databrickspart1

(https://databricks.com)

Setup

Modules

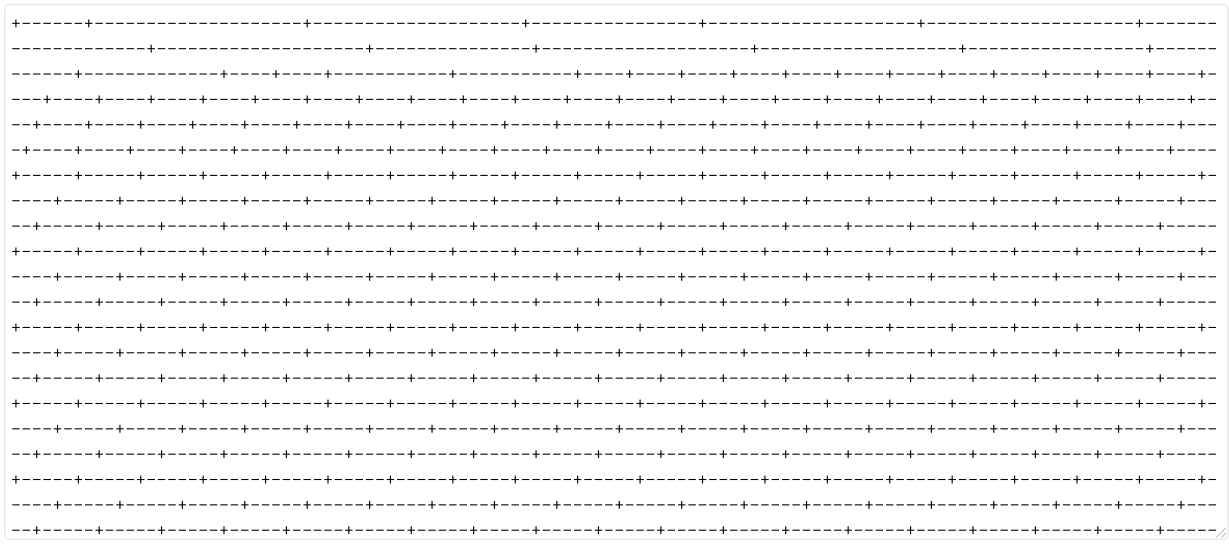
```
# modules
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
```

Data

```
# skill2vec_50k data
skill2vec_raw = sqlContext.read.format("csv").option("header",
"false").load("/FileStore/tables/skill2vec_50K.csv.gz")

# technology skills data
technology_raw = sqlContext.read.format("csv").option("header",
"true").option("delimiter","\t").load("/FileStore/tables/Technology_Skills.txt")
```

```
# check skill data
skill2vec_raw.show(10)
```



```
# check technology data
technology_raw.show(10)
```

```
+--------------+--------------------+--------------+--------------------+--------------+----------+
|O*NET-SOC Code|             Example|Commodity Code|     Commodity Title|Hot Technology|In Demand|
+--------------+--------------------+--------------+--------------------+--------------+----------+
|    11-1011.00|Adobe Systems Ado...|      43232202|Document manageme...|             Y|         N|
|    11-1011.00|     AdSense Tracker|      43232306|Data base user in...|             N|         N|
|    11-1011.00|      Atlassian JIRA|      43232201|Content workflow ...|             Y|         N|
|    11-1011.00|Blackbaud The Rai...|      43232303|Customer relation...|             N|         N|
|    11-1011.00|ComputerEase cons...|      43231601| Accounting software|             N|         N|
|    11-1011.00|Database reportin...|      43232305|Data base reporti...|             N|         N|
|    11-1011.00|             Databox|      43232306|Data base user in...|             N|         N|
|    11-1011.00|      Email software|      43233501|Electronic mail s...|             N|         N|
|    11-1011.00|Enterprise resour...|      43231602|Enterprise resour...|             N|         N|
|    11-1011.00|Exact Software Ma...|      43231605|Time accounting s...|             N|         N|
+--------------+--------------------+--------------+--------------------+--------------+----------+
only showing top 10 rows
```

```
# register as tables to use in SQL
sqlContext.registerDataFrameAsTable(skill2vec_raw, "skill2vec_raw")
sqlContext.registerDataFrameAsTable(technology_raw, "technology_raw")
```

Data cleaning and preparation

Skill data

(a) Data checking

```
# check null values
skill2vec_raw.select([count(when(col(c).isNull(), c)).alias(c) for c in skill2vec_raw.columns]).display()
```

Table

	_c0	_c1	_c2	_c3	_c4	_c5	_c6	_c7	_c8
1	0	0	3367	5827	8476	11612	15048	18456	21802

1 row

```
# check data by _c1 name order
skill2vec_raw.sort(col('_c1').asc()).display()
```

Table

	_c0	_c1	_c
1	50329	#Expertise of multi plant handling#	nu
2	52031	&quot ; AR Analyst&quot ;	&
3	76414	&quot ; ASP.Net MVC	nu
4	71494	&quot ; Accounts Executive	nu
5	12327	&quot ; Accounts&quot ;	&
6	14484	&quot ; Android	nu
7	52297	&quot ; As400 Administrator&quot ;	is

419 rows | Truncated data

(b) Data restructuring

```
# save columns of skill2vec_raw
cols = skill2vec_raw.columns
cols.remove('_c0')

# skill2vec_list
skill2vec_temp = skill2vec_raw.withColumn('skill_list', array(cols))
skill2vec_temp = skill2vec_temp.withColumn('skill_list', array_except('skill_list', array(lit(None))))
skill2vec_temp = skill2vec_temp.withColumn('size', size(col('skill_list')))
skill2vec_list = skill2vec_temp.select(col('_c0').alias('id'),'skill_list','size')


# check skill2vec_list data
skill2vec_list.show()
```

```
+------+--------------------+----+
|    id|          skill_list|size|
+------+--------------------+----+
|125720|[HR Executive, sc...|  11|
|112708|[Special Teacher,...|   3|
|115226|[consulting, fres...|  17|
| 19805|[diploma, machini...|  10|
| 80208|[Compensation, Be...|  10|
| 64086|[Storage Administ...|   1|
| 48468|[HR Operations, E...|  13|
|122729|[Simulink, statef...|  13|
```

```
| 36721|[development, inf...|  10|
|  9342|[software_develop...|   6|
| 78148|[Tableau, Analyti...|   9|
| 31313|[Investment Banki...|  10|
|123378|            [sap sd]|   1|
|  3574|[factset, portfol...|  31|
| 64830|[Import, Export, ...|   6|
|  4772|[gis, analysis, g...|  32|
| 44923|[Full Stack Devel...|   8|
```

```
# skill2vec_processed
skill2vec_temp = skill2vec_raw.drop('_c0')
skill2vec_processed = skill2vec_list.withColumn('skill', explode('skill_list'))
skill2vec_processed = skill2vec_processed.select('skill')


# check skill2vec_processed data
skill2vec_processed.show()
```

```
+--------------------+
|               skill|
+--------------------+
|        HR Executive|
|           screening|
|           selection|
|           Interview|
|                  HR|
|           Recruiter|
|        IT Recruiter|
|            Sourcing|
|recruitment execu...|
|          onboarding|
|      IT Recruitment|
|     Special Teacher|
|             Teaching|
|           Education|
|          consulting|
|             fresher|
|         IT helpdesk|
|Techincal Trouble...|
```

(c) Addition of lowercase column

```
# create a lowercase column of skills
skill2vec_processed = skill2vec_processed.withColumn('skill_lower', lower('skill'))


# check created columns
skill2vec_processed.show()
```

```
+--------------------+--------------------+
|               skill|         skill_lower|
+--------------------+--------------------+
|        HR Executive|        hr executive|
|           screening|           screening|
|           selection|           selection|
|           Interview|           interview|
|                  HR|                  hr|
|           Recruiter|           recruiter|
|        IT Recruiter|        it recruiter|
|            Sourcing|            sourcing|
|recruitment execu...|recruitment execu...|
|          onboarding|          onboarding|
|      IT Recruitment|      it recruitment|
|     Special Teacher|     special teacher|
|             Teaching|             teaching|
|           Education|           education|
|          consulting|          consulting|
|             fresher|             fresher|
|         IT helpdesk|         it helpdesk|
|Techincal Trouble...|techincal trouble...|
```

(d) Replacement of invalid values

```
# check data before replacing invalid values ascending
skill2vec_processed.sort(col('skill').asc()).display()
```

Table

	skill	skill_lower
1	clinical data analyst	clinical data analyst
2	#Expertise of multi plant handling#	#expertise of multi
3	&	&
4	& TDSAccounting.	& tdsaccountir
5	&quot ; 802.11a&quot ;	&quot ; 802.11
6	&quot ; 802.11g&quot ;	&quot ; 802.11
7	&quot ; 802.11n&quot ;	&quot ; 802.11
8	&quot ; AR Analyst&quot ;	&quot ; ar ana
9	&quot ; AR Analysts	&quot ; ar ana
10	&quot ; ASP.Net MVC	&quot ; asp.ne
11	&quot ; Accounts Executive	&quot ; accou
12	&quot ; Accounts Receivable	&quot ; accou
13	&quot ; Accounts Receivable	&quot ; accou

10,000 rows | Truncated data

```
# check data before replacing invalid values descending
skill2vec_processed.sort(col('skill').desc()).display()
```

Table

	skill	skill_lower
1	¾ñ´¾ïâ□»_	¾ñ´¾ïâ□»_
2	¾ñ´¾ïâ□»_	¾ñ´¾ïâ□»_
3	□óù□ô_□ô¬¾ïù	□óù□ô_□ô¬¾ïù
4	zynga	zynga
5	zuora	zuora
6	zuora	zuora
7	zuora	zuora
8	zuora	zuora
9	zuora	zuora

10,000 rows | Truncated data

```
# create a new dataframe 'skill2vec_replaced' and a new column 'skill_replaced' to save the replaced values
skill2vec_replaced = skill2vec_processed.withColumn('skill_replaced', skill2vec_processed['skill'])


# replace invalid values that have certain patterns
skill2vec_replaced = skill2vec_replaced.withColumn("skill_replaced",\
       when(col("skill_replaced").like('%&amp;quot ; %'), regexp_replace(col("skill_replaced"),"&amp;quot ; ",""))\
       .when(col("skill_replaced").like('%&amp;quot ;%'), regexp_replace(col("skill_replaced"),"&amp;quot ;",""))\
       .when(col("skill_replaced").like('%&amp;%'), regexp_replace(col("skill_replaced"),"&amp;","&"))\
       .when(col("skill_replaced").like('%&amp; %'), regexp_replace(col("skill_replaced"),"&amp; ","&"))\
       .when(col("skill_replaced").like('%quot ; %'), regexp_replace(col("skill_replaced"),"quot ; ",""))\
       .when(col("skill_replaced").like('%quot ;%'), regexp_replace(col("skill_replaced"),"quot ;",""))\
       .otherwise(col("skill_replaced")))

skill2vec_replaced = skill2vec_replaced.withColumn("skill_replaced",\
       when(col("skill_replaced").like('%&amp;quot ;%'), regexp_extract(col("skill_replaced"), '.*(?=[\&])', 0))\
       .when(col("skill_replaced").like('%&%'), regexp_extract(col("skill_replaced"), '.+\s+\&+\s+.*', 0))\
       .when(col("skill_replaced").like('*%'), regexp_extract(col("skill_replaced"), '[^\*\s](.*)', 0))\
       .when(col("skill_replaced").like('- %'), regexp_extract(col("skill_replaced"), '[^-\s](.*)', 0))\
       .when(col("skill_replaced").like('%...%'), regexp_replace(col("skill_replaced"),"...",""))\
       .when(col("skill_replaced").like('. %'), regexp_replace(col("skill_replaced"),". ",""))\
       .otherwise(col("skill_replaced")))
```

```
# replace invalid values indiviaully
skill2vec_replaced = skill2vec_replaced.withColumn("skill_replaced",\
       when(col("skill_replaced").like('#Expertise of multi plant handling#'),
regexp_replace(col("skill_replaced"),"#Expertise of multi plant handling#","Expertise of multi plant handling"))\
       .when(col("skill_replaced").like('&'), regexp_replace(col("skill_replaced"),"&",""))\
       .when(col("skill_replaced").like('& TDSAccounting.'), regexp_replace(col("skill_replaced"),"&
TDSAccounting.","TDSAccounting"))\
       .when(col("skill_replaced").like('- Sales & Marketing'), regexp_replace(col("skill_replaced"),"- Sales &
Marketing","Sales & Marketing"))\
       .when(col("skill_replaced").like('. Managing Training & UAT'), regexp_replace(col("skill_replaced"),".
Managing Training & UAT","Managing Training & UAT"))\
       .when(col("skill_replaced").like('C&amp;R'), regexp_replace(col("skill_replaced"),"C&amp;R","C & R"))\
       .when(col("skill_replaced").like('HR &amp; Admin'), regexp_replace(col("skill_replaced"),"HR &amp; Admin","HR
& Admin"))\
       .when(col("skill_replaced").like('Office Assistant &amp; Receptionist'),
regexp_replace(col("skill_replaced"),"Office Assistant &amp; Receptionist","Office Assistant & Receptionist"))\
       .when(col("skill_replaced").like('collections &amp; recovery'),
regexp_replace(col("skill_replaced"),"collections &amp; recovery","collections & recovery"))\
       .when(col("skill_replaced").like('trade&amp; forex'), regexp_replace(col("skill_replaced"),"trade&amp;
forex","trade & forex"))\
       .when(col("skill_replaced").like('Devising & implementing pre & post marketing activities...'),
regexp_replace(col("skill_replaced"),"Devising & implementing pre & post marketing activities...","Devising &
implementing pre & post marketing activities"))\
       .when(col("skill_replaced").like('General Manager Sales & Marketing - Hospital Business...'),
regexp_replace(col("skill_replaced"),"General Manager Sales & Marketing - Hospital Business...","General Manager
Sales & Marketing - Hospital Business"))\
       .when(col("skill_replaced").like("' ; Customer Support'"), regexp_replace(col("skill_replaced"),"' ; Customer
Support'","Customer Support"))\
       .when(col("skill_replaced").like('.JAVA preferably multiple technologies'),
regexp_replace(col("skill_replaced"),".JAVA preferably multiple technologies","JAVA preferably multiple
technologies"))\
       .otherwise(col("skill_replaced")))

# filter invalid values
skill2vec_replaced = skill2vec_replaced.filter(col("skill_replaced")!='')\
       .filter(col("skill_replaced")!='.')\
       .filter(col("skill_replaced")!='..')\
       .filter(col("skill_replaced")!='¾ñ´¾ïâ»_')\
       .filter(col("skill_replaced")!='óùô_ôô¬¾ïù')


# create a lowercase column of skill_replaced
skill2vec_replaced = skill2vec_replaced.withColumn('skill_replaced_lower', lower('skill_replaced'))

# leave only replaced columns
skill2vec_replaced = skill2vec_replaced.select(col('skill_replaced'), col('skill_replaced_lower'))


# check replaced values ascending
skill2vec_replaced.sort(col('skill').asc()).show()
```

```
+--------------------+--------------------+
|      skill_replaced|skill_replaced_lower|
+--------------------+--------------------+
| clinical data an...| clinical data an...|
|Expertise of mult...|expertise of mult...|
|             802.11a|             802.11a|
|             802.11g|             802.11g|
|             802.11n|             802.11n|
|          AR Analyst|          ar analyst|
|         AR Analysts|         ar analysts|
|         ASP.Net MVC|         asp.net mvc|
|  Accounts Executive|  accounts executive|
| Accounts Receivable| accounts receivable|
| Accounts Receivable| accounts receivable|
| Accounts Receivable| accounts receivable|
| Accounts Receivable| accounts receivable|
| Accounts Receivable| accounts receivable|
|            Accounts|            accounts|
|    Active Directory|    active directory|
|           Analytics|           analytics|
|             Android|             android|
```

```
# check replaced values descending
skill2vec_replaced.sort(col('skill').desc()).show()
```

```
+--------------+--------------------+
|skill_replaced|skill_replaced_lower|
+--------------+--------------------+
|         zynga|               zynga|
|         zuora|               zuora|
|         zuora|               zuora|
|         zuora|               zuora|
|         zuora|               zuora|
|         zuora|               zuora|
|         zuora|               zuora|
|         zuora|               zuora|
|         zuora|               zuora|
|         zuora|               zuora|
|         zuken|               zuken|
|           zte|                 zte|
|           zte|                 zte|
|           zte|                 zte|
|           zte|                 zte|
|           zte|                 zte|
|           zte|                 zte|
|           zte|                 zte|
```

(e) Table registration

```
sqlContext.registerDataFrameAsTable(skill2vec_list, "skill2vec_list")
sqlContext.registerDataFrameAsTable(skill2vec_processed, "skill2vec_processed")
sqlContext.registerDataFrameAsTable(skill2vec_replaced, "skill2vec_replaced")
```

```
%sql
SELECT *
FROM skill2vec_list
```

Table

	id	skill_list	si
1	125720	▸ ["HR Executive", "screening", "selection", "Interview", "HR", "Recruiter", "IT Recruiter", "Sourcing", "recruitment executive", "onboarding", "IT Recruitment"]	11
2	112708	▸ ["Special Teacher", "Teaching", "Education"]	3
3	115226	▸ ["consulting", "fresher", "IT helpdesk", "Techincal Troubleshooting", "international voice", "international BPO", "technical support", "outsourcing", "call center", "BBA fresher", "Bcom fresher", "Tech support", "voice calling", "BPO", "SME", "BCA fresher", "MBA fresher"]	17
4	19805	▸ ["diploma", "machining", "cnc m", "mould", "conventional machines", "die making", "knowledge", "tool", "cipet", "assembly"]	10
5	80208	▸ ["Compensation", "Benefits", "HR Functions", "Alm", "Payroll", "ESS", "Core HR", "QC", "QA", "SQL"]	10
6	64086	▸ ["Storage Administrator"]	1
7	48468	▸ ["HR Operations", "Exit Formalities", "Shortlisting", "Screening", "Interviewing", "Verbal Communication", "End to end recruitment", "IT Recruitment", "Hiring", "Core HR", "Sourcing", "recruit", "recruitment"]	13
8	122729	▸ ["Simulink", "stateflow", "Matlab developer", "targetlink", "matlab programmer", "simulink developer", "matlab software engineer", "matlab designer", "matlab software developer", "stateflow developer", "mathcad developer", "Embedded C", "MATLAB"]	13
9	36721	▸ ["development", "information technology", "api", "business intelligence", "problem solving", "quality assurance", "soa", "siebel", "informatica", "microsoft certified"]	10
10	9342	▸ ["software_development", "product_development_life-cycle", "pdlc", "systems_development_life_cycle", "sdlc", "development_manager"]	6
11	78148	▸ ["Tableau", "Analytics", "Financial regulation", "compliance", "Business Intelligence", "Microstrategy", "Cognos", "Reporting", "Risk management"]	9
12	31313	▸ ["Investment Banking", "Secretarial Activities", "Accounting", "Business Finance", "Company Secretary", "Auditing", "Taxation", "Credit Risk", "Risk Management", "Credit Control"]	10
13	123378	▸ ["sap sd"]	1

10,000 rows | Truncated data

```
%sql
SELECT *
FROM skill2vec_processed
```

Table

	skill	skill_lower
1	HR Executive	hr executive

2	screening	screening
3	selection	selection
4	Interview	interview
5	HR	hr
6	Recruiter	recruiter
7	IT Recruiter	it recruiter

10,000 rows | Truncated data

```
%sql
SELECT *
FROM skill2vec_replaced
```

Table

	skill_replaced	skill_replaced_lower
1	HR Executive	hr executive
2	screening	screening
3	selection	selection
4	Interview	interview
5	HR	hr
6	Recruiter	recruiter
7	IT Recruiter	it recruiter

10,000 rows | Truncated data

technology data

(a) Data checking

```
# check null values
technology_raw.select([count(when(col(c).isNull(), c)).alias(c) for c in technology_raw.columns]).show()
```

```
+--------------+-------+--------------+---------------+--------------+---------+
|O*NET-SOC Code|Example|Commodity Code|Commodity Title|Hot Technology|In Demand|
+--------------+-------+--------------+---------------+--------------+---------+
|             0|      0|             0|              0|             0|        0|
+--------------+-------+--------------+---------------+--------------+---------+
```

```
# check data by Example name order
technology_raw.sort(col('Example').asc()).display()
```

Table

	O*NET-SOC Code	Example
1	11-3111.00	!Trak-it Solutions !Trak-it HR
2	17-3011.00	100 Plus Hatch Pattern Library
3	13-2072.00	1003 Uniform Residential Loan Application
4	13-2011.00	1099 ProsSoftware
5	17-2011.00	1CadCam Unigraphics
6	17-2141.00	1CadCam Unigraphics
7	17-2141.02	1CadCam Unigraphics

10,000 rows | Truncated data

```
# check white space in column names
technology_raw.columns
```

```
Out[170]: ['O*NET-SOC Code',
 'Example',
 'Commodity Code',
 'Commodity Title',
 'Hot Technology',
 'In Demand']
```

(b) Replacement of invalid values

```
# replace an invalid value
technology_processed = technology_raw.withColumn("Example",\
       when(col("Example").like('!Trak-it Solutions !Trak-it HR'),\
        regexp_replace(col("Example"),"!Trak-it Solutions !Trak-it HR","Trak-it Solutions Trak-it HR"))\
       .otherwise(col('Example')))


# check data after replacing the invalid value by ordering example ascending
technology_processed.sort(col('Example').asc()).display()
```

Table

	O*NET-SOC Code	Example
1	17-3011.00	100 Plus Hatch Pattern Library
2	13-2072.00	1003 Uniform Residential Loan Application
3	13-2011.00	1099 ProsSoftware
4	17-2011.00	1CadCam Unigraphics
5	17-2141.00	1CadCam Unigraphics
6	17-2141.02	1CadCam Unigraphics
7	17-3012.00	1CadCam Unigraphics

10,000 rows | Truncated data

```
# check replaced value
technology_processed.where(col('Example')=='Trak-it Solutions Trak-it HR').show()
```

```
+--------------+--------------------+--------------+--------------------+--------------+---------+
|O*NET-SOC Code|             Example|Commodity Code|     Commodity Title|Hot Technology|In Demand|
+--------------+--------------------+--------------+--------------------+--------------+---------+
|    11-3111.00|Trak-it Solutions...|      43231505|Human resources s...|             N|        N|
+--------------+--------------------+--------------+--------------------+--------------+---------+
```

(c) Modification of column names

```
# replace white spaces with underbars
technology_processed = technology_processed.select(col("O*NET-SOC Code").alias("soc_code")\
                                          , col("Example").alias("example")\
                                          , col("Commodity Code").alias("commodity_code")\
                                          , col("Commodity Title").alias("commodity_title")\
                                          , col("Hot Technology").alias("hot_technology")\
                                          , col("In Demand").alias("in_demand"))


# check changed column names
technology_processed.columns
```

```
Out[175]: ['soc_code',
 'example',
 'commodity_code',
 'commodity_title',
 'hot_technology',
 'in_demand']
```

(d) Addition of lowercase column

```
# create a lower case column
technology_processed = technology_processed.withColumn('example_lower', lower('example'))


# check after creating lower case column
technology_processed.select('example', 'example_lower').show()
```

```
+--------------------+--------------------+
|             example|       example_lower|
+--------------------+--------------------+
|Adobe Systems Ado...|adobe systems ado...|
```

```
|     AdSense Tracker|     adsense tracker|
|      Atlassian JIRA|      atlassian jira|
|Blackbaud The Rai...|blackbaud the rai...|
|ComputerEase cons...|computerease cons...|
|Database reportin...|database reportin...|
|             Databox|             databox|
|      Email software|      email software|
|Enterprise resour...|enterprise resour...|
|Exact Software Ma...|exact software ma...|
|Extensible markup...|extensible markup...|
|Fund accounting s...|fund accounting s...|
|Graphic presentat...|graphic presentat...|
|        Halogen e360|        halogen e360|
|    Halogen ePraisal|    halogen epraisal|
|       HCSS HeavyBid|       hcss heavybid|
|       HCSS HeavyJob|       hcss heavyjob|
```

(e) Table registration

```
sqlContext.registerDataFrameAsTable(technology_processed, "technology_processed")
```

Table

	soc_code	example	commodity_code	commod
1	11-1011.00	Adobe Systems Adobe Acrobat	43232202	Documen
2	11-1011.00	AdSense Tracker	43232306	Data base
3	11-1011.00	Atlassian JIRA	43232201	Content v
4	11-1011.00	Blackbaud The Raiser's Edge	43232303	Customer
5	11-1011.00	ComputerEase construction accounting software	43231601	Accountir
6	11-1011.00	Database reporting software	43232305	Data base
7	11-1011.00	Databox	43232306	Data base

10,000 rows | Truncated data