



Дипломная работа

DATA ENGINEER

Задание на диплом

- ▶ **Цель:** составить документацию процессов ETL на основе предложенного датасета
 - ▶ Обработайте и проанализируйте данные
 - ▶ Сформируйте нормализованную схему данных (NDS)
 - ▶ Сформируйте состав таблиц фактов и измерений (DDS)
 - ▶ Сформируйте ETL-процессы: для заливки данных в NDS и для создания витрин
 - ▶ Сформируйте набор метрик и дашбордов на их основе
 - ▶ Оформите результаты, сформулируйте выводы

1. Анализ исходных данных

- ▶ **Invoice ID** - уникальный идентификатор чека
- ▶ **Branch** - филиал магазина (A, B, C)
- ▶ **City** - город расположения филиала
- ▶ **Customer type** - тип покупателя (Member/Normal)
- ▶ **Gender** - пол покупателя
- ▶ **Product line** - категория товара
- ▶ **Unit price** - цена за единицу товара
- ▶ **Quantity** - количество товаров в чеке
- ▶ **Tax 5%** - сумма налога (5%)
- ▶ **Total** - общая сумма чека
- ▶ **Date** - дата покупки
- ▶ **Time** - время покупки
- ▶ **Payment** - способ оплаты
- ▶ **cogs** - себестоимость проданных товаров
- ▶ **gross margin percentage** - процент валовой маржи
- ▶ **gross income** - валовая прибыль
- ▶ **Rating** - оценка покупателя

1. Анализ исходных данных

- ▶ Для первичного анализа исследуем датасет с помощью Jupyter Notebook
 - ▶ Подсчет количества пропущенных значений – пропущенных значений нет

```
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Invoice ID             1000 non-null   object
1   Branch                 1000 non-null   object
2   City                   1000 non-null   object
3   Customer type          1000 non-null   object
4   Gender                 1000 non-null   object
5   Product line           1000 non-null   object
6   Unit price             1000 non-null   float64
7   Quantity               1000 non-null   int64
8   Tax 5%                 1000 non-null   float64
9   Total                  1000 non-null   float64
10  Date                   1000 non-null   object
11  Time                   1000 non-null   object
12  Payment                1000 non-null   object
13  cogs                   1000 non-null   float64
14  gross margin percentage 1000 non-null   float64
15  gross income           1000 non-null   float64
16  Rating                 1000 non-null   float64
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB
```

Пропущенные значения:

```
[5]: Invoice ID             0
      Branch              0
      City                 0
      Customer type        0
      Gender               0
      Product line         0
      Unit price           0
      Quantity             0
      Tax 5%               0
      Total                0
      Date                 0
      Time                 0
      Payment              0
      cogs                 0
      gross margin percentage 0
      gross income         0
      Rating               0
dtype: int64
```

2. Обработка данных для дальнейшего анализа

- ▶ Добавляем вспомогательные временные колонки
- ▶ Присваиваем тип данных категории

```
: df['Date'] = pd.to_datetime(df['Date'], format='%m/%d/%Y')

df['Month'] = df['Date'].dt.month
df['DayOfWeek'] = df['Date'].dt.dayofweek # 0-понедельник, 6-воскресенье
df['Hour'] = pd.to_datetime(df['Time'], format='%H:%M').dt.hour

cat_cols = ['Branch', 'City', 'Customer type', 'Gender', 'Product line', 'Payment']
df[cat_cols] = df[cat_cols].astype('category')

df.info()
```

3. Общие показатели

► Рассчитываем значения общих показателей продаж компании

```
metrics = {  
    'Total Sales': df['Total'].sum(),  
    'Number of Invoices': df['Invoice ID'].nunique(),  
    'Average Rating': df['Rating'].mean(),  
    'Average Transaction Value': df['Total'].mean(),  
    'Total Quantity Sold': df['Quantity'].sum()  
}  
  
pd.DataFrame.from_dict(metrics, orient='index', columns=['Value'])
```

	Value
Total Sales	322966.75
Number of Invoices	1000.00
Average Rating	6.97
Average Transaction Value	322.97
Total Quantity Sold	5510.00

3. Общие показатели

- ▶ **Пытаемся выявить динамику продаж в зависимости от времени года. Но данных в датасете слишком мало, чтобы сделать вывод о наиболее прибыльных сезонах**



3. Общие показатели

- ▶ Рассмотрим продуктивность филиалов и городов компании.
- ▶ Вывод: каждому городу соответствует отдельный филиал. Таким образом анализ конкретного филиала соответствует показателям города продаж компании
- ▶ Филиал С (город Nauryitaw) является самым рейтинговым и прибыльным

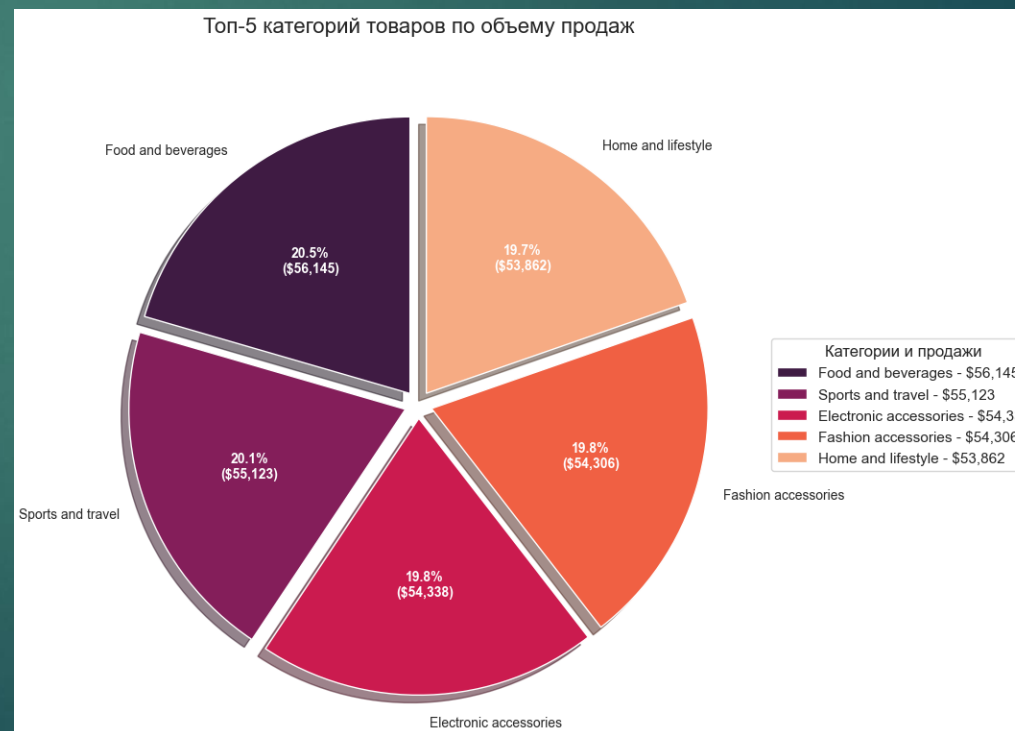
		Total Sales	Avg Transaction	Transactions	Avg Rating
Branch	City				
A	Mandalay	0.00	NaN	0	NaN
	Naypyitaw	0.00	NaN	0	NaN
	Yangon	106200.37	312.35	340	7.03
B	Mandalay	106197.67	319.87	332	6.82
	Naypyitaw	0.00	NaN	0	NaN
	Yangon	0.00	NaN	0	NaN
C	Mandalay	0.00	NaN	0	NaN
	Naypyitaw	110568.71	337.10	328	7.07
	Yangon	0.00	NaN	0	NaN



4. Анализ товаров

- ▶ Самой прибыльной продуктовой линейкой является еда
- ▶ Топ 5 товаров продаются в равном объеме с погрешностью около 1%

	Product line	Total
0	Food and beverages	56144.84
1	Sports and travel	55122.83
2	Electronic accessories	54337.53
3	Fashion accessories	54305.89
4	Home and lifestyle	53861.91



4. Анализ товаров

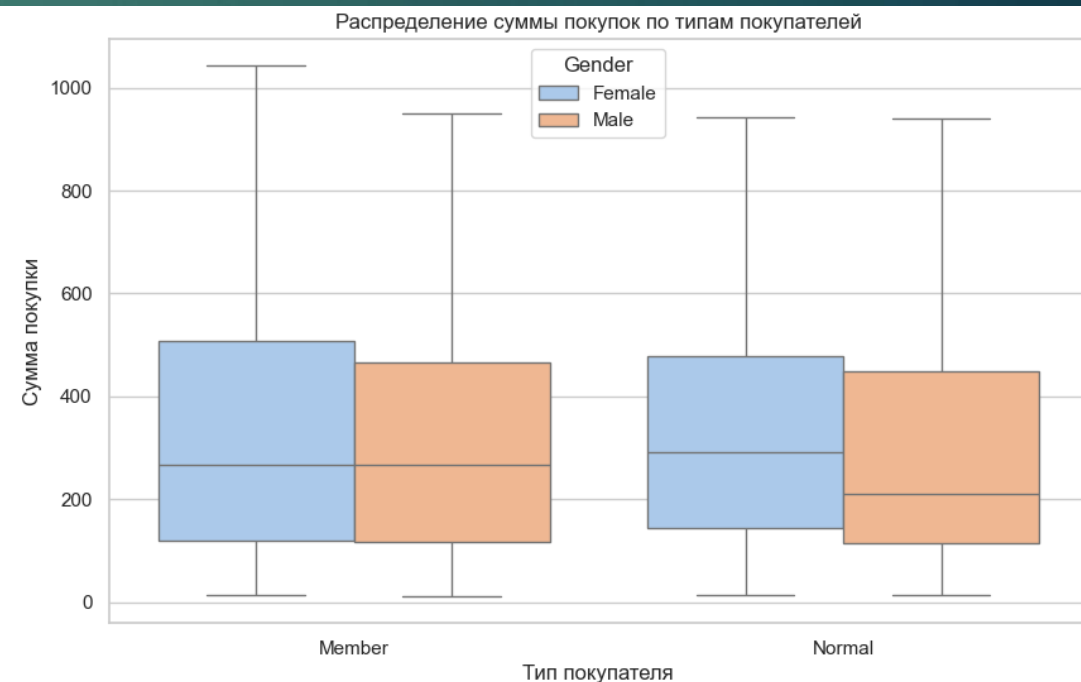
- ▶ Маржинальность товаров одинакова
- ▶ Меньше всего продаются товары из категории «Здоровье и красота»

	Total	gross income	Margin %
Product line			
Health and beauty	49193.74	2342.56	4.76
Electronic accessories	54337.53	2587.50	4.76
Fashion accessories	54305.89	2585.99	4.76
Food and beverages	56144.84	2673.56	4.76
Home and lifestyle	53861.91	2564.85	4.76
Sports and travel	55122.83	2624.90	4.76

5. Анализ покупателей

- ▶ Более высокие оценки у покупателей мужского пола без подписки
- ▶ Покупатели с подпиской получают в среднем оценки ниже
- ▶ Нет сильной взаимосвязи между полом, наличием подписки и рейтингом
- ▶ Женщины без подписки покупают товары дороже чем другие покупатели

		Transactions	Total	Rating
Customer type	Gender			
Member	Female	261	88146.94	6.94
	Male	240	76076.50	6.94
Normal	Female	240	79735.98	6.99
	Male	259	79007.32	7.02



6. Анализ внешних факторов (API)

- ▶ Для дополнительного анализа используем API сервис <https://holidayapi.com>
- ▶ Сервис позволяет понять является ли день праздником и название праздника
- ▶ Так как по бесплатной подписке можно выгрузить лишь 2024 год, а в датасете даты 2019 года – выгружаем данные из 2024 года и по числу месяца матчим с данными в датасете

```
# Преобразуем даты праздников 2024 года
holidays['date'] = pd.to_datetime(holidays['date'])
holidays['month_day'] = holidays['date'].dt.strftime('%m-%d') # Извлекаем месяц и день

# Создаем словарь для быстрого поиска
holiday_dict = dict(zip(holidays['month_day'], holidays['name']))

# Добавляем информацию о праздниках для всех годов
df['date'] = pd.to_datetime(df['Date'])
df['month_day'] = df['date'].dt.strftime('%m-%d')

df['is_holiday'] = df['month_day'].isin(holiday_dict)
df['holiday_name'] = df['month_day'].map(holiday_dict)

# Удаляем временные колонки
df.drop(['date', 'month_day'], axis=1, inplace=True)
```

Делаем запрос к HolidayAPI (2024 год)...

Получено 39 праздников за 2024 год

Результат добавления праздников:

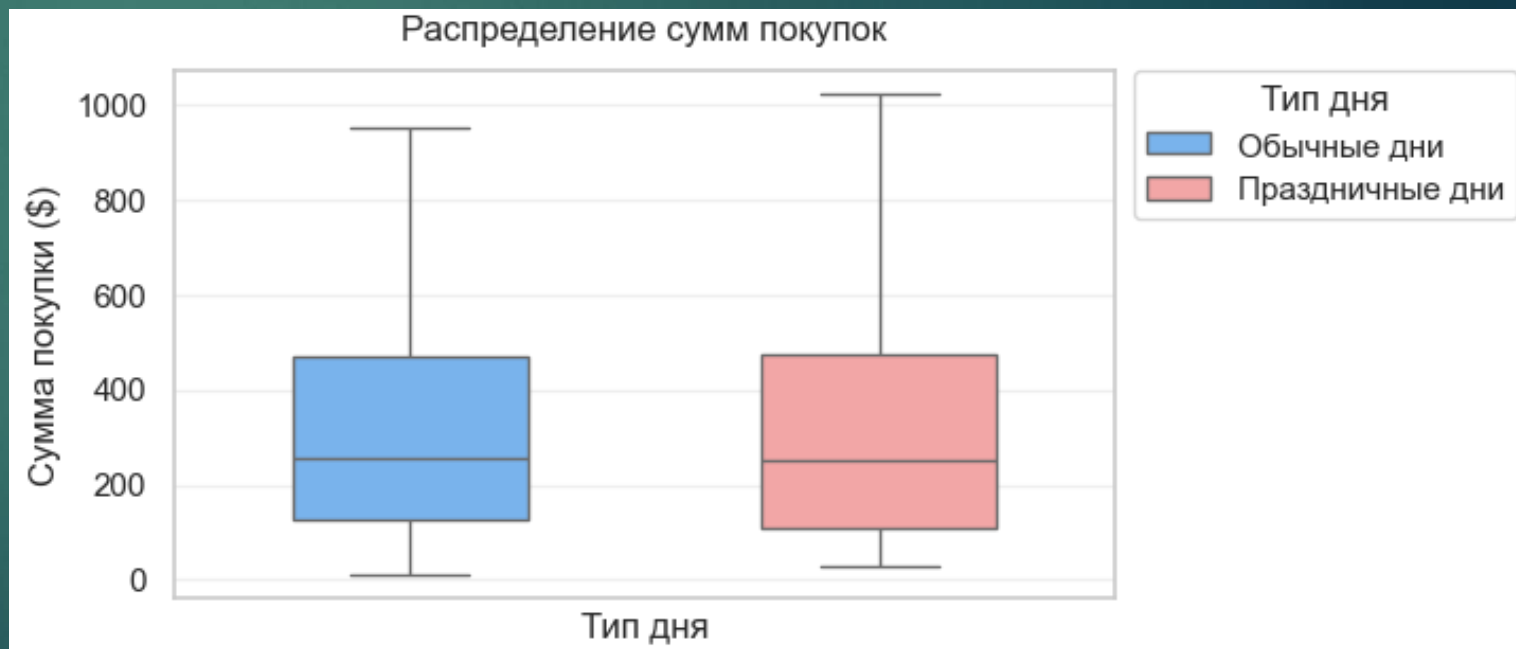
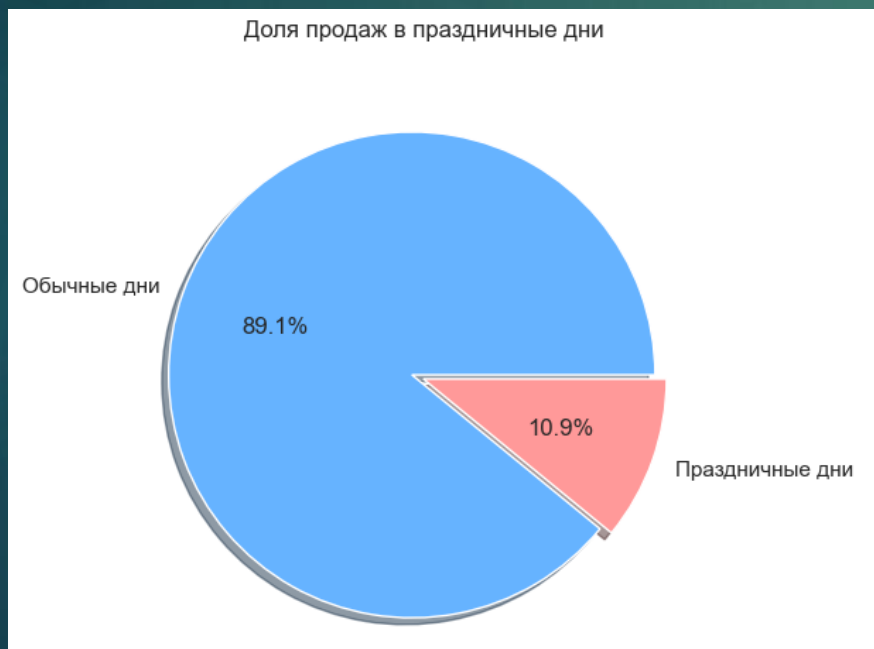
Всего праздничных дней в данных: 108

Примеры праздников в данных:

	Date	holiday_name
17	2019-01-01	New Year's Day
156	2019-01-04	Independence Day
225	2019-01-11	Kayin New Year
12	2019-02-12	Union Day
24	2019-03-02	Peasants' Day
1	2019-03-08	Maha Shivaratri
212	2019-03-20	March Equinox
108	2019-03-24	Full Moon Day of Tabaung
5	2019-03-25	Festival of Colors
56	2019-03-27	Armed Forces Day

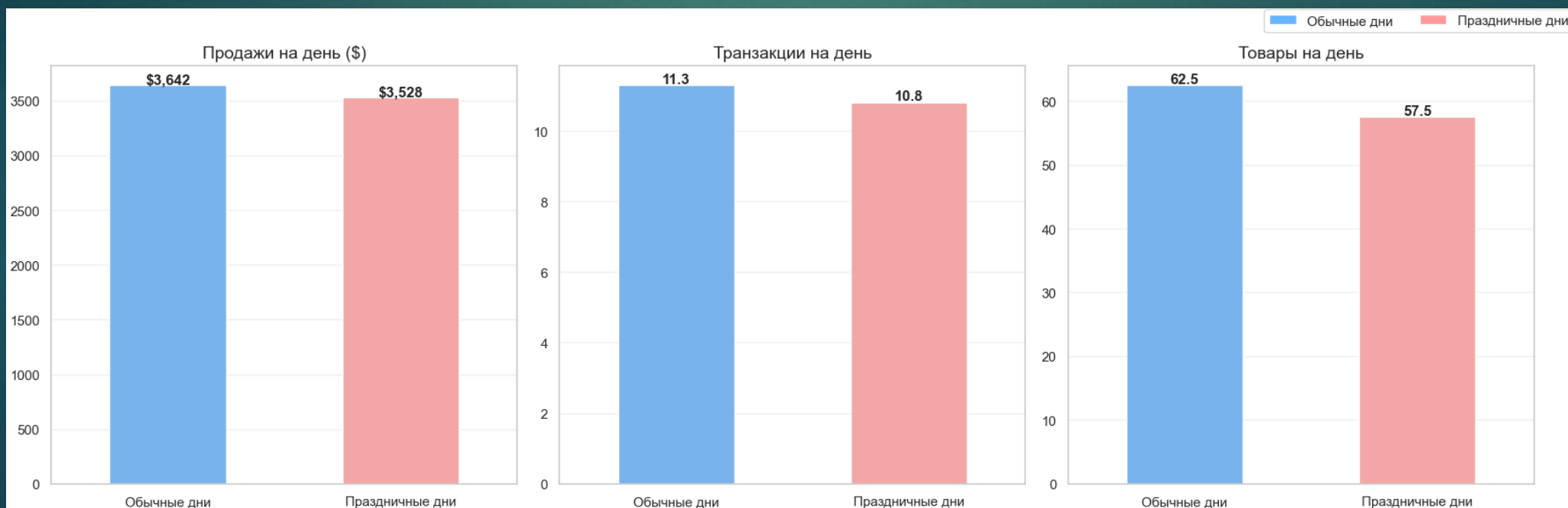
6. Анализ внешних факторов (API)

- ▶ На долю праздничных дней приходится около 11 % продаж
- ▶ В праздники растет средний чек и максимальная и минимальная суммы покупок



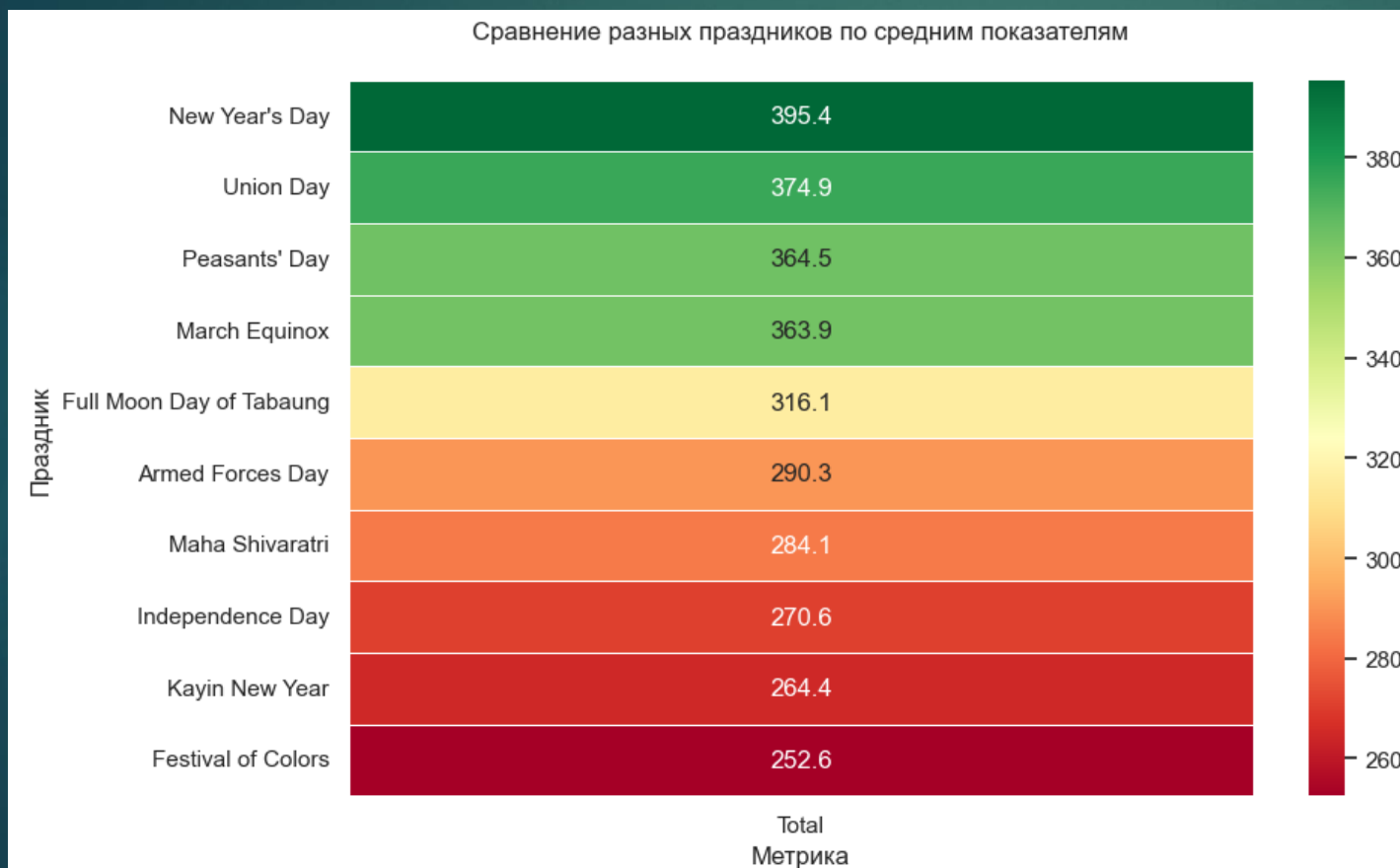
6. Анализ внешних факторов (API)

- ▶ Однако, общее количество продаж в праздничные дни падает



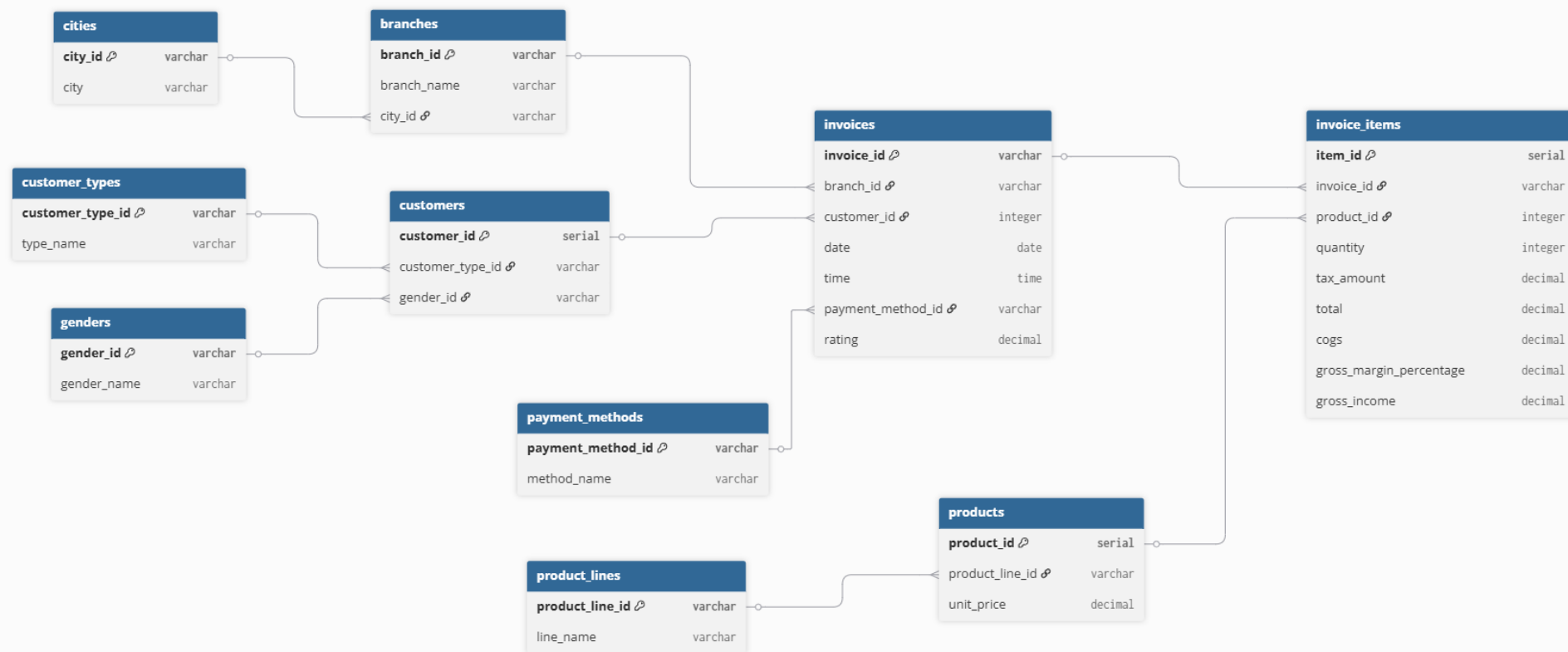
6. Анализ внешних факторов (API)

- ▶ Наиболее прибыльным праздником является Новый Год



7. NDS

► Проектируем NDS слой хранения данных



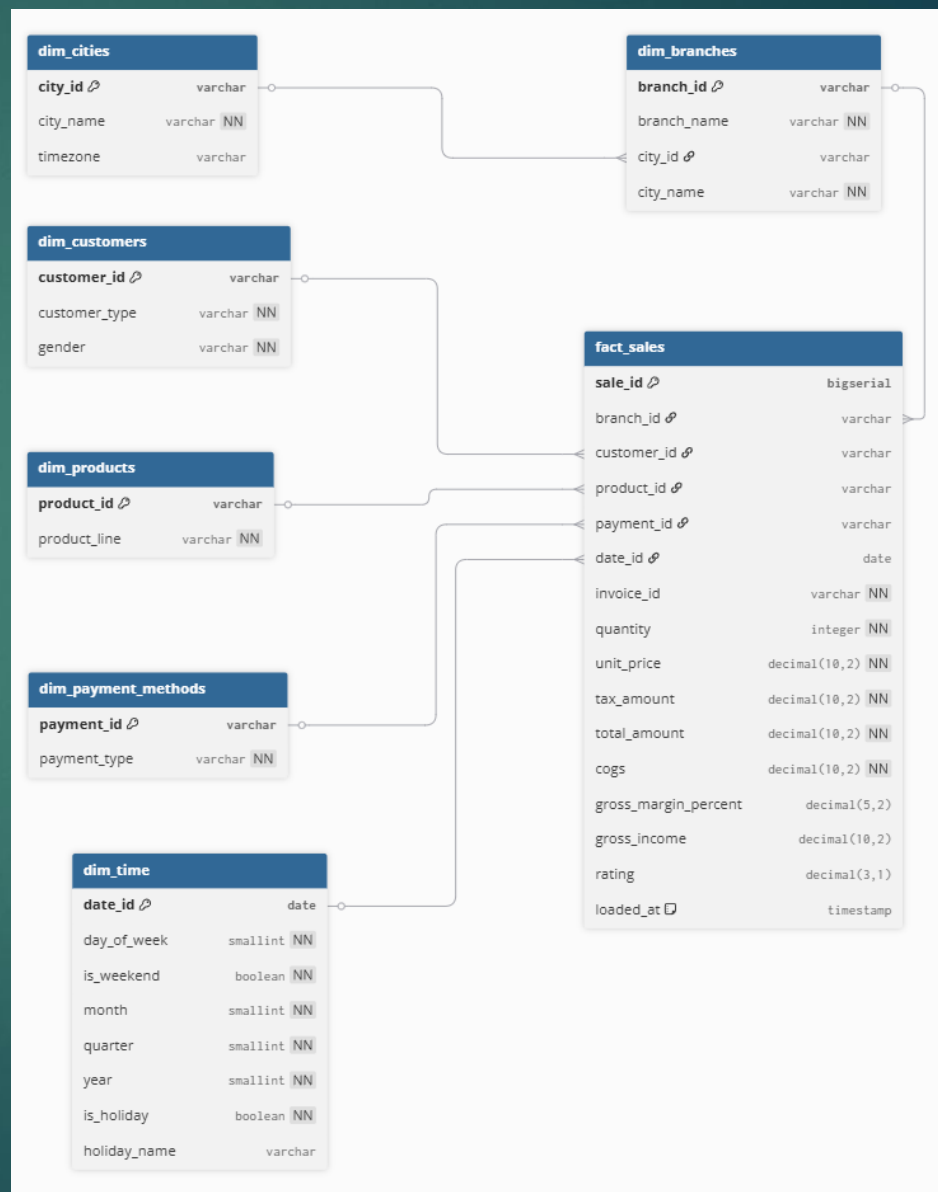
7. DDS

► Проектируем DDS слой хранения данных



7. DDS

► Проектируем DDS слой хранения данных




8. Docker



































▶ Запускаем том docker с airflow и clickhouse

```
PS C:\NetoDiplom> docker compose up -d
time="2025-08-11T15:16:28+05:00" level=warning msg="C:\\NetoDiplo
m\\docker-compose.yml: the attribute 'version' is obsolete, it wi
ll be ignored, please remove it to avoid potential confusion"
[+] Running 7/7
✓Container netodiplom-redis-1           Running      0.0s
✓Container netodiplom-postgres-1        Healthy      0.5s
✓Container netodiplom-clickhouse-1       Started      0.2s
✓Container netodiplom-airflow-init-1     Started      0.2s
✓Container netodiplom-airflow-worker-1   Started      0.1s
✓Container netodiplom-airflow-webserver-1 Started      0.2s
✓Container netodiplom-airflow-scheduler-1 Started      0.2s
```

Container CPU usage ⓘ 4.50% / 2000% (20 CPUs available)

Container memory usage ⓘ 2.39GB / 15.16GB [Show charts](#)

Search  ☐ Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last st	Actions
<input type="checkbox"/>	 netodiplom	-	-	-	4.57%	1 minut	  
<input type="checkbox"/>	 clickhouse-1	938769d4a36e	yandex/clic	8123:8123  Show all ports (3)	1.36%	1 minut	  
<input type="checkbox"/>	 redis-1	c9ffecdd7217	redis:latest		0.21%	2 minut	  
<input type="checkbox"/>	 postgres-1	e3d28b8af252	postgres:1		0.71%	2 minut	  
<input type="checkbox"/>	 airflow-init-1	b95153386f3e	apache/airf		0%	1 minut	  
<input type="checkbox"/>	 airflow-webs	7f145fbe1b26	apache/airf	8080:8080 	0.12%	1 minut	  
<input type="checkbox"/>	 airflow-worke	0956e7220a43	apache/airf		0.18%	1 minut	  
<input type="checkbox"/>	 airflow-sched	46031549d6f7	apache/airf		1.99%	1 minut	  

9. Dag.NDS

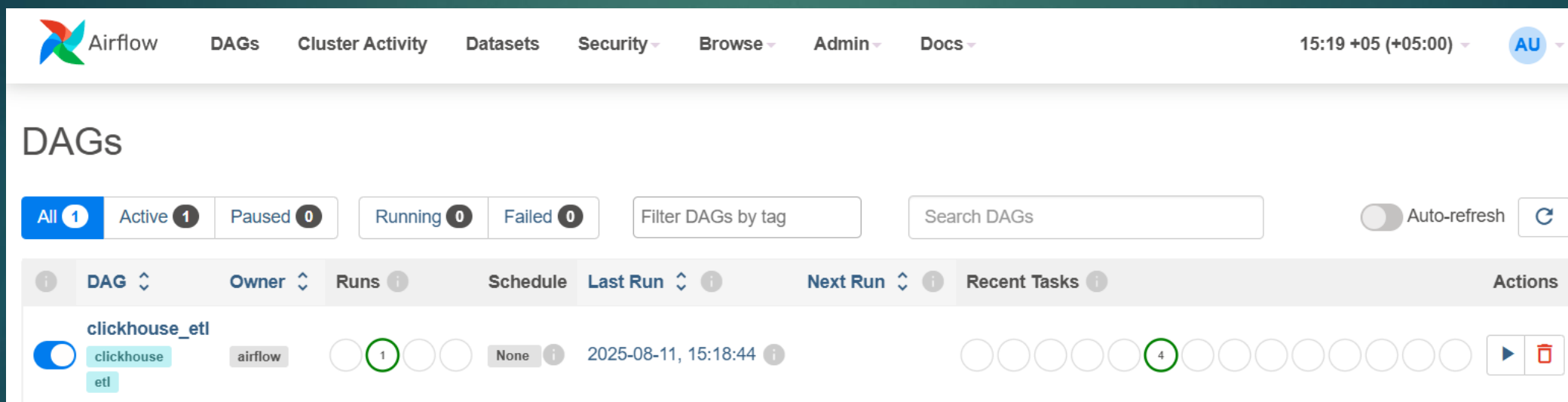
- ▶ **Инициализация структуры БД (init_clickhouse_db)**
 - ▶ Создает базы данных retail_nds (сырой слой) и retail_dwh (витрины данных).
 - ▶ Определяет таблицы для нормализованных данных (города, клиенты, продукты и т.д.) и витрин (измерения и факты).
 - ▶ Использует движок ReplacingMergeTree для автоматического удаления дубликатов.
- ▶ **Загрузка сырых данных (load_raw_data_to_nds)**
 - ▶ Читает CSV-файл (sales_data.csv), преобразует даты и проверяет обязательные колонки.
 - ▶ Загружает данные в таблицу raw_sales слоя NDS.
- ▶ **Нормализация данных (load_normalized_data_to_nds)**
 - ▶ Заполняет справочники (города, типы клиентов) и основные таблицы (филиалы, инвойсы) из raw_sales.
 - ▶ Использует хеширование (cityHash64) для генерации ID.

9. Dag.DDS

- ▶ Подготовка измерений (transform_and_load_to_dds)
 - ▶ Таблица времени (dim_time):
 - ▶ Извлекает день недели, месяц, квартал, отмечает праздники (через API HolidayAPI).
 - ▶ Другие измерения:
 - ▶ dim_cities, dim_branches (филиалы с регионами), dim_products (сегменты цен).
- ▶ Связь задач в Airflow: Линейный граф (init_db → raw → normalized → DDS).

10. Airflow

- ▶ Запускаем dag в airflow
- ▶ Запуск прошел успешно



The screenshot displays the Apache Airflow web interface. At the top, there is a navigation bar with the Airflow logo and several menu items: DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs. On the right side of the navigation bar, the current time is shown as 15:19 +05 (+05:00) and a user profile icon labeled 'AU'.

Below the navigation bar, the main heading is 'DAGs'. Under this heading, there are several filters and controls:

- Buttons for DAG status: All (1), Active (1), Paused (0), Running (0), and Failed (0).
- A text input field labeled 'Filter DAGs by tag'.
- A search input field labeled 'Search DAGs'.
- An 'Auto-refresh' toggle switch and a refresh icon.

The main content area is a table with the following columns: DAG, Owner, Runs, Schedule, Last Run, Next Run, Recent Tasks, and Actions.

The first row of the table shows a DAG named 'clickhouse_etl' with the following details:

- DAG:** clickhouse_etl (with a toggle switch set to 'On').
- Owner:** airflow.
- Runs:** A series of circles representing task runs. The first circle is highlighted with a green border and contains the number '1'.
- Schedule:** None.
- Last Run:** 2025-08-11, 15:18:44.
- Next Run:** (empty).
- Recent Tasks:** A series of circles representing task instances. The fourth circle is highlighted with a green border and contains the number '4'.
- Actions:** A play button and a trash icon.

11. Power BI

- ▶ Подключаемся к clickhouse через Power BI (в Tableau Community нет соответствующего коннектора)
- ▶ Убеждаемся что оба слоя прогрузились в ClickHouse, все таблицы на месте

Получить данные

✕

Все

База данных

Все

ClickHouse

ClickHouse

Host ⓘ

Port ⓘ

Database (необязательно) ⓘ

Режим подключения к данным ⓘ

☐ импорт

☒ DirectQuery

Навигатор

Отобразить параметры ▾

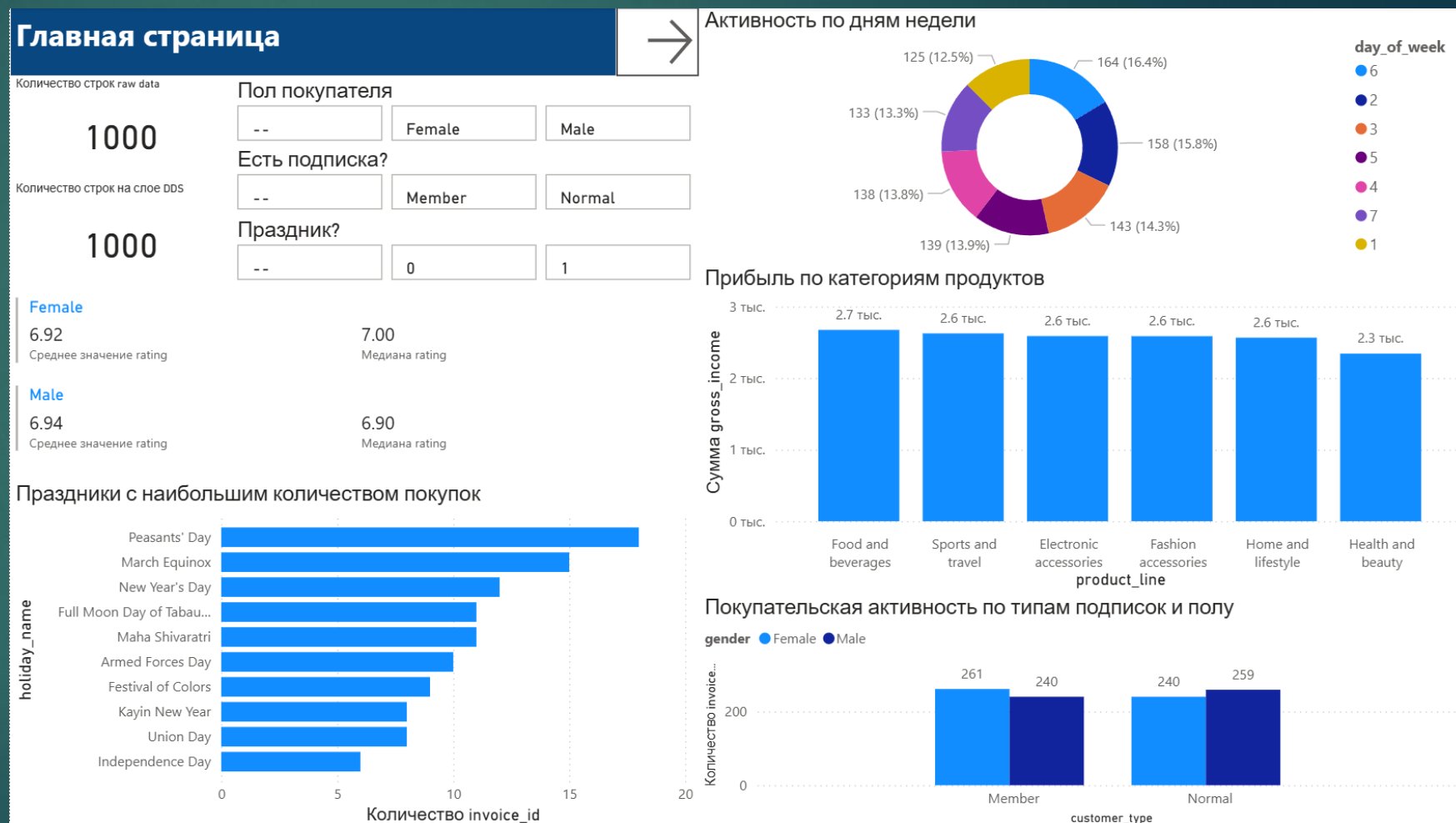
- localhost: 8123 [4]
 - default
 - retail_dwh [7]
 - ☐ dim_branches
 - ☐ dim_cities
 - ☐ dim_customers
 - ☐ dim_payment_methods
 - ☐ dim_products
 - ☐ dim_time
 - ☐ fact_sales
 - retail_nds [11]
 - ☐ branches
 - ☐ cities
 - ☐ customer_types
 - ☐ customers
 - ☐ genders
 - ☐ invoice_items
 - ☐ invoices
 - ☐ payment_methods
 - ☐ product_lines

Элементы для предпросмотра не выбраны.

OK Отмена

11. Power BI

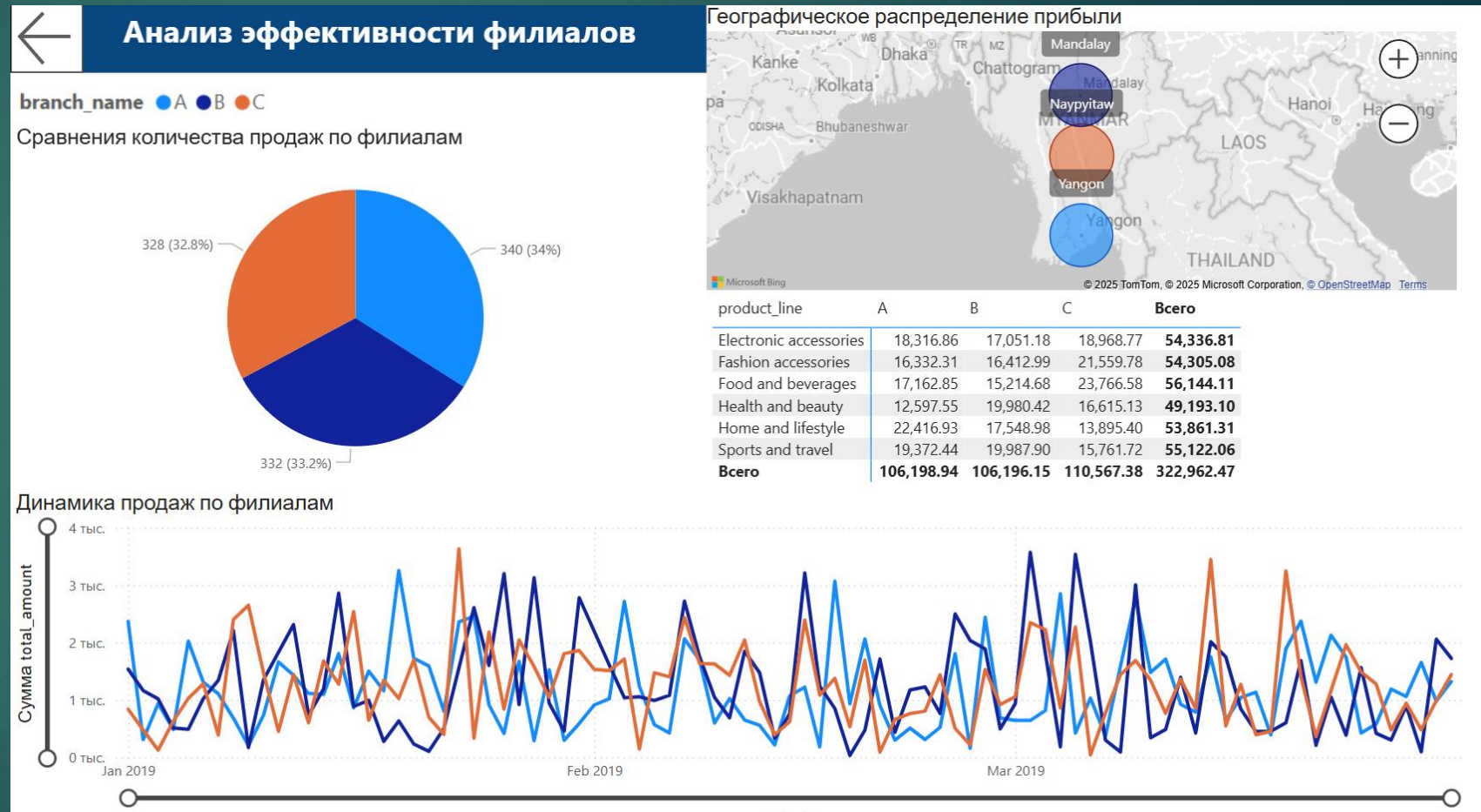
- ▶ Делим дашборды на 2 страницы
- ▶ На главной странице:
 - ▶ Проверка загрузки данных
 - ▶ Анализ поведения разных типов покупателей
 - ▶ Анализ продаж по праздничным дням и разным дням недели
 - ▶ Анализ продаж по типам покупателей



11. Power BI

▶ На страницы филиалов :

- ▶ Сравнение показателей филиалов
- ▶ Распределение прибыли с привязкой к географии
- ▶ Распределение продаж в зависимости от филиала и продуктовой линейки
- ▶ Динамика продаж



ВЫВОДЫ

- ▶ 1. Достижение цели
 - ▶ В рамках дипломного проекта успешно реализован ETL-процесс для розничных продаж, включая:
 - ▶ Обработку и анализ исходных данных (проверка на пропуски, добавление временных колонок, расчет метрик).
 - ▶ Проектирование NDS (нормализованный слой) и DDS (витрины данных) в ClickHouse.
 - ▶ Автоматизацию ETL с помощью Airflow (даг clickhouse_etl), обеспечивающий загрузку данных в NDS и преобразование в DDS.
 - ▶ Интеграцию внешних данных (праздники через HolidayAPI) для расширенной аналитики.
- ▶ 2. Ключевые результаты анализа данных
 - ▶ Филиалы и города:
 - ▶ Филиал С (Nauryitaw) — самый прибыльный и рейтинговый.
 - ▶ Каждому городу соответствует один филиал, что упрощает анализ.
 - ▶ Товары:
 - ▶ Наиболее прибыльная категория — еда, наименее популярная — «Здоровье и красота».
 - ▶ Маржинальность товаров равномерная, топ-5 товаров продаются с разницей ~1%.
 - ▶ Покупатели:
 - ▶ У мужчин без подписки рейтинг выше, но женщины без подписки покупают дороже.
 - ▶ Подписка не влияет на рейтинг значимо.
 - ▶ Внешние факторы:
 - ▶ В праздники (11% продаж) средний чек растет, но количество покупок снижается.
 - ▶ Самый прибыльный праздник — Новый Год.

ВЫВОДЫ

- ▶ 3. Техническая реализация
 - ▶ ClickHouse:
 - ▶ Оптимизированное хранение с ReplacingMergeTree для борьбы с дубликатами.
 - ▶ Разделение на NDS (нормализованные таблицы) и DDS (витрины для аналитики).
 - ▶ Airflow:
 - ▶ Линейный даг с этапами: инициализация БД → загрузка в NDS → нормализация → формирование DDS.
 - ▶ Обработка ошибок (повторные попытки, логирование)
 - ▶ Визуализация:
 - ▶ Дашборды в Power BI с анализом продаж по филиалам, праздникам, типам покупателей.
- ▶ 4. Ограничения
 - ▶ Недостаточно данных для сезонного анализа (только 2019 год).
 - ▶ API праздников предоставил данные за 2024 год, что потребовало условного сопоставления.
- ▶ Проект подтвердил эффективность ETL-подхода для розничной аналитики: от сырых данных до готовых метрик. Решение готово к масштабированию — например, для обработки данных из новых филиалов или интеграции с CRM.