Tony Nahra

Cell: (319) 594-1358 . **LinkedIn**: linkedin.com/in/TonyNahra
Web: tonynahra.github.io . **email**: tony-nahra@uiowa.edu

SKILLS

Summary: 25 years experience developing software, managing systems, reporting, planning, evaluating and budgeting
Technologies: MS-SQL, PowerBI, Fabric, Epic systems, Oracle, Azure, AWS, Web Services, Excel, Access, SharePoint
Programming: Python**,** Pandas, ETL, OLAP, Tableau, C#, JavaScript, PHP, SQL, VBA, APEX, XML, HTML5, Office365 Add-ins
Project Management: Agile, ITIL, PMI, Atlassian Jira, Github, Microsoft Projects

PROFESSIONAL EXPERIENCE

UI Health Care: CCA division **Jul. 2023 – Present**
Coralville, Iowa, 52241, USA
Application Developer

Application development and administration of various systems:

- Epic Unified Communication, Voalte administration, Message Media, Securitas RTLS and MS Teams
- Member of two governance groups: MS-Teams and Message Media

Patient Education Institute **Jul. 2001 – June 2023**
Coralville, Iowa, 52241, USA
Senior Analyst Programmer

Interactive Tutorial Engine/Player with LMS integration:

- Designed and developed a tutorial player with voice narration, questions, feedback, glossary, scoring features and multilingual capabilities to improve learning outcomes and increase user engagement
- Database Driven LMS with completion records, response analysis and usage reporting
- Worked with major healthcare institutions: NLM, Kaiser Permanente, Cleveland Clinic, HCA and others
- Managed projects across continents and in different languages
- Generated training material and plans

Abu Dhabi Investment Authority **Feb. 1994 – Jun. 2001**
Abu Dhabi, U.A.E.
IT Education Consultant

Financial systems data analysis:

- Assisted financial analysts build models, data warehouse, OLAP and drill down
- Automated tasks and consolidated data from Reuters, Bloomberg and other sources
- Generated periodic reports, statistical regression, back-testing, and time-series seasonality analysis

CERTIFICATION

Epic Systems	**EpicCare Inpatient Clinical Documentation**	**Oct 2024**
Epic Systems	**Data Courier Mover Badge**	**Nov 2024**
Microsoft	**Fabric Analytics Engineer Associate**	**Jun 2025**
Oracle University	**APEX cloud developer certified professional**	**May 2025**
Oracle University	**AI vector search certified professional**	**May 2025**
Atlassian	**Agile Project Management Professional**	**Jun 2025**

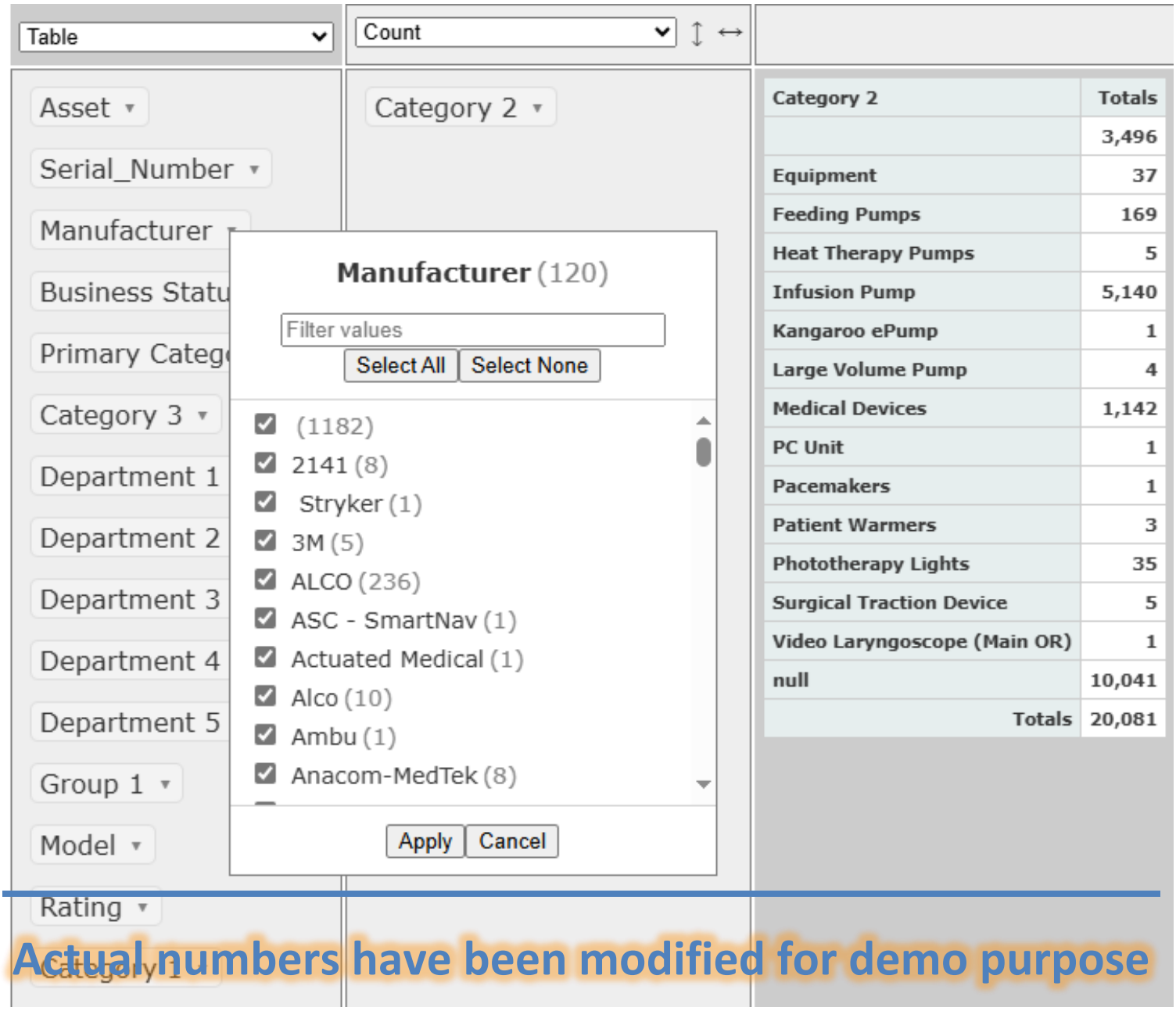
EDUCATION

American University of Beirut	**Beirut, Lebanon**	**Aug 1988**
Bachelor of Engineering	Electrical Engineering	

More details and sample projects at **https://tonynahra.github.io/cv** **Last Update:** 06/20/2025

Screenshots of sample projects that I designed and developed

A RTLS DrillDown report to show the count of RTLS tags per category.



Voalte report to calculate monthly average Time_To_Read messages then use UNION to combine with other months

```
SELECT
  "04-24" AS [ "MM-YY" ],
  int( Max(
      24 * 60 * 60 * ( IowaText_Apr2024.Read - IowaText_Apr2024.Delivered )
        )/ 3600 / 24 ) AS Max_TTR_days
FROM
  IowaText_Apr2024
WHERE
  IowaText_Apr2024.Delivered > 0 AND IowaText_Apr2024.Read > 0 AND IowaText_Apr2024.[Distribution Type] = "One-to-One"
```

SELECT * from Avg_TTR_Jan_2024 Union SELECT * from Avg_TTR_Feb_2024 Union SELECT * from Avg_TTR_Mar_2024 UNION SELECT * from Avg_TTR_Apr_2024

A DrillDown RTLS report using the LastSeen field to analyze who has not been using their RTLS tag per Dept. / Building

```python
def DateRange12Month(X):
    if X > 180 and X < 365 :
        return "X"
    else:
        return ""
def DateRangeInYear(X):
    if X > 365 :
        return "X"
    else:
        return ""

GRP=dataset.groupby(['Department']).agg(Cnt=('TagID',len),Last=('LastDate',np.max))
GRP["LastSeen"] = ( dt.datetime.now() - GRP['Last']  ).dt.days
GRP["Within_1_month"] = GRP['LastSeen'].apply(DateRange1Month)
GRP["Within_3_months"] = GRP['LastSeen'].apply(DateRange3Month)
GRP["Within_year"] = GRP['LastSeen'].apply(DateRange12Month)
GRP["more_than_year"] = GRP['LastSeen'].apply(DateRangeInYear)
GRP = GRP.drop( ['Last'] , axis=1)
```

Python Pandas code used to generate the PIVOT below

Table | Count | Last_seen_within_1_month | Last_seen_within_3_

StaffName | TagID | Location | LastDate | LastSeen

Department

Department	Last_seen_within_1_month	Last_seen_within_3_months	Last_seen_within_year	Last_seen_more_than_year	Totals
	X	X	X	X	
	87	14	14	56	171
1JPW		1	1	4	6
2JCP				1	1
2JPE				3	3
2JPW		1	2	10	13
2RCP				1	1

Count numbers are clickable to drill into TagIDs

Table | Count | Name | Unit | AT

Building

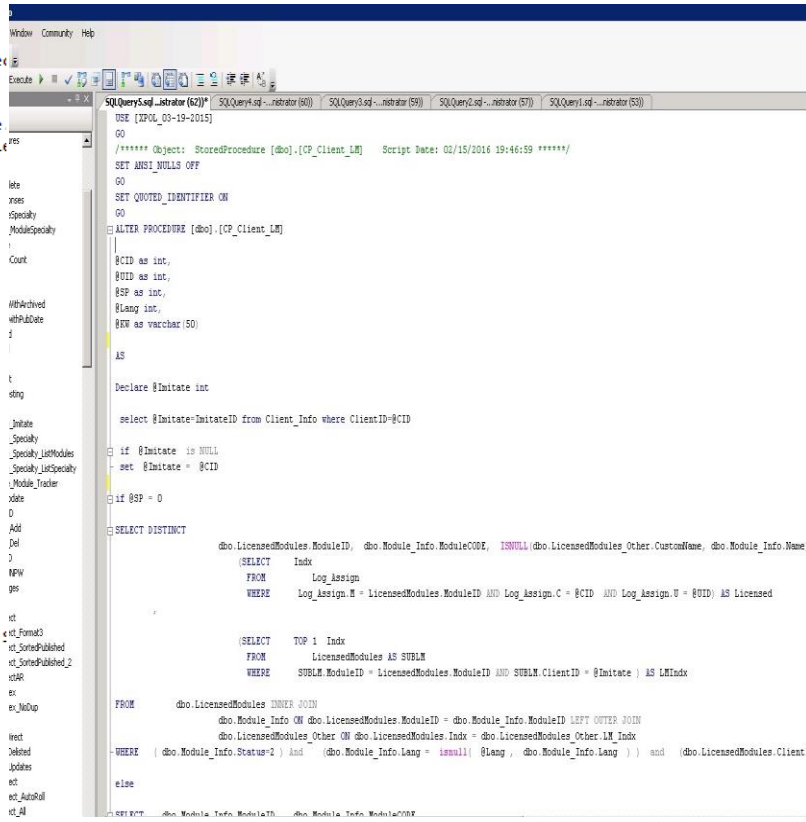
Building	AT	--D	-M-	-MD	1-D	1MD	null	Totals
Ambulatory Clinics		57		68	449	135		709
Anesthesia		1		38	91	70		200
Capital Management				3				3
Cardiothoracic Surgery				2		1		3
Care Coordination Division				2	9	13		24
Carver College of Medicine (CCOM)				6	2	3		11
Center for Disabilities and Development		4		14	34	22		74
Central Sterilizing Services				6	149	48		203
Clinical Quality, Safety & Performance Improvement				8	2	16		26
Compassus				2				2
Compliance						2		2
Dermatology		2		5	3	5		15
Emergency Medicine		9		8	142	71		230
Engineering Services				40	5	8		53
Environmental Services		1		6	211	7		225
Family Medicine		1		39	8	45		93
Food & Nutrition Services		3		45	149	49		246
Guest Services		7			54	2		63
HCIS		21		96	53	71		241
Heart and Vascular Center (HVC)		6		31	113	44		194
Holden Comprehensive Cancer Center (HCCC)		21		61	89	98		269
Hospital Administration		3		18	7	6		34
Hospital Dentistry					1	6		7
Housewide Services		5		405	84	861		1,355
[illegible] Units		1			277	7		285
[illegible]		1			1	1		24
Integrated Call Center (ICC)		18		3	28	3		52
Internal Medicine		7		129	21	83		240
[illegible]				8				8
Iowa River Landing		5		12	374	97		488
Joint Office of Patient Financial Services		65		4	31	6		106
Marketing and Communication				6				6

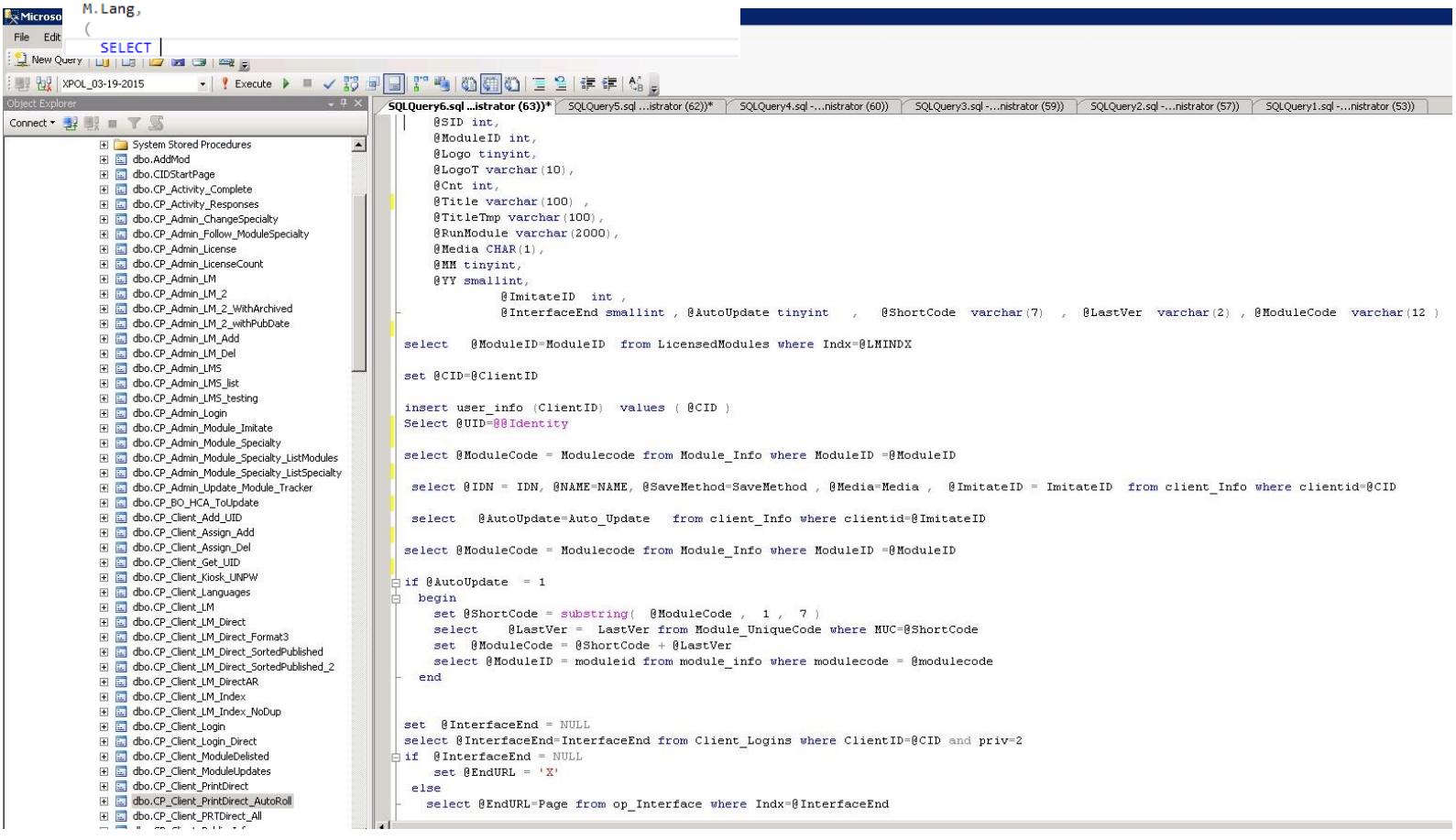
Voalte PIVOT report for who is using which format:
1: VoalteOne
M: Mobile
D: Desktop

Actual numbers have been modified for demo purpose

MS-SQL stored procedures that I developed for an online training system documenting completion

```
  module_info.Name as Title,
  module_info.Type as Type,
  @CID as ClientID,
  dbo.Specialties.SPname as DespSpecialty
FROM
  dbo.Module_Info
  INNER JOIN dbo.Module_Specialty ON dbo.Module_Info.ModuleID = dbo.Module_Spec
  INNER JOIN dbo.Module_Type ON dbo.Module_Info.Type = Module_Type.Indx
  INNER JOIN dbo.Specialties Spe ON dbo.Module_Info.Specialty = Spe.SP
  INNER JOIN dbo.Module_Language ON dbo.Module_Info.Lang = dbo.Module_Language
  INNER JOIN dbo.Specialties ON dbo.Module_Specialty.Specialty = dbo.Specialti
WHERE
  (
    dbo.Module_Info.ModuleCODE = @FiltK
  )
  and (dbo.Module_Info.Status <> 3) else if @SpDest =-2
SELECT
  TOP 100 PERCENT M.ModuleID AS ModuleID,
  M.ModuleCODE,
  M.Lang,
  1 AS Licensed,
  Specialties_1.SPName AS Specialty,
  dbo.Module_Language.[Language],
  M.Name AS Module_Title,
  MT.M_Type AS asType,
  dbo.LicensedModules.ClientID,
  Specialties_1.SP AS DestSpecialty
FROM
  dbo.Module_Info M
  INNER JOIN dbo.Module_Language ON M.Lang = dbo.Module_Language.Indx
  INNER JOIN dbo.Module_Type MT ON M.Type = MT.Indx
  INNER JOIN dbo.Specialties Spe ON M.Specialty = Spe.SP
  INNER JOIN dbo.LicensedModules ON M.ModuleID = dbo.LicensedModules.ModuleID
  INNER JOIN dbo.Specialties Specialties_1 ON dbo.LicensedModules.Specialty =
WHERE
  (
    M.Type = ISNULL(@Type, M.Type)
  )
  AND M.Status <> 3
  AND dbo.LicensedModules.ClientID = @CID
  and M.Specialty = isnull(@FiltS, M.Specialty)
  and M.[Lang] = isnull(@FiltL, M.[Lang])
  and M.ModuleCode + M.Name + M.keywords like '%' + isnull(@FiltK, '%') + '%'
ORDER BY
  M.Sorting,
  M.Name else
SELECT
  TOP 100 PERCENT M.ModuleID AS ModuleID,
  M.ModuleCODE,
  M.Lang,
  (
    SELECT
```



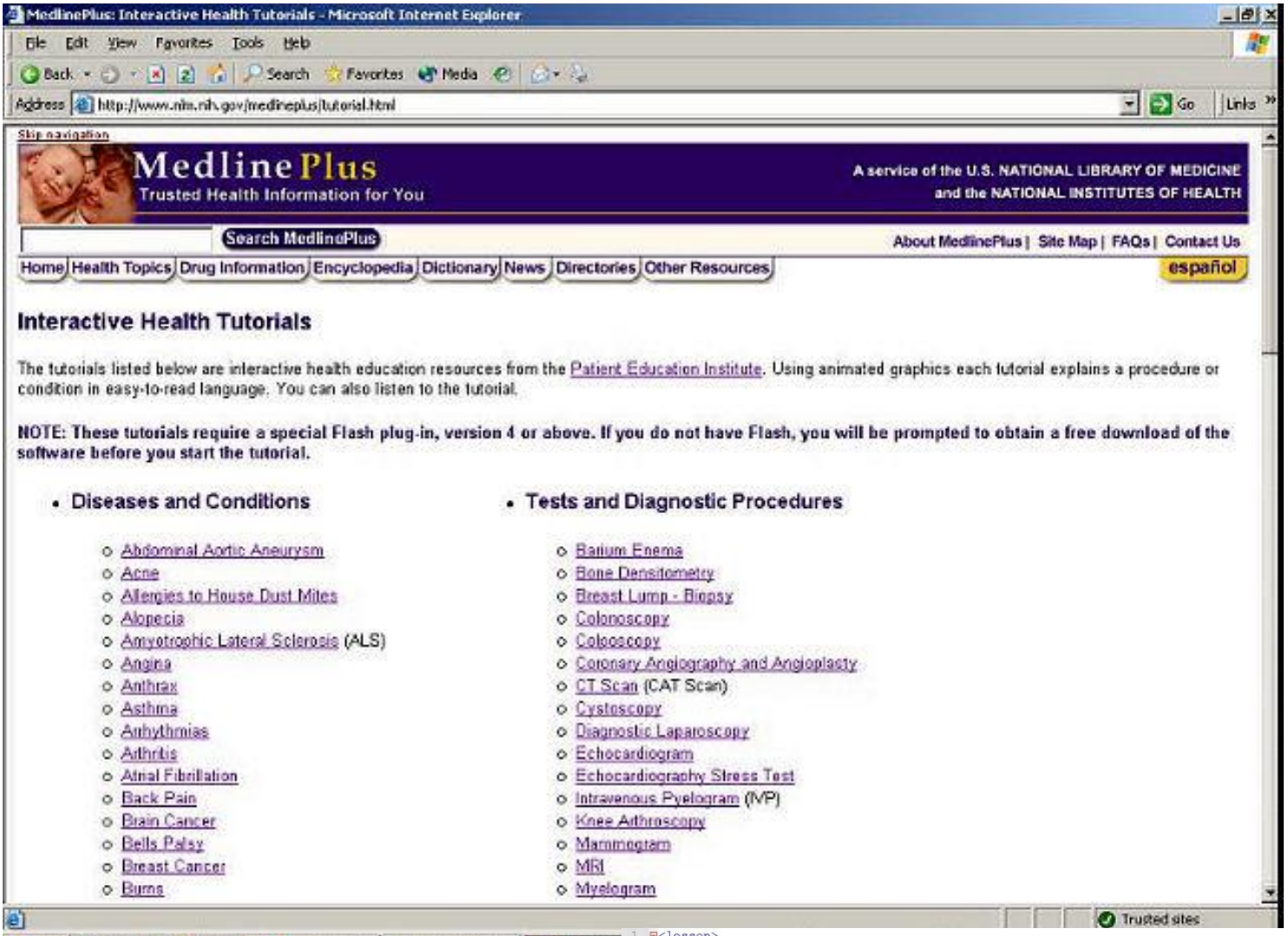


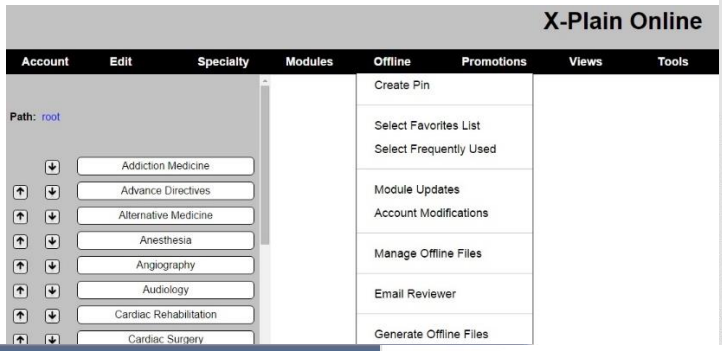
mySql (that does not support PIVOT function) query for cross tabulation

```
SELECT
  TOP 100 PERCENT Specialties.SPName AS Specialty,
  SUM( CASE WHEN module_info.Lang = 1 THEN 1 ELSE 0 END ) AS English,
  SUM( CASE WHEN module_info.Lang = 2 THEN 1 ELSE 0 END ) AS Spanish,
  SUM( CASE WHEN module_info.Lang = 3 THEN 1 ELSE 0 END ) AS Arabic,
  SUM( CASE WHEN module_info.Lang = 4 THEN 1 ELSE 0 END ) AS Vietnamese
FROM
  LicensedModules
  INNER JOIN Module_Info ON LicensedModules.ModuleID = Module_Info.ModuleID
  INNER JOIN Specialties ON LicensedModules.Specialty = Specialties.SP
  INNER JOIN Module_Language ON Module_Info.Lang = Module_Language.Indx
WHERE LicensedModules.ClientID = 91
GROUP BY
  Specialties.SPName,
  Specialties.Type,
  Module_Info.Status
HAVING
  Specialties.Type = 1 AND Module_Info.Status = 2
ORDER BY
  Specialty
```

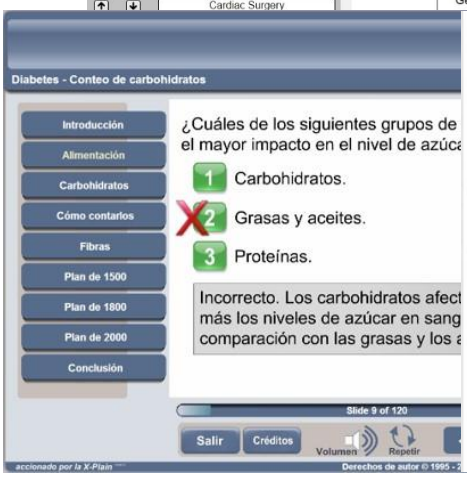
```
ALTER PROCEDURE [CP_Survey_Analysis]
@ClientID smallint,
@ModID varchar(10),
@d1 smalldatetime,
@d2 smalldatetime AS
SELECT
  TOP 100 PERCENT Activity_Responses.QuestionNumber,
  Activity_Responses.UserResponse,
  COUNT(Activity_Responses.Indx) AS cnt,
  Module_Questions.Text AS Question,
  Module_Answers.Text AS Reply,
  Log_Activity.ClientID
FROM
  Activity_Responses
  INNER JOIN Log_Activity ON Activity_Responses.ActivityIndex = Log_Activity.Indx
  INNER JOIN Module_Info ON Log_Activity.ModuleID = Module_Info.ModuleID
  INNER JOIN Module_Questions ON Activity_Responses.QuestionNumber = Module_Questions.Qcode
  AND Module_Info.ModuleCODE = Module_Questions.ModuleCode
  INNER JOIN Module_Answers ON Activity_Responses.QuestionNumber = Module_Answers.Qcode
  AND Activity_Responses.UserResponse = Module_Answers.Answer
  AND Module_Info.ModuleCODE = Module_Answers.ModuleCode
WHERE
  Activity_Responses.DT BETWEEN @d1 AND @d2
GROUP BY
  Activity_Responses.QuestionNumber,
  Activity_Responses.UserResponse,
  Module_Info.ModuleCODE,
  Module_Questions.Text,
  Module_Answers.Text,
  Log_Activity.ClientID
HAVING
  Log_Activity.ClientID = @ClientID
  AND Module_Info.ModuleCODE = @ModID
ORDER BY
  Activity_Responses.QuestionNumber,
  Activity_Responses.UserResponse
```

Database export to XML for a MedlinePlus static website project






```
<lesson>
 <info>
   <Title>Diabetes &#45; Eye Complications</Title>
   <Author>Diabetes &#45; Eye Complications</Author>
   <Published>08/30/2004</Published>
   <PassScore>60</PassScore>
   <Language>English</Language>
   <ModuleID>db019105</ModuleID>
   <folder>diabetes/db010105</folder>
   <AutoAdvance>-</AutoAdvance>
   <RevDate>11/29/2012</RevDate>
 </info>
<s f="1_0" p="s"  t="6" s="Introduction" />
<s f="1_1" p="o"  t="9" />
<s f="1_2" p="o"  t="9" />
<s f="1_3" p="o"  t="9" />
<s f="2_1" p="s"  t="2" s="How the Eye Works" />
<s f="2_2" p="o"  t="6" />
<s f="2_3" p="o"  t="5" />
<s f="2_4" p="o"  t="7" />
<s f="2_5" p="o"  t="5" />
```



MessageMedia OPT_OUT report generator

Instructions

Generate a Message Media csv export
Drop the csv file in csv subfolder
Run the Python script below
The generated CSV file will be stored in 'Message Media/Reports' folder

Subject : Messagemedia OPT_OUT for 2025-01-12_to_2025-01-20

Hello

Here is the messagemedia OPT_OUT report.

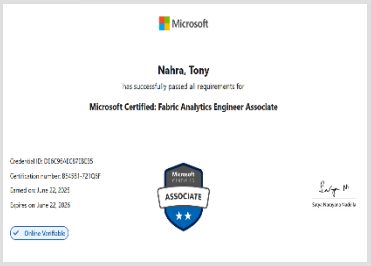
In [1]:
```
import pandas as pd
import os
import glob
import datetime as dt
import re

SRC_FILE = sorted( filter( os.path.isfile, glob.glob('csv/*.csv')))[0]
print(SRC_FILE)
rep = pd.read_csv( SRC_FILE , chunksize=10000,
                      low_memory=False ,
                      index_col=False ,
                      header=0 ,
                      usecols=[
                           'source_address',
                           'destination_address',
                           'action',
                           'timestamp_localtime',
                           'content',
                           'timestamp',
                           'account_name'
                      ] ,
                      dtype="string"
                  )

df = pd.concat(
    (x.query("action in ['OPT_OUT','OPT_IN'] ") for x in rep),
    ignore_index=True
    )
FN  = df['timestamp'].str[0:10].str.strip().min() + '_'
FN += df['timestamp'].str[0:10].str.strip().max()

RepPath = "file:///                                                  /Reports/"
RepPath += FN + ".csv"
print( RepPath )

df['source_address'] = df['source_address'].str[1:]
df = df.query(" `account_name` == '                    " )

dfDupe = df[df.duplicated(subset=['source_address'], keep=False)]
dfDupe = dfDupe.sort_values(
    'timestamp',
```

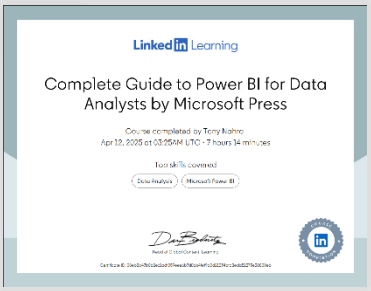
```
    ascending=False).drop_duplicates(['source_address'] ,
    keep='first'
    )

dfDupe = dfDupe.reset_index(drop=True)
dfDupe = dfDupe.drop( dfDupe.query(" `action`=='OPT_OUT' ").index)

print('')
print('Duplicates')
print('OPT_IN ----------------------------------------------------')

if dfDupe.empty:
    print('No duplicates')
else:
    #print(dfDupe.to_markdown())
    dfDupe.to_csv( 'output/' + FN + ".csv" , mode='a')

df = df.sort_values(
    ['timestamp','source_address'],
    ascending=[False, True]
    ).drop_duplicates(['source_address'])

a_index = df.set_index('source_address').index
b_index = dfDupe.set_index('source_address').index

mask = ~a_index.isin(b_index)
df = df.loc[mask]
df = df.sort_values('timestamp', ascending=True )
df = df.reset_index(drop=True)
df = df.drop(['destination_address'], axis=1)
df = df.drop(['timestamp'], axis=1)

print('')
print('Result ----------------------------------------------------------------')
df.to_csv( 'output/' + FN + ".csv" , index=False )
df.to_csv(
    "[redacted] FN + ".csv" ,
    index=False
    )
#print(df.to_markdown())
```

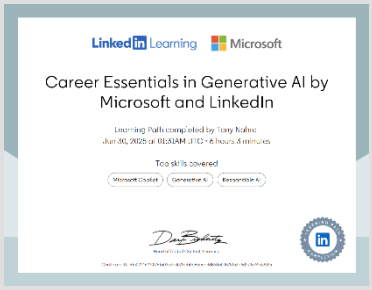
```
csv\2025-02-10_02-08-16_detail_all.csv
file:///S[redacted]eports/2025-02-02_2025-02-
08.csv

Duplicates
OPT_IN ----------------------------------------------------

Result ----------------------------------------------------------------
```

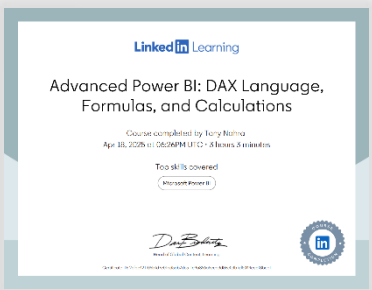
In [ ]:

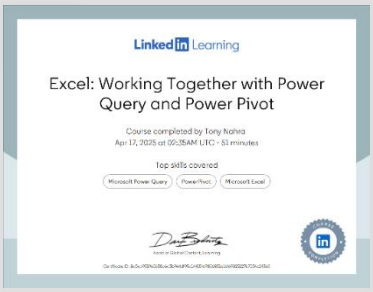
Certification

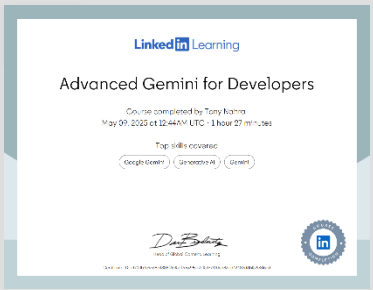


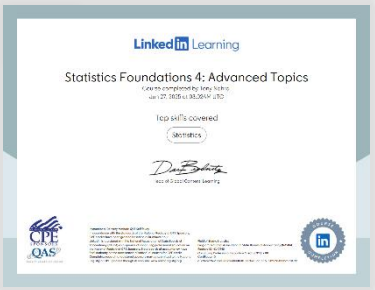


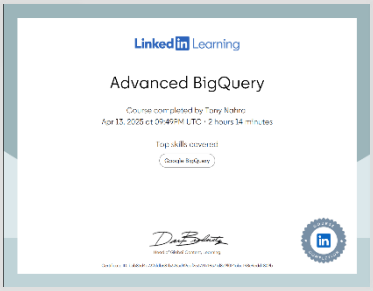


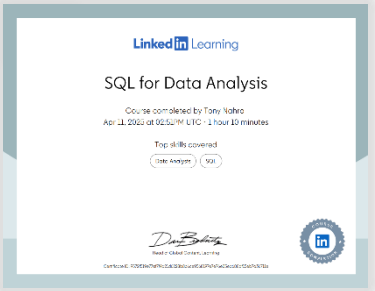












More certificates at https://tonynahra.github.io/cv/cert