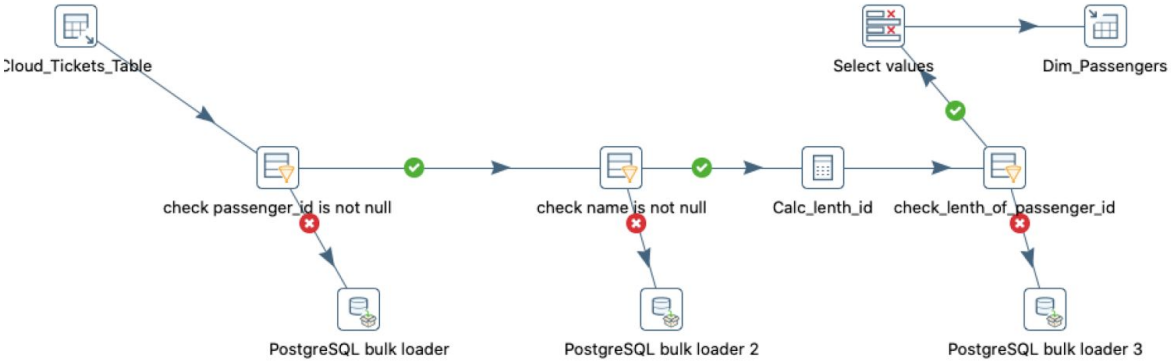
Промежуточное задание DWH

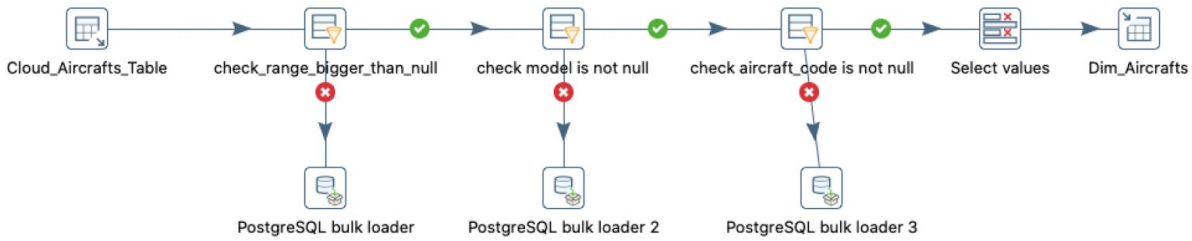
ETL состоит из 6 трансформаций:
Dim_Passengers - создание справочника пассажиров
Dim_Aircrafts - создание справочника самолетов
Dim_Airports - создание справочника аэропортов
Dim_Tariff - создание справочника тарифов
Stage_Flights - создание промежуточной версии таблицы фактов в БД
Fact_Flights - обновление ссылок на справочники и запись в БД

Dim_Passengers



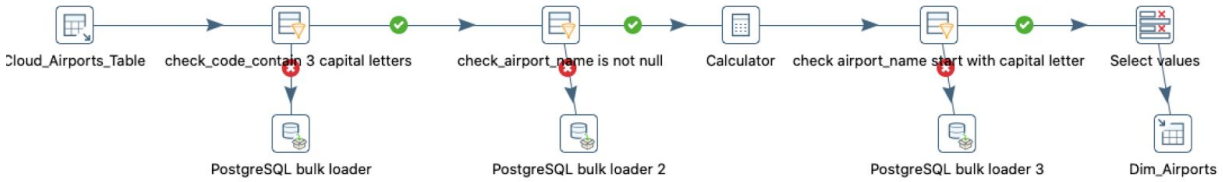
Забираем данные из облака яндекс из таблицы tickets и проверяем на то что passenger_id(пасспорт) не null, имя не пустое, а также что длина номера пасспорта соответствует 13 символам. Далее выбираем поля для записи в таблицу - это номер пасспорта и имя и фамилия

Dim_Aircrafts



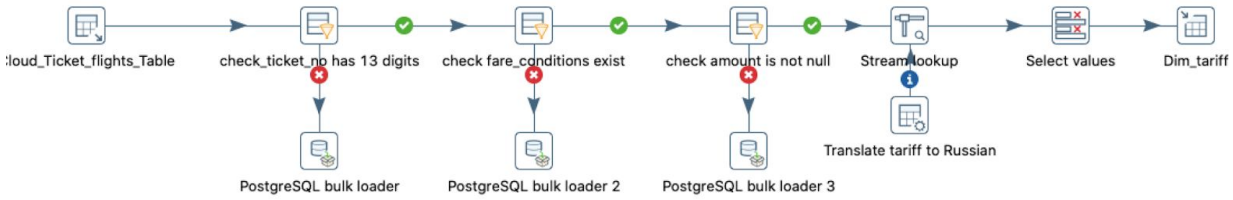
Забираем данные из облака яндекс и таблицы aircrafts. Проверяем на то чтобы дальность полета (range) была больше нуля, чтобы была модель самолета и существовал код самолета. Выбираем поля для записи в таблицу aircraft_code, model,range

Dim_Airports



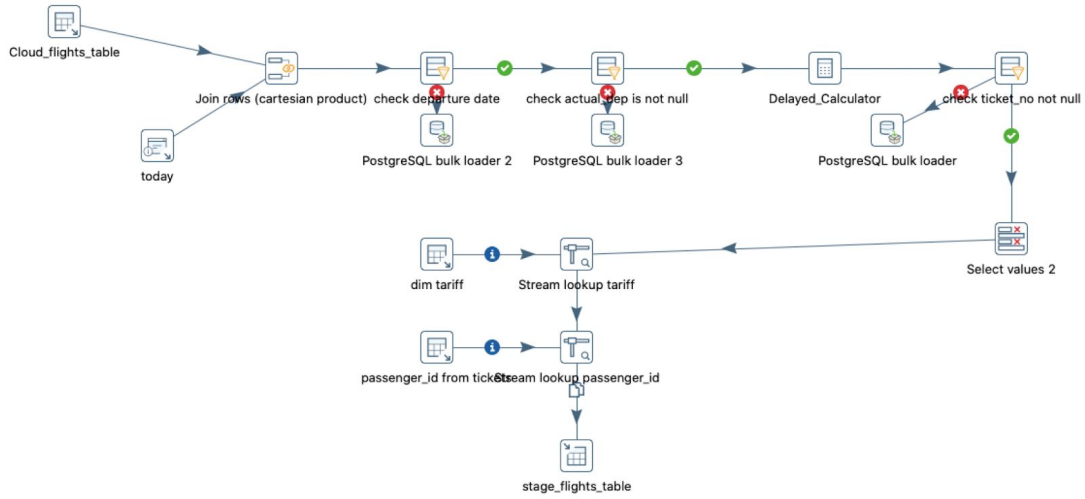
Забираем данные из облака яндекс и таблицы airports. Проверяем с помощью регулярного выражения то чтобы код аэропорта был с 3 заглавных букв, а также чтобы название аэропорта начиналось с заглавной буквы. Третья проверка на то чтобы название аэропорта было не пустое. Записываем в таблицу значения airport_code, name, city.

Dim_Tariff



Забираем данные из облака яндекс и таблицы tickets_fligts. Делаем проверку на то что номер билета состоит из 13 цифр, что есть указание на класс, а также что стоимость билета не пустая, также переводим класс с английского на русский используя stream lookup и данные для перевода. Записываем в таблицу ticket_no, flight_id, tariff, price.

Stage_Flights



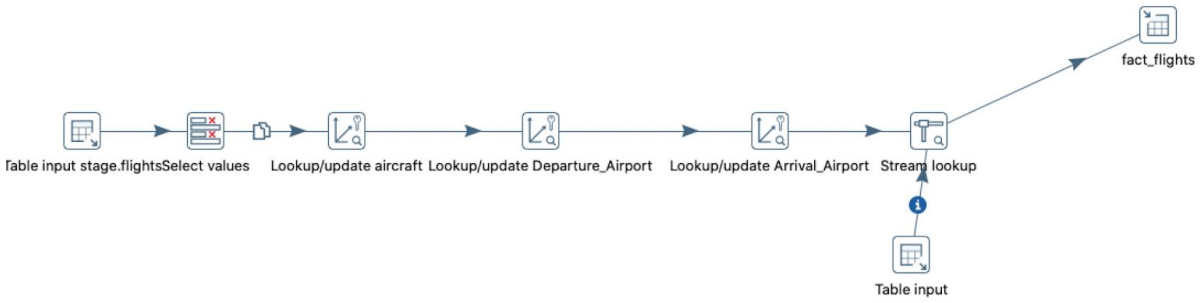
Забираем данные из облака яндекс и таблицы flights, которая соединена в запросе с

таблицей ticket_flights с помощью left join, а также выбраны значения только прилетевших самолетов, так как в задании нужны только совершенные перелеты

```
select *
from flights
left join ticket_flights using (flight_id)
where status='Arrived'
order by flight_id asc;
```

Добавлена с помощью join rows сегодняшняя дата и проведена проверка что дата отправления в прошлом, на то что значение actual departure не пустое, произведен подсчет с помощью калькулятора значения задержки фактического вылета и прилета в секундах. Далее произведена проверка, на то что ticket_no для совершенного полета не содержит пустое значение. С помощью первого stream lookup добавляем информацию из справочника тарифов производя lookup по полю ticket_no. С помощью второго stream lookup добавляем паспорт пассажира, также по полю ticket_no. Записываем получившиеся значения в таблицу stage.films при этом поменяв формат даты на yyyyDDmm, убрав actual departure and arrival. Данный stage был создан для того, чтобы не терять в производительности, так как более быстрые шаги вначале ждут пока пройдет lookup\update и скорость записи стремительно уменьшается.

Fact_Flights



Читаем из таблицы stage.fligths и обновляем ссылку на ключ с помощью lookup/update для самолета, аэропортов вылета и прилета. Для ключа пассажиров пришлось использовать stream lookup так как в таком случае производительность намного выше. Записываем обновленную информацию в таблицу фактов fact.flights.

Все трансформации объединены в одно задание

Требуется около 25 минут для создания всех справочников и таблицы фактов.