1. Описание кейса
   1. **Предметная область:**

Сфера услуг

* 1. **Введение:**

Каждая модель автомобиля может иметь свои характерные дефекты, которые приводят к износу или выходу из строя определенных деталей. Для эффективного обслуживания и ремонта автомобилей необходима система, способная анализировать информацию о наиболее часто выходящих из строя деталях для каждой модели автомобиля.

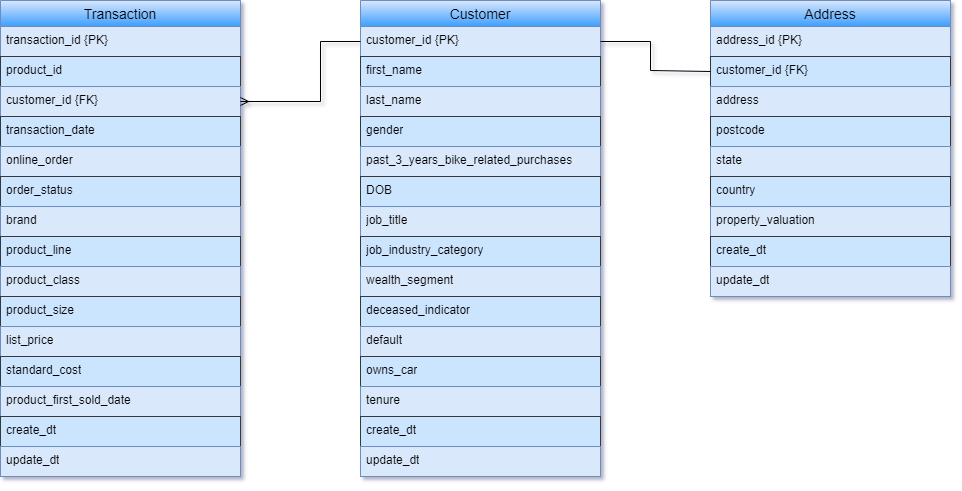
* 1. **Цель:**

Целью проекта является разработка системы анализа дефектов автомобилей на основе информации о наиболее часто выходящих из строя деталях. Система должна помочь при диагностике и техническом обслуживании автомобилей, позволяя с большей вероятностью обнаружить дефекты и принять соответствующие меры по их устранению.

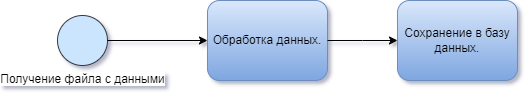
* 1. **Проблема:**

Проблема заключается в том, что владельцы автомобилей, дилерские и ремонтные центры, а также производители не всегда имеют полную информацию о типичных дефектах, которые могут возникнуть у конкретной модели автомобиля. В результате, при диагностике и техническом обслуживании могут быть пропущены значимые проблемы, что приводит к повышенным затратам на ремонт и снижению доверия владельцев к автомобилю.

1. ER-диаграмма



1. BPMN-нотация:



1. Архитектура
   1. База данных
      * + SQLite
   2. Компоненты
      * + Python
        + SQL

Архитектура пайплайна представляет собой последовательную обработку таблиц исходного файла формата (excel) с последующей записью данных в базу (SqlLight). Код обработки данных написан на языке Python и SQL. База данных представляет собой двухуровневое хранение данных (стейдинговый слой и основной). Отношения сущностей в базе представлено во 2НФ.

1. Выбор СУБД

Исходя из того, что исходные данные предоставляются в табличном формате, было принято решение использовать реляционную базу данных SQLite.

1. Выбор СХД
   1. блочная
   2. объектная
   3. файловая
2. В этом проекте были использованы следующие алгоритмы и методы:

**Обработка файлов:**

- Получение списка файлов в директориях "input\_data/" и "archive/".

- Копирование файлов с добавлением расширения ".dump".

**Работа с данными:**

- Чтение данных из файлов Excel с использованием библиотеки Pandas.

- Извлечение данных из разных листов (sheets) в файле Excel.

**Алгоритм ETL (Extract, Transform, Load):**

- Извлечение данных из исходных источников (Excel-файлов).

- Преобразование данных в формат, подходящий для загрузки в хранилище данных.

- Загрузка данных во временные таблицы (STG) в базе данных.

**Работа с базой данных:**

- Создание и использование класса "EtlTask" для выполнения операций с базой данных.

- Инициализация временных таблиц (STG) для загрузки данных перед обработкой.

- Исполнение SQL-скриптов для создания структуры хранилища данных (Data Warehouse).

- Загрузка данных в хранилище данных с помощью SQL-скриптов.

**Управление файлами:**

- Использование библиотеки "shutil" для копирования файлов.

**Структурированный программный код:**

- Организация кода в функции, обеспечивающие логическую структуру проекта.

- Использование условных операторов для выбора соответствующих таблиц и скриптов в зависимости от имени листа (sheet\_name).

Общий алгоритм проекта включает в себя процессы извлечения данных из файлов, преобразования данных, загрузки их во временные таблицы, а также создания и загрузки данных в хранилище данных (Data Warehouse).

1. Проекту могут угрожать различные модели угроз, связанные с безопасностью данных, производительностью и надежностью системы.

**Несанкционированный доступ к данным:**

Злоумышленники могут попытаться получить несанкционированный доступ к файлам данных или базе данных, что может привести к утечке конфиденциальных данных.

**Инъекции SQL:**

Атаки типа SQL-инъекции могут позволить злоумышленникам выполнить вредоносные SQL-запросы через пользовательский ввод, нарушая целостность и безопасность данных.

**Отказ в обслуживании (DDoS):**

Массированные атаки на сервера могут вызвать отказ в обслуживании, что приведет к недоступности системы и потере производительности.

**Недостаточная аутентификация и авторизация:**

Слабые механизмы аутентификации и авторизации могут позволить злоумышленникам получить доступ к данным, на которые у них не должно быть прав.

**Утечка данных:**

Некорректная обработка данных, недостаточное шифрование или недостаточные меры безопасности могут привести к утечке конфиденциальных данных.

**Вредоносное программное обеспечение:**

Вредоносные программы могут попасть на серверы или рабочие станции, что может привести к потере данных или нарушению функционирования системы.

**Ошибка в коде:**

Некорректно написанный код может привести к уязвимостям в системе и стать точкой входа для злоумышленников.

**Недостаточное мониторинг и аудит:**

Отсутствие мониторинга и аудита может привести к невозможности обнаружения и предотвращения инцидентов безопасности.

Чтобы защитить проект от этих угроз, необходимо внедрить соответствующие меры безопасности, такие как хорошая аутентификация, шифрование данных, применение паттернов безопасного программирования, регулярное обновление и мониторинг системы, а также обучение сотрудников основам кибербезопасности.