Traffic Forecaster Design Notes

Tim Rowe

May 15, 2014

Abstract

Contents

1 Introduction **4**

2 Populating Data **5**
2.1 General 5
2.2 Support Tools 5
2.2.1 Airports 5
2.2.2 Countries 6
2.2.3 Regions 6
2.2.4 Market List 7

3 Database Structure **8**
3.1 Overview 8
3.2 Schedule 8
3.2.1 Aircraft 8
3.2.2 Airport 12
3.2.3 Country 14
3.2.4 Region 17
3.3 Forecast 18
3.3.1 Project 18
3.3.2 Scenario 19
3.3.3 User 19

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 2

Populating Data

2.1 General

The data is imported from the files `Schedule Database Model v2.xlsm`, ??? and the internet.

The Excel files are too large for any of the current Ruby Excel importers, so the individual worksheets have to be exported to CSV format. Calculated data that depends on other worksheets is invalidated by this process, but that does not matter because it is recalculated once it is imported.

2.2 Support Tools

2.2.1 Airports

The `airports.rb` program reads a CSV file of airport details in the original spreadsheet format, and outputs a rails seed file to populate the database Airport table.

Note The spreadsheet wrongly codes Queensland, Australia as 'QL'. It should be 'QLD'.

Note The spreadsheet uses an 'SF' region code in Russia. This is not an ISO code, and needs to be replaced by the correct regional code for each airport.

Note The spreadsheet uses an 'IF' region code in Russia. This is not an ISO code, and needs to be replaced by the correct regional code for each airport.

Note The database incorrectly has 'SC' instead of 'SCT' for Scotland

Note The database wrongly has the Channel Islands as a region of the UK. They are autonomous countries with their own ISO codes.

Note Garberville, WI, Shingletown, CA, Window Rock, AZ, Santa Rosa, CA wrongly identified as being in the UK.

Note London St. Pancras misspelled.

2.2.2 Countries

The `countries.rb` program interrogates the internet for a current list of ISO 3166-1 country codes. If run from the command line[1] it produces a comma-separated list of all countries.

Because it interrogates the internet for its data, it is vulnerable to changes to the required pages and to internet time-outs. This is why it does not update the traffic forecaster database automatically; the data should be checked for correctness before use.

See the program documentation for current details.

Note Taiwan is imported incorrectly. The seeds file has to be updated by hand.

Note The spreadsheet has "AND" as the Alpha3 code for Andorra, but the ISO code is "ADO". If the spreadsheet is used as the sole source of country data, the seeds file will have to be updated by hand.

Note The spreadsheet has an incorrect Alpha2 code for Kosovo, which causes a spurious entry merging Kosovo and Sri-Lanka that must be removed by hand.

Note A spurious entry for Romania with the incorrect Alpha3 code of 'ROM' is created and must be deleted by hand.

Note A spurious entry for Timor-Leste with the incorrect Alpha3 code of 'TLS' is created and must be deleted by hand.

Note The program does not find the Alpha2 code for Samoa ('WS'), which must therefore be added by hand.

2.2.3 Regions

The `regions.rb` program interrogates the internet for a current list of ISO 3166-2 region codes.

If run from the command line[2] it produces a comma-separated list of all countries.

Because it interrogates the internet for its data, it is vulnerable to changes to the required pages and to internet time-outs. This is why it does not update the traffic forecaster database automatically; the data should be checked for correctness before use.

The program options for `import_regions` are:

`-h, -help` Print usage information instead of running the import

[1] requires a ruby interpreter; run with `ruby countries.rb` if the computer does not recognise it as a program file

[2] requires a ruby interpreter; run with `ruby regions.rb` if the computer does not recognise it as a program file

`-l[logfile]` Record progress information, warnings and errors to `[logfile]`. if `-l` is specified with no `logfile`, logging goes to `STDOUT`

`-o[overrides]` Read in a file of overrides for regions. The file contains one override per line, each override comprising an Alpha-2 country code, a region code and a region name, comma separated (parsed according to usual CSV rules). If the program finds an entry on the internet for an overriden country and code, it will use the name given in the override file whatever the internet seems to say (because it just isn't worth catering for every possible variation of Wikipedia table format...)

`-p` Put progress dots to `STDERR`

`-r[n]` Sets the number of page load retries to `n`

See the program documentation for current details.

Note Great Britain is not imported correctly. The region codes have to be corrected manually.

2.2.4 Market List

Creates a CSV equivalent of the Market List table in the Schedule Database spreadsheet model. It optionally takes a table of basic country information, as produced by `Countries.rb`, and will use that instead of the spreadsheet for official country names if given.

Chapter 3

Database Structure

3.1 Overview

The database is organised into two conceptual groups:

Schedule corresponds closely with the 'Schedule database model' spreadsheet from which this database is adapted; and

Forecast corresponds closely with the 'Air traffic forecast tool' from which this database is adapted.

3.2 Schedule

The Schedule portion of the database is structured as an OLAP star schema, although OLAP operations are not used to extract data. The `schedule` table is the fact table, and all others are dimension tables.

3.2.1 Aircraft

AIRCRAFT (identifier, manufacturer, body_type, family, aircraft_type, equip_code, aircraft_name, wingspan, mtow, aircraft_code, icao_code, max_mtow, weight_category)
Specification of aircraft types

identifier

Type Text

Can be null? No

Description A unique name for the aircraft type

Constraints Longest value on original import was 33 characters. 50 characters allowed for field

Source Originally imported from [Schedule database model]![Aircraft List]![$J]

manufacturer

Type text

Can be null? Yes

Description The aircraft manufacturer

Constraints Longest value on original import was 17 characters. 25 characters allowed for field.

Source Originally imported from [Schedule database model]![Aircraft List]![$A]

body_type

Type Text

Can be null? Yes

Description The aircraft body type

Constraints Longest value on original import was 10 characters. 15 characters allowed for field

Source Originally imported from [Schedule database model]![Aircraft List]![$B]

family

Type Text

Can be null? Yes

Description The aircraft family

Constraints Longest value on original import was 19 characters. 25 characters allowed for field.

Source Originally imported from [Schedule database model]![Aircraft List]![$C]

aircraft_type

Type Text

Can be null? Yes

Description The aircraft type name

Constraints Longest value on original import was 29 characters. 40 characters allowed for field

Source Originally imported from [Schedule database model]![Aircraft List]![$D]

equip_code

Type Text

Can be null? Yes

Description The aircraft equipment code (note: not unique)

Constraints Length = 3

Source Originally imported from [Schedule database model]![Aircraft List]![$E]

aircraft_name

Type Text

Can be null? Yes

Description The name of the aircraft type (note: not unique).

Constraints Longest value on original import was 39 characters. 50 characters allowed for field

Source Originally imported from [Schedule database model]![Aircraft List]![$F]

wingspan

Type Decimal

Can be null? Yes

Description The aircraft wingspan in metres (null if not fixed-wing).

Constraints Scale = 2

Source Originally imported from [Schedule database model]![Aircraft List]![$G]

mtow

Type Text

Can be null? Yes

Description Maximum take off weight. In some cases this is a range of weights, presumably relating to different configurations, so the field is stored in a string rather than a numeric type.

Constraints Longest value on original import was 24 characters. 30 characters allowed for field

Source Originally imported from [Schedule database model]![Aircraft List]![$H]

aircraft_code

Type Text

Can be null? Yes

Description Aircraft type code (usually corresponds to icao_code [3.2.1])

Constraints Length = 1

Source Originally imported from [Schedule database model]![Aircraft List]![$I]

icao_code

Type Text

Can be null? Yes

Description Aircraft ICAO design group code.

Constraints Longest value on original import was 10 characters. 15 characters allowed for field.

Source Originally imported from [Schedule database model]![Aircraft List]![$K]

max_mtow

Type Integer

Can be null? Yes

Description The maximum take-off weight. If the mtow field [3.2.1] is a range, this represents the maximum value of the range, otherwise it should be identical to Mtow but expressed as a numeric type. Consistency with the mtow field is not checked or maintained

Constraints None

Source Originally imported from [Schedule database model]![Aircraft List]![$L]

weight_category

Type Text

Can be null? Yes

Description The ICAO weight category

Constraints Longest value on original import was 23 characters. 30 characters allowed for field.

Source Originally imported from [Schedule database model]![Aircraft List]![$M]

3.2.2 Airport

AIRPORT (code, country_id, name, city, region_id, latitude, longitude, wac, notes)

- belongs_to country
- has_many projects
- has_many scenarios

code

Type Text

Can be null? No

Description The IATA code for the airport

Constraints Length = 3

Source Originally imported from [Schedule database model]![Airport List]![$A]

country_id

Type Internally generated

Can be null? No

Description A reference to the relevant entry in the Country table.

Constraints N/A

Source Internally generated

name

Type

Can be null?

Description

Constraints

Source

city

Type

Can be null?

Description

Constraints

Source

state

Type

Can be null?

Description

Constraints

Source

latitude

Type

Can be null?

Description

Constraints

Source

longitude

Type

Can be null?

Description

Constraints

Source

region_id

Type

Can be null?

Description

Constraints

Source

wac

Type

Can be null?

Description

Constraints

Source

notes

Type

Can be null?

Description

Constraints

Source

3.2.3 Country

COUNTRY (alpha3, alpha2, iso_name, srs_name, global_region, european_route_markets, eu_member, oecd_member, un_member, economy)

- has_many regions
- has_many airports

Represents a country of the world.

Some of the ISO 3166-1 country codes relate to regions, not individual countries, and so there may be no matching entry in the SRS analyser data. Because of this, all fields derived from the SRS analyser may be NULL.

alpha3

Type Text

Can be null? No

Description The ISO 3166-1 Alpha-3 code for the country.

Constraints Length = 3

Source Originally imported from the internet using the tool `import_countries.rb`.

alpha2

Type Text

Can be null? No

description The ISO 3166-1 Alpha-2 code for the country.

Constraints Length = 2

Source Originally imported from the internet using the tool `import_countries.rb`

iso_name

Type Text

Can be null? No

Description The ISO 3166-1 short country name.

Constraints The longest value on original import was 46 characters; 75 characters allowed in the database.

Source Originally imported from the internet using the tool `countries.rb`.

srs_name

Type Text

Can be null? Yes

Description The country name as shown in the SRS analyser.

Constraints The longest value on original import was 46 characters; 75 characters allowed in the database.

Source Originally imported from [ScheduleDatabaseModel]![Market List]![$E].

global_region

Type Text

Can be null? Yes

Description The global region of the country, as identified in the SRS analyser.

Constraints The longest value on original import was 15 characters; 25 characters allowed in the database.

Source Originally imported from [ScheduleDatabaseModel]![Market List]![$E].

european_route_markets

Type Text.

Can be null? Yes

Description The European route market of the country as identified in the SRS analyser.

Constraints The longest value on original import was 17 characters; 25 characters allowed in the database.

Source Originally imported from [ScheduleDatabaseModel]![Market List]![$F].

eu_member

Type Yes/No

Can be null? Yes

Description Whether the country is a member of the EU.

Constraints None

Source Originally imported from [ScheduleDatabaseModel]![Market List]![$G].

oecd_member

Type Yes/No

Can be null? Yes

Description Whether the country is a member of the OECD.

Constraints None

Source Originally imported from [ScheduleDatabaseModel]![Market List]![$H].

un_member

Type Yes/No

Can be null? Yes

Description Whether the country is a member of the UN.

Constraints None

Source Originally imported from [ScheduleDatabaseModel]![Market List]![$I].

economy

Type String

Can be null? Yes

Description The type of the country's economy.

Constraints "developed" | "transition" | "developing"

Source Originally imported from [ScheduleDatabaseModel]![Market List]![$J].

3.2.4 Region

REGION (country_id, region_code, name)

- belongs_to country

Represents a region of a country.

country_id

Type Text

Can be null? No

Description The alpha3 country identifier of the country containing this recion.

Constraints Length = 3

Source http://en.wikipedia.org/wiki/ISO_3166-1

region_id

Type Text

Can be null? No

Description The ISO 3166-2 region code

Constraints Length $\leq$ 3

Source Pages linked from http://en.wikipedia.org/wiki/ISO_3166-1

name

Type Text

Can be null? No

Description The name of the region

Constraints The longest entry on original import was 15 characters. 25 characters allowed in database.

Source

3.3 Forecast

3.3.1 Project

PROJECT (name, airport_id)

- has_many users
- has_many scenarios

A project is a package for multiple scenarios.

name

Type String

Can be null? No

Description The project name

Constraints None

Source Created by the DBA

airport_id

Type Application defined

Can be null? No

Description The airport

Constraints None

Source Created by the DBA

3.3.2 Scenario

SCENARIO (name, base_year, min_r2, local_res_domestic, local_res_international, dom_developed_elasticity, short_developed_elasticity, short_developing_elasticity, medium_developed_elasticity, medium_developing_elasticity, long_developed_elasticity, long_developing_elasticity, ultra_developed_elasticity, ultra_developing_elasticity, dom_developed_saturation, short_developed_saturation, short_developing_saturation, medium_developed_saturation, medium_developing_saturation, long_developed_saturation, long_developing_saturation, ultra_developed_saturation, ultra_developing_saturation)

- scenario belongs_to project
- scenario has_many airports

name

Type Text

Can be null? No

Description A name for the scenario

Constraints None

Source Created by the user

3.3.3 User

USER (name, admin, password_digest)

- has_many projects

A user is simply a person who is authorised to use the tool.

name

Type String

Can be null? No

Description The user name

Constraints None

Source Created by the DBA

admin

Type Boolean

Can be null? No

Description Whether the user has admin privileges

Constraints None

Source Created by the DBA

password_digest

Type String

Can be null? No

Description A secure code used to test the user password

Constraints None

Source Created (indirectly) by the DBA