

Dealing with missing data

CLEANING DATA IN SQL SERVER DATABASES



Miriam Antona
Software Engineer

Understanding missing data

registration_code	airport_code	carrier_code	canceled	on_time	delayed	diverted
...
000000119	JFK	AA	74	819	233	13
000000130	JFK	HA	NULL	NULL	NULL	NULL
000000131	JFK	HA	NULL	NULL	NULL	NULL
000000132	MSP	YV	0	6	1	0
...

Understanding missing data

registration_code	airport_code	carrier_code	canceled	on_time	delayed	diverted
...
000000119	JFK	AA	74	819	233	13
000000130	JFK	HA				
000000131	JFK	HA				
000000132	MSP	YV	0	6	1	0
...

Understanding missing data - Reasons

- Intentionally
- Error when inserting data
- Data doesn't exist.

Understanding missing data - Solutions

- Recommended:
 - Investigate to get the missing values
- Depending on the business:
 - Leave as it is
 - Remove rows with missing values
 - Fill with other value (text, average, etc.)

Remove missing values - IS NOT NULL

airport_code	airport_name	airport_city	airport_state
...
DFW	Dallas/Fort Worth International	Dallas/Fort Worth	Texas
BOS	Logan International	Boston	Massachusetts
SEA	Seattle/Tacoma International	NULL	NULL
PHL	Philadelphia International	Philadelphia	NULL
SLC	Salt Lake City International	NULL	Utah
MCO	Orlando International	Orlando	Florida
TPA	Tampa International	Tampa	FL
FLL	Fort Lauderdale-Hollywood International	Fort Lauderdale	FL
...

Remove missing values - IS NOT NULL

```
SELECT * FROM airports
WHERE airport_state IS NOT NULL
```

airport_code	airport_name	airport_city	airport_state
...
DFW	Dallas/Fort Worth International	Dallas/Fort Worth	Texas
BOS	Logan International	Boston	Massachusetts
SLC	Salt Lake City International	NULL	Utah
MCO	Orlando International	Orlando	Florida
TPA	Tampa International	Tampa	FL
FLL	Fort Lauderdale-Hollywood International	Fort Lauderdale	FL
...

Remove missing values - IS NOT NULL

```
SELECT * FROM airports
WHERE airport_state IS NULL
```

airport_code	airport_name	airport_city	airport_state
SEA	Seattle/Tacoma International	NULL	NULL
PHL	Philadelphia International	Philadelphia	NULL

Remove missing values - <> "

```
SELECT * FROM airports
```

airport_code	airport_name	airport_city	airport_state
...
DFW	Dallas/Fort Worth International	Dallas/Fort Worth	Texas
BOS	Logan International	Boston	Massachusetts
SEA	Seattle/Tacoma International	NULL	
PHL	Philadelphia International	Philadelphia	
SLC	Salt Lake City International	NULL	Utah
MCO	Orlando International	Orlando	Florida
TPA	Tampa International	Tampa	Fl
FLL	Fort Lauderdale-Hollywood International	Fort Lauderdale	FL
...

Remove missing values - <> ''

```
SELECT * FROM airports
WHERE airport_state <> ''
```

airport_code	airport_name	airport_city	airport_state
...
DFW	Dallas/Fort Worth International	Dallas/Fort Worth	Texas
BOS	Logan International	Boston	Massachusetts
SLC	Salt Lake City International	NULL	Utah
MCO	Orlando International	Orlando	Florida
TPA	Tampa International	Tampa	FL
FLL	Fort Lauderdale-Hollywood International	Fort Lauderdale	FL
...

Fill with other value - ISNULL

```
ISNULL ( check_expression , replacement_value )
```

- ISNULL <> IS NULL

Fill with other value - ISNULL

```
SELECT
  airport_code,
  airport_name,
  airport_city,
  ISNULL(airport_state, 'Unknown') AS airport_state
FROM airports
```

airport_code	airport_name	airport_city	airport_state
...
SEA	Seattle/Tacoma International	NULL	Unkown
PHL	Philadelphia International	Philadelphia	Unkown
SLC	Salt Lake City International	NULL	Utah
MCO	Orlando International	Orlando	Florida
TPA	Tampa International	Tampa	FL
FLL	Fort Lauderdale-Hollywood International	Fort Lauderdale	FL
...

Fill with other value - ISNULL with AVG

registration_code	airport_code	carrier_code	canceled
...
000000128	JFK	B6	181
000000129	JFK	EV	18
000000130	JFK	HA	NULL
000000131	JFK	HA	NULL
000000132	MSP	YV	0
000000133	MSP	AA	15
...

- Replace `NULL` with the average.

Fill with other value - ISNULL with AVG

```
SELECT registration_code, airport_code, carrier_code,  
ISNULL(canceled, (SELECT AVG(canceled) FROM flight_statistics)) AS canceled_fixed  
FROM flight_statistics  
GROUP BY registration_code, airport_code, carrier_code, canceled
```

registration_code	airport_code	carrier_code	canceled
...
000000128	JFK	B6	181
000000129	JFK	EV	18
000000130	JFK	HA	65
000000131	JFK	HA	65
000000132	MSP	YV	0
000000133	MSP	AA	15
...

Fill with other value - COALESCE

```
COALESCE ( arg1, arg2, arg3, ... )
```

SELECT

```
airport_code,  
airport_city,  
airport_state,  
COALESCE (airport_state, airport_city, 'Unknown') AS airport_state_fixed  
FROM airports
```

airport_code	airport_city	airport_state	airport_state_fixed
...
SLC	NULL	Utah	Utah
PHL	Philadelphia	NULL	Philadelphia
SEA	NULL	NULL	Unknown
...

Let's practice!

CLEANING DATA IN SQL SERVER DATABASES

Avoiding duplicate data

CLEANING DATA IN SQL SERVER DATABASES



Miriam Antona
Software Engineer

What is duplicate data?

registration_code	airport_code	carrier_code	registration_date	canceled	on_time	delayed	diverted	statistician_name	statistician_surname
000000134	MSP	DL	2014-01-01	61	3148	925	4	Anne	Johnson
000000134	MSP	DL	2014-01-01	61	3148	925	4	Anne	Johnson

registration_code	airport_code	carrier_code	registration_date	canceled	on_time	delayed	diverted	statistician_name	statistician_surname
000000134	MSP	DL	2014-01-01	61	3148	925	4	Anne	Johnson
000000150	MSP	DL	2014-01-01	61	3148	925	4	Anne	Johnson

registration_code	airport_code	carrier_code	registration_date	canceled	on_time	delayed	diverted	statistician_name	statistician_surname
000000134	MSP	DL	2014-01-01	61	3148	925	4	Anne	Johnson
000000150	MSP	DL	2014-01-01	3148	61	4	925	Anne	Johnson

- Duplicate data can interfere in our analysis!

Finding repeating groups

flight_statistics

```
| registration_code | airport_code | carrier_code | registration_date | canceled | on_time | delayed | diverted | statistician_name | statistician_surname |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
```

No problem to exclude duplicate rows:

```
| registration_code | airport_code | carrier_code | registration_date | canceled | on_time | delayed | diverted | statistician_name | statistician_surname |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 000000134        | MSP         | DL          | 2014-01-01       | 61       | 3148    | 925     | 4        | Anne             | Johnson              |
| 000000150        | MSP         | DL          | 2014-01-01       | 61       | 3148    | 925     | 4        | Bernard          | Ross                 |
```

```
| registration_code | airport_code | carrier_code | registration_date | canceled | on_time | delayed | diverted | statistician_name | statistician_surname |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 000000134        | MSP         | DL          | 2014-01-01       | 61       | 3148    | 925     | 4        | Anne             | Johnson              |
| 000000134        | MSP         | DL          | 2014-01-01       | 61       | 3148    | 925     | 4        | Anne             | Johnson              |
```

Finding repeating groups

- We have a problem to exclude!

registration_code	airport_code	carrier_code	registration_date	canceled	on_time	delayed	diverted	statistician_name	statistician_surname
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
000000134	MSP	DL	2014-01-01	61	3148	925	4	Anne	Johnson
000000150	MSP	DL	2014-01-01	3148	61	4	925	Bernard	Ross

Investigate!

Detecting duplicate data - ROW_NUMBER()

```
ROW_NUMBER () OVER (  
    [ PARTITION BY  
        value_expression ,  
        ... [ n ] ]  
    order_by_clause )
```

- Partitions = repeating groups
- Returns a number starting at 1 for the first row in every partition
- Returns sequential number for each row within the same partition

Detecting duplicate data - ROW_NUMBER()

```
SELECT *,  
    ROW_NUMBER() OVER (  
        PARTITION BY  
            airport_code,  
            carrier_code,  
            registration_date  
        ORDER BY  
            airport_code,  
            carrier_code,  
            registration_date  
    ) row_num  
FROM flight_statistics
```

Detecting duplicate data - ROW_NUMBER()

registration_code	airport_code	carrier_code	registration_date	canceled	...	statistician_name	statistician_surname	row_num
-----	-----	-----	-----	-----	---	-----	-----	-----
...
000000135	MSP	AS	2014-01-01	0	...	Anne	Johnson	1
000000136	MSP	DL	2014-01-01	61	...	Bernard	Ross	1
000000134	MSP	DL	2014-01-01	61	...	Anne	Johnson	2
000000137	MSP	EV	2014-01-01	117	...	Michael	Andersen	1
000000138	MSP	F9	2014-01-01	0	...	Anne	Johnson	1
...

Detecting duplicate data - ROW_NUMBER()

registration_code	airport_code	carrier_code	registration_date	canceled	...	statistician_name	statistician_surname	row_num
...
000000135	MSP	AS	2014-01-01	0	...	Anne	Johnson	1
000000136	MSP	DL	2014-01-01	61	...	Bernard	Ross	1
000000134	MSP	DL	2014-01-01	61	...	Anne	Johnson	2
000000137	MSP	EV	2014-01-01	117	...	Michael	Andersen	1
000000138	MSP	F9	2014-01-01	0	...	Anne	Johnson	1
...

Detecting duplicate data - ROW_NUMBER()

registration_code	airport_code	carrier_code	registration_date	canceled	...	statistician_name	statistician_surname	row_num
...
000000135	MSP	AS	2014-01-01	0	...	Anne	Johnson	1
000000136	MSP	DL	2014-01-01	61	...	Bernard	Ross	1
000000134	MSP	DL	2014-01-01	61	...	Anne	Johnson	2
000000137	MSP	EV	2014-01-01	117	...	Michael	Andersen	1
000000138	MSP	F9	2014-01-01	0	...	Anne	Johnson	1
...

Detecting duplicate data - ROW_NUMBER()

registration_code	airport_code	carrier_code	registration_date	canceled	...	statistician_name	statistician_surname	row_num
...
000000135	MSP	AS	2014-01-01	0	...	Anne	Johnson	1
000000136	MSP	DL	2014-01-01	61	...	Bernard	Ross	1
000000134	MSP	DL	2014-01-01	61	...	Anne	Johnson	2
000000137	MSP	EV	2014-01-01	117	...	Michael	Andersen	1
000000138	MSP	F9	2014-01-01	0	...	Anne	Johnson	1
...

- **Get** duplicate rows:
 - `row_num > 1`
- **Exclude** duplicate rows:
 - `row_num = 1`

Getting only duplicate rows

```
WITH cte AS (  
    SELECT *,  
        ROW_NUMBER() OVER (  
            PARTITION BY  
                airport_code,  
                carrier_code,  
                registration_date  
            ORDER BY  
                airport_code,  
                carrier_code,  
                registration_date  
        ) row_num  
    FROM flight_statistics  
)  
SELECT * FROM cte  
WHERE row_num > 1;
```

Getting only duplicate rows

registration_code	airport_code	carrier_code	registration_date	cancelled	...	statistician_name	statistician_surname	row_num
-----	-----	-----	-----	-----	---	-----	-----	-----
000000131	JFK	EV	2014-02-28	18	...	Anne	Johnson	2
000000134	MSP	DL	2014-01-01	61	...	Anne	Johnson	2
000000142	MSP	00	2014-01-01	76	...	Anne	Johnson	2
000000143	MSP	00	2014-01-01	76	...	Anne	Johnson	3

Excluding duplicate rows

```
WITH cte AS (  
    SELECT *,  
        ROW_NUMBER() OVER (  
            PARTITION BY  
                airport_code,  
                carrier_code,  
                registration_date  
            ORDER BY  
                airport_code,  
                carrier_code,  
                registration_date  
        ) row_num  
    FROM flight_statistics  
)  
SELECT * FROM cte  
WHERE row_num = 1;
```

Excluding duplicate rows

registration_code	airport_code	carrier_code	registration_date	anceled	...	statistician_name	statistician_surname	row_num
-----	-----	-----	-----	-----	...	-----	-----	-----
...	1
000000126	JFK	VX	2014-01-31	12	...	Bryan	Page	1
000000133	MSP	AA	2014-01-01	15	...	Peter	Johnson	1
000000135	MSP	AS	2014-01-01	0	...	Anne	Johnson	1
000000136	MSP	DL	2014-01-01	61	...	Bernard	Ross	1
000000137	MSP	EV	2014-01-01	117	...	Michael	Andersen	1
000000138	MSP	F9	2014-01-01	0	...	Anne	Johnson	1
000000139	MSP	FL	2014-01-01	8	...	Anne	Johnson	1
000000140	MSP	MQ	2014-01-01	20	...	Anne	Johnson	1
000000141	MSP	00	2014-01-01	76	...	Anne	Johnson	1
000000132	MSP	YV	2013-12-01	0	...	Michael	Andersen	1
...

Let's practice!

CLEANING DATA IN SQL SERVER DATABASES

Dealing with different date formats

CLEANING DATA IN SQL SERVER DATABASES



Miriam Antona
Software Engineer

Different date formats

- US English (month/day/year)
 - 04/15/2008
- Spanish (day/month/year)
 - 15/04/2008
- Italian (year/month/day)
 - 2008/04/15
- ...

Checking the date format of our tables

```
SELECT *  
FROM pilots
```

pilot_code	pilot_name	pilot_surname	carrier_code	entry_date
1	Thomas	Peters	HA	2011-10-01
2	Hiroki	Konoe	MQ	2011-01-21
3	Arturo	Montero	UA	2012-12-28
4	David	Captain	US	2000-10-01
5	Ainhua	Guerrera	VX	2000-10-05
6	Alvin	Andersen	00	2012-01-15
7	William	Champy	F9	2011-03-15
...

- Format: yyyy-MM-dd

Functions to modify the date formats

- `CONVERT`
- `FORMAT`

CONVERT

```
CONVERT(data_type[(length)], expression [, style])
```

CONVERT

SELECT

```
CONVERT(VARCHAR(11), CAST(entry_date AS DATE), 0) AS '0',  
CONVERT(VARCHAR(10), CAST(entry_date AS DATE), 1) AS '1',  
CONVERT(VARCHAR(10), CAST(entry_date AS DATE), 101) AS '101',  
CONVERT(VARCHAR(10), CAST(entry_date AS DATE), 2) AS '2',  
CONVERT(VARCHAR(10), CAST(entry_date AS DATE), 120) AS '202'
```

FROM pilots

0	1	101	2	202
-----	-----	-----	-----	-----
Oct 1 2011	10/01/11	10/01/2011	11.10.01	2011-10-01
Jan 21 2011	01/21/11	01/21/2011	11.01.21	2011-01-21
Dec 28 2012	12/28/12	12/28/2012	12.12.28	2012-12-28
Oct 1 2000	10/01/00	10/01/2000	00.10.01	2000-10-01
...

CONVERT

Without century (yy)	With century (yyyy)	Standard	Input/Output
-	0 or 100 (1,2)	Default for datetime and smalldatetime	mon dd yyyy hh:miAM (or PM)
1	101	U.S.	1 = mm/dd/yy 101 = mm/dd/yyyy
2	102	ANSI	2 = yy.mm.dd 102 = yyyy.mm.dd
3	103	British/French	3 = dd/mm/yy 103 = dd/mm/yyyy
5	105	Italian	5 = dd-mm-yy 105 = dd-mm-yyyy
10	110	USA	10 = mm-dd-yy 110 = mm-dd-yyyy
12	112	ISO	12 = yymmdd 112 = yyyyymmdd
...

FORMAT

```
FORMAT ( value, format [, culture ] )
```

- More flexible than `CONVERT`
- Worse performance

FORMAT

```
SELECT FORMAT (CAST(entry_date AS DATE), 'd', 'en-US' ) AS 'US English Result',  
        FORMAT (CAST(entry_date AS DATE), 'd', 'de-de' ) AS 'German Result',  
        FORMAT (CAST(entry_date AS DATE), 'D', 'en-US' ) AS 'US English Result',  
        FORMAT (CAST(entry_date AS DATE), 'dd/MM/yyyy') AS 'DateTime Result'  
from pilots
```

US English Result	German Result	US English Result	DateTime Result
10/1/2011	01.10.2011	Saturday, October 1, 2011	01/10/2011
1/21/2011	21.01.2011	Friday, January 21, 2011	21/01/2011
12/28/2012	28.12.2012	Friday, December 28, 2012	28/12/2012
...

Let's practice!

CLEANING DATA IN SQL SERVER DATABASES