXT

);

-- Конструкции (объекты обследования)

CREATE TABLE structures (

structure\_id INTEGER PRIMARY KEY,

name TEXT NOT NULL,

location TEXT,

inspection\_date TEXT NOT NULL,

operator\_id INTEGER NOT NULL,

FOREIGN KEY (operator\_id) REFERENCES operators(operator\_id)

);

-- Типы трещин

CREATE TABLE crack\_types (

crack\_type\_id INTEGER PRIMARY KEY,

name TEXT NOT NULL,

description TEXT

);

-- Обнаруженные трещины

CREATE TABLE cracks (

crack\_id INTEGER PRIMARY KEY,

structure\_id INTEGER NOT NULL,

crack\_type\_id INTEGER NOT NULL,

x\_coord INTEGER,

y\_coord INTEGER,

length\_mm REAL,

width\_mm REAL,

severity TEXT,

FOREIGN KEY (structure\_id) REFERENCES structures(structure\_id),

FOREIGN KEY (crack\_type\_id) REFERENCES crack\_types(crack\_type\_id)

);

-- Камеры

CREATE TABLE cameras (

camera\_id INTEGER PRIMARY KEY,

model TEXT,

resolution TEXT,

location TEXT

);

-- Снимки трещин

CREATE TABLE images (

image\_id INTEGER PRIMARY KEY,

structure\_id INTEGER NOT NULL,

camera\_id INTEGER,

image\_path TEXT NOT NULL,

captured\_at TEXT,

FOREIGN KEY (structure\_id) REFERENCES structures(structure\_id),

FOREIGN KEY (camera\_id) REFERENCES cameras(camera\_id)

);  
Дополнительно:   
CREATE INDEX idx\_inspection\_date ON Inspection(inspection\_date);

CREATE INDEX idx\_defect\_type ON Defect(defect\_type);

Партиционирование и настройки хранилища — **не требуются** для SQLite.

### 3.5. Бизнес-правила и ограничения:

* NOT NULL на всех обязательных полях.
* CHECK(result IN ('годен', 'не годен')).
* ON DELETE CASCADE — при удалении изделия, удаляются все проверки и дефекты.

### 3.6. Доступ и безопасность:

* **Роли в SQLite** отсутствуют — файл БД доступен в рамках ОС.
* Настройка прав доступа — средствами файловой системы.
* Шифрование — через **SQLite с расширением SEE** или внешними средствами (не требуется по заданию).
* Бэкап — регулярное копирование .db файла по расписанию (скриптом или вручную).

3.7. Примеры запросов:

-- Найти все дефекты по изделию с серийным номером 'A001':  
SELECT d.defect\_type, d.location, d.photo\_path  
FROM Defect d  
JOIN Inspection i ON d.inspection\_id = i.inspection\_id  
JOIN Product p ON i.product\_id = p.product\_id  
WHERE p.serial\_number = 'A001';  
-- Добавление изделия:  
INSERT INTO Product(serial\_number, date\_created)  
VALUES ('A001', DATE('now'));  
-- Добавление проверки:  
INSERT INTO Inspection(product\_id, inspection\_date, inspector\_name, result)  
VALUES (1, DATETIME('now'), 'Иванов И.И.', 'годен');  
  
-- Добавление дефекта:  
INSERT INTO Defect(inspection\_id, defect\_type, location, photo\_path, comment)  
VALUES (1, 'трещина', 'центр', '/photos/defect1.jpg', 'Мелкая трещина');

3.8. Руководство администратора

* **Бэкап**: копировать файл db.sqlite3 ежедневно в отдельную папку или облако.
* **Мониторинг**: вручную или через утилиты sqlite3/Python-скрипты.
* **Восстановление**: скопировать нужную версию .db обратно.
* **Ошибки**: использовать PRAGMA integrity\_check; для диагностики.

### 3.9. Словарь данных

| Таблица | Поле | Тип | Описание |
| --- | --- | --- | --- |
| Product | product\_id | INTEGER | Уникальный ID изделия |
|  | serial\_number | TEXT | Серийный номер |
|  | date\_created | DATE | Дата создания |
| Inspection | inspection\_id | INTEGER | ID проверки |
|  | product\_id | INTEGER | FK на изделие |
|  | inspection\_date | DATETIME | Дата и время проверки |
|  | inspector\_name | TEXT | Имя оператора |
|  | result | TEXT | Годен/Не годен |
| Defect | defect\_id | INTEGER | ID дефекта |
|  | inspection\_id | INTEGER | FK на проверку |
|  | defect\_type | TEXT | Тип дефекта |
|  | location | TEXT | Зона дефекта |
|  | photo\_path | TEXT | Путь к изображению |
|  | comment | TEXT | Комментарий |

# ЗАКЛЮЧЕНИЕ

В ходе прохождения производственной практики в ООО «Малленом Системс» были получены ценные практические навыки в области администрирования баз данных и серверов, а также углублены теоретические знания, полученные в рамках образовательной программы. Практика позволила ознакомиться с реальной организацией работы ИТ-отдела, архитектурой вычислительной и сетевой инфраструктуры, а также с используемыми в компании технологиями и средствами автоматизации.

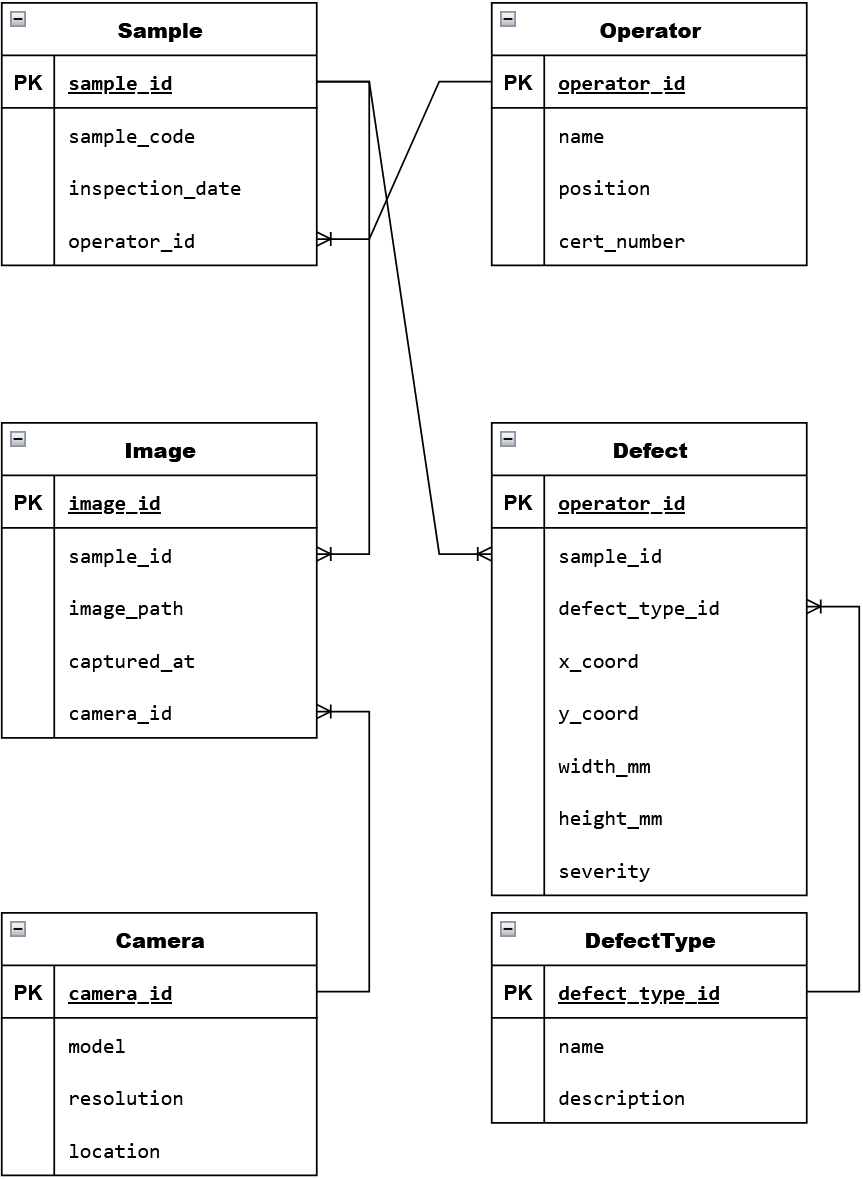
Особое внимание в процессе практики уделялось обеспечению бесперебойной работы серверного оборудования, мониторингу состояния баз данных, выполнению задач резервного копирования, управлению доступом пользователей и анализу проблем, возникающих в процессе эксплуатации. Кроме того, был выполнен проект по проектированию и разработке базы данных для хранения результатов оптического контроля качества керамики, включающий в себя анализ предметной области, создание логической и физической моделей, реализацию СУБД и описание механизмов обеспечения целостности данных.

Полученные знания и опыт будут полезны в дальнейшей профессиональной деятельности и формируют прочную основу для работы в сфере системного администрирования, разработки и сопровождения информационных систем.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* PostgreSQL [Электронный ресурс] // Википедия. – URL: <https://ru.wikipedia.org/wiki/PostgreSQL> (дата обращения: 09.07.2025).
* База данных [Электронный ресурс] // Википедия. – URL: <https://ru.wikipedia.org/wiki/База_данных> (дата обращения: 09.07.2025).
* Система управления базами данных [Электронный ресурс] // Википедия. – URL: <https://ru.wikipedia.org/wiki/Система_управления_базами_данных> (дата обращения: 09.07.2025).
* SQL [Электронный ресурс] // Википедия. – URL: <https://ru.wikipedia.org/wiki/SQL> (дата обращения: 09.07.2025).
* Нормализация базы данных [Электронный ресурс] // Википедия. – URL: <https://ru.wikipedia.org/wiki/Нормализация_базы_данных> (дата обращения: 09.07.2025).
* Резервное копирование [Электронный ресурс] // Википедия. – URL: <https://ru.wikipedia.org/wiki/Резервное_копирование> (дата обращения: 09.07.2025).
* Информационная безопасность [Электронный ресурс] // Википедия. – URL: <https://ru.wikipedia.org/wiki/Информационная_безопасность> (дата обращения: 09.07.2025).
* ER-модель [Электронный ресурс] // Википедия. – URL: <https://ru.wikipedia.org/wiki/ER-модель> (дата обращения: 09.07.2025).
* Docker [Электронный ресурс] // Википедия. – URL: <https://ru.wikipedia.org/wiki/Docker> (дата обращения: 09.07.2025).
* ООО «Малленом Системс» [Электронный ресурс] // Rusprofile. – URL: https://www.rusprofile.ru/id/4878368 (дата обращения: 09.07.2025).

# ПРИЛОЖЕНИЯ

Рисунок 1 - ER-диаграмма.