Joanikij Chulev

“This is all my own work. I have not knowingly allowed others to copy my work. This work has not been submitted in any other context. ”

Project 3-Data Cleansing

SCICOMP201 DB

Column patno

This a difficult column to fix. There were non numerical inputs and duplicate inputs etc. I assumed the original inputs of patno:123 and patno:321 are VALID, due to the reason they might be just very high patient numbers and they do check all the requirements.

Rules:

-When the input has two or one X characters in the input(and future inputs) they get replaced with zeros.

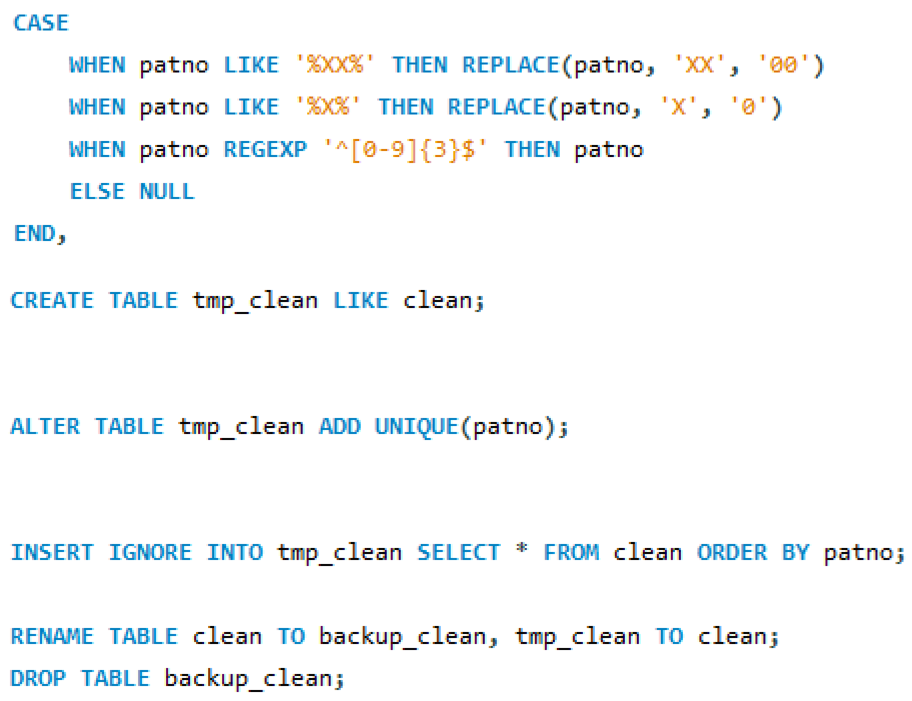
(XX>00,X>0).

-The input only allows for exactly 3 digits raining from 0 to 9.

(digits: (0-9)+(0-9)+(0-9)).

-Leave only UNIQUE values for patno. This removes any duplicates. Than order the table by patno.

-Otherwise the input gets set as NULL.



```
CASE
    WHEN patno LIKE '%XX%' THEN REPLACE(patno, 'XX', '00')
    WHEN patno LIKE '%X%' THEN REPLACE(patno, 'X', '0')
    WHEN patno REGEXP '^[0-9]{3}$' THEN patno
    ELSE NULL
END,

CREATE TABLE tmp_clean LIKE clean;


ALTER TABLE tmp_clean ADD UNIQUE(patno);


INSERT IGNORE INTO tmp_clean SELECT * FROM clean ORDER BY patno;

RENAME TABLE clean TO backup_clean, tmp_clean TO clean;
DROP TABLE backup_clean;
```

Column gender

Rules:

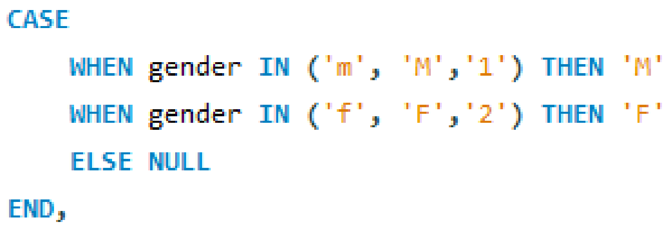
- If gender input is m or 1 it than is changed to M (Male).

-If gender is M it is left as M.

- If gender input is f or 2 it than is changed to F (Female).

-If gender is F it is left as F.

-Otherwise the input gets set as NULL.



```
CASE
    WHEN gender IN ('m', 'M','1') THEN 'M'
    WHEN gender IN ('f', 'F','2') THEN 'F'
    ELSE NULL
END,
```

Column visit

This Column also very proved difficult to clean. It had to be divided in multiple strings and a lot of checks needed to be done for each part due to the variety of the data mistakes in the patient table in the visit column.

Rules:

-If the first part of the date string is a month with 30 days (date input being:1-30 and is between the years 1995 and 2023 we allow for the visit input.

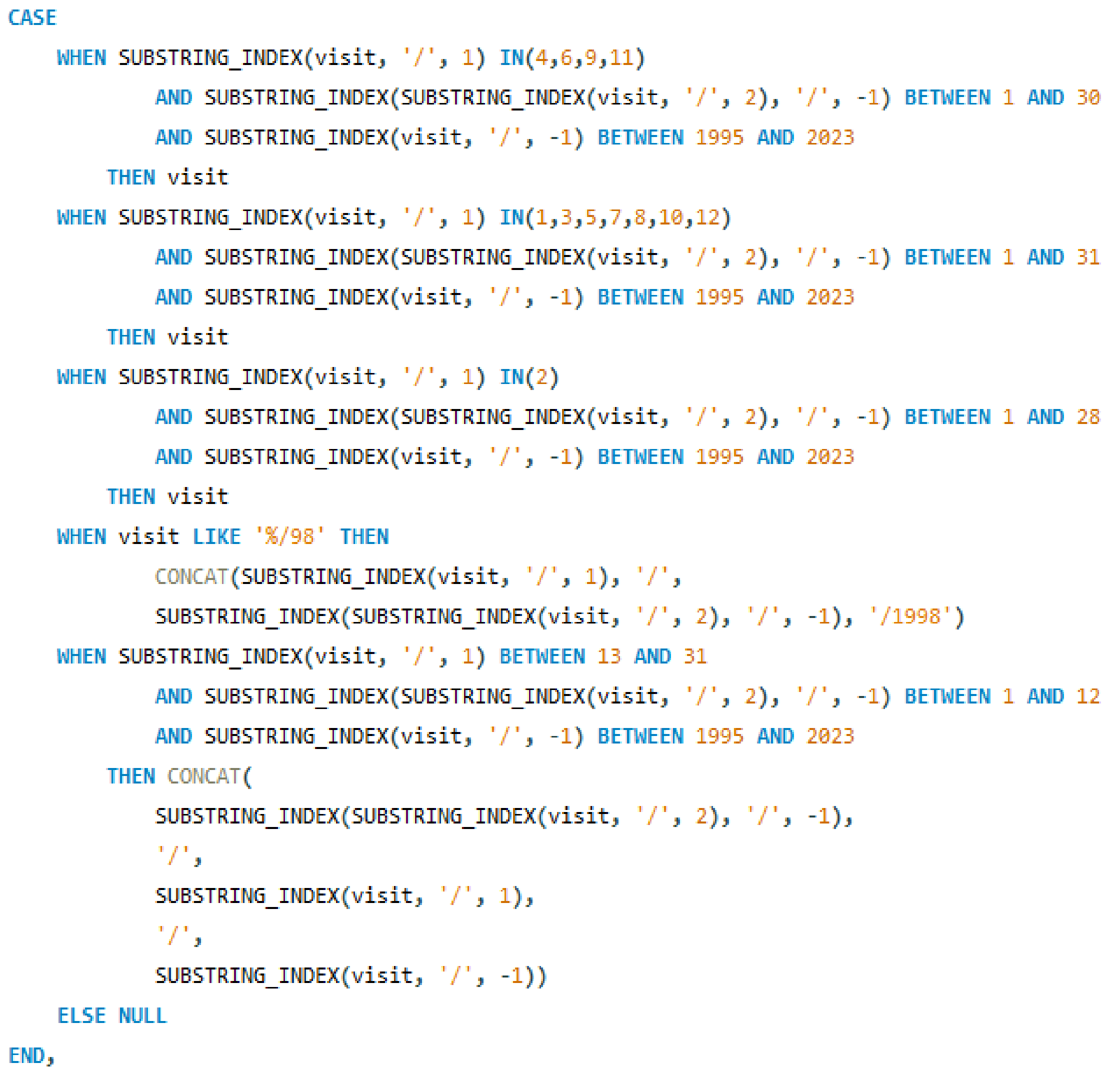
-If the first part of the date string is a month with 31 days (date input being:1-31 and is between the years 1995 and 2023 we allow for the visit input.

-If the month is February('2') with 1-28 days and the year value is between the years 1995 and 2023 we allow for the visit input.

-If the last part of the date string ends with 98 it than gets replaced by 1998 for the correct year.

-If the first part of the date string for month is a value between 13 and 31 and fo the second part of the string for day we have a value between 1 and 12 and the year is between 1995 and 2023, we swap the month and day values to get the correct date input.

-Otherwise the input gets set as NULL.



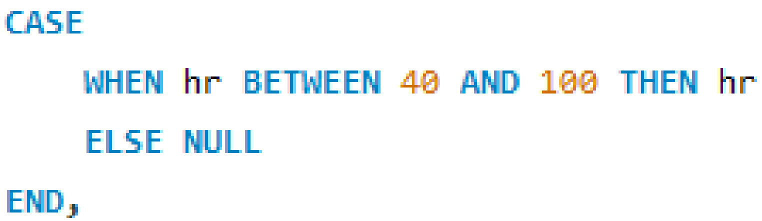
```
CASE
    WHEN SUBSTRING_INDEX(visit, '/', 1) IN(4,6,9,11)
            AND SUBSTRING_INDEX(SUBSTRING_INDEX(visit, '/', 2), '/', -1) BETWEEN 1 AND 30
            AND SUBSTRING_INDEX(visit, '/', -1) BETWEEN 1995 AND 2023
        THEN visit
    WHEN SUBSTRING_INDEX(visit, '/', 1) IN(1,3,5,7,8,10,12)
            AND SUBSTRING_INDEX(SUBSTRING_INDEX(visit, '/', 2), '/', -1) BETWEEN 1 AND 31
            AND SUBSTRING_INDEX(visit, '/', -1) BETWEEN 1995 AND 2023
        THEN visit
    WHEN SUBSTRING_INDEX(visit, '/', 1) IN(2)
            AND SUBSTRING_INDEX(SUBSTRING_INDEX(visit, '/', 2), '/', -1) BETWEEN 1 AND 28
            AND SUBSTRING_INDEX(visit, '/', -1) BETWEEN 1995 AND 2023
        THEN visit
    WHEN visit LIKE '%/98' THEN
            CONCAT(SUBSTRING_INDEX(visit, '/', 1), '/',
            SUBSTRING_INDEX(SUBSTRING_INDEX(visit, '/', 2), '/', -1), '/1998')
    WHEN SUBSTRING_INDEX(visit, '/', 1) BETWEEN 13 AND 31
            AND SUBSTRING_INDEX(SUBSTRING_INDEX(visit, '/', 2), '/', -1) BETWEEN 1 AND 12
            AND SUBSTRING_INDEX(visit, '/', -1) BETWEEN 1995 AND 2023
        THEN CONCAT(
            SUBSTRING_INDEX(SUBSTRING_INDEX(visit, '/', 2), '/', -1),
            '/',
            SUBSTRING_INDEX(visit, '/', 1),
            '/',
            SUBSTRING_INDEX(visit, '/', -1))
    ELSE NULL
END,
```

Column hr

Rules:

-Only allows for heart rate between 40 and 100 bpm.

-Otherwise the input gets set as NULL.



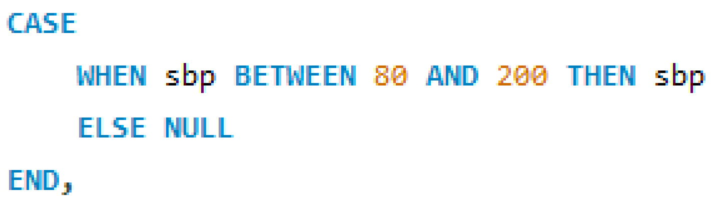
```
CASE
    WHEN hr BETWEEN 40 AND 100 THEN hr
    ELSE NULL
END,
```

Column sbp

Rules:

-Only allows for heart rate between 80 and 200.

-Otherwise the input gets set as NULL.



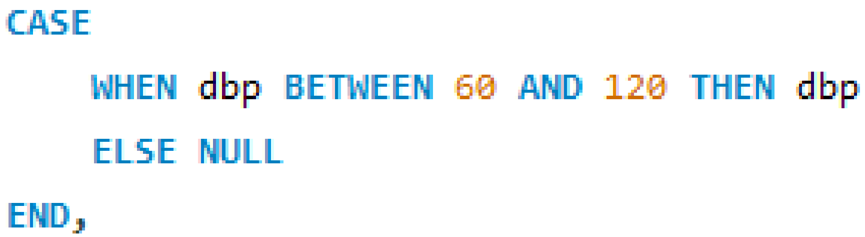
```
CASE
    WHEN sbp BETWEEN 80 AND 200 THEN sbp
    ELSE NULL
END,
```

Column dbp

Rules:

-Only allows for heart rate between 60 and 120.

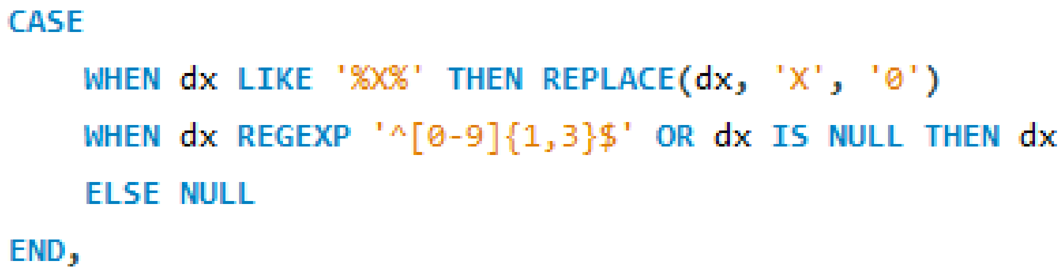
-Otherwise the input gets set as NULL.



```
CASE
    WHEN dbp BETWEEN 60 AND 120 THEN dbp
    ELSE NULL
END,
```

Column dx

This column cleaned will follow the digit rule but also I assumed that in general the input of X represents 0 as seen in the patno column. Thus the same method of replacing Xs’ with 0s’ has been utilized.



```
CASE
    WHEN dx LIKE '%X%' THEN REPLACE(dx, 'X', '0')
    WHEN dx REGEXP '^[0-9]{1,3}$' OR dx IS NULL THEN dx
    ELSE NULL
END,
```

Rules:

-When the input has an X character it gets set to 0.

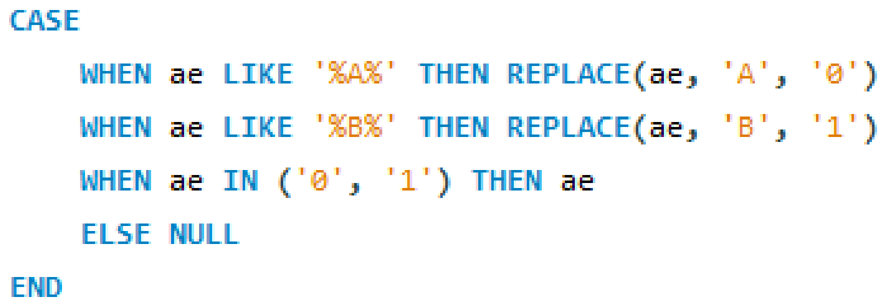
-The input only allows for 1,2 or 3 digits raining from 0 to 9.

(digits example: 1 or 12 or 123, but not 1111).

-Otherwise the input gets set as NULL.

Column ae

Here I assumed that the input value of A was mistaken for 0 like for example in the gender column as 1 was represented as male and 2 as female. Correspondingly to provide additional measures B was assumed to be 1.



```
CASE
    WHEN ae LIKE '%A%' THEN REPLACE(ae, 'A', '0')
    WHEN ae LIKE '%B%' THEN REPLACE(ae, 'B', '1')
    WHEN ae IN ('0', '1') THEN ae
    ELSE NULL
END
```

Rules:

-When the input has an A character it gets set to 0.

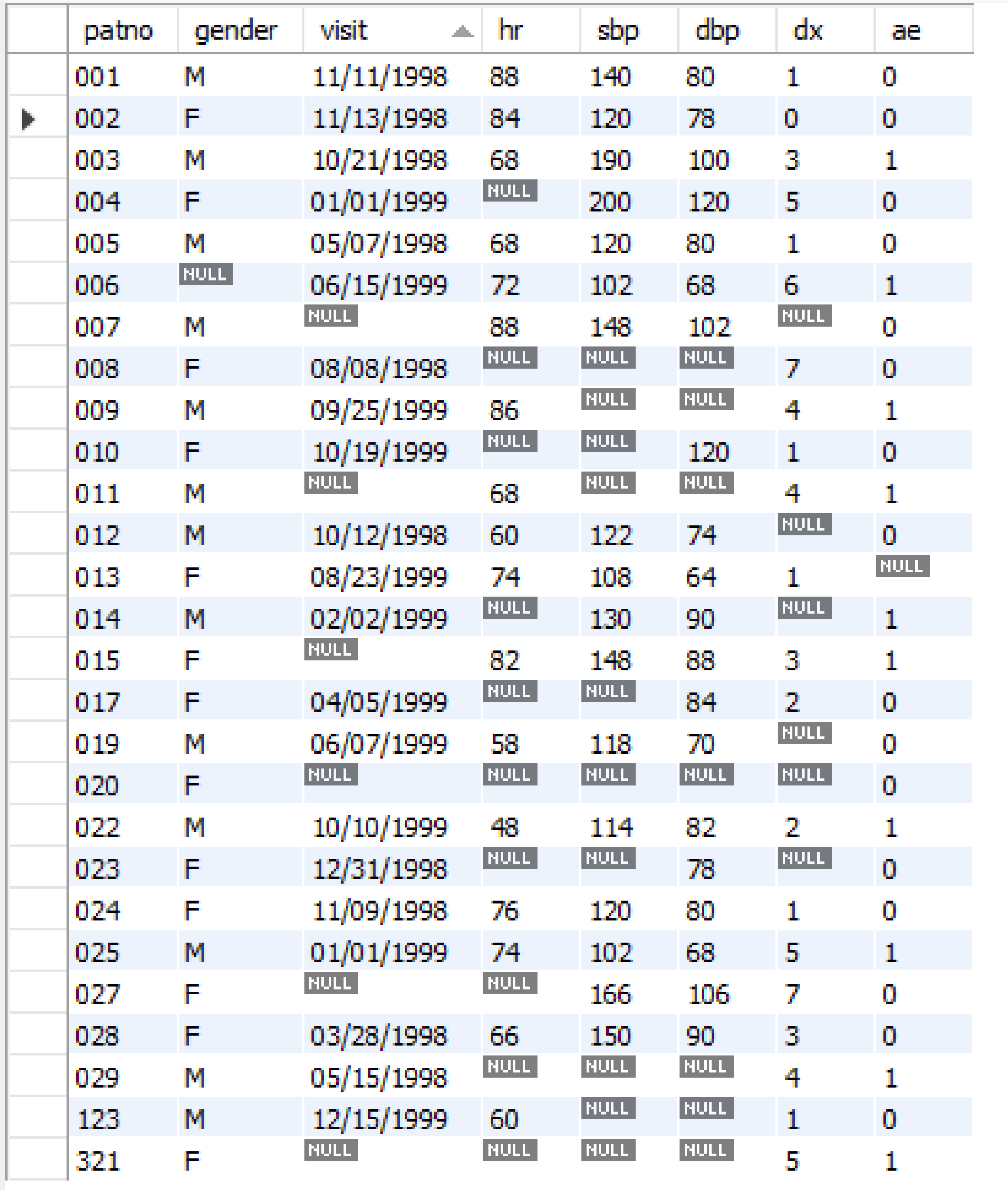
-When the input has a B character it gets set to 1.

-Only allows for 0 or 1 input (0 being for NO to adverse event and 1 being YES).

-Otherwise the input gets set as NULL.

Table of cleansed data-clean TABLE

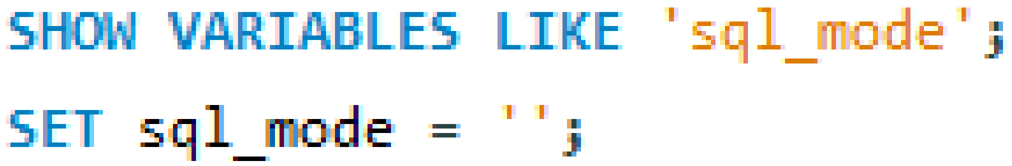
As we can see we had to NULL a bunch of STRICT values for hr, sbp, dbp because these are just incorrect integer values that are hard to interpret.



patno	gender	visit	hr	sbp	dbp	dx	ae
001	M	11/11/1998	88	140	80	1	0
002	F	11/13/1998	84	120	78	0	0
003	M	10/21/1998	68	190	100	3	1
004	F	01/01/1999	NULL	200	120	5	0
005	M	05/07/1998	68	120	80	1	0
006	NULL	06/15/1999	72	102	68	6	1
007	M	NULL	88	148	102	NULL	0
008	F	08/08/1998	NULL	NULL	NULL	7	0
009	M	09/25/1999	86	NULL	NULL	4	1
010	F	10/19/1999	NULL	NULL	120	1	0
011	M	NULL	68	NULL	NULL	4	1
012	M	10/12/1998	60	122	74	NULL	0
013	F	08/23/1999	74	108	64	1	NULL
014	M	02/02/1999	NULL	130	90	NULL	1
015	F	NULL	82	148	88	3	1
017	F	04/05/1999	NULL	NULL	84	2	0
019	M	06/07/1999	58	118	70	NULL	0
020	F	NULL	NULL	NULL	NULL	NULL	0
022	M	10/10/1999	48	114	82	2	1
023	F	12/31/1998	NULL	NULL	78	NULL	0
024	F	11/09/1998	76	120	80	1	0
025	M	01/01/1999	74	102	68	5	1
027	F	NULL	NULL	166	106	7	0
028	F	03/28/1998	66	150	90	3	0
029	M	05/15/1998	NULL	NULL	NULL	4	1
123	M	12/15/1999	60	NULL	NULL	1	0
321	F	NULL	NULL	NULL	NULL	5	1

Acknowledgements

A thank you to Professor Brooks for fixing an issue with the code regarding > Error Code: 1292. Truncated incorrect DOUBLE value. It was a very niche problem that did not stem for programing logic or rules.



```
SHOW VARIABLES LIKE 'sql_mode';

SET sql_mode = '';
```

This code fixed the issue after restarting the MYSQL service and opening it once more.