

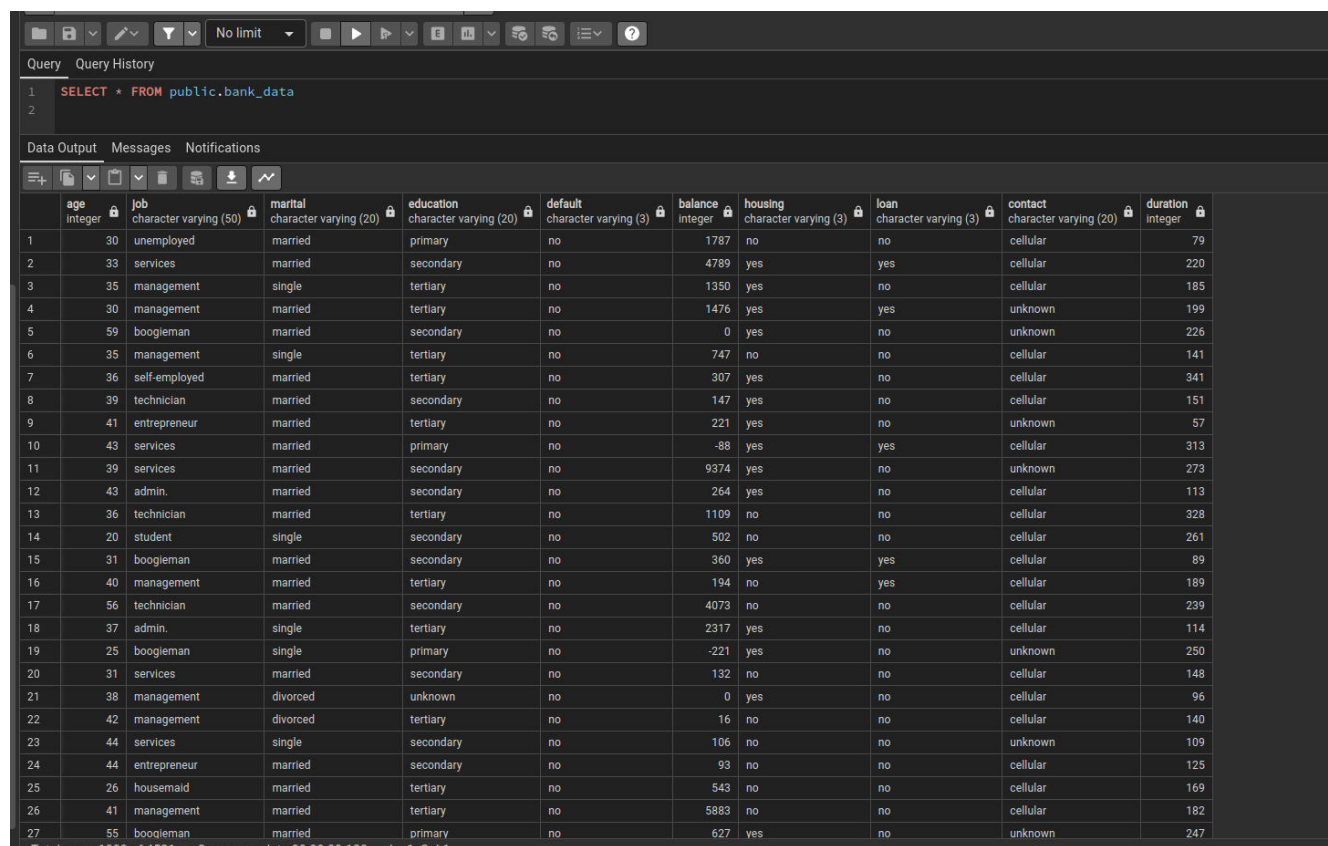


DATA QUALITY HOMEWORK

Defect reporting

1.

For bank.csv file I'm going to use PGAdmin4 to make analysis, I have created bank_data table and populated it using out file.



The screenshot shows the PGAdmin4 interface. At the top, there's a toolbar with various icons. Below it, the 'Query' tab is active, showing a SQL query: `SELECT * FROM public.bank_data`. The 'Data Output' tab is also visible, showing a table with 11 columns and 27 rows of data. The columns are: age (integer), job (character varying (50)), marital (character varying (20)), education (character varying (20)), default (character varying (3)), balance (integer), housing (character varying (3)), loan (character varying (3)), contact (character varying (20)), and duration (integer). The data rows show various customer profiles, such as unemployed, services, management, boogleman, technician, entrepreneur, student, divorced, and housemaid, with their respective ages, marital statuses, education levels, and other financial details.

	age integer	job character varying (50)	marital character varying (20)	education character varying (20)	default character varying (3)	balance integer	housing character varying (3)	loan character varying (3)	contact character varying (20)	duration integer
1	30	unemployed	married	primary	no	1787	no	no	cellular	79
2	33	services	married	secondary	no	4789	yes	yes	cellular	220
3	35	management	single	tertiary	no	1350	yes	no	cellular	185
4	30	management	married	tertiary	no	1476	yes	yes	unknown	199
5	59	boogleman	married	secondary	no	0	yes	no	unknown	226
6	35	management	single	tertiary	no	747	no	no	cellular	141
7	36	self-employed	married	tertiary	no	307	yes	no	cellular	341
8	39	technician	married	secondary	no	147	yes	no	cellular	151
9	41	entrepreneur	married	tertiary	no	221	yes	no	unknown	57
10	43	services	married	primary	no	-88	yes	yes	cellular	313
11	39	services	married	secondary	no	9374	yes	no	unknown	273
12	43	admin.	married	secondary	no	264	yes	no	cellular	113
13	36	technician	married	tertiary	no	1109	no	no	cellular	328
14	20	student	single	secondary	no	502	no	no	cellular	261
15	31	boogleman	married	secondary	no	360	yes	yes	cellular	89
16	40	management	married	tertiary	no	194	no	yes	cellular	189
17	56	technician	married	secondary	no	4073	no	no	cellular	239
18	37	admin.	single	tertiary	no	2317	yes	no	cellular	114
19	25	boogleman	single	primary	no	-221	yes	no	unknown	250
20	31	services	married	secondary	no	132	no	no	cellular	148
21	38	management	divorced	unknown	no	0	yes	no	cellular	96
22	42	management	divorced	tertiary	no	16	no	no	cellular	140
23	44	services	single	secondary	no	106	no	no	unknown	109
24	44	entrepreneur	married	secondary	no	93	no	no	cellular	125
25	26	housemaid	married	tertiary	no	543	no	no	cellular	169
26	41	management	married	tertiary	no	5883	no	no	cellular	182
27	55	boogleman	married	primary	no	627	yes	no	unknown	247

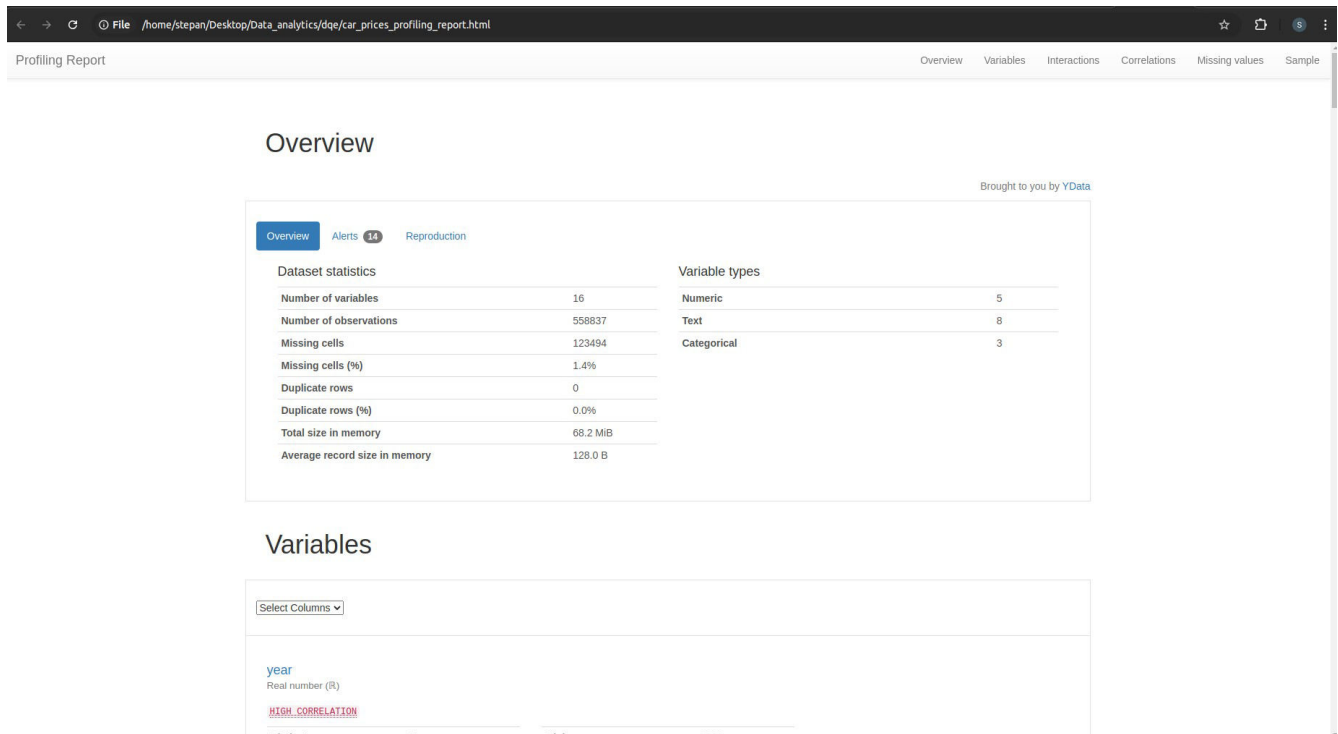
For car_prices.parquet file I will use [ydata-profiling](#) library , so I have installed it in my envioment and executed script below.

```
import pandas as pd
```

```
df = pd.read_parquet('car_prices.parquet')
```

```
df.to_csv('car_prices.csv', index=False)
```

After executing we got an html file.



About parquet:

Parquet is column-oriented data file format designed for efficient data storage and retrieval, good for storing big data of any kind, saves on cloud storage space by using highly efficient column-wise compression, and flexible encoding schemes for columns with different data types. Increased data throughput and performance using techniques like data skipping, whereby queries that fetch specific column values need not read the entire row of data. It is implemented using record-shredding and assembly algorithm. Columnar storage like Apache Parquet is designed to bring efficiency compared to row-based files like CSV. When querying, columnar storage you can skip over the non-relevant data very quickly. As a result, aggregation queries are less time-consuming compared to row-oriented databases. This way of storage has translated into hardware savings and minimized latency for accessing data. Apache Parquet works best with interactive and serverless technologies like AWS Athena, Amazon Redshift Spectrum, Google BigQuery and Google Dataproc. Its 34 x faster compared with csv files, no much costs.

2.1

After conversion parquet to csv file I noticed that the csv file is 91.9MB while parquet is 17.7 MB, it is because parquet files build using compression algorithms, also encoding of the parquet files is different. These techniques minimize storage needs by encoding repeated or similar values compactly.

2.2

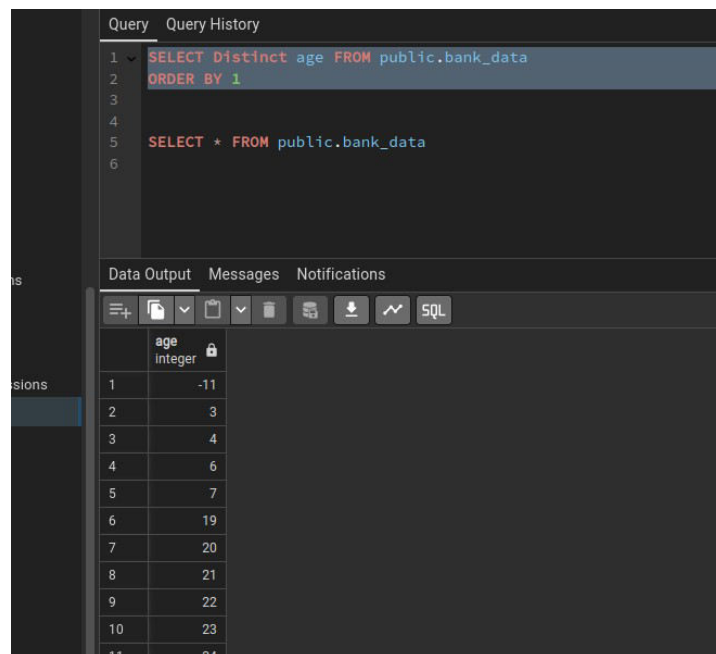
Bank dataset contains information about customers of the bank , we have age of the customer , job description ,marital status ,education level , nor sure about default column what it contains , customer balance , housing info yes or no(also not quite clear what is this column for) , loan exists or not , contact type , and duration of something I guess it can be number of days after becoming bank customer , I mean current date – first_date. Business can get analysis from this dataset about how many customers are from different job spheres , how many of the has primary secondary or tertiary education , also most the most loyal customers ,etc.

Car prices dataset contains sale record of cars , basically we have all information , vehicle production date , make, model , trim , body type , transmission type, vin code , state of store who made a sale, condition rank(I guess this needs to be provided by business for proper analysis) , odometer record , body color , interior color , seller name, MMR (Manheim Market Report (MMR) is the leading indicator of wholesale prices across the industry) , selling price and sale date. Business can get a lot of information about trending vehicles, most solded ones , differenced between MMR and actuall sale prices , analysis by date range (for example 2015 year sales) ,Top vehicle sold seller, conditions of vehicles, etc.

2.3

Bank dataset anomalies:

1. Age column contains negative value which is not expected and also there is so high values but lets assume it's ok.



The screenshot shows a SQL query editor with the following query:

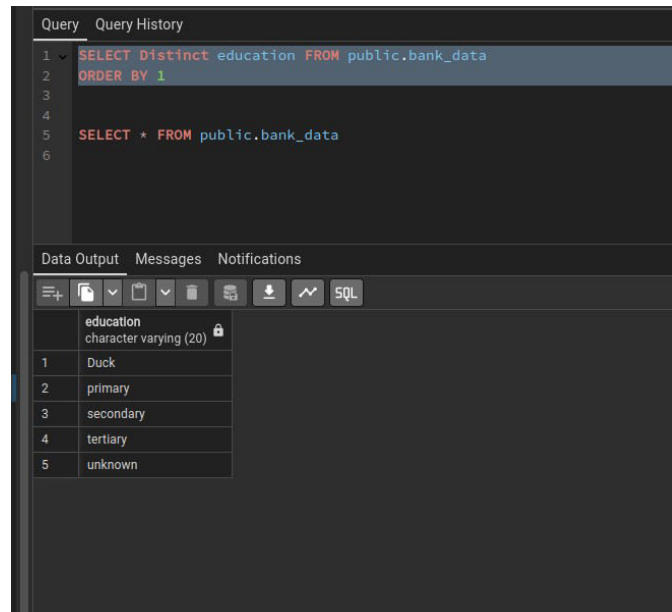
```
1 SELECT Distinct age FROM public.bank_data
2 ORDER BY 1
3
4
5 SELECT * FROM public.bank_data
6
```

The results pane shows the output of the query, displaying a list of ages from -11 to 24. The column is labeled 'age' and 'Integer'.

age
-11
3
4
6
7
19
20
21
22
23
24

2. job column has value “admin.” and I think its also anomaly, there shouldn't be any symbols in this column.

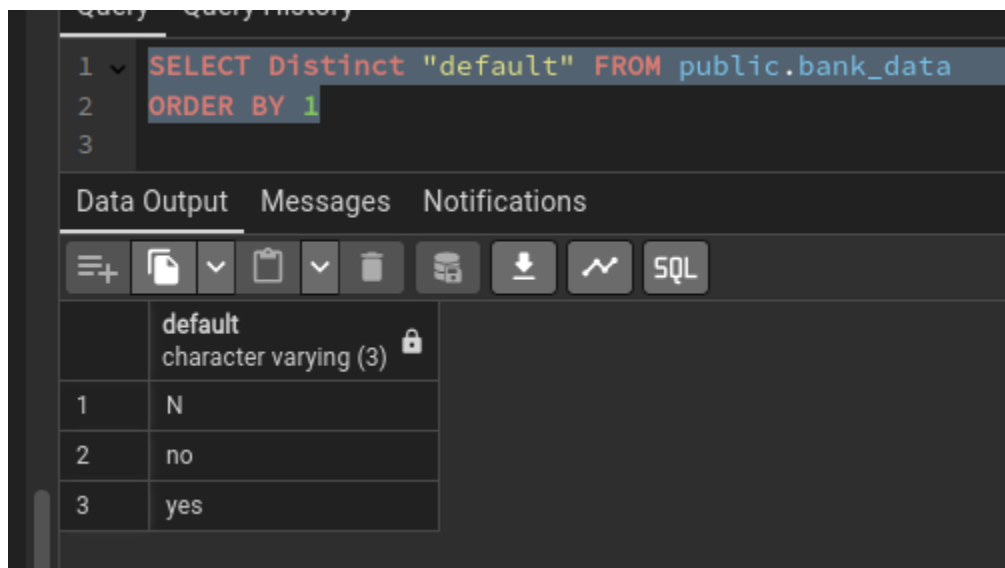
3. Column education has value Duck , which is not education level name :)



The screenshot shows a SQL IDE with a query editor and a results pane. The query editor contains two SQL queries. The first query is highlighted and is: `SELECT Distinct education FROM public.bank_data ORDER BY 1`. The second query is: `SELECT * FROM public.bank_data`. The results pane shows the output of the first query. It has a table with 5 rows and 1 column named 'education'. The data values are 'Duck', 'primary', 'secondary', 'tertiary', and 'unknown'.

education
1 Duck
2 primary
3 secondary
4 tertiary
5 unknown

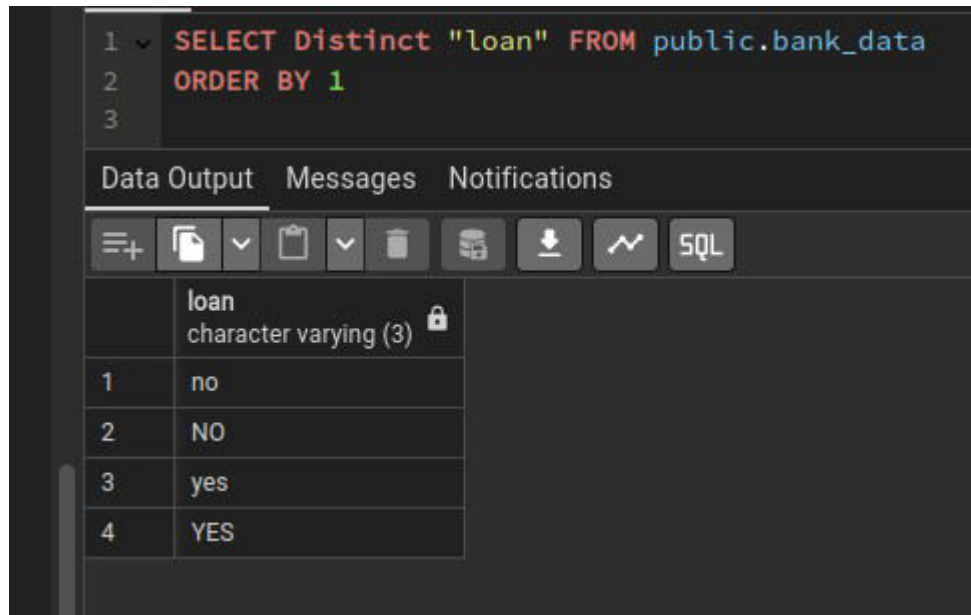
4. The name of the column default is problematic for DBMS's because the word default is used in sql language so it cause some conflicts , also it's contains value "N" which don't have any meaning in this case ,we need to have only "yes" or "no" .



The screenshot shows a SQL IDE with a query editor and a results pane. The query editor contains a SQL query: `SELECT Distinct "default" FROM public.bank_data ORDER BY 1`. The results pane shows the output of the query. It has a table with 3 rows and 1 column named 'default'. The data values are 'N', 'no', and 'yes'.

default
1 N
2 no
3 yes

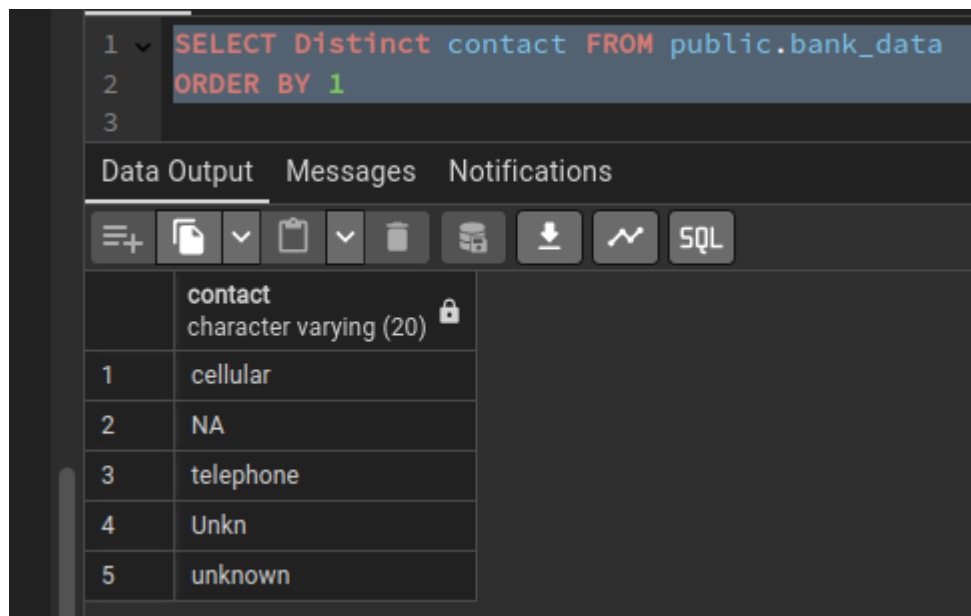
5. Column loan contains values with the same word but in different cases.



```
1 SELECT Distinct "loan" FROM public.bank_data
2 ORDER BY 1
3
```

	loan character varying (3) 🔒
1	no
2	NO
3	yes
4	YES

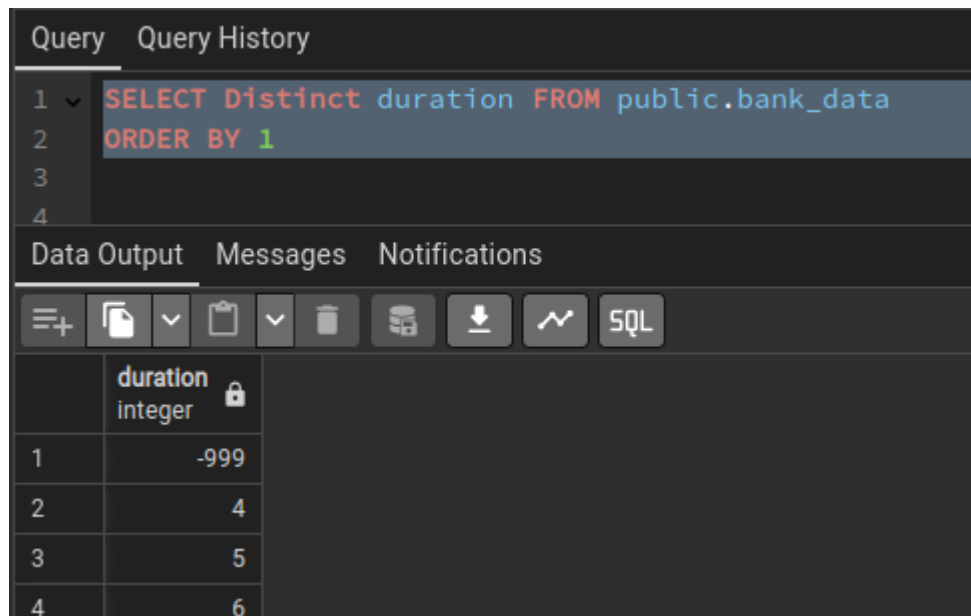
6. Column contact has 5 values but 3 of them are most likely same , NA , Unkn and unknown , all should be unknown .



```
1 SELECT Distinct contact FROM public.bank_data
2 ORDER BY 1
3
```

	contact character varying (20) 🔒
1	cellular
2	NA
3	telephone
4	Unkn
5	unknown

7. Duration column has value -999 but if we don't have any arrangement then it shouldn't be like that .



Query Query History

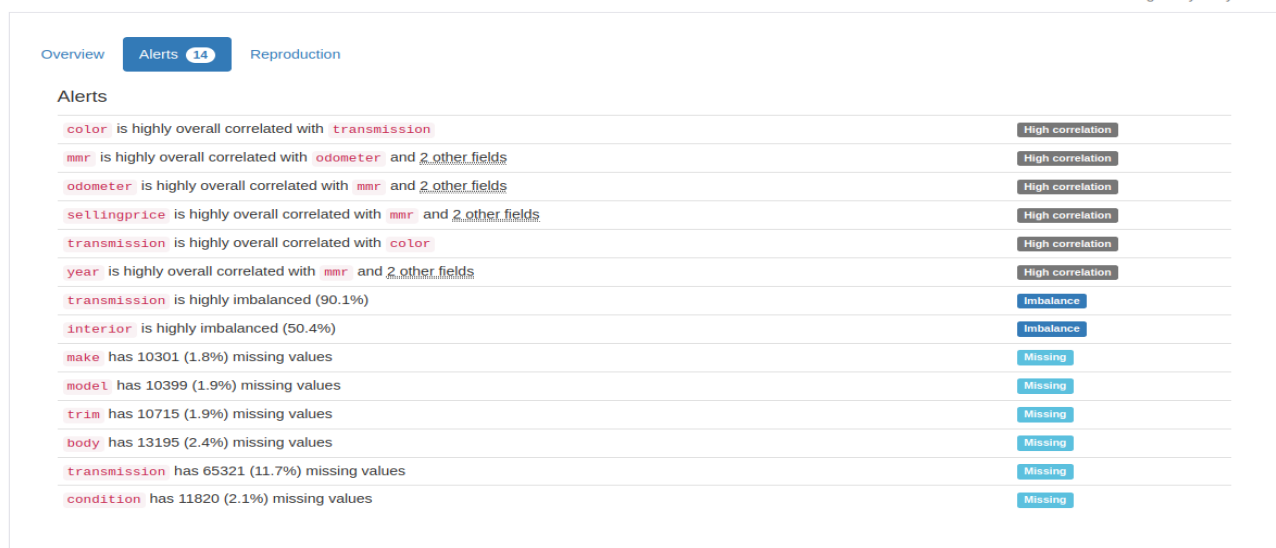
```
1 SELECT Distinct duration FROM public.bank_data
2 ORDER BY 1
3
4
```

Data Output Messages Notifications

	duration integer
1	-999
2	4
3	5
4	6

Car prices dataset :

In Ydata html report we have this Alerts`



Overview Alerts 14 Reproduction

Alerts

color is highly overall correlated with transmission	High correlation
mmr is highly overall correlated with odometer and 2 other fields	High correlation
odometer is highly overall correlated with mmr and 2 other fields	High correlation
sellingprice is highly overall correlated with mmr and 2 other fields	High correlation
transmission is highly overall correlated with color	High correlation
year is highly overall correlated with mmr and 2 other fields	High correlation
transmission is highly imbalanced (90.1%)	Imbalance
interior is highly imbalanced (50.4%)	Imbalance
make has 10301 (1.8%) missing values	Missing
model has 10399 (1.9%) missing values	Missing
trim has 10715 (1.9%) missing values	Missing
body has 13195 (2.4%) missing values	Missing
transmission has 65321 (11.7%) missing values	Missing
condition has 11820 (2.1%) missing values	Missing

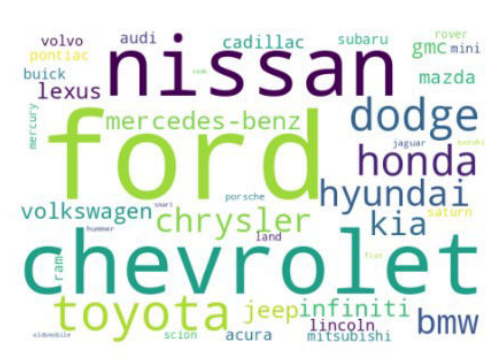
- 1. We have missing column name for year .
- 2. Column make has same values in different cases and has null values .

make

Text

MISSING

Distinct	96
Distinct (%)	< 0.1%
Missing	10301
Missing (%)	1.8%
Memory size	4.3 MiB



Word cloud showing car brands: nissan, ford, chevrolet, toyota, mercedes-benz, chrysler, volkswagen, honda, dodge, jeep, infiniti, bmw, audi, cadillac, subaru, gmc, mini, mazda, lexus, buick, pontiac, volvo, mercury, jaguar, hummer, land, porsche, volkswagen, toyota, jeep, infiniti, bmw, audi, cadillac, subaru, gmc, mini, mazda, lexus, buick, pontiac, volvo, mercury, jaguar, hummer, land, porsche.

More details

Overview

Words

Characters

Length	Characters and Unicode	Unique	Sample
Max length13	Total characters3288596	Unique8	1st rowKia
Median length11	Distinct characters49	Unique (%)< 0.1%	2nd rowKia
Mean length5.9952236	Distinct categories1		3rd rowBMW
Min length2	Distinct scripts1		4th rowVolvo
	Distinct blocks1		5th rowBMW

The Unicode Standard assigns character properties to each code point, which can be used to analyse textual variables.

- 3. Column model contains some dates in it which is not expected , also other values has case problems.

12

13

14

SELECT DISTINCT model FROM public.car_sales

Data Output

Messages

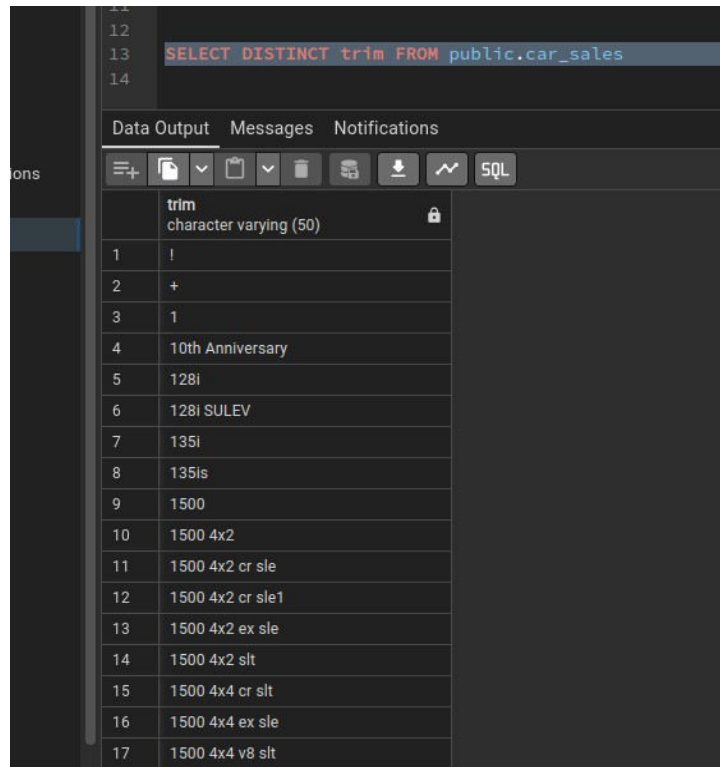
Notifications

+

SQL

model
character varying (100)
09-Mar
09-May
1
1500
190-Class
1 Series
200
200SX
2500
2 Series
3
300
3000GT
300-Class
300e
300M
300ZX

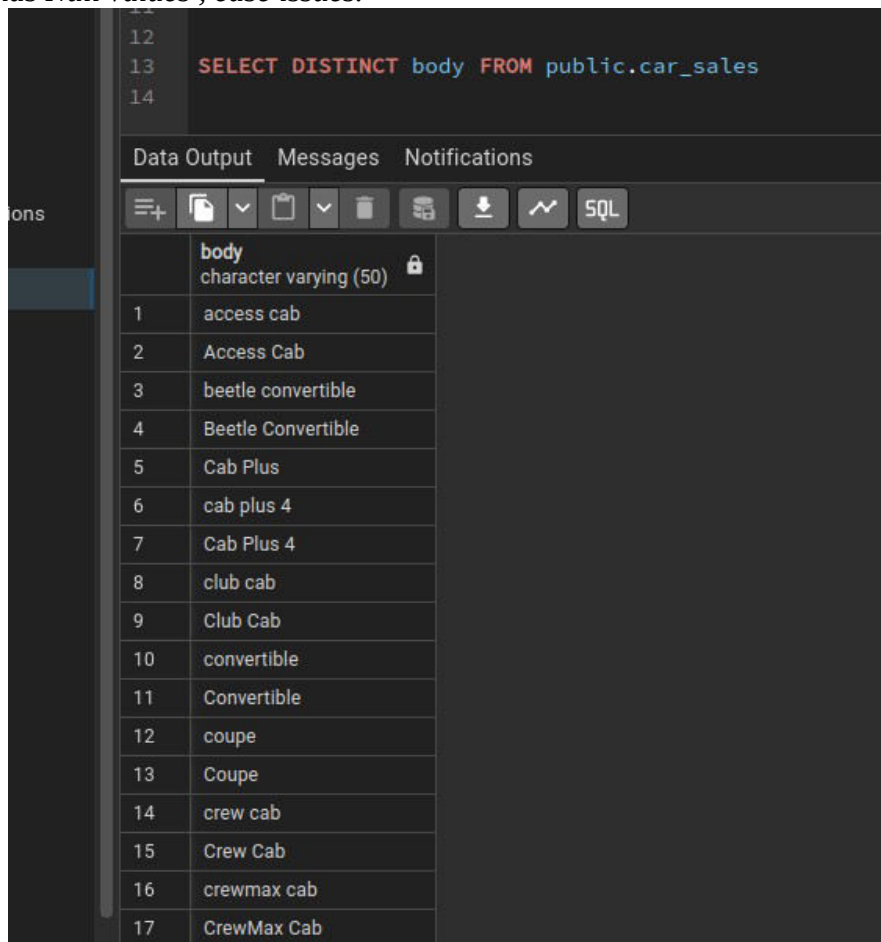
4. Trim column contains !,+ symbols , upper lower case issue.



The screenshot shows a SQL IDE with a query editor at the top containing the SQL statement: `SELECT DISTINCT trim FROM public.car_sales`. Below the editor, the 'Data Output' tab is active, displaying a table with 17 rows of results. The table has two columns: 'trim' (character varying (50)) and an empty column. The results show various car models with their trim levels, including '!', '+', '1', '10th Anniversary', '128i', '128i SULEV', '135i', '135is', '1500', '1500 4x2', '1500 4x2 cr sle', '1500 4x2 cr sle1', '1500 4x2 ex sle', '1500 4x2 slt', '1500 4x4 cr slt', '1500 4x4 ex sle', and '1500 4x4 v8 slt'.

	trim	
1	!	
2	+	
3	1	
4	10th Anniversary	
5	128i	
6	128i SULEV	
7	135i	
8	135is	
9	1500	
10	1500 4x2	
11	1500 4x2 cr sle	
12	1500 4x2 cr sle1	
13	1500 4x2 ex sle	
14	1500 4x2 slt	
15	1500 4x4 cr slt	
16	1500 4x4 ex sle	
17	1500 4x4 v8 slt	

5. Body column has Null values , case issues.



The screenshot shows a SQL IDE with a query editor at the top containing the SQL statement: `SELECT DISTINCT body FROM public.car_sales`. Below the editor, the 'Data Output' tab is active, displaying a table with 17 rows of results. The table has two columns: 'body' (character varying (50)) and an empty column. The results show various car models with their body styles, including 'access cab', 'Access Cab', 'beetle convertible', 'Beetle Convertible', 'Cab Plus', 'cab plus 4', 'Cab Plus 4', 'club cab', 'Club Cab', 'convertible', 'Convertible', 'coupe', 'Coupe', 'crew cab', 'Crew Cab', 'crewmax cab', and 'CrewMax Cab'.

	body	
1	access cab	
2	Access Cab	
3	beetle convertible	
4	Beetle Convertible	
5	Cab Plus	
6	cab plus 4	
7	Cab Plus 4	
8	club cab	
9	Club Cab	
10	convertible	
11	Convertible	
12	coupe	
13	Coupe	
14	crew cab	
15	Crew Cab	
16	crewmax cab	
17	CrewMax Cab	

6. Column transmission has null values and also contains values are not related to transmission type.

11
12
13
14

SELECT DISTINCT transmission FROM public.car_sales

Data OutputMessagesNotifications

≡+📄▼📋▼🗑️🗄️⬇️📈SQL

	transmission character varying (20) 🔒
1	automatic
2	horse-driven
3	manual
4	sedan
5	Sedan
6	[null]

transmission

Categorical

HIGH CORRELATIONIMBALANCEMISSING

Distinct	5
Distinct (%)	< 0.1%
Missing	65321
Missing (%)	11.7%
Memory size	4.3 MIB

automatic475677

manual17539

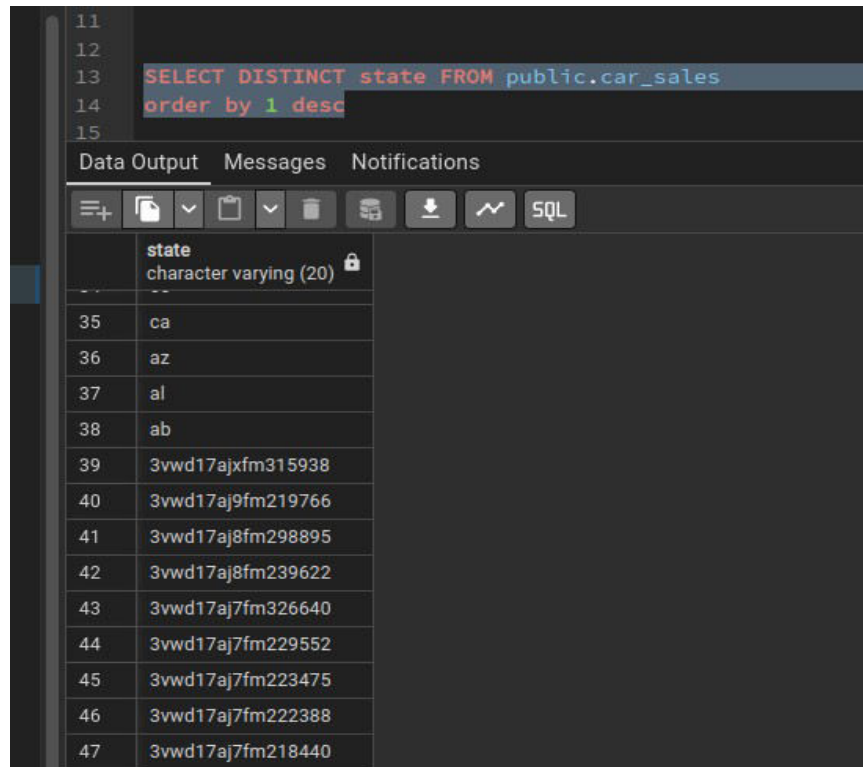
horse-driven274

sedan15

Sedan11

More details

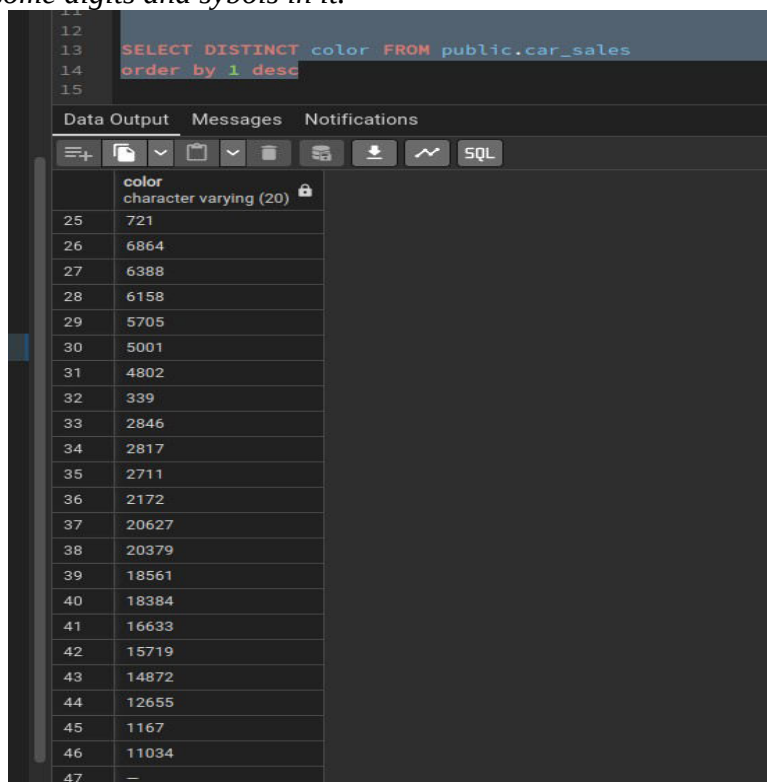
7. Column state has vin code values in it.



```
11
12
13 SELECT DISTINCT state FROM public.car_sales
14 order by 1 desc
15
```

	state
35	ca
36	az
37	al
38	ab
39	3vwd17ajxfm315938
40	3vwd17aj9fm219766
41	3vwd17aj8fm298895
42	3vwd17aj8fm239622
43	3vwd17aj7fm326640
44	3vwd17aj7fm229552
45	3vwd17aj7fm223475
46	3vwd17aj7fm222388
47	3vwd17aj7fm218440

8. Color column has some digits and sybols in it.



```
11
12
13 SELECT DISTINCT color FROM public.car_sales
14 order by 1 desc
15
```

	color
25	721
26	6864
27	6388
28	6158
29	5705
30	5001
31	4802
32	339
33	2846
34	2817
35	2711
36	2172
37	20627
38	20379
39	18561
40	18384
41	16633
42	15719
43	14872
44	12655
45	1167
46	11034
47	-

