МИНИСТЕРСТВО ТРАНСПОРТА РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА (МИИТ)»

Кафедра ЦТУТП

**Отчёт**

**По лабораторной работе №3**  
по дисциплине «Информационные хранилища и аналитические системы»

Тема: «ТЕХНОЛОГИЯ ПОСТРОЕНИЯ АНАЛИТИЧЕСКИХ ОТЧЕТОВ»

Выполнил: Шедания В. М.

Группа: УИС-411

Преподаватель: доц. Козьяков П. О.

­

Москва 2025 г.

**СОДЕРЖАНИЕ**

[ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ 3](#_Toc195648333)

[ВАРИАНТ ЗАДАНИЯ 4](#_Toc195648334)

[ИСХОДНЫЕ ФАЙЛЫ 6](#_Toc195648335)

[ИСХОДНЫЙ КОД ПРОГРАММЫ 10](#_Toc195648336)

[РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ 15](#_Toc195648337)

[ВЫВОД 20](#_Toc195648338)

# ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ

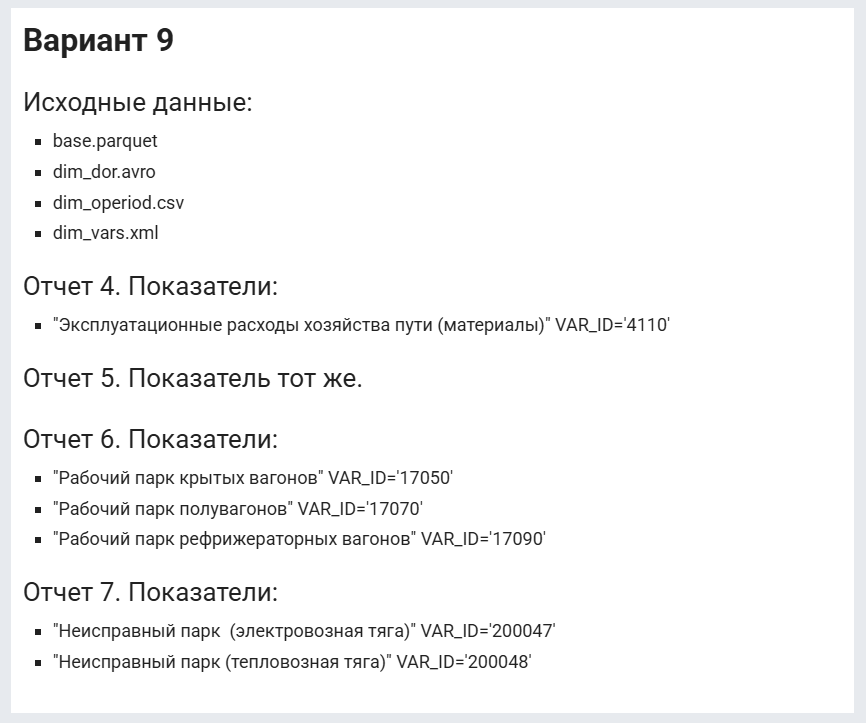
Овладеть навыками построения отчетов различного формата при помощи языка Python и библиотеки Pandas.

# ВАРИАНТ ЗАДАНИЯ

**Вариант №9**

Средствами Python и библиотеки Pandas подключить к проекту данные, образующие схему "Звезда", построить на основе них отчеты табличного и графического вида и экспортировать результаты в виде документов.

1. Создать проект и подключить необходимые библиотеки (NumPy, Pandas, MatPlotLib).
2. Добавить данные в проект согласно варианту.
3. Создать словари данных.
4. Подготовить списочный отчет показателей по варианту, который выводит итоговое значение показателя (показатели по вариантам, используя отчетность H, но с нарастающим итогом) по годам для каждой дороги с применением словарей. Значение нарастающего итога вывести в форматированном виде с использованием символа валюты ₽.
5. Подготовить отчет со статистиками по показателю (показатели по вариантам, используя отчетность H) по годам по дорогам  с применением словарей.
6. Подготовить сводный отчет, который выводит итоговое значения показателя (используя отчетность H, с нарастающим итогом) по дорогам по годам для каждого показателя (показатели по вариантам) с применением словарей. Значение нарастающего итога вывести в форматированном виде с использованием символа валюты ₽ или обозначения "ед."
7. Подготовить графический отчет в виде столбчатой диаграммы (Bar Chart), который будет выводить значение показателя (показатели по вариантам, используя отчетность P, ненарастающий итог) по дорогам за весь 2003 год с применением словарей.
8. Экспортировать полученные результаты пунктов 4-7 в виде листов единого файла Excel.



# ИСХОДНЫЕ ФАЙЛЫ

Файл base.parquet представляет собой основной фактологический набор данных в формате Apache Parquet — это колонко-ориентированный бинарный формат, оптимизированный для хранения и обработки больших объёмов табличных данных. В этом файле содержатся значения показателей (переменная fact) по железнодорожным дорогам (DOR\_ID) за разные даты (DATE) и по различным видам отчётности (OPERIOD). Каждая запись относится к определённому показателю (VAR\_ID) и времени. Этот файл служит основой для построения всех аналитических отчётов.

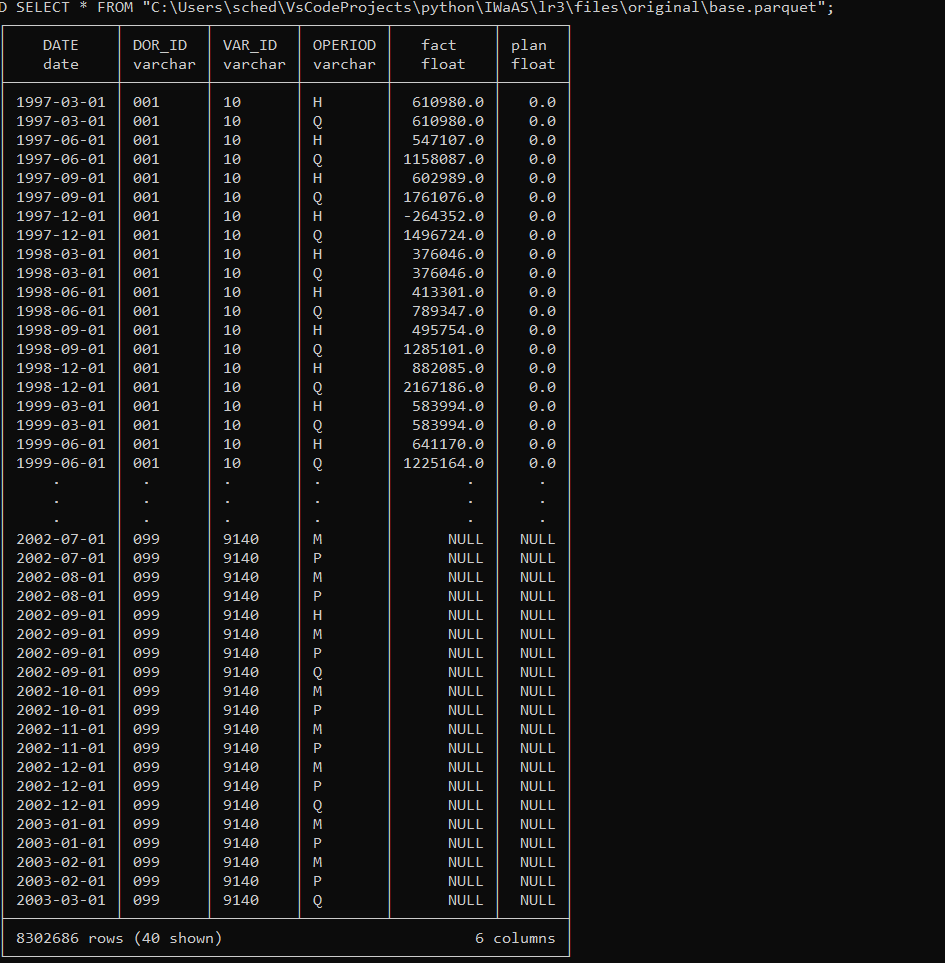


Рисунок 1 – Файл base.parquet

Файл dim\_vars.xml — это справочник показателей, представленный в формате XML. В нём содержатся описания переменных (показателей) с полями VAR\_ID (уникальный код показателя) и NAME (наименование показателя на русском языке). Этот файл используется для преобразования числовых идентификаторов показателей в человекочитаемый формат при формировании выходных отчётов и графиков. Формат XML позволяет структурировать данные и легко извлекать их с помощью библиотеки ElementTree.

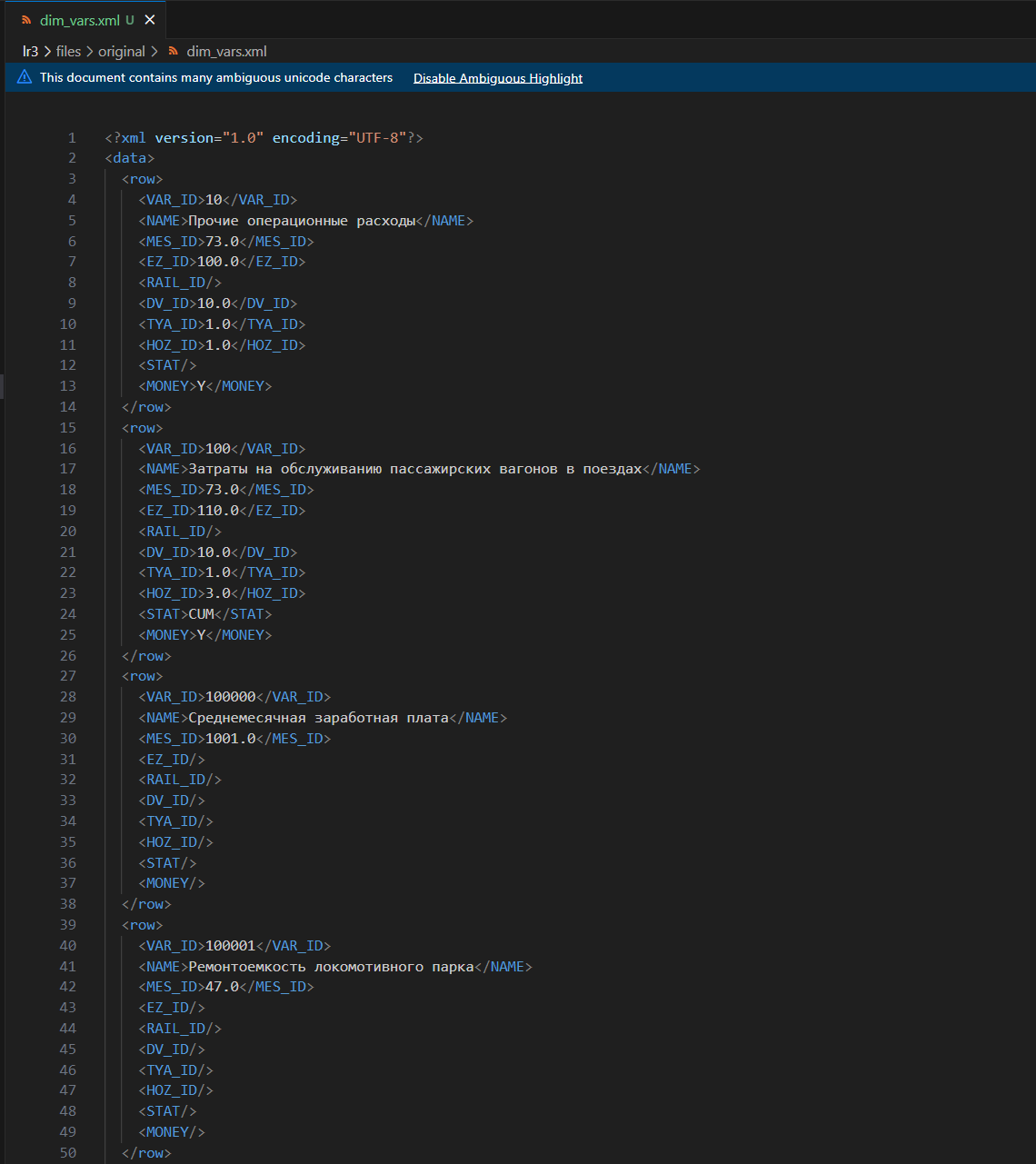


Рисунок 2 – Файл dim\_vars.xml

Файл dim\_operiod.csv содержит справочную информацию об отчётных периодах. Он записан в формате CSV (comma-separated values), что делает его пригодным для работы в любой среде аналитики. Каждая строка описывает один тип периода отчётности, например, месячный (P) или полугодовой (H), и содержит его описание или метку. Этот файл используется при интерпретации поля OPERIOD в фактовой таблице.

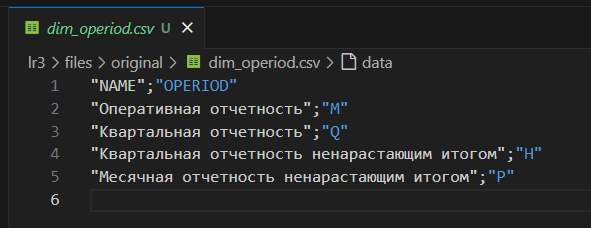


Рисунок 3 – Файл dim\_operiod.csv

Файл dim\_dor.avro является справочником железных дорог, записанным в формате Apache Avro — это компактный и быстро читаемый бинарный формат с встроенной схемой. Содержит пары DOR\_ID и NAME, где каждому коду дороги соответствует её полное наименование. Используется для расшифровки кодов дорог в итоговых отчётах, что позволяет выводить не цифровые идентификаторы, а полные названия (например, "Московская", "Октябрьская").

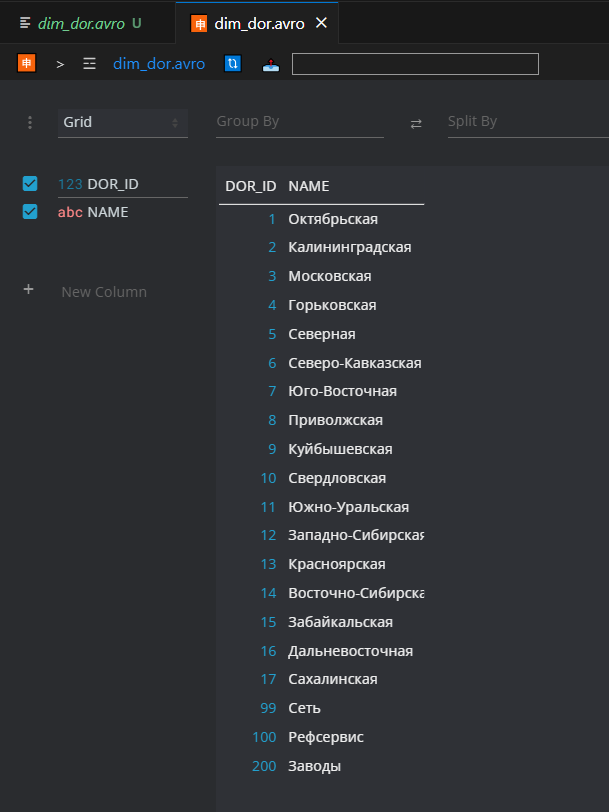


Рисунок 4 – Файл dim\_dor.avro

# ИСХОДНЫЙ КОД ПРОГРАММЫ

import pandas as pd

import json

import xml.etree.ElementTree as ET

from pathlib import Path

BASE\_DIR = Path(\_\_file\_\_).resolve().parent

data\_dir = BASE\_DIR / "files" / "original" / "view"

def load\_dictionaries():

    # Словарь DOR\_ID → название дороги

    with open(data\_dir / "dim\_dor\_output.json", encoding='utf-8') as f:

        dor\_data = json.load(f)

        dor\_dict = {str(row["DOR\_ID"]).zfill(3): row["NAME"] for row in dor\_data}

    # Словарь VAR\_ID → название показателя

    tree = ET.parse(data\_dir / "dim\_vars.xml")

    root = tree.getroot()

    vars\_dict = {

        row.find("VAR\_ID").text.strip(): row.find("NAME").text.strip()

        for row in root.findall("row")

    }

    # Словарь OPERIOD → описание (если нужно)

    operiod = pd.read\_csv(data\_dir / "dim\_operiod.csv", sep=",", encoding="utf-8")

    operiod\_dict = operiod.set\_index("NAME").to\_dict(orient="index")

    return dor\_dict, vars\_dict, operiod\_dict

import pandas as pd

import fastavro

import xml.etree.ElementTree as ET

from pathlib import Path

# === Пути ===

BASE\_DIR = Path(\_\_file\_\_).resolve().parent

original\_dir = BASE\_DIR / "files" / "original"

processed\_dir = BASE\_DIR / "files" / "processed"

processed\_dir.mkdir(parents=True, exist\_ok=True)

# === Чтение данных ===

base = pd.read\_parquet(original\_dir / "base.parquet")

base["VAR\_ID"] = base["VAR\_ID"].astype(str).str.strip()

base["DOR\_ID"] = base["DOR\_ID"].astype(str).str.zfill(3)

# === DOR словарь из .avro ===

with open(original\_dir / "dim\_dor.avro", "rb") as f:

    reader = fastavro.reader(f)

    dor\_data = list(reader)

    dor\_dict = {

        str(row["DOR\_ID"]).zfill(3): row["NAME"]

        for row in dor\_data

    }

# === VAR словарь из XML ===

tree = ET.parse(original\_dir / "dim\_vars.xml")

root = tree.getroot()

vars\_dict = {

    row.find("VAR\_ID").text.strip(): row.find("NAME").text.strip()

    for row in root.findall("row")

}

# === Подготовка ===

base["DOR\_NAME"] = base["DOR\_ID"].map(dor\_dict)

base["VAR\_NAME"] = base["VAR\_ID"].map(vars\_dict)

base["DATE"] = pd.to\_datetime(base["DATE"])

base["Year"] = base["DATE"].dt.year

# === Отчёт 4: Списочный отчёт по VAR\_ID=4110 (₽, накопит.) ===

df\_4110 = base[(base["OPERIOD"] == "H") & (base["VAR\_ID"] == "4110")].copy()

report\_4 = df\_4110.groupby(["Year", "DOR\_NAME"])["fact"].sum().reset\_index()

report\_4["Cumulative\_Sum"] = report\_4.groupby("DOR\_NAME")["fact"].cumsum()

report\_4["Formatted\_Value"] = report\_4["Cumulative\_Sum"].apply(lambda x: f"{x:,.2f} ₽")

# === Отчёт 5: Статистика по тому же показателю ===

report\_5 = df\_4110.groupby("Year", as\_index=False)["fact"].mean().rename(columns={"fact": "Average\_Fact"}).sort\_values("Year")

# === Отчёт 6: Сводный по VAR\_ID=[17050, 17070, 17090] (ед., накопит.) ===

summary\_vars = ["17050", "17070", "17090"]

df\_summary = base[(base["OPERIOD"] == "H") & (base["VAR\_ID"].isin(summary\_vars))].copy()

df\_summary["Cumulative\_Sum"] = df\_summary.groupby(["VAR\_ID", "DOR\_NAME"])["fact"].cumsum()

df\_summary["Formatted\_Value"] = df\_summary["Cumulative\_Sum"].apply(lambda x: f"{x:,.2f} ед.")

df\_summary["VAR\_NAME"] = df\_summary["VAR\_ID"].map(vars\_dict)

report\_6 = df\_summary.pivot\_table(

    index=["DOR\_NAME", "VAR\_NAME"],

    columns="Year",

    values="Cumulative\_Sum",

    aggfunc="sum"

).fillna(0).reset\_index()

# === Отчёт 7: График по VAR\_ID=[200047, 200048] (2003 год, P, без накоп.) ===

graph\_vars = ["200047", "200048"]

df\_graph = base[

    (base["OPERIOD"] == "P") &

    (base["VAR\_ID"].isin(graph\_vars)) &

    (base["DATE"].dt.year == 2003)

].copy()

df\_graph\_grouped = df\_graph.groupby(["DOR\_NAME", "VAR\_ID"])["fact"].sum().reset\_index()

report\_7 = df\_graph\_grouped.pivot\_table(

    index="DOR\_NAME",

    columns="VAR\_ID",

    values="fact",

    aggfunc="sum"

).fillna(0).reset\_index()

report\_7.rename(columns={v: vars\_dict.get(v, v) for v in graph\_vars}, inplace=True)

# === Экспорт в Excel с графиком ===

output\_excel = processed\_dir / "report.xlsx"

with pd.ExcelWriter(output\_excel, engine='xlsxwriter') as writer:

    report\_4.to\_excel(writer, sheet\_name="Списочный отчёт", index=False)

    report\_5.to\_excel(writer, sheet\_name="Стат. отчёт", index=False)

    report\_6.to\_excel(writer, sheet\_name="Сводный отчёт", index=False)

    report\_7.to\_excel(writer, sheet\_name="График (данные)", index=False)

    workbook = writer.book

    worksheet = writer.sheets["График (данные)"]

    chart = workbook.add\_chart({'type': 'column'})

    categories = ['График (данные)', 1, 0, len(report\_7), 0]

    series\_1 = ['График (данные)', 1, 1, len(report\_7), 1]

    series\_2 = ['График (данные)', 1, 2, len(report\_7), 2]

    chart.add\_series({

        'name': list(report\_7.columns)[1],

        'categories': categories,

        'values': series\_1,

        'fill': {'color': 'blue'}

    })

    chart.add\_series({

        'name': list(report\_7.columns)[2],

        'categories': categories,

        'values': series\_2,

        'fill': {'color': 'red'}

    })

    chart.set\_x\_axis({'name': 'Дороги'})

    chart.set\_y\_axis({'name': 'Значение показателя (ед.)'})

    chart.set\_title({'name': 'График неисправного парка, 2003'})

    chart.set\_legend({'position': 'bottom'})

    chart\_sheet = workbook.add\_worksheet("График")

    chart\_sheet.insert\_chart("B2", chart)

print(f"[✓] Отчёт успешно сформирован: {output\_excel}")

# РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

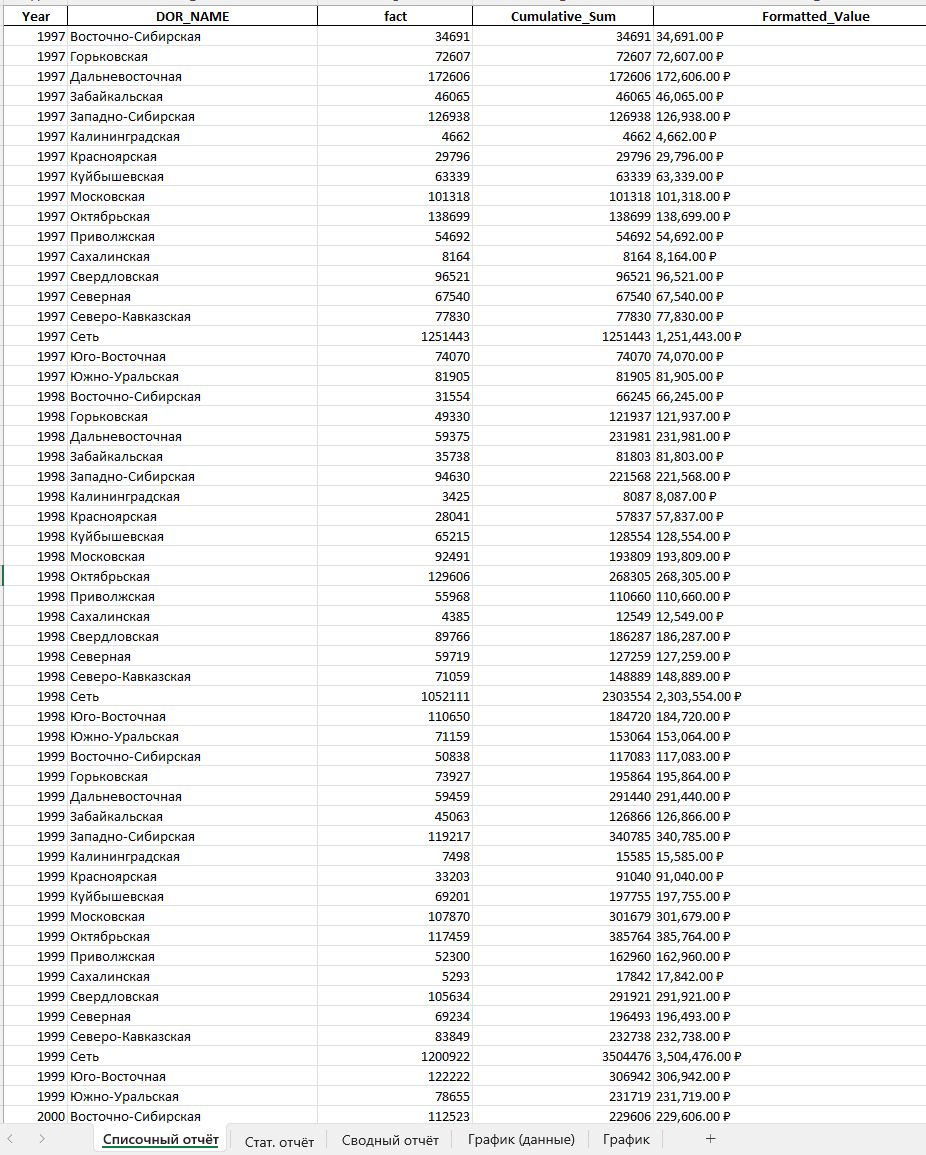
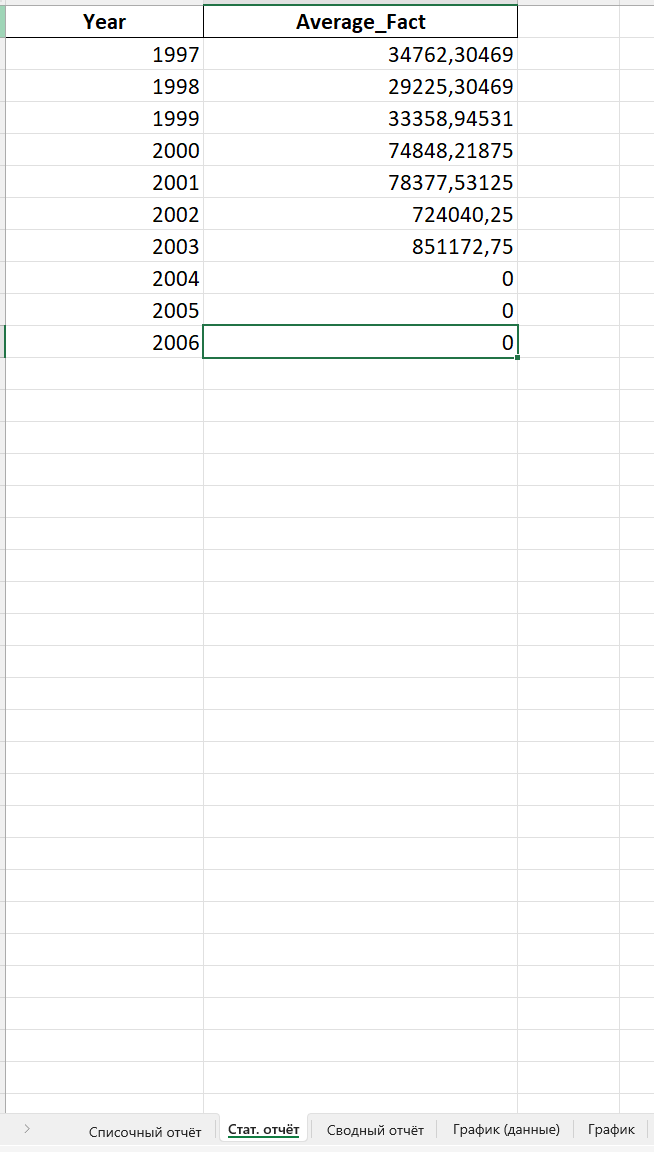


Рисунок 5 – Отчет 4 (Списочный отчет)

  
  
Рисунок 6 – Отчет 5 (Стат. отчет)

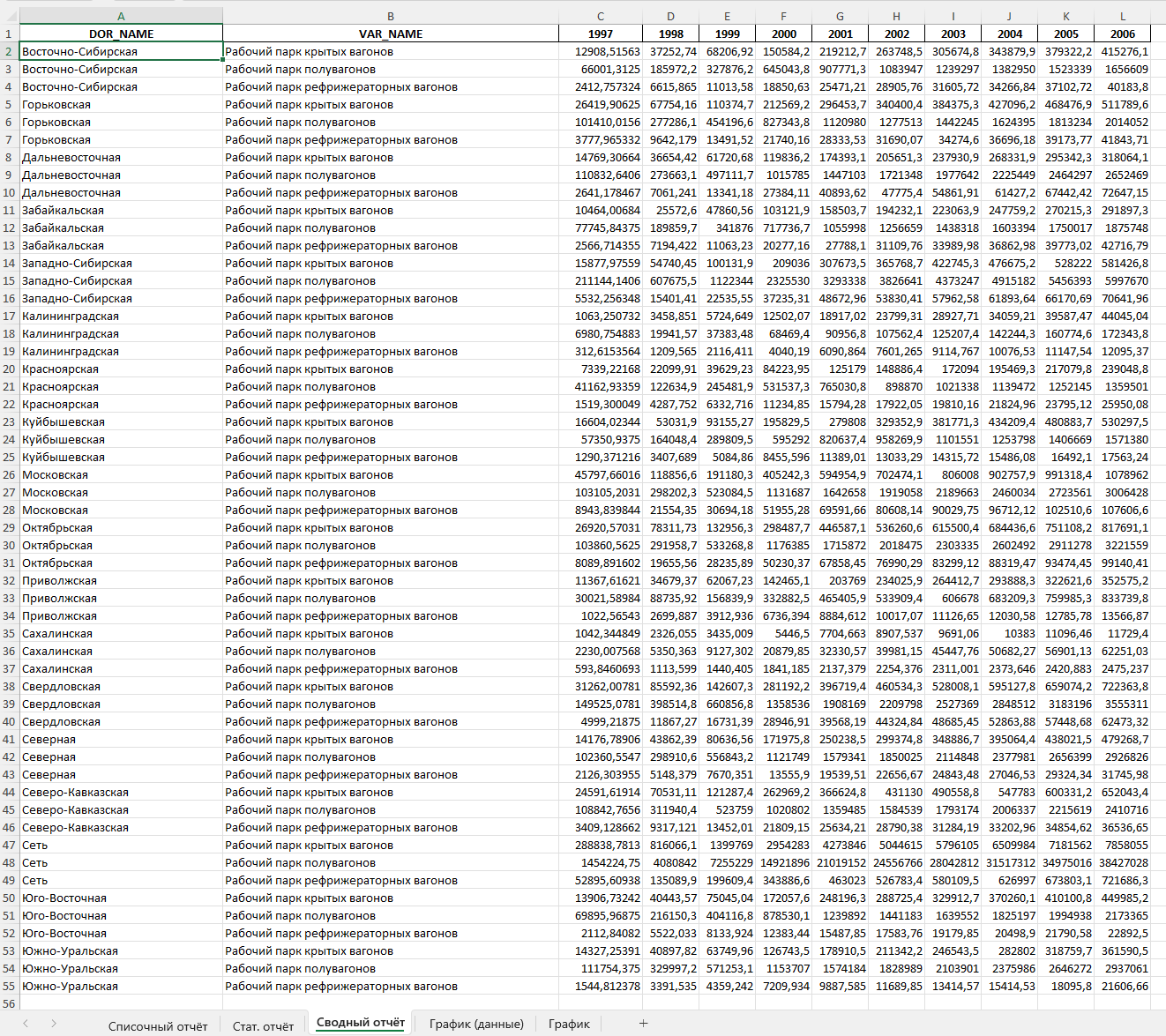


Рисунок 7 – Отчет 6 (Сводный отчет)

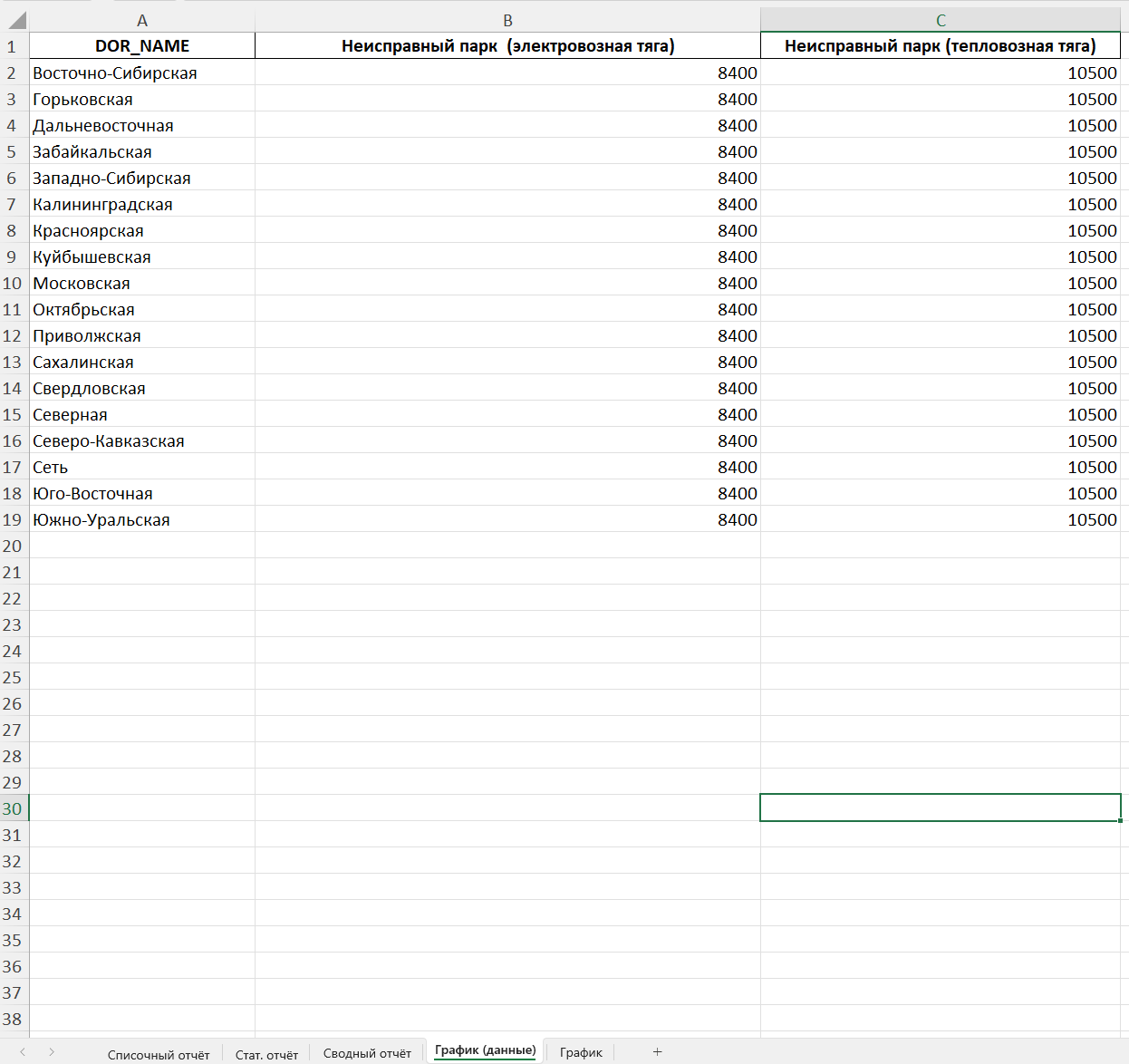


Рисунок 8 – Данные для отчета 7 (График)

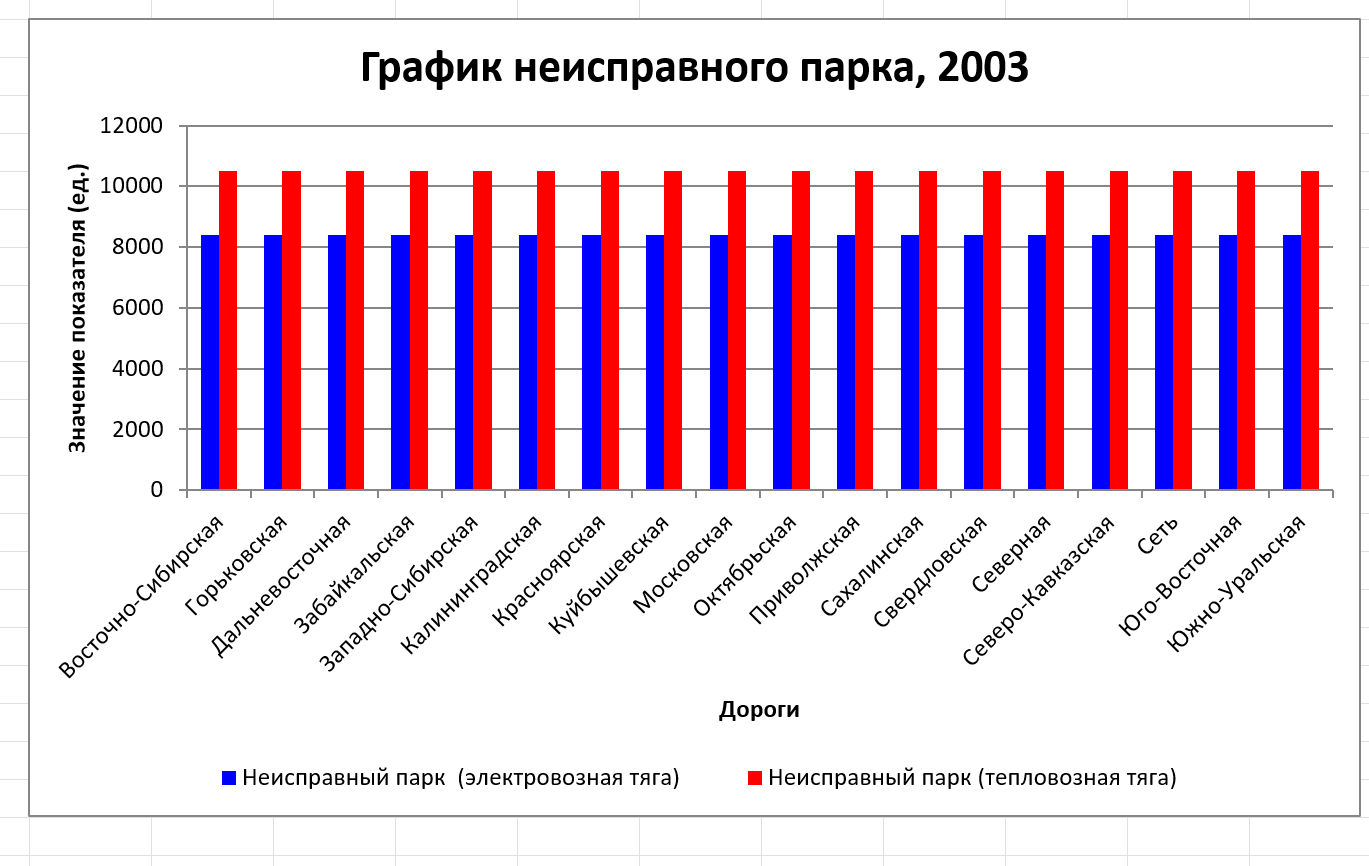


Рисунок 9 – Отчет 7 (График)

# ВЫВОД

В ходе выполнения лабораторной работы была разработана и реализована система построения аналитических отчётов на основе схемы данных «звезда», включающей таблицу фактов и таблицы измерений. Были освоены методы загрузки и обработки данных различных форматов — Parquet, Avro, XML и CSV — с использованием языка Python и библиотек Pandas, FastAvro и ElementTree.

Сформированы словари расшифровки кодов дорог и показателей, что позволило создать человекочитаемые отчёты. Построены списочный отчёт с нарастающим итогом, статистический отчёт по средним значениям, сводный отчёт по нескольким показателям с агрегацией по годам, а также графический отчёт в виде столбчатой диаграммы. Все результаты экспортированы в единый Excel-файл с удобной структурой и визуализацией.

Работа позволила закрепить навыки построения аналитических витрин, применения кумулятивных операций, сводных таблиц, а также интеграции справочников для повышения читаемости данных. Полученные результаты соответствуют требованиям современной аналитики и демонстрируют практическую ценность технологий обработки и визуализации информации в задачах управления железнодорожной отраслью.