

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	2
1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ.....	3
Общая информация.....	3
Основные используемые технологии.....	3
2. ОСНОВНЫЕ НАПРАВЛЕНИЯ АДМИНИСТРИРОВАНИЯ БАЗ ДАННЫХ И СЕРВЕРОВ.....	5
2.1. Технические проблемы, возникающие в процессе эксплуатации баз данных.....	5
2.2. Администрирование отдельных компонент серверов.....	5
2.3. Требования, предъявляемые к конфигурации локальных компьютерных сетей и серверного оборудования.....	6
2.4. Аудит систем безопасности БД и серверов.....	6
2.5. Регламенты по защите информации баз данных.....	6
3. ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ.....	8
ЗАКЛЮЧЕНИЕ.....	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	18
ПРИЛОЖЕНИЯ.....	19

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 интеллекта.

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: GitHub Projects.

Инструменты CI/CD: GitHub 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)

а. 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.

б. 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.

с. 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)

2. 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")

3. 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

б. 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

а. 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

б. Нормализация:

Таблицы соответствуют 3НФ:

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

с. 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 представлена на Рисунке 1

3.4. Физическая модель (реализация в СУБД SQLite)

SQL-скрипт создания таблиц:

-- Таблица изделий

```
CREATE TABLE Product (  
    product_id INTEGER PRIMARY KEY AUTOINCREMENT,  
    serial_number TEXT NOT NULL UNIQUE,  
    date_created DATE  
);
```

-- Таблица проверок

```
CREATE TABLE 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  
);
```

-- Таблица дефектов

```

CREATE TABLE 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
);

```

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.
- Возможна реализация триггера, например, логирования:

```

CREATE TRIGGER log_defect_insert

AFTER INSERT ON Defect

BEGIN

```

```
INSERT INTO LogTable(event_type, event_time, message)
VALUES ('insert', DATETIME('now'), 'Добавлен дефект');
END;
```

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	Годеен/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

Таблица	Поле	Тип	Описание
	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

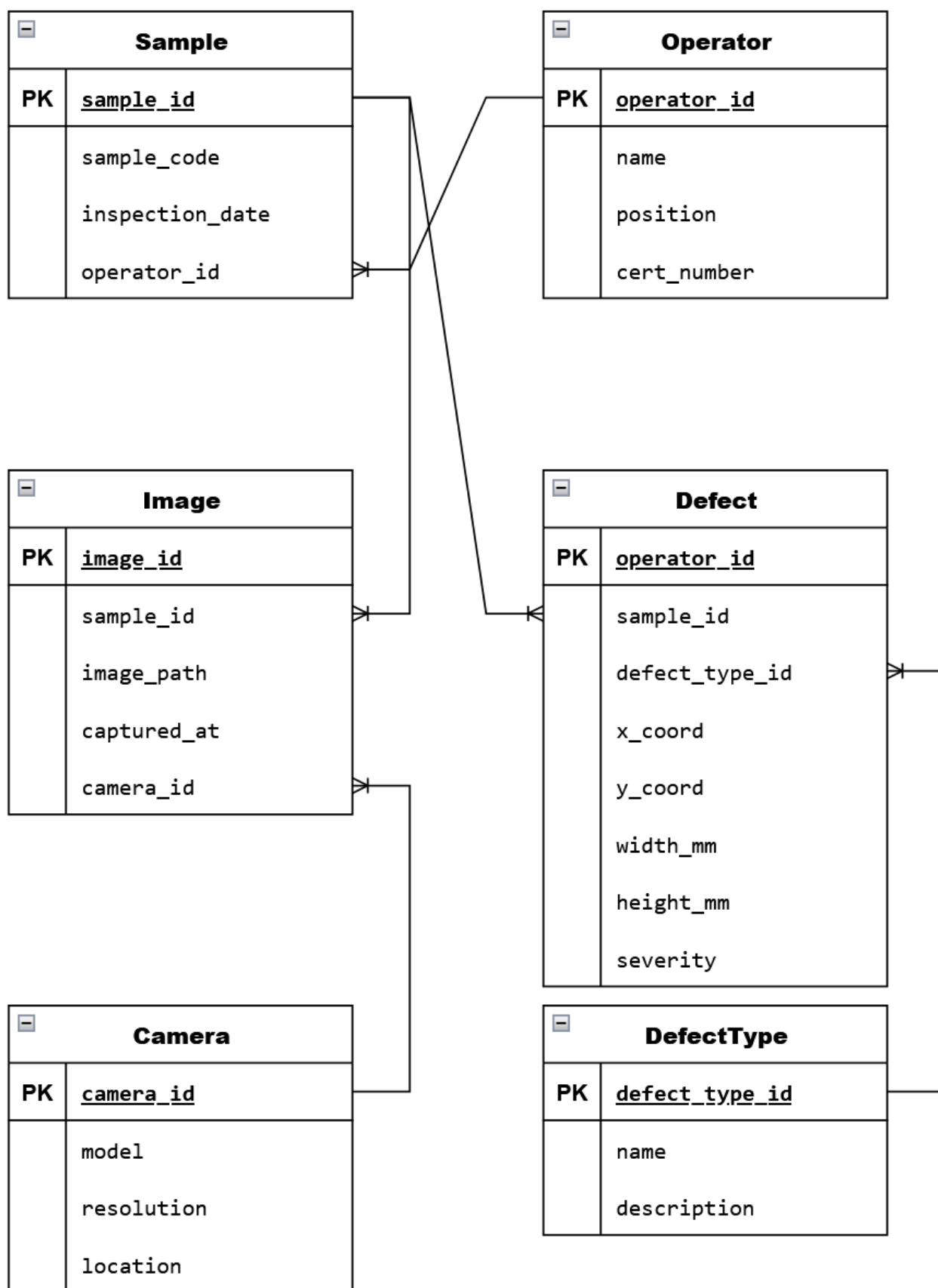


Рисунок 1 - ER-диаграмма.