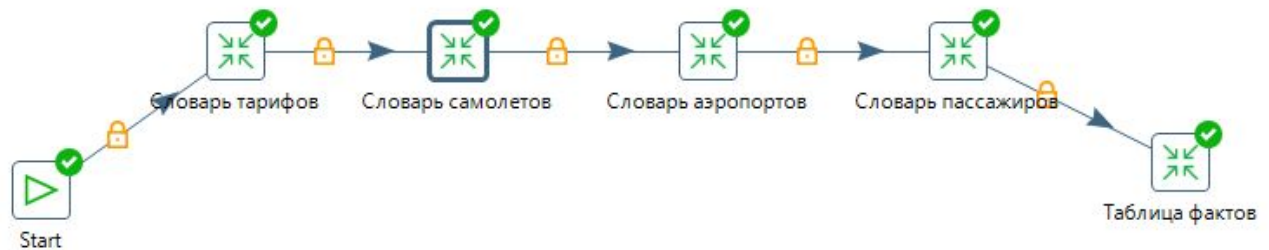


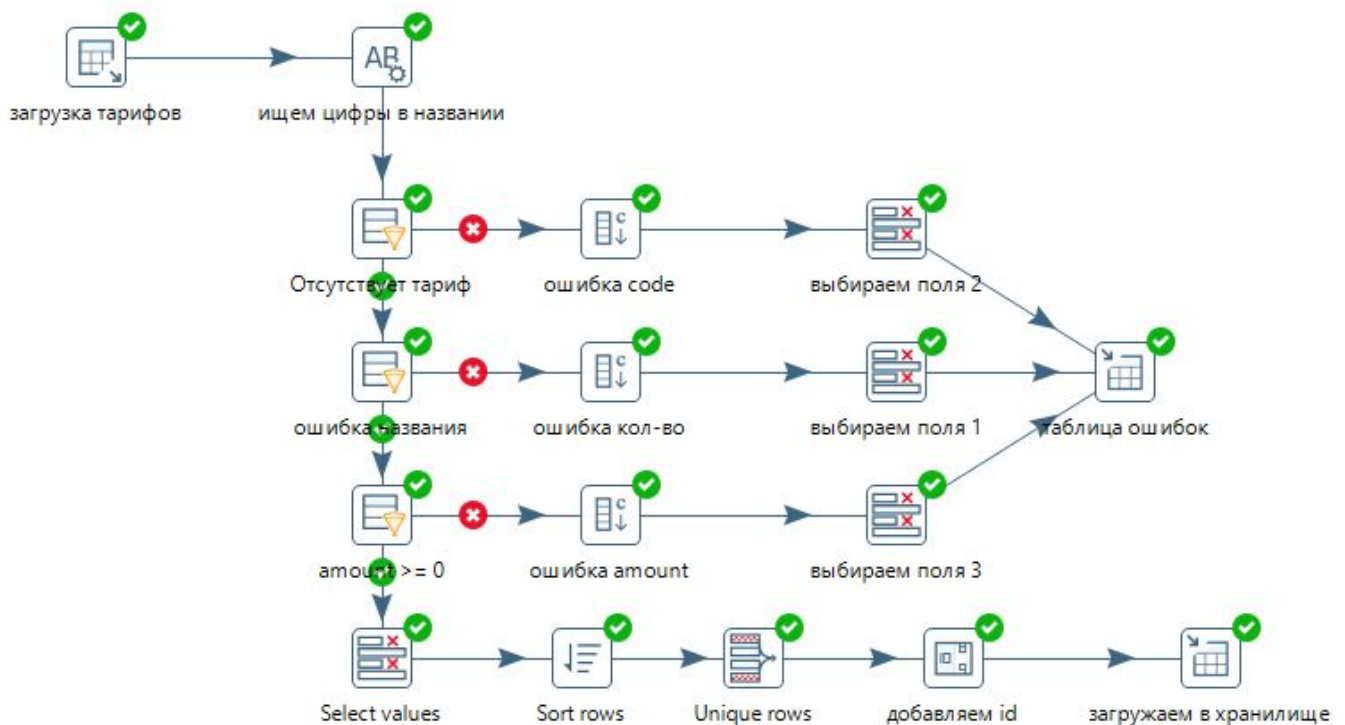
# Документация ETL-загрузки данных полетов из схемы bookings

В основе ETL находится job ETL.kjb



Первые 4 шага после Start - это загрузка словарей в хранилище. Последний - загрузка данных в таблицу фактов

## 1. Словарь тарифов

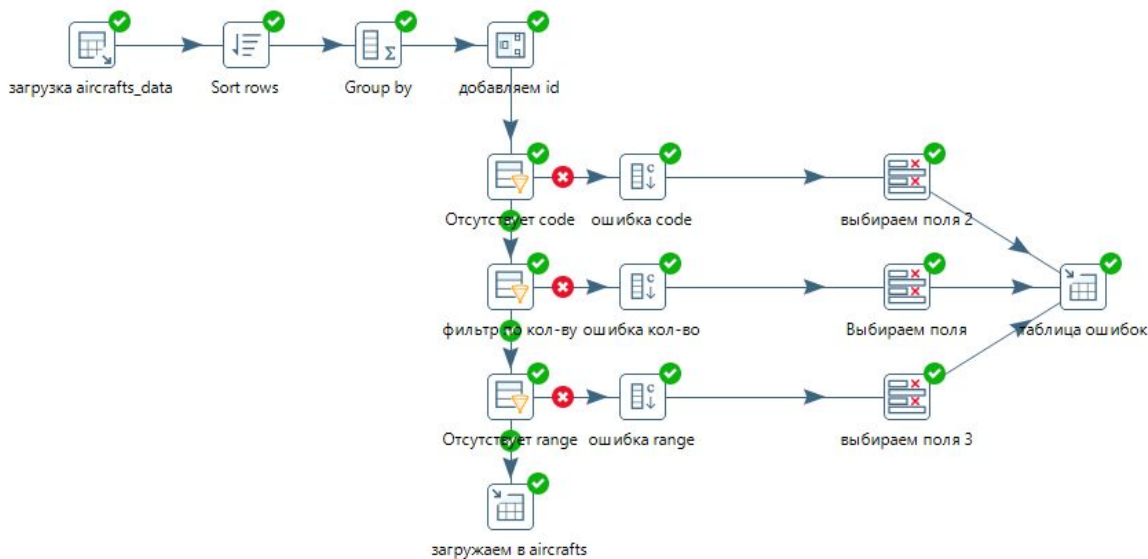


Первым шагом подключаемся к таблице **bookings.ticket\_flights** и достаём уникальные записи тарифов. Далее добавляем уникальный id и загружаем в таблицу хранилища **dim.tariff** после проверки.

Перед загрузкой делаем три проверки на качество данных:

1. Проверяем наличие записи о тарифе
2. Проверяем, есть ли ошибки в названии в виде цифр. Т.е. могли ли ввести лишние значения
3. Проверяем, что стоимость  $\geq 0$ , чтобы исключить отрицательные значения.

## 2. Словарь самолетов

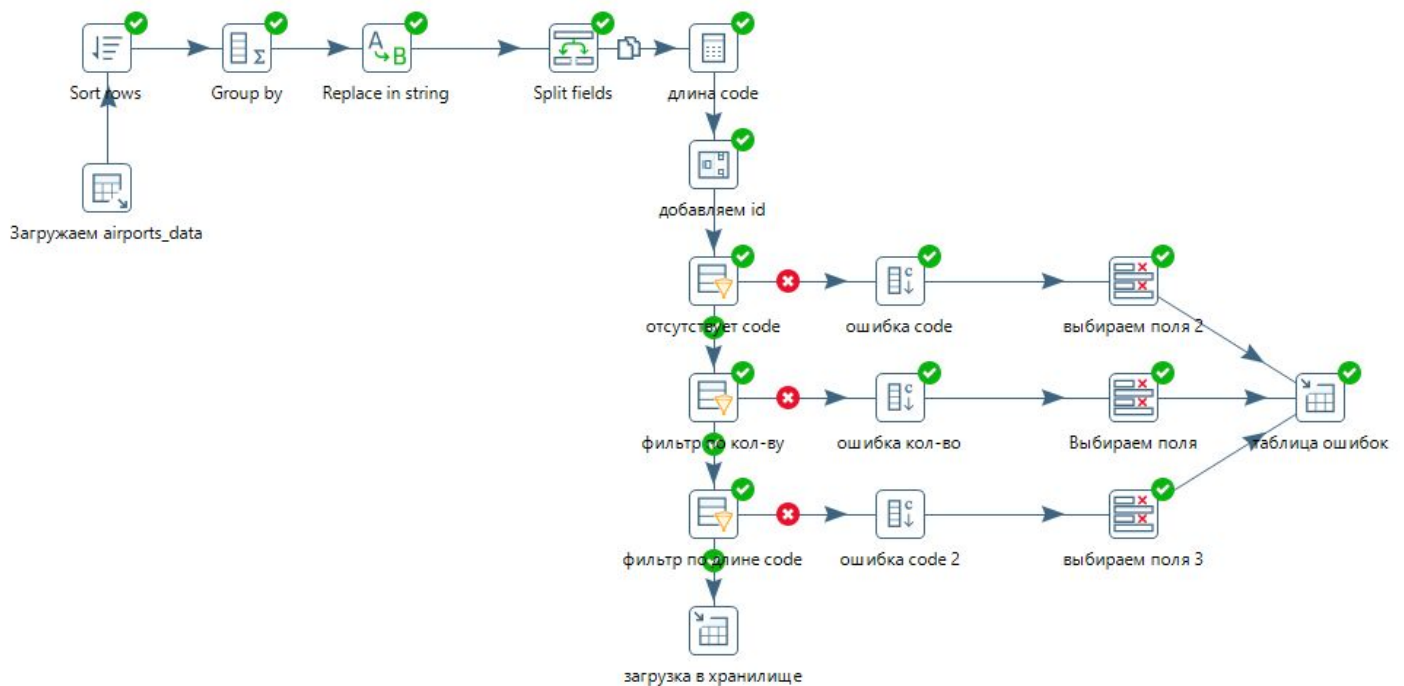


Первым шагом подключаемся к таблице **bookings.aircrafts\_data** и добавляем кол-во записей по code для проверки на лишние записи-дубли. Далее добавляем уникальный id и загружаем в таблицу хранилища **dim.aircrafts** после проверки.

Перед загрузкой делаем три проверки на качество данных:

1. Проверяем наличие code в записи.
2. Проверяем, есть ли ошибки в виде задвоений записей по code.
3. Проверяем, что дальность полета заполнена.

## 4. Словарь аэропортов

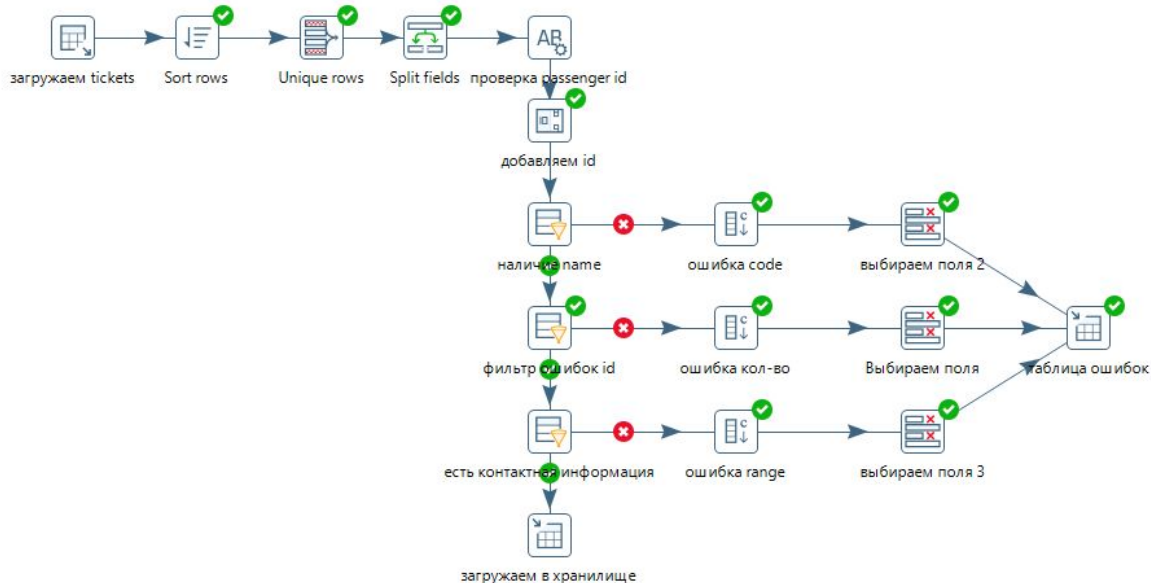


Первым шагом подключаемся к таблице **bookings.airports\_data**, разбиваем координаты на две части, проверяем длину code и добавляем кол-во записей по code для проверки на лишние записи-дубли. Далее добавляем уникальный id и загружаем в таблицу хранилища **dim.airports** после проверки.

Перед загрузкой делаем три проверки на качество данных:

1. Проверяем наличие code в записи.
2. Проверяем, есть ли ошибки в виде задвоений записей по code.
3. Проверяем, что все code длины в 3 символа, что соответствует стандартной кодировке.

## 5. Словарь пассажиров



Первым шагом подключаемся к таблице **bookings.tickets**. Берем уникальные записи для словаря, далее добавляем уникальный id, смотрим наличие непонятных цифр в именах и загружаем в таблицу хранилища **dim.passengers** после проверки.

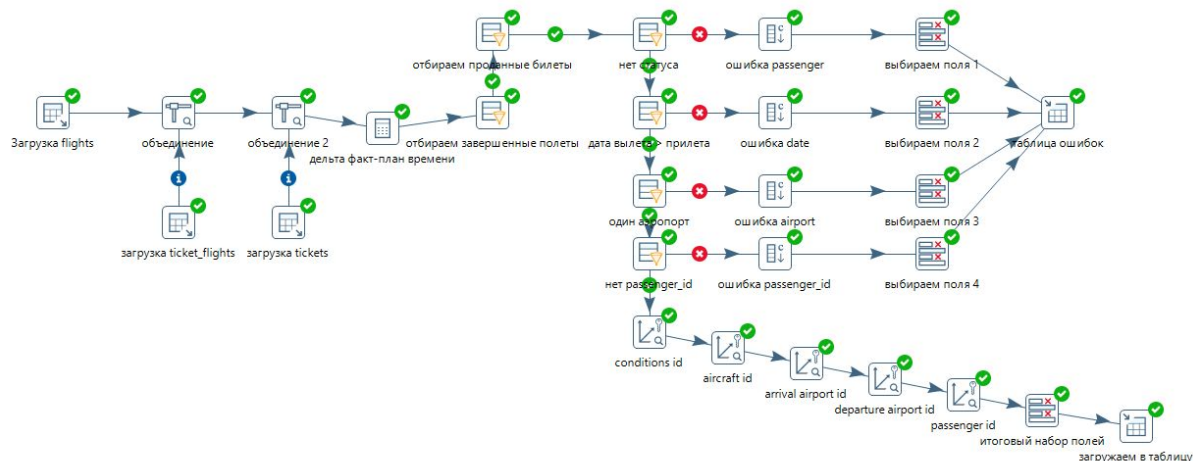
Перед загрузкой делаем три проверки на качество данных:

1. Проверяем наличие имени и фамилии.
2. Проверяем, есть ли ошибки в id в виде буквенных символов.
3. Проверяем, что заполнена контактная информация.

## 6. Словарь дат

Формируется в файле sql запросом.

## 7. Таблица фактов



Первым шагом загружаются данные из таблицы **bookings.flights**. Дальше к нему джойним таблицу **bookings.tickets\_flights** и **bookings.tickets**. Рассчитываем дельту между временем прибытия и вылета в секундах. Фильтруем только завершённые полеты по наличию даты прибытия. Фильтруем только проданные билеты по одновременному наличию данных book\_ref, ticket\_no и passenger\_id, т.е. убираем пустые места без пассажиров.

Перед загрузкой делаем проверку качества данных:

1. Проверяем заполненного поля статуса рейса.
2. Проверяем корректность дат, дата вылета раньше даты прибытия.
3. Проверяем отсутствие задвоения в аэропортах отправления и прибытия. Не должно быть одного и того же значения.
4. Проверяем наличие passenger\_id. Т.е. должен быть пассажир

Дальше 5 шагов заменяют id в таблице на технический id из словарей. Выбираем итоговый набор полей для загрузки и загружаем в таблицу.