Application Note

GradientOne

Provide Finance Team With Insight Into Engineering Asset Utilization

Use SQLITE To Join A CSV of Test Equipment Utilization Data With Your Company's Fixed Asset Register

Summary

This Application Note describes how to join table from two sources: An engineering database that contains results and utilization data collected from test equipment and a financial database that tracks the value of a company's assets. You will learn how to load two CSV files into SQL database tables. Join the two tables. And produce a new table that combines utilization data and asset value to estimate potential savings from improved utilization. As an illustration of how this technique can be used a s a tool, the SQL solution will be extended using Python

Background

Engineering teams require significant investment in fixed assets such as test equipment to bring products to market. Finance teams track these fixed assets for reporting on, depreciation, condition, maintenance status, location, and more. Using fixed asset tracking software helps simplify many aspects of keeping the fixed asset register up to date.

However the fixed asset register won't tell you if the asset is actually being utilized.

GradientOne supports rich interactions between instruments and cloud databases. These include industry standard protocols such as SCPI, VISA, and VXI11. Data derived from the frequency of instrument's interaction with the database as well as information in the embedded operating system support a range of utilization estimates.

These utilization calculations are available as customizable reports, downloadable CSV files, or programmatic APIs.

Example

In this example we start with data commonly found in a fixed asset register. Obtain this data from your Finance team, it is stored in a database or cloud application and is typically downloadable via a CSV file for use in spreadsheets or scripts for analysis. In this example (fig. 1), columns in fixedassets.csv include:

- name
- purchase date
- purchase cost
- book value
- manufacturer
- serial number
- cost center

Fixed Asset Register Example Data (cost in USD)

Name	Purchase	PurchaseCost	BookValue	Manufacturer	SerialNumber	CostCenter
Tektronix_MSO73304DX	8/28/2017	300000	189,978	Tektronix	B260614	Support
Keysight_DSAV334A	9/22/2017	350000	227,476	Keysight	MY55170132	R&D
Tektronix_DPO72004C	9/28/2017	65000	42,245	Tektronix	B030139	R&D
R&S_CMW500	11/10/2017	150000	99,990	Rohde & Schwarz	R51921521A	Production
Keysight_PNAN5225A	11/14/2017	250000	166,650	Keysight	MY51451131	R&D
Keysight_E5071C	11/30/2017	35000	23,331	Keysight	MY46521121	R&D
M8020_JBERT	1/15/2018	350000	244,979	Keysight	MY55301937	R&D
Keysight_E5071C	1/22/2018	35000	24,498	Keysight	MY46528143	R&D
MP1800_BERT	1/25/2018	300000	209,982	Anritsu	A620151941	R&D
R&S_CMW500	2/12/2018	150000	107,492	Rohde & Schwarz	R51921511A	Production
Keysight_DSOV334A	3/1/2018	300000	219,984	Keysight	MY55301970	R&D
Keysight_PNAN5225A	3/8/2018	250000	183,320	Keysight	MY51451972	R&D
R&S_CMW500	3/12/2018	150000	109,992	Rohde & Schwarz	R51922020G	Production
Keysight_PNAN5225A	6/27/2018	250000	195,823	Keysight	MY51451098	R&D
M8020_JBERT	7/19/2018	350000	279,986	Keysight	MY49510077	R&D
MP1800_BERT	8/3/2018	300000	244,989	Anritsu	A14300110	R&D
Keysight_DSOV334A	9/5/2018	300000	249,990	Keysight	MY57221914	R&D
Keysight_DSOV334A	11/13/2018	300000	259,992	Keysight	MY58121918	Support
R&S_CMW500	11/22/2018	150000	129,996	Rohde & Schwarz	R51922330A	Production

Fig. 1

Next, download the GradientOne utilization data (file name is utilization.csv). It is accessible via APIs as JSON, but also is available via CSV. The data includes (Fig. 2):

- name
- serial number
- lab
- last 90 days utilization
- last 30 days utilization
- last 7 days utilization

Test Equipment Utilization Example Data

Name	SerialNumber	Lab	Last90DaysUtilization	Last30DaysUtilization	Last7DaysUtilization
TektronixMSO73304DX	B260614	Fremont	9%	8%	25%
KeysightDSAV334A	MY55170132	Kansas City	55%	32%	43%
TektronixDPO72004C	B030139	Kansas City	48%	63%	60%
R&SCMW500	R51921521A	Taipei	68%	64%	54%
KeysightPNAN5225A	MY51451131	San Francisco	13%	20%	33%
KeysightE5071C	MY46521121	San Francisco	44%	45%	39%
KeysightE5071C	MY46528143	Kansas City	30%	52%	56%
R&SCMW500	R51921511A	Taipei	73%	68%	83%
KeysightPNAN5225A	MY51451098	Kansas City	0%	0%	0%
M8020JBERT	MY49510077	Kansas City	6%	12%	9%
MP1800BERT	A14300110	San Diego	6%	8%	10%
KeysightDSOV334A	MY57221914	San Diego	20%	30%	49%
KeysightDSOV334A	MY58121918	Fremont	5%	5%	3%
R&SCMW500	R51922330A	Taipei	62%	66%	70%
M8020JBERT	MY55301937	San Diego	0%	0%	0%
MP1800BERT	A620151941	Kansas City	52%	38%	7%
KeysightDSOV334A	MY55301970	San Diego	17%	23%	40%
KeysightPNAN5225A	MY51451972	San Francisco	12%	17%	23%
R&SCMW500	R51922020G	Taipei	66%	62%	73%

Fig. 2

After both data sets are available in a CSV file, use SQLITE to generate a new CSV combining the two sets of data using the serial number as the common parameter. It may be unrealistic in many organizations to have a 100% utilization target, so in this case we will use 70% as the utilization target. This example will calculate the potential value recapture for equipment being utilized less than 70% by: (0.7 – utilization) * BookValue. (The book value assumes a basic 5 year straight line depreciation). For piece of equipment with a 17% utilization and a book value of $219,984, the potential value recapture is: (0.7-0.17)*219,984 = $116,592.

You can download SQLITE here: **https://www.sqlite.org/download.html**

See below for sample code (Fig. 3) that generates a CSV file (Fig. 4) representing the joined data.

```
.mode csv
.open joinedutil.db
.import fixedassets.csv assets
.import utilization.csv util
.headers on
.output potentialvalue.csv
select a.CostCenter,a.Manufacturer,a.SerialNumber,a.BookValue*(0.7-u.Last90DaysUtilization) as Potential from assets
a join util u on a.SerialNumber = u.SerialNumber;
.exit
```

Fig. 3

Resulting CSV (Potential in USD)

CostCenter	Manufacturer	SerialNumber	Potential
Support	Tektronix	B260614	115886.58
R&D	Keysight	MY55170132	34121.4
R&D	Tektronix	B030139	9293.9
Production	Rohde & Schwarz	R51921521A	1999.8
R&D	Keysight	MY51451131	94990.5
R&D	Keysight	MY46521121	6066.06
R&D	Keysight	MY55301937	171485.3
R&D	Keysight	MY46528143	9799.2
R&D	Anritsu	A620151941	37796.76
Production	Rohde & Schwarz	R51921511A	-3224.76
R&D	Keysight	MY55301970	116591.52
R&D	Keysight	MY51451972	106325.6
Production	Rohde & Schwarz	R51922020G	4399.68
R&D	Keysight	MY51451098	137076.1
R&D	Keysight	MY49510077	179191.04
R&D	Anritsu	A14300110	156792.96
R&D	Keysight	MY57221914	124995
Support	Keysight	MY58121918	168994.8
Production	Rohde & Schwarz	R51922330A	10399.68

Fig. 4

You can also turn the previous sqlite commands into a script (Fig 5.):

```
$ cat calcpotential.sh
#!/bin/bash
rm -f potentialvalue.csv
rm -f joinedutil.db
echo '.mode csv
.import fixedassets.csv assets
.import utilization.csv util
.headers on
.output potentialvalue.csv
select
a.CostCenter,a.Manufacturer,a.SerialNumber,a.BookValue*(0.7-
u.Last90DaysUtilization) as Potential
from assets a
join util u on a.SerialNumber = u.SerialNumber
;' | sqlite3 joinedutil.db
```

Fig. 5

The SQL solution is extended even further using Python with the following script:

```python
import sqlite3
from sqlite3 import Error

def createConnection(dbFile):
    """ create a database connection to the SQLite database
        specified by the dbFile
    :param dbFile: database file
    :return: Connection object or None
    """
    try:
        conn = sqlite3.connect(dbFile)
        return conn
    except Error as e:
        print(e)

    return None

queryTemplate = """
SELECT a.CostCenter,a.Manufacturer,a.SerialNumber,
       round(a.BookValue*{unusedFactor}, 2) as potential
 FROM assets a
   JOIN util u
    ON a.SerialNumber = u.SerialNumber;
"""

conn = createConnection('joinedutil.db')
cur = conn.cursor()
q = queryTemplate.format(unusedFactor = "(0.7 - u.last90DaysUtilization)")
cur.execute(q)
rows = cur.fetchall()
for row in rows:
    print(row)
```

Fig. 6

Running this Python script results in (Fig. 7):

```
>python calclutil.py
(u'Support', u'Tektronix', u'B260614', 115886.58)
(u'R&D', u'Keysight', u'MY55170132', 34121.4)
(u'R&D', u'Tektronix', u'B030139', 9293.9)
(u'Production', u'Rohde & Schwarz', u'R51921521A', 1999.8)
(u'R&D', u'Keysight', u'MY51451131', 94990.5)
(u'R&D', u'Keysight', u'MY46521121', 6066.06)
(u'R&D', u'Keysight', u'MY55301937', 171485.3)
(u'R&D', u'Keysight', u'MY46528143', 9799.2)
(u'R&D', u'Anritsu', u'A620151941', 37796.76)
(u'Production', u'Rohde & Schwarz', u'R51921511A', -3224.76)
(u'R&D', u'Keysight', u'MY55301970', 116591.52)
(u'R&D', u'Keysight', u'MY51451972', 106325.6)
(u'Production', u'Rohde & Schwarz', u'R51922020G', 4399.68)
(u'R&D', u'Keysight', u'MY51451098', 137076.1)
(u'R&D', u'Keysight', u'MY49510077', 179191.04)
(u'R&D', u'Anritsu', u'A14300110', 156792.96)
(u'R&D', u'Keysight', u'MY57221914', 124995.0)
(u'Support', u'Keysight', u'MY58121918', 168994.8)
(u'Production', u'Rohde & Schwarz', u'R51922330A', 10399.68)
```

Fig. 7

The resulting CSV or script output provides a column named “Potential” highlighting the dollar impact of unused and under utilized equipment. Reports like this put a dollar value on improved utilization. Finance and engineering teams can use these reports to take specific actions that improve productivity and save money.

GradientOne is revolutionizing test engineering!

From monitoring test equipment utilization to setting up and running tests, to storing data and screenshots, to visualizing waveforms, GradientOne makes the job quick and easy. And once you have data, tasks like performing analysis, generating a report, and sharing those results are just a click away. If you'd like to see how we can help simplify test engineering, you can start using GradientOne for free.