Описание ETL

**Наполнение Dim\_Calendar - справочник дат**

Скриптом SQL

**Наполнение Dim\_Passengers - справочник пассажиров**

1. Шаг Table input

SELECT passenger\_id, passenger\_name, contact\_data->>'phone' as phone, contact\_data->>'email' as email FROM tickets

1. Шаг Data validation - проверки
2. passenger\_name

- regex expected to match: [\w+\s]+ (только буквы и пробелы)

- not null

- max string length: 400

1. passenger\_id

- regex expected to match: \d{4}\s\d{6} (шаблон 4 цифры пробел 6 цифр)

- not null

1. Шаг Table output для записи некорректных данных в таблицу passengers\_reject
2. Шаги Sort rows + Unique rows – отбор только уникальных значений для первичного ключа passenger\_id
3. Шаг Table output – запись в таблицу Dim\_Passengers
4. Шаг Table output log – ошибки заполнения таблицы на шаге 5 записываются в таблицу output\_log

**Наполнение Dim\_Aircrafts - справочник самолетов**

1. Шаг Table input   
   SELECT aircraft\_code as aircraft\_code, model->>'ru' as aircraft\_model, range as aircraft\_range FROM aircrafts\_data
2. Шаг Data validation - проверки
3. aircraft\_code

- regex not allowed to match: \W (без спецсимволов)

- not null

- max string length: 100

1. aircraft\_model

- not null

- max string length: 100

с) range

- only numeric data expected

1. Шаг Table output Reject для записи некорректных данных в таблицу aircrafts\_reject
2. Шаги Sort rows + Unique rows – отбор только уникальных значений для первичного ключа aircraft\_code
3. Шаг Select values Range to Str – range преобразуется в string
4. Шаг Table output – запись в таблицу Dim\_ Aircrafts
5. Шаг Table output log – ошибки заполнения таблицы на шаге 6 записываются в таблицу output\_log

**Наполнение Dim\_Airports - справочник аэропортов**

1. Шаг Table input   
   SELECT airport\_code as airport\_code, airport\_name->>'ru' as airport\_name, city->>'ru' as city, coordinates::varchar as coordinates,timezone as timezone FROM airports\_data
2. Шаг Data validation - проверки
3. airport\_code

- regex not allowed to match: \W

- not null

- max string length: 100

1. airport\_name

- not null

- max string length: 200

- regex expected to match: [А-Я|а-я|ё|Ё|\-\*|\s\*]+ (русские буквы с тире и пробелами)

с) city

- max string length: 200

- regex expected to match: [А-Я|а-я|ё|Ё|\-\*|\s\*]+

1. Шаг Table output Reject для записи некорректных данных в таблицу airports\_reject
2. Шаги Sort rows + Unique rows – отбор только уникальных значений для первичного ключа airport\_code
3. Шаг Replace in string – в строке coordinates убираются скобки \(|\)
4. Шаг Table output – запись в таблицу Dim\_Airports
5. Шаг Table output log – ошибки заполнения таблицы на шаге 6 записываются в таблицу output\_log

**Наполнение Dim\_Tariff - справочник тарифов**

1. Шаг Table input   
   select distinct (fare\_conditions) from ticket\_flights
2. Шаги Data Grid, Stream Lookup – дополнение кодов тарифов названиями
3. Шаг Data validation - проверки
4. fare\_conditions

- regex not allowed to match: \W

- not null

- max string length: 100

1. tariff\_name

- not null

- max string length: 200

1. Шаг Table output Reject для записи некорректных данных в таблицу tariff\_reject
2. Шаги Sort rows + Unique rows – отбор только уникальных значений для первичного ключа tariff\_code
3. Шаг Table output – запись в таблицу Dim\_Tariff
4. Шаг Table output log – ошибки заполнения таблицы на шаге 6 записываются в таблицу output\_log

**Наполнение Fact\_Flights – таблицы фактов о перелетах**

1. Шаг Table input

select t.passenger\_id as passenger\_id, f.actual\_departure as departure\_time, to\_char(f.actual\_departure, 'YYYYMMDD') as departure\_date, f.actual\_arrival as arrival\_time,to\_char(f.actual\_arrival, 'YYYYMMDD') as arrival\_date, extract(epoch from (f.actual\_departure-f.scheduled\_departure))::int4 as dep\_delay, extract(epoch from (f.actual\_arrival-f.scheduled\_arrival))::int4 as arr\_delay, f.aircraft\_code as aircraft\_code,f.departure\_airport AS dep\_airport, f.arrival\_airport as arr\_airport, tf.fare\_conditions as tariff\_code, tf.amount as amount

from tickets t join ticket\_flights tf on tf.ticket\_no = t.ticket\_no join flights f on tf.flight\_id = f.flight\_id

1. Шаг Select values – соответствие типов
2. Шаг If field value is null для полей dep\_delay и arr\_delay для задания 0 вместо null
3. Шаг Filter rows – проверка что departure\_time < arrival\_time или один из них null
4. Шаг Filter rows 2 – удаление строк, где departure\_time is null and arrival\_time is not null (строки отброшенные фильтрами отправляются в таблицу flights\_reject)
5. Шаги Table input Cal, Table input Pass, Table input Aic, Table input Aip, Table input Tar выбирают первичные ключи из справочников и передают их шагу 7.
6. Шаг Data validation - проверки
7. dep\_delay, и аналогично для arr\_delay   
   - Integer

- only numeric data expected

- maximum value: 30000 (подозрительно большая задержка, м.б. ошибка данных)

- minimum value: -1000 (большое опережение расписания, м.б. ошибка данных)

max string length: 100

1. departure\_date и arrival\_date  
   - список допустимых значений берется из поля id справочника дат

- null

1. passenger\_id

- список допустимых значений берется из поля passenger\_id справочника пассажиров

- not null

1. aircraft\_code

- список допустимых значений берется из поля aircraft\_code справочника самолетов

- null

1. dep\_airport и arr\_airport

- список допустимых значений берется из поля airport\_code справочника самолетов

- not null

1. tariff\_code

- список допустимых значений берется из поля tariff\_code справочника тарифов

- null

1. amount

- min 0, max 1000000

- null

1. Шаг Table output Reject для записи некорректных данных в таблицу flights\_reject (собирает в таблицу flights\_reject строки отброшенные шагами 4, 5 и 7)
2. Шаг Table output – запись в таблицу Fact\_Flights
3. Шаг Table output log – ошибки заполнения таблицы на шаге 9 записываются в таблицу output\_log