## Общая структура

ETL процедуры разработаны с использованием Pentaho Data Integration. Преобразования включают несколько трансформаций (transformation), выполнение которых прописано в соответствующем задании (job).

Трансформации можно разделить на две группы:

* Создание/обновление измерений
* Заполнение таблиц данных

Создание таблиц измерений и таблицы фактов вынесено в отдельный SQL-скрипт.

Процедура выполняется на локальном инстансе PostgreSQL в базе данных «avia», схеме «bookings».

## Основное задание (job)

Основное задание представляет собой задачи по последовательному выполнению трансформаций в рамках ETL-процедуры

Первые 4 трансформации – заполнение и обновление таблиц измерений. При этом каждое измерение вынесено в отдельную трансформацию. Это упрощает мониторинг выполнения трансформаций и их редактирование.

Оставшиеся трансформации – формирование таблицы фактов. При этом формирование таблицы фактов осуществляется в 2 шага:

* Формирование данных о выполненных перелетах
* Формирование данных о пассажирах и билетах

## Трансформации измерений (dimension)

В рамках ETL-процесса используются следующие таблицы измерений:

* dim\_calendar
* dim\_passenger
* dim\_aircrafts
* dim\_airports
* dim\_tariff

### Calendar

Содержит данные о датах с указанием ряда дополнительных признаков:

* Год
* Месяц
* День
* День недели
* Номер недели в году
* Признак выходного/праздничного дня

Данная таблица формируется с помощью SLQ-скрипта.

### Passenger

Содержит данные о пассажире:

* Идентификатор записи (id, serial)
* Уникальный код пассажира (passenger\_code, varchar)
* Имя и фамилия (name, varchar)
* Контактные данные (contact\_data, jsonb)

Источник данных – tickets.

Проверки данных:

1. Для пассажира указан код (не пустая строка)
2. Указано имя пассажира (не пустая строка)
3. Указаны контактные данные (не пустая строка)

Строки с ошибками записываются в таблицу – rej\_tickets с указанием времени добавления записи и причины (поля) ошибки.

### Aircrafts

Содержит данные о самолетах:

* Идентификатор записи (id, serial)
* Уникальный код самолета (aircraft\_code, varchar)
* Модель самолета (model, varchar)
* Дальность полета (range, int)

Источник данных – aircrafts.

Проверки данных:

1. Код самолета содержит ровно 3 символа
2. Указана модель самолета (не пустая строка)
3. Дальность полет больше ноля

Строки с ошибками записываются в таблицу – rej\_aircrafts с указанием времени добавления записи и причины (поля) ошибки.

### Airports

Содержит данные о самолетах:

* Идентификатор записи (id, serial)
* Уникальный код аэропорта (airport\_code, varchar)
* Название (airport\_name, varchar)
* Город (city, varchar)
* Долгота и широта (longitude, latitude, double)
* Часовой пояс (timezone, varchar)

Источник данных – airports.

Проверки данных:

1. Код аэропорта содержит ровно 3 символа
2. Указан город (не пустая строка)
3. Указаны координаты (долгота и широта)

Строки с ошибками записываются в таблицу – rej\_airports с указанием времени добавления записи и причины (поля) ошибки.

### Tariff

Содержит данные о классах обслуживания:

* Идентификатор записи (id, serial)
* Название тарифа (tariff, varchar)

Источник данных –ticket\_flights.

Проверки данных:

1. Указано название тарифа (не пустая строка)

Строки с ошибками записываются в таблицу – rej\_ticket\_flights с указанием времени добавления записи и причины (поля) ошибки.

## Трансформации таблицы фактов (fact)

В связи с большим количеством записей о пассажирах и билетах, принято решение разделить формирование таблицы фактов на два шага.

На первом шаге формируется таблица с данными о фактически выполненных перелетах (с установкой связей на измерения).

На втором шаге формируется массив данных о пассажирах, классе обслуживания и стоимости перелета. Также на этом шаге производится объединение данных о перелетах и данных о пассажирах и билетах.

### Flights

Формируется массив строк, содержащий:

* Идентификатор перелета
* Дату вылета (внешний ключ, dim\_calendar, departure\_date)
* Время вылета (departure\_time, timestamp)
* Задержку вылета в секундах (departure delay, int)
* Дату прибытия (внешний ключ, dim\_calendar, arrival\_date,)
* Время прибытия (arrival\_time, timestamp)
* Задержку прибытия в секундах (arrival\_delay, int)
* Код самолета (внешний ключ, dim\_aircrafts, aircraft\_code)
* Код аэропорта вылета (внешний ключ, dim\_airports, departure\_airport)
* Код аэропорта прибытия (внешний ключ, dim\_airports, arrival\_airport)

Данный массив передается на следующий этап заданий для добавления данных о пассажирах.

Источник данных - flights

Проверки данных:

1. Указаны фактические даты вылета и прибытия
2. Дата/время прибытия больше чем дата/время вылета
3. Для рейса указан статус «arrived»

Строки с ошибками записываются в таблицу – rej\_flights с указанием времени добавления записи и причины (поля) ошибки.

### Tickets+Flights

Изображение выглядит как стол

Автоматически созданное описание

На данном этапе формируется массив данных о пассажирах и билетах:

* Идентификатор перелета
* Код пассажира (внешний ключ, dim\_passengers, passenger\_code)
* Класс обслуживания (внешний ключ, dim\_tariff, tariff)
* Стоимость билета (cost)

Источники данных:

* Tickets
* Ticket\_flights

К полученному массиву данных добавляется информация о выполненных перелетах. Связь осуществляется по идентификатору перелета (flight\_id).

Полученный массив данных записывается в результирующую таблицу – fact\_flights.