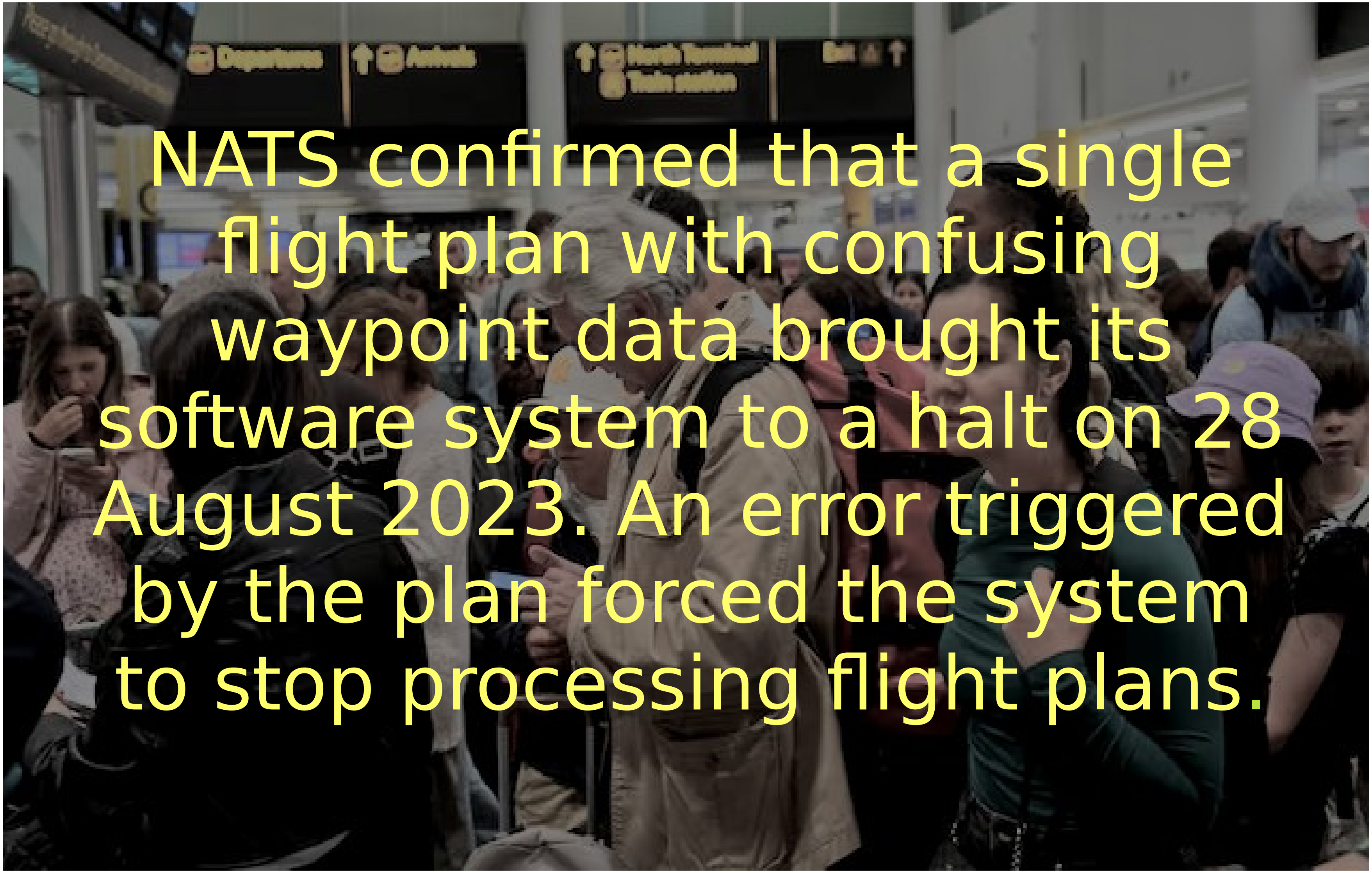
Agile On The Beach 2024
Data Needs Testing Too

Ron Ballard

4 July 2024 15:00

Data Needs Testing Too



NATS confirmed that a single flight plan with confusing waypoint data brought its software system to a halt on 28 August 2023. An error triggered by the plan forced the system to stop processing flight plans.

Data Needs Testing Too

When all your tests pass and you release an application, you know that your code will not change until the next release.

Data changes all the time

Data can hit your application from:

- User input
- Databases
- Files
- API calls
- Automated feeds from devices
- Parameters from files, startup commands or environment variables

Data from User Input

- Whenever data is entered by a user, it should be checked on the form (form-controls and JavaScript).

- Dates should normally be entered using a calendar. The user can type the date, but it should be shown on a calendar for passive confirmation.

- Address lookups are cheap and widely available, so always use them to enter, or, at least confirm, addresses.

- Email addresses should always be checked. JavaScript with a regular expression can, at least check that the email address is plausible.

- User training may be needed to explain why they can’t just type anything anywhere.

Data From Databases

- Data from databases can be guaranteed to be clean in various ways ...
- … but only if you are using a proper relational database …
- … and only if you are using it properly

We’ll come back to this

Data from Files

- Data from files may have come from a database…
- … more likely, it has come from Microsoft Excel

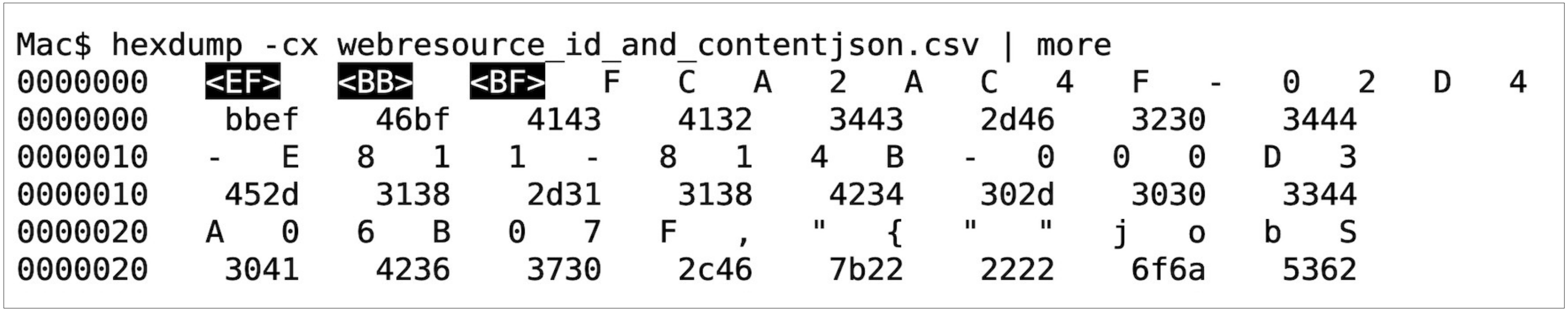
Excel – The Helpful Vandal

```
name,                     real,         date,        mobile phone,    timestamp
Nadia Redding,            13321.113598, 28/12/23,    07856392214,     2023-12-28 09:30:56.763
charles.walters@cm.com,   12.5,         20231228,    +3597889684456,  2023-12-28 09:31:00
Brontë Café,              143.86,       21/28/23,    07128324727,     20231228093223
```

before Excel ⇧ ⇩ after Excel

```
name,                     real,         date,        mobile phone,    timestamp
Nadia Redding,            13321.1136,   28/12/2023,  7856392214,      30:56.8
charles.walters@cm.com,   12.5,         20231228,    3.59789E+12,     28/12/2023 09:31
BrontÃ« CafÃ©,            143.86,       21/28/23,    7128324727,      2.02312E+13
```

Character encoding – BOM misused by Excel



```
Mac$ hexdump -cx webresource_id_and_contentjson.csv | more
0000000   <EF>  <BB>  <BF>  F   C   A   2   A   C   4   F   -   0   2   D   4
0000000     bbef    46bf    4143    4132    3443    2d46    3230    3444
0000010   -   E   8   1   1   -   8   1   4   B   -   0   0   0   D   3
0000010     452d    3138    2d31    3138    4234    302d    3030    3344
0000020   A   0   6   B   0   7   F   ,   "   {   "   "   j   o   b   S
0000020     3041    4236    3730    2c46    7b22    2222    6f6a    5362
```

“But Excel is such a useful tool”…

So is a grater, but it can take your knuckles off. Be careful!

Listen to Tim Harford’s “More or Less” BBC Radio 4 11 Feb 2023
https://www.bbc.co.uk/sounds/play/p0f2cytq

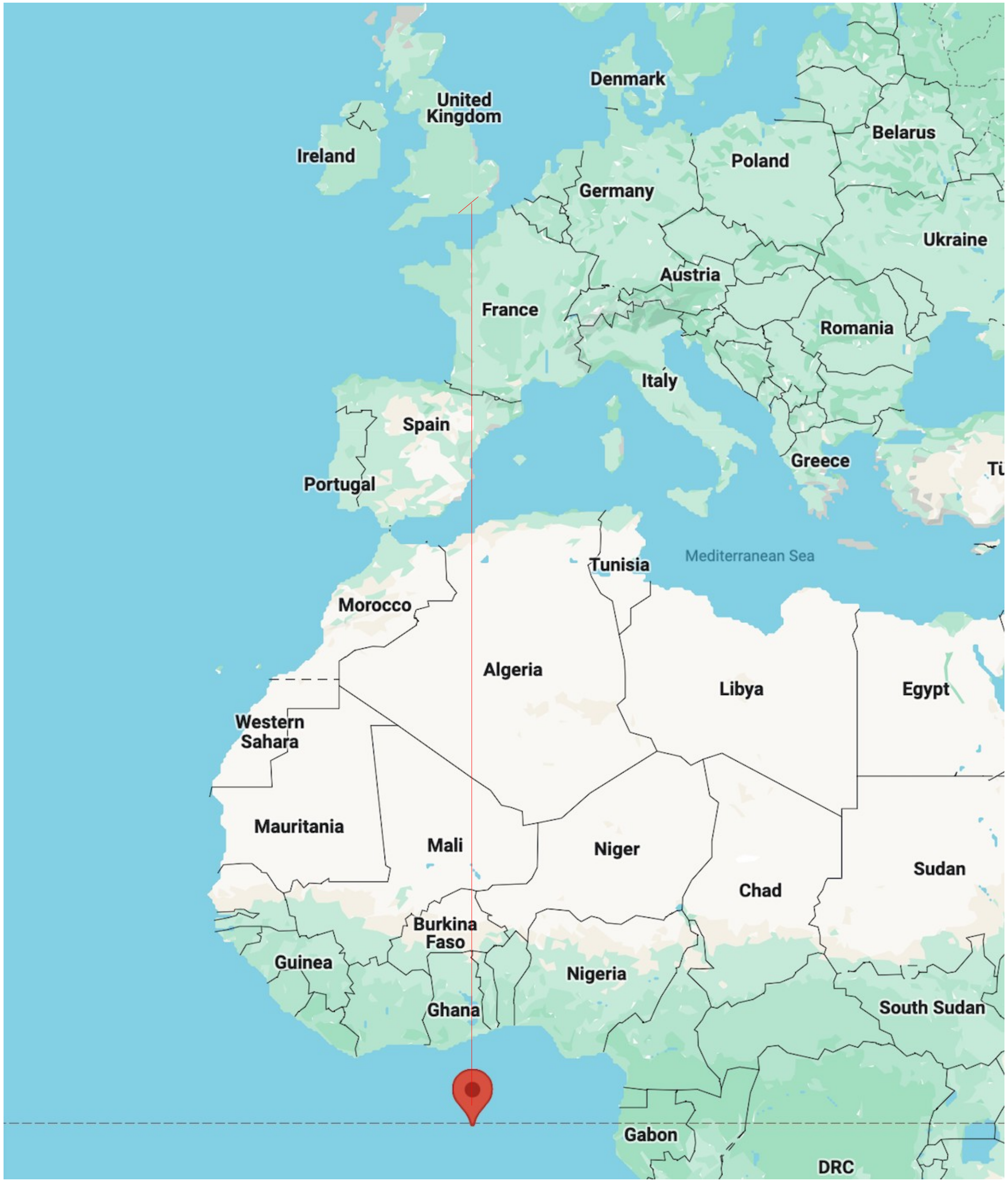
Data From API Calls

- API calls usually mean data delivered as XML or JSON.
- XML and JSON include metadata as well as data, but the metadata does not enforce any rules on the content.
- API data therefore needs to be validated before being used in an application.

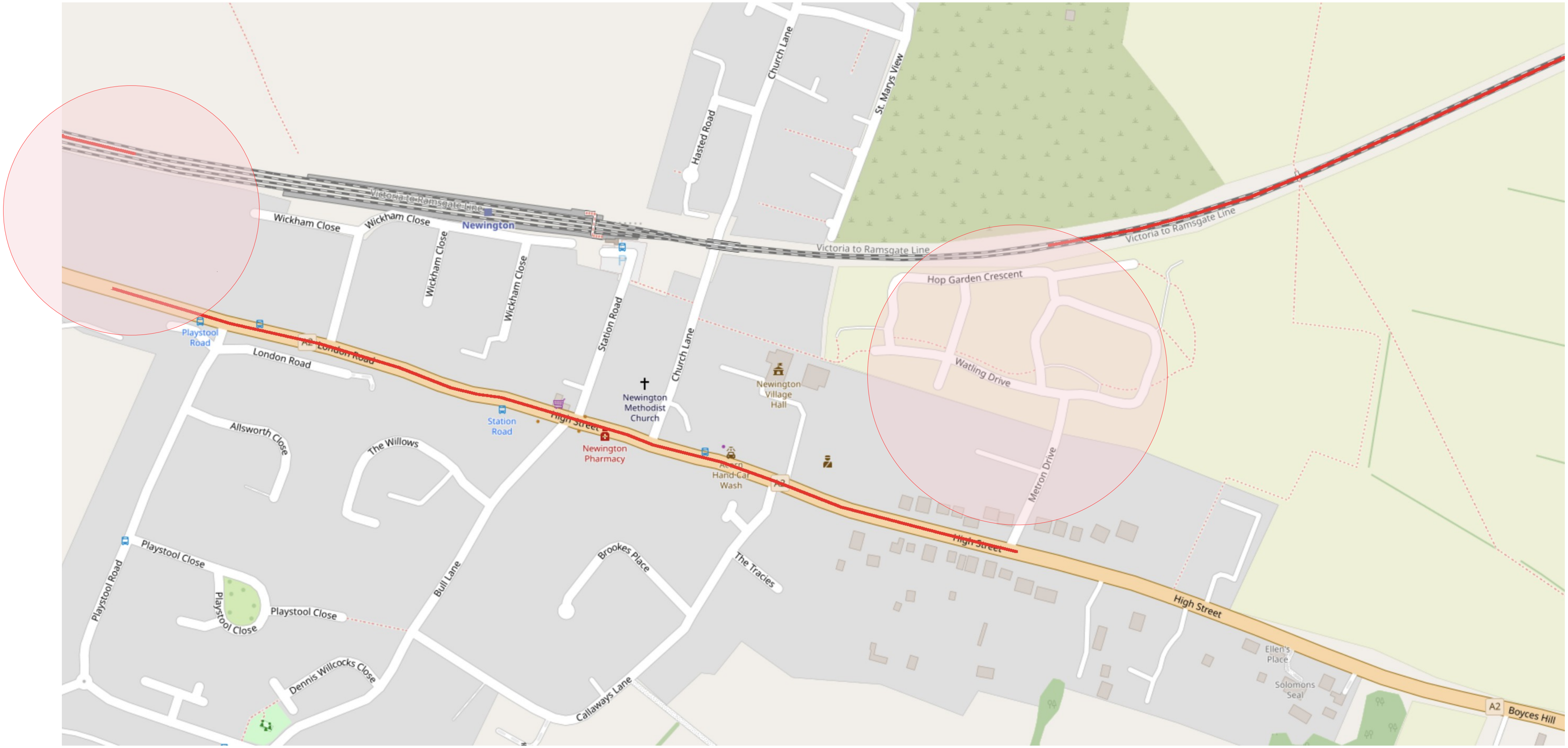
Automated Feeds From Devices (Internet of Things)

Example:

a car-insurance company that installed boxes in policy-holders’ cars to record location and acceleration data, which was sent back to a central database.


Denmark
United Kingdom
Ireland
Belarus
Poland
Germany
Ukraine
Austria
France
Romania
Italy
Spain
Portugal
Greece
Tu
Tunisia
Mediterranean Sea
Morocco
Algeria
Libya
Egypt
Western Sahara
Mauritania
Mali
Niger
Sudan
Chad
Burkina Faso
Guinea
Nigeria
Ghana
South Sudan
Gabon
DRC

Clever Correction
(too clever!)

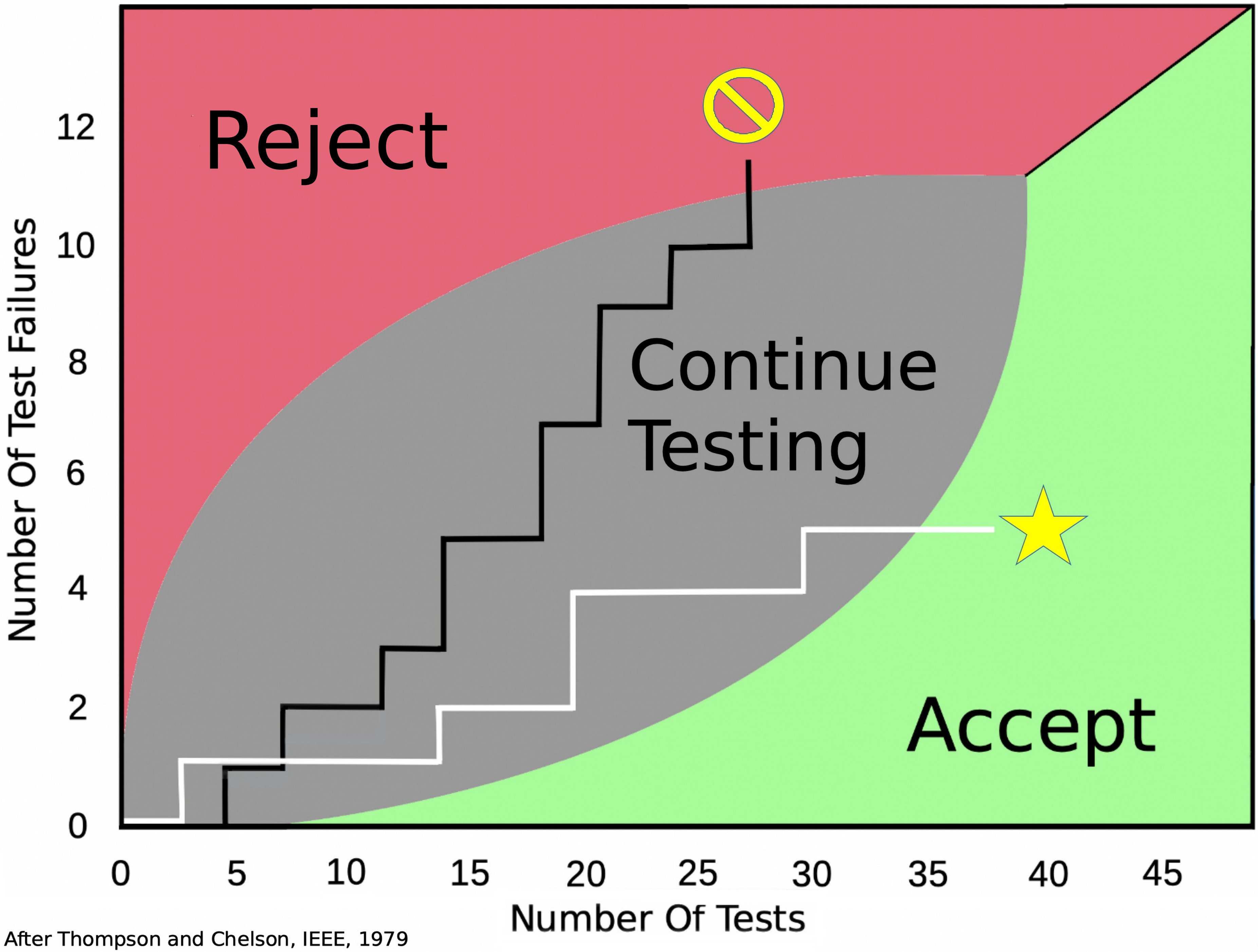


Parameters from Files, Startup Commands or Environment Variables

If you have any of these, they should be controlled by scripts which are in your version control system as part of the release.

Defect Clustering

- Typically, in a system with many components, around 5% of the components will account for about 50% of the defects.
- Many components will be defect-free from the day they were written.
- The probable number of ***remaining defects*** in a tested component will be proportional to the number of defects already found and removed.
- This applies to code and to data



After Thompson and Chelson, IEEE, 1979

How Can We Test Data?

User input	Form controls and JavaScript
Databases	More to say
Files	More to say
API calls	Gatekeeper code
Feeds from devices	Gatekeeper code
Parameters from files, startup commands or environment variables	Part of your test suite

Testing Databases and Files

The easiest way to test data that arrives in a file is to load it into a database, and then use the database facilities and techniques described in the next few slides.

Getting File Data into a Database

- Do you have a suitable database? PostgreSQL is free and it runs on Linux, Windows and Mac. It is robust and scalable.
- The PostgreSQL copy command is easy and efficient. Feeding your data through an API or another program is more work and is far less efficient.
- CSV (comma separated variables) is the easiest and most efficient file format for loading. Beware eccentric CSV-like formats.

Case Study

A leisure company does a deal with a large supermarket chain.

The supermarket’s customers get information and discounts on the leisure services.

The leisure company gets a list of contacts who may be interested in its services.

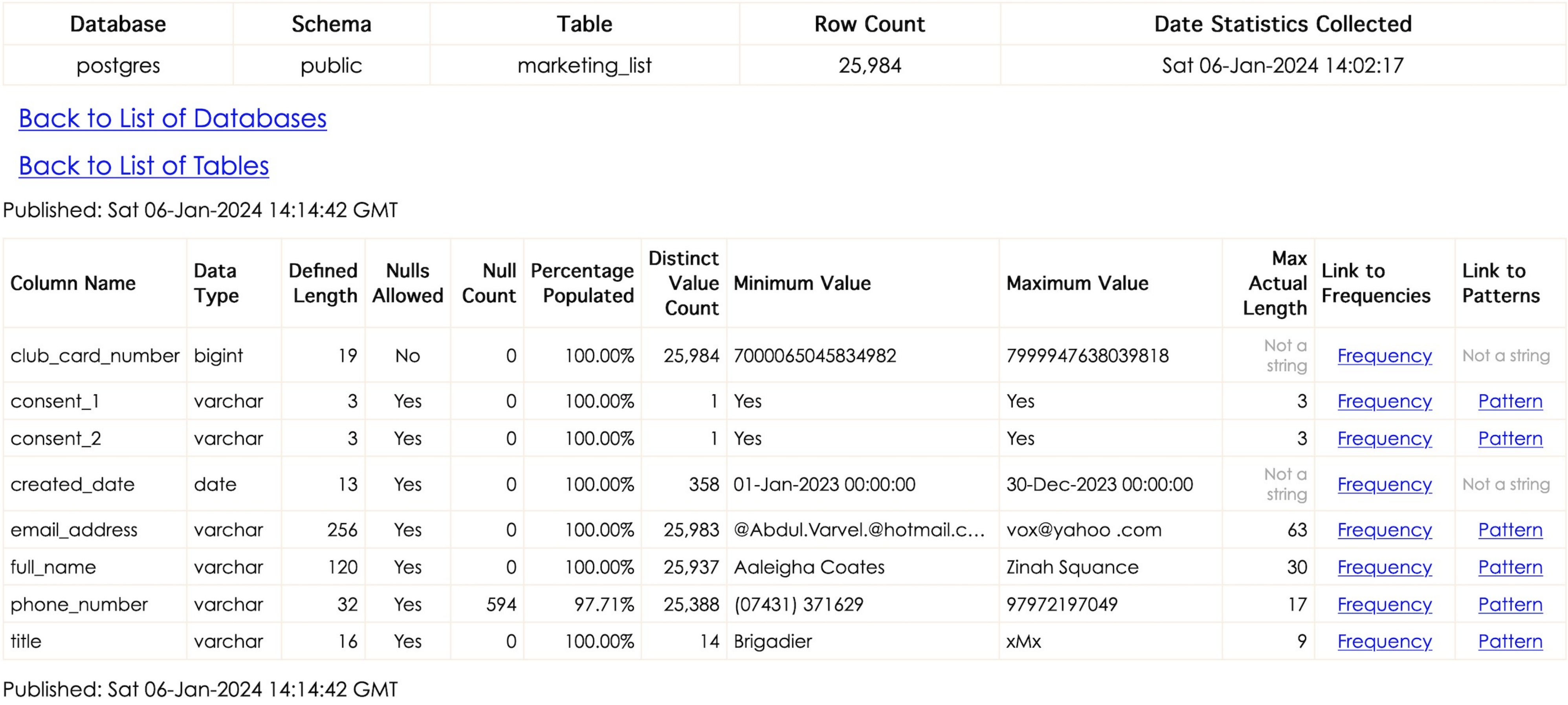
Case Study

As “the data guy” I get a file of 26,000 contacts, to add to our contact database.

A quick look at the file shows that is a CSV file with the following columns:

- club_card_number
- title
- full_name
- email_address
- phone_number
- consent_1
- date
- consent_2

Case Study



Database	Schema	Table	Row Count	Date Statistics Collected
postgres	public	marketing_list	25,984	Sat 06-Jan-2024 14:02:17

Back to List of Databases

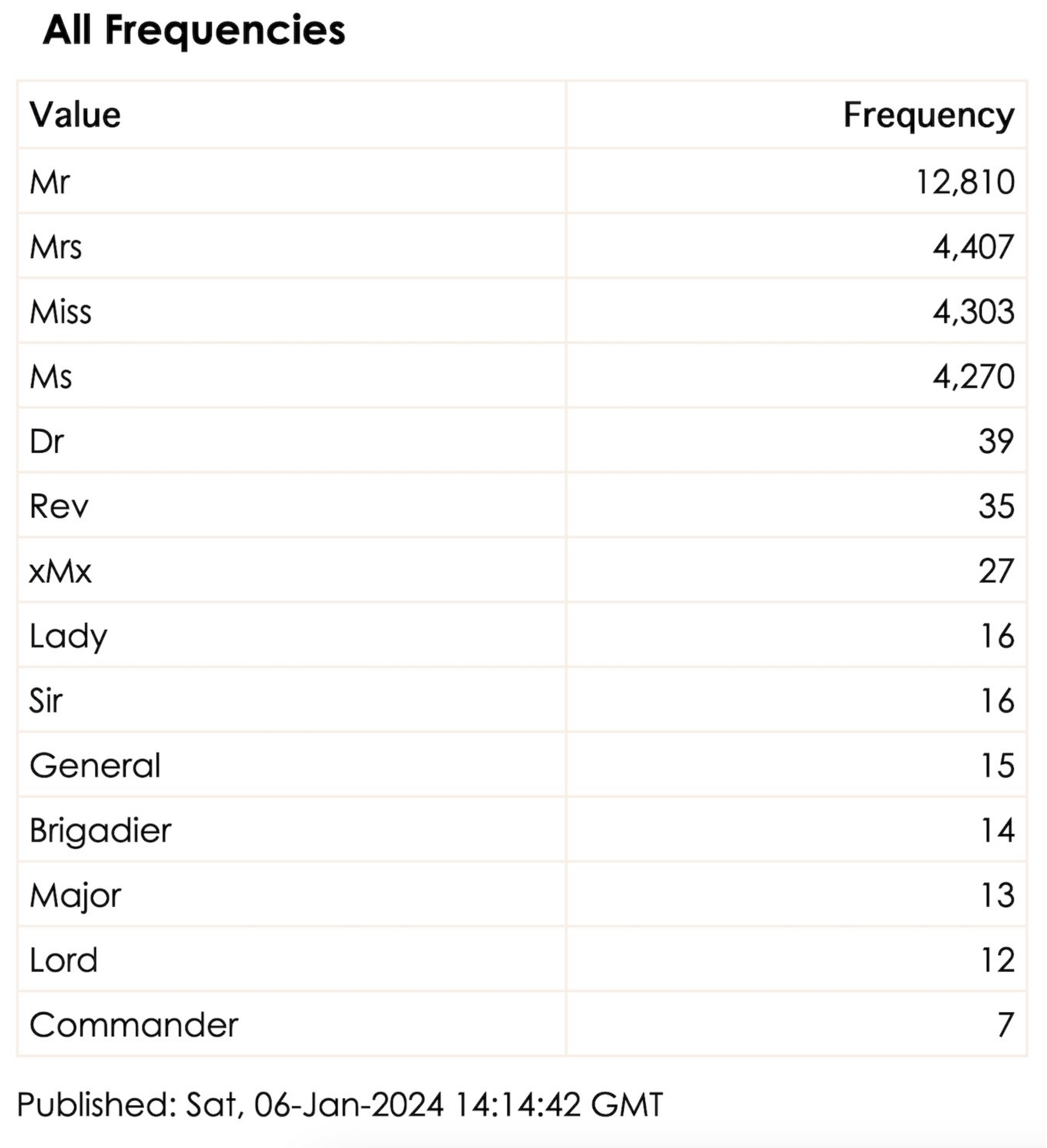
Back to List of Tables

Published: Sat 06-Jan-2024 14:14:42 GMT

Column Name	Data Type	Defined Length	Nulls Allowed	Null Count	Percentage Populated	Distinct Value Count	Minimum Value	Maximum Value	Max Actual Length	Link to Frequencies	Link to Patterns
club_card_number	bigint	19	No	0	100.00%	25,984	7000065045834982	7999947638039818	Not a string	Frequency	Not a string
consent_1	varchar	3	Yes	0	100.00%	1	Yes	Yes	3	Frequency	Pattern
consent_2	varchar	3	Yes	0	100.00%	1	Yes	Yes	3	Frequency	Pattern
created_date	date	13	Yes	0	100.00%	358	01-Jan-2023 00:00:00	30-Dec-2023 00:00:00	Not a string	Frequency	Not a string
email_address	varchar	256	Yes	0	100.00%	25,983	@Abdul.Varvel.@hotmail.c…	vox@yahoo .com	63	Frequency	Pattern
full_name	varchar	120	Yes	0	100.00%	25,937	Aaleigha Coates	Zinah Squance	30	Frequency	Pattern
phone_number	varchar	32	Yes	594	97.71%	25,388	(07431) 371629	97972197049	17	Frequency	Pattern
title	varchar	16	Yes	0	100.00%	14	Brigadier	xMx	9	Frequency	Pattern

Published: Sat 06-Jan-2024 14:14:42 GMT

Case Study

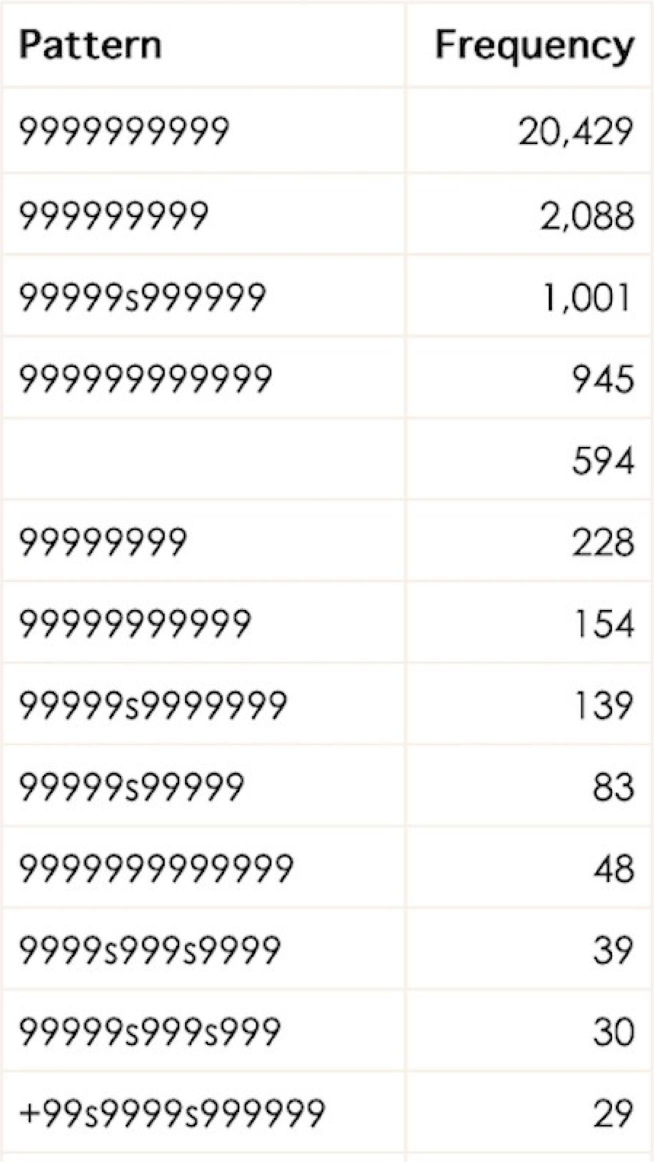


All Frequencies

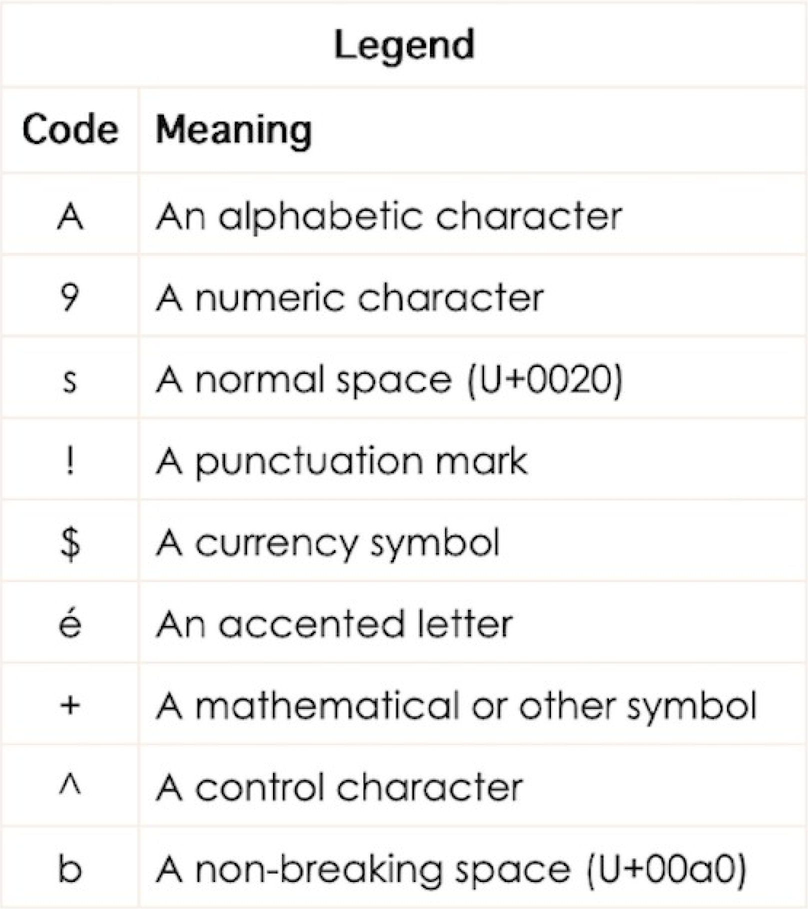
Value	Frequency
Mr	12,810
Mrs	4,407
Miss	4,303
Ms	4,270
Dr	39
Rev	35
xMx	27
Lady	16
Sir	16
General	15
Brigadier	14
Major	13
Lord	12
Commander	7

Published: Sat, 06-Jan-2024 14:14:42 GMT

Case Study



Pattern	Frequency
9999999999	20,429
999999999	2,088
99999s999999	1,001
999999999999	945
	594
99999999	228
99999999999	154
99999s9999999	139
99999s99999	83
9999999999999	48
9999s999s9999	39
99999s999s999	30
+99s9999s999999	29



Legend	
Code	**Meaning**
A	An alphabetic character
9	A numeric character
s	A normal space (U+0020)
!	A punctuation mark
$	A currency symbol
é	An accented letter
+	A mathematical or other symbol
^	A control character
b	A non-breaking space (U+00a0)

Case Study

What we learned from the profile:

- Club Card Number is unique and always starts with 7
- Consent 1 and Consent 2 are not interesting (always “Yes”)
- Created Date is always in 2023 - all valid dates
- Email address has some errors (we’ll look more closely)
- Full name looks OK
- Phone number inconsistently formatted

  (patterns show many formats - more checks needed)
- Titles are OK, except for “xMx”, which should just be “Mx”

Case Study – Email Address

- Email addresses are defined by several RFCs (3696, 5321, 5322, 6530)
- Maximum length is 320 characters
- Some characters are not permitted in some places
- Actual accepted addresses are often more restricted
- Dozens of validations are available – some reject valid email addresses
- The validation shown in the next slide does find many genuine errors and we have not found a valid email address that it rejects. There probably are some, so it is useful but may not be infallible.

Case Study – email address

```
select
  email_address
from
  marketing_list
where
  club_card_number in
  (
    select club_card_number from marketing_list
    except
    select club_card_number
    from
    (
      select
        club_card_number,
        regexp_matches
        (
          trim(email_address),
          '^[\w\-\.]+@([\w-]+\.)+[\w-]{2,}$', 'gm'
        )
      from
        marketing_list
    ) x
  );
```

https://regex101.com/

Case Study – email address

```
Glare@-hotmail@co.uk
Irving.Ford@gmail
Irene.Kempson.)@hotmail.com
Padraig.Rutter11@@hotmail.com
nectare@yahoo   .com
uenistis5@hotmail co uk
Erna.Edwards@� hotmail.com
subiere@.net
Malik58@bt internet.com
Kelsie.Goody@.net
Carolina.Hemley@yahoo/co/uk
Janeil.Cowles@gmail..com
Ullrick.Steedley@yahoo/co/uk
Wade.Bathmaker@yahoo    .com
Tiarne Drake@fervet.com
```

Case Study – email address

```
Glare@-hotmail@co.uk
Irving.Ford@gmail    
Irene.Kempson.)@hotmail.com
Padraig.Rutter11@@hotmail.com
nectare@yahoo →.com
uenistis5@hotmail co uk
Erna.Edwards@� hotmail.com
subiere@.net
Malik58@bt internet.com
Kelsie.Goody@.net
Carolina.Hemley@yahoo/co/uk
Janeil.Cowles@gmail..com
Ullrick.Steedley@yahoo/co/uk
Wade.Bathmaker@yahoo    .com
Tiarne Drake@fervet.com
```

Telephone Numbers
Flexibility and Ambiguity

Telephone numbers are surprisingly complicated. The rules defining valid phone numbers vary within and across countries and over time. And there are many rules.

People write their phone numbers in different ways. There are official recommendations, but there is a lot of flexibility in spacing and punctuation.

This flexibility can lead to ambiguity, for example:

+7797 273 743 could be:

+77 9727 3743 (A Kazakhstan corporate network) or

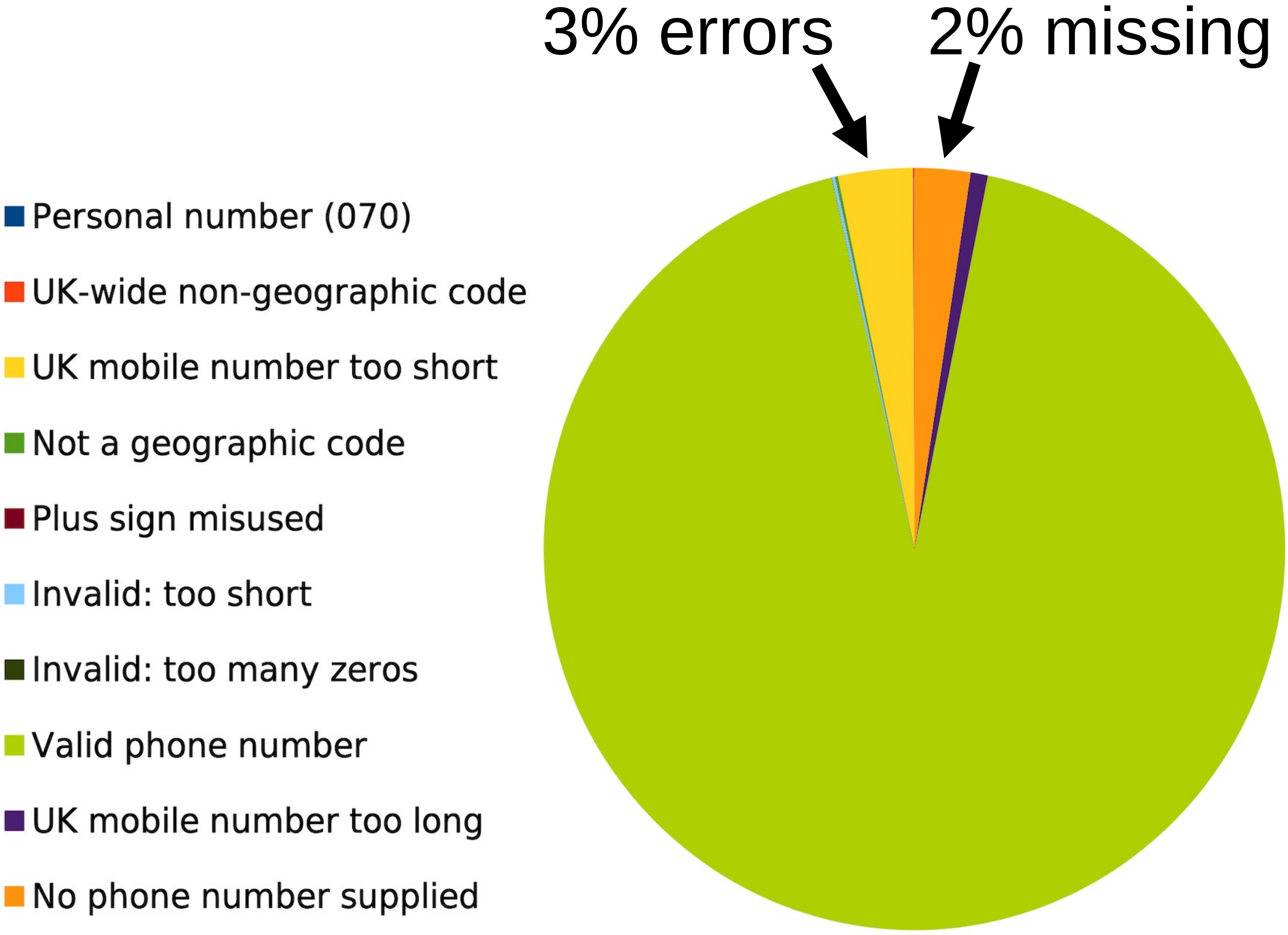
07797 273 743 (A UK mobile)

Tools for Validating Telephone Numbers

A comprehensive tool, recommended by the UK Government Design Guide, is Google’s ***`libphonenumber`*** – a Java library containing about 4,000 files of reference data and code. Even this is not foolproof, or completely up-to-date, but it does look good.

A simpler solution, for the UK only, and currently available for PostgreSQL only, is available from my website (with source code). It does find a variety of errors that we have seen in phone numbers. It also returns the phone number in a standardised format for human beings to read, as well as a digits-only format which is best for automated dialing and database searching.

Case Study – Phone Number



3% errors

2% missing

Duplication

You may have duplication:

- In your database
- In files and other inputs to your database
- Between your database and data that is to be added

Duplicate Records

How ***not*** to detect duplicates

Microsoft’s “sophisticated” duplicate detection in Dynamics 365:
“If you specified Same First Characters or Same Last Characters, in the No. of Characters column, select Enter Value, and then enter the number of characters to compare.”

First 3= ‘Har’; Last 3 = ‘son’

Harris-Wilson
Harbinson
Harvison
Harkinson
Harkison
Harvey-Richardson
Harrington-Mason
Harrison
Hart-Thomson
Harper-Benson
Harrier-Wilson
Haroldson

First 3= ‘Bra’; Last 3 = ‘ell’

Bracknell
Branwell
Brazewell
Bradwell
Bratchell
Brakell
Braun-dorrell
Bramwell
Brassell
Bracewell
Brattell
Brackwell

Duplicate Records

A Good Way to Detect Duplicates

This uses the Levenshtein Edit Distance function which is provided as standard in PostgreSQL and in other databases and languages. The function calculates the number of single character edits necessary to turn one string into another.

For example:
levenshtein('banana', 'Panama') gives result: 2
levenshtein('pomegranate','Uzbekistan') gives result: 9

The results below show pairs of records with 1- or 2-letter differences in last name and first name. By checking other details, such as the address and postcode we can be very confident that most of these are indeed duplicates.

```
|  last_name_a |   last_name_b | first_name_a | first_name_b | address_town_a | address_town_b | address_street_a  |  address_street_b   | postcode_a | postcode_b
+--------------+---------------+--------------+--------------+----------------+----------------+-------------------+---------------------+------------
+------------
| BABESHA      | BADESHA       | KOMAL        | KOMAL        | BECKENHAM      | BECKENHAM      | 66 EAST WAY       | 66 EAST WAY         | BR3 4XT    | BR3 4XT
| WALTER-CLARK | WEALTER-CLARK | ROBERT       | ROBERT       | BEDFORD        | BEDFORD        | 30 ABINGDON  ROAD | 30 ABINGDON ROAD    | MK45 1AF   | MK45 1AF
| DE NIRO      | DE-NIRO       | MICHAEL      | MICHAEL      | BIRMINGHAM     | BIRMINGHAM     | 1 NEWTOWN ROAD    | 1 NEWTOWN ROAD      | B10 3PW    | B10 3PW
| DE-NIRO      | DE NIRO       | MICHAEL      | MICHAEL      | BIRMINGHAM     | BIRMINGHAM     | 1 NEWTOWN ROAD    | 1 NEWTOWN ROAD      | B10 3PW    | B10 3PW
| PREECE       | PRICE         | ELIZA        | ELIZA        | BRIGHTON       | BRIGHTON       |                   | 104 PILCHARD AVENUE |            | BN1 5GD
| MOYES        | MOYSE         | JULIAN       | JULIAN       | LONDON         | LONDON         | 22 FORBES ROAD    | 22 FORBES ROAD      | SW11 6RS   | SW11 6RS
| PRASAD       | RRASAD        | PRIYA        | PRIYA        |                |                | 71C BUXTON ROAD   | 71C BUXTON ROAD     | GL3 5QU    | GL3 5QU
```

Much more detail, including the SQL to produce these results is on The Data Studio website: https://www.thedatastudio.net/duplicates.htm

Missing Data

This suggests that data exists somewhere, but is missing from our particular application database.

If we’re synchronising with another system we need to run regular reconciliations.

Missing Data

How ***not*** to do it.

I asked the project manager of an international $48 billion consultancy how he would reconcile the transaction table (80 million records) when it was migrated to his proposed system. He looked me in the eye, smiled, and said, “we’ll eye-ball a few records, of course”!

Missing Data - Reconciled

It can be a challenge to reconcile data, record-by-record in two different databases. It could take a long time, and synchronising frequently-changing systems can be particularly difficult.

Some feasible aproaches include:

- Matching total record counts
- Matching record counts by year or month or day
- Matching record counts by another partitioning of the data.
- Matching total values of financial records by year or month or day or other partitions
- Select values from random records in one system, and compare all the values with the same records in the other system
- Use hash values (as with file downloads)

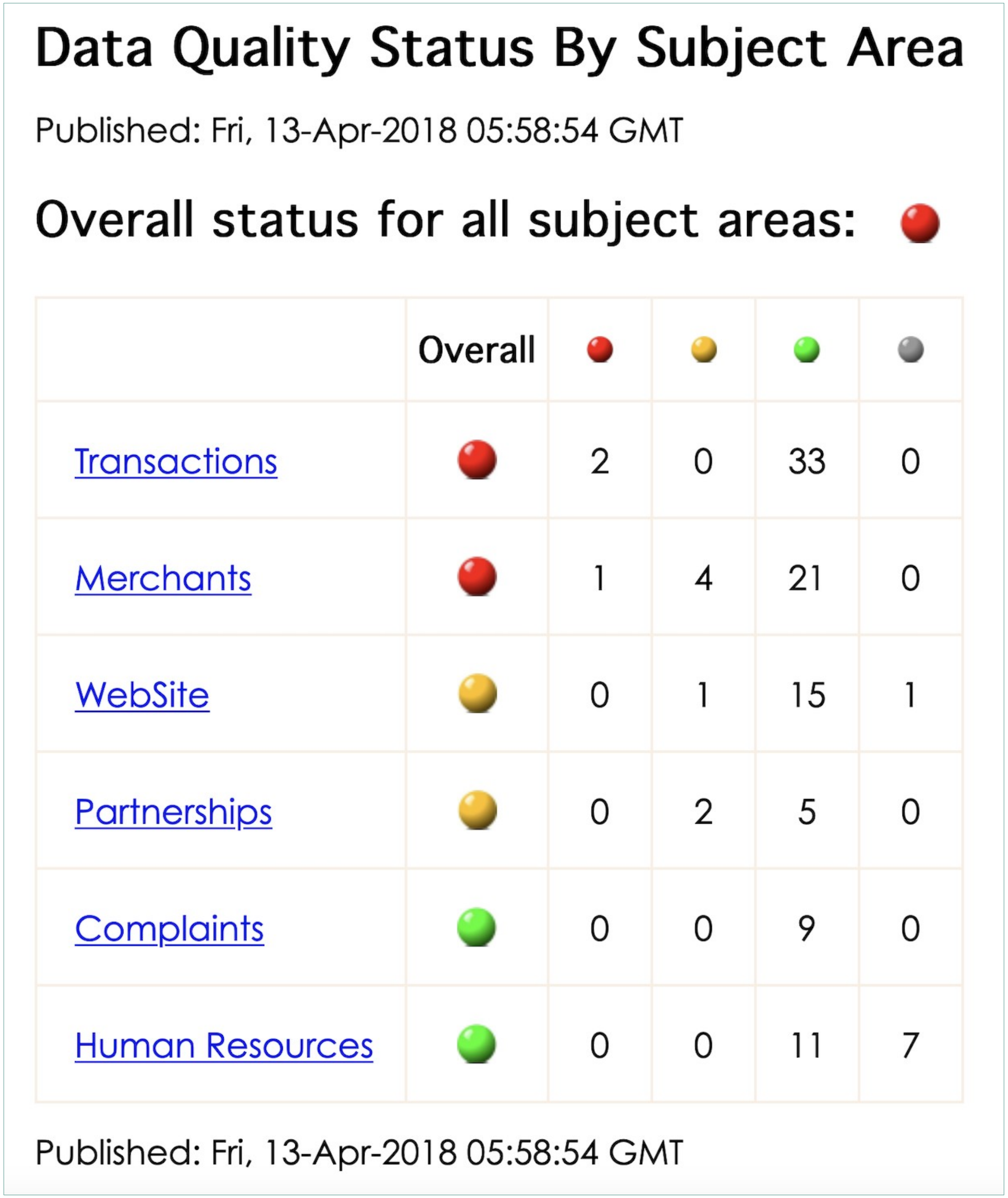
Monitoring

We have emphasized the need to trap bad data on the way into the system. We can make improvements, but we cannot expect to catch every error.

We can often limit the impact of data errors by detecting and acting on any unusual data patterns or events.

When we find another error we add it to the monitoring, and keep that check even (especially) after the error has been fixed.

Monitoring



Data Quality Status By Subject Area

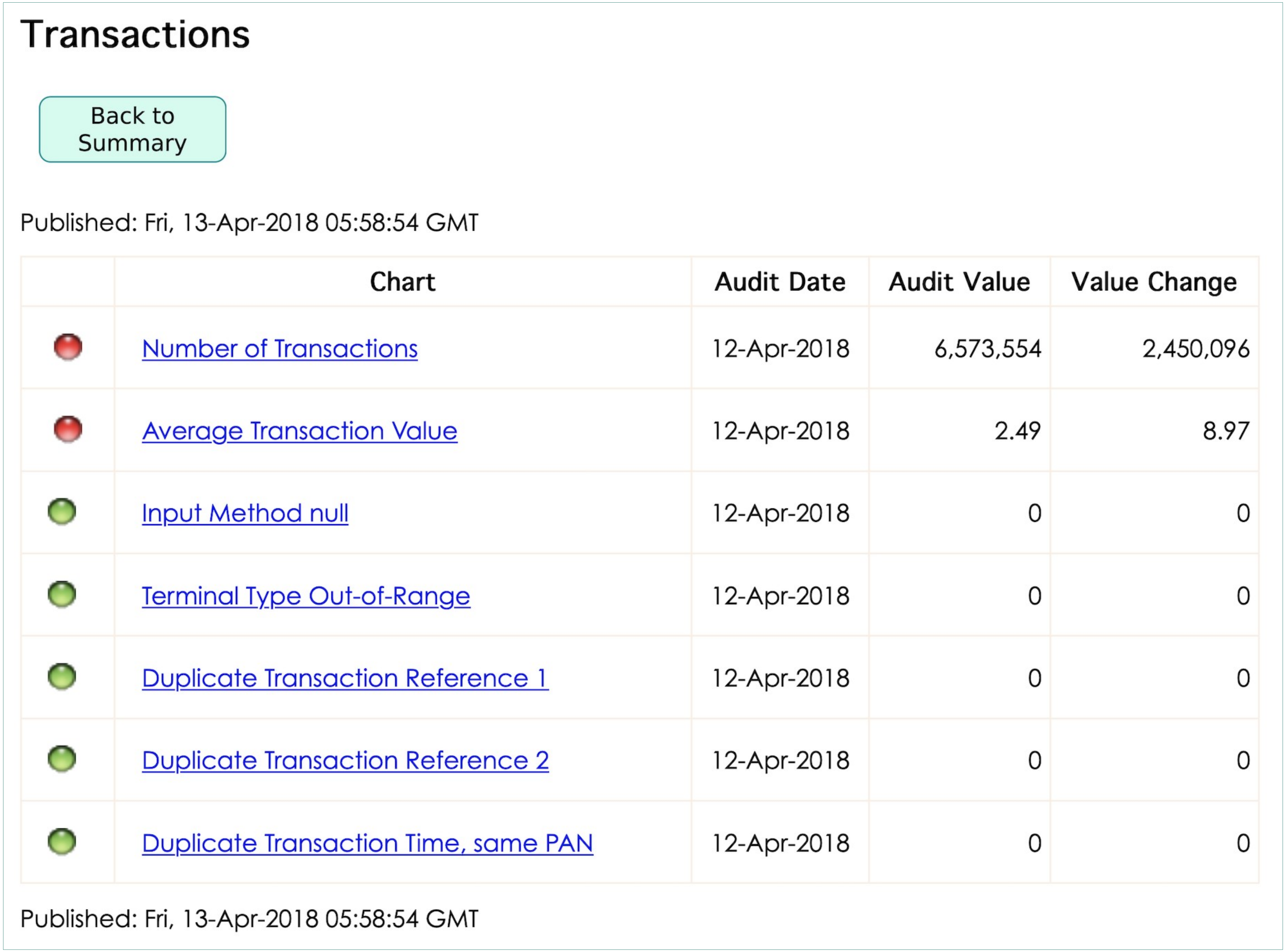
Published: Fri, 13-Apr-2018 05:58:54 GMT

Overall status for all subject areas: 🔴

	Overall	🔴	🟡	🟢	⚪
Transactions	🔴	2	0	33	0
Merchants	🔴	1	4	21	0
WebSite	🟡	0	1	15	1
Partnerships	🟡	0	2	5	0
Complaints	🟢	0	0	9	0
Human Resources	🟢	0	0	11	7

Published: Fri, 13-Apr-2018 05:58:54 GMT

Monitoring



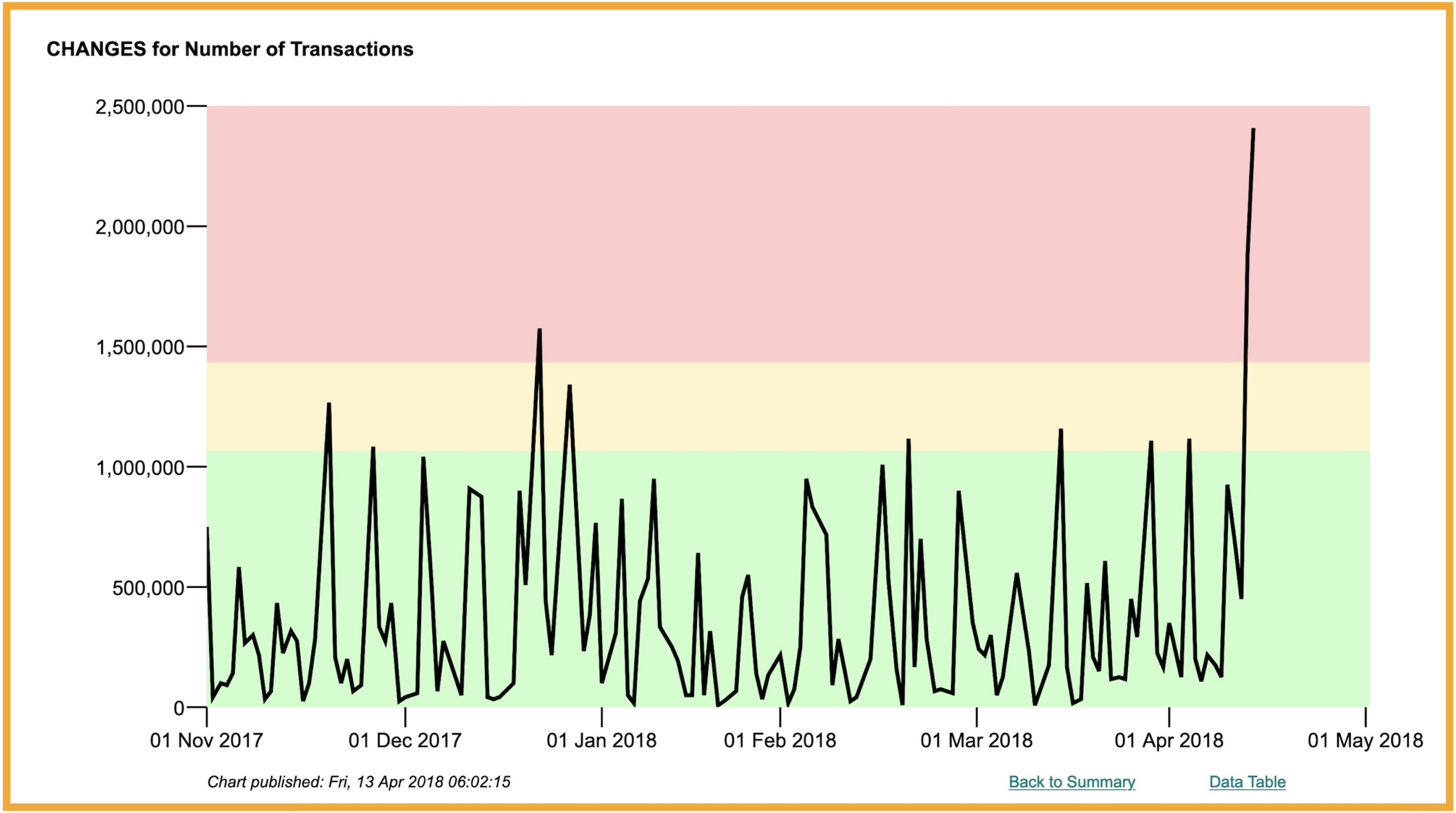
Transactions

Published: Fri, 13-Apr-2018 05:58:54 GMT

	Chart	Audit Date	Audit Value	Value Change
🔴	Number of Transactions	12-Apr-2018	6,573,554	2,450,096
🔴	Average Transaction Value	12-Apr-2018	2.49	8.97
🟢	Input Method null	12-Apr-2018	0	0
🟢	Terminal Type Out-of-Range	12-Apr-2018	0	0
🟢	Duplicate Transaction Reference 1	12-Apr-2018	0	0
🟢	Duplicate Transaction Reference 2	12-Apr-2018	0	0
🟢	Duplicate Transaction Time, same PAN	12-Apr-2018	0	0

Published: Fri, 13-Apr-2018 05:58:54 GMT

Monitoring



CHANGES for Number of Transactions
2,500,000
2,000,000
1,500,000
1,000,000
500,000
0
01 Nov 2017
01 Dec 2017
01 Jan 2018
01 Feb 2018
01 Mar 2018
01 Apr 2018
01 May 2018
Chart published: Fri, 13 Apr 2018 06:02:15
Back to Summary
Data Table

Monitoring

- This monitor is an automated test tool
- The tests are are written in SQL
- You can use the automatic detection of unusual situations, and you can create tests with specific limits
- Documentation and code are on my website, and are free.
- You can contact me if you need help.

Monitoring Capacity

- Since we are usually testing data in a live system, we must be careful about the impact we are having on that system.
- Often there is a quiet time at night when can check everything.
- If we are getting close to the time we have available for testing, then we can apply classic risk analysis.

Risk

Probability and Impact

How do we judge the probability of a so-far-unidentified defect?

Impact can be very high:

- Post Office / Fujitsu / Horizon scandal
- NATS flight-plan error
- And many others

Cost of Monitoring

- Our default setting is to run bulk tests every night.
- If that takes too much time, we can run tests less frequently, running different tests on different days.
- If we are confident that some data is static, we can test it less often.
- Apply the risk analysis, realistically.

Improving Data Quality

Behaviour

Policy

Design

Tools

Improvement | Behaviour

- Train users to respect data: no test data in live system and correct data for each field.
- Make sure users have features they need so they don't have to improvise.
- Make sure users can quickly find the record they need so they don't create duplicates.

Improvement | Policy

- Do not use Excel to prepare data
- Character encoding: use UTF-8 everywhere
- When you find a data error, add it to the gate-keeping and monitoring
- Avoid large packaged solutions (Dynamics, Salesforce, etc.)
- Do Not Use a "Common Data Model" - a Common Data Model is an expensive myth.
- Do not use Windows: Use Unix/Linux

Improvement | Design

- Trap data errors at the gate
- Validate and clean bulk-data inserts
- No XML or JSON for structured data
- Do not re-purpose
- Do not deprecate; refactor immediately
- Create flexibility only where there is a specific requirement for it
- Store anonymised data separately

Improvement | Tools

- Use database constraints to enforce unique keys
- Use database constraints to avoid broken links
- Use the most restrictive data-type that is applicable
- Use not-null constraints
- Use third normal form for all tables
- Reconcile bulk data changes
- Profile new data sources
- Set up automated data monitoring

Questions?