Configuration Manual

MSc Research Project
Cloud Computing

Richard McDonald
Student ID: 17123437

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Richard McDonald
Student ID:	17123437
Programme:	Cloud Computing
Year:	2021
Module:	MSc Research Project
Supervisor:	Mr. Vikas Sahni
Submission Due Date:	20/12/2021
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th August 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Richard McDonald
17123437

1 Hardware configuration for Hypervisor

Our lab environment has been prepared by Cherry Servers [1]

1.1 Hardware configuration

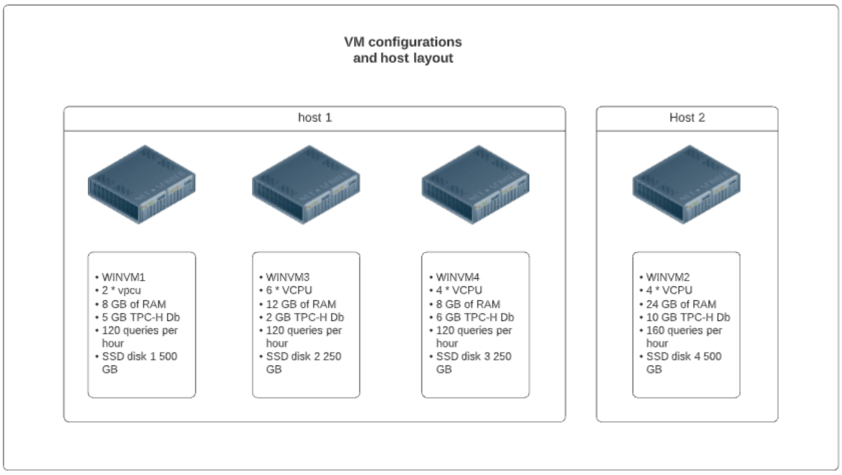
The lab environment contains two physical hosts running the VMware Esxi Version 6.50 (build 4887370). Host 1 running 6 * Intel Xeon CPU @3.6 GHZ and 64gb of Memory. Host 2 running 4 * Intel Xeon CPU at 3.5 GHZ and 32gb of memory. The initial plan was to run all VMs with larger workloads on one host. This changed once we ran multiple VMs running queries across different storage tiers on a single host. After several efforts I had to introduce a separate host to complete log collection and experimentation.

Each VMrunning in the lab is situated on its own dedicated SSD drive.

1.2 VM and MySql environments

In Figure.1 the overall configuration of each VM is shown. Each Machine is configured with random hardware configurations, database sizes and combinations of hourly queries. Three of the VMs are running on the larger host with the largest VM running on its own specialised server. The hardware configuration for each of the host servers is contained in the configuration manual. This information is also available in table format in Table1

Figure 1: Vm configuration and host layout



[1] Cherry Servers:www.cherryservers.com

name	OS	storage tier	Memory	CPU	Db size
winvm1	win 2012	SSD1	8GB	2 * 3GHz	5GB
winvm2	win 2012	SSD4	24GB	4 * 3GHz	10GB
winvm3	win 2012	SSD2	8GB	6 * 3GHZ	2GB
winvm4	WIN 2012	SSD3	8GB	4 * 3GHZ	5GB

Table 1: VMs running on Hypervisor

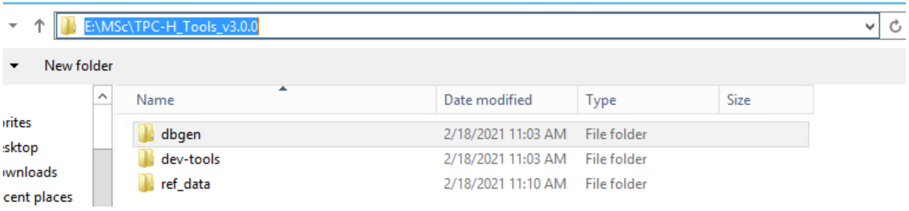
2 TPC-H configuration

2.1 Visual Studio install

Visual Studio is used to compile the DBGen and Qgen utilities for the TPCD data and queries. To begin installation install accept all default installation folders. The version we used for this project is Visual Studio 2015 Community Edition

Once the version of visual studio has been installed. Open the Visual studio Tool and prepare to import the project files from the TPC-H project. The files can be retrieved from the TPC website. [2] the default location for binaries in all of my servers is located in the next image Fig 2.

Figure 2: Default TPC-H folder

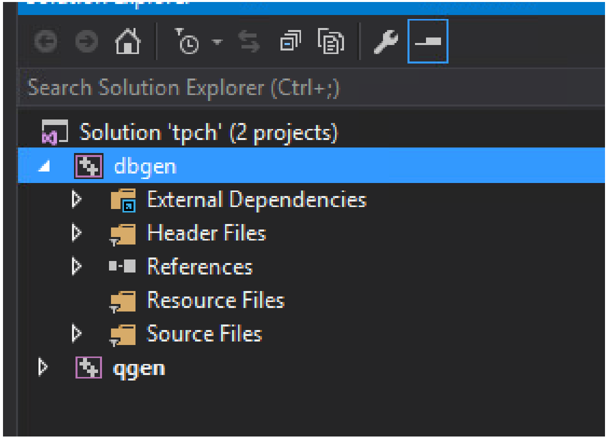


E:\MSc\TPC-H_Tools_v3.0.0

Name	Date modified	Type	Size
dbgen	2/18/2021 11:03 AM	File folder	
dev-tools	2/18/2021 11:03 AM	File folder	
ref_data	2/18/2021 11:10 AM	File folder	

Visual studio will ask to update projects. Approve the request and you will be asked to install open sdk 8.1 and C++ binaries for Visual Studio. Open and compile the TPCH.sln file located in the Dbgen folder. The following projects are shown in the solution window. See next image Fig 3.

Figure 3: Solution Visual studio

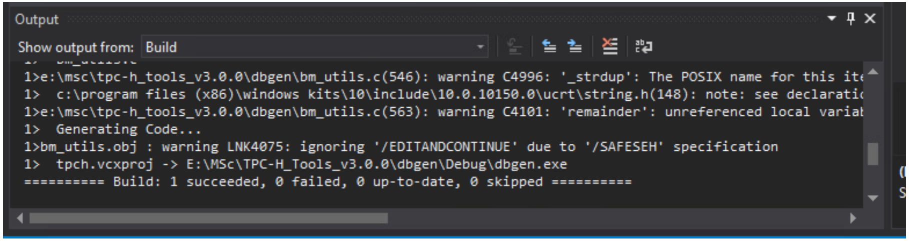


Solution 'tpch' (2 projects)
- dbgen
  - External Dependencies
  - Header Files
  - References
  - Resource Files
  - Source Files
- qgen

[2] TPC-H Benchmark: `http://www.tpc.org/tpch/`

Right click on Dbgen project and click build. Project will compile in the output window in Fig 4.

Figure 4: Solution Visual studio



```
1>e:\msc\tpc-h_tools_v3.0.0\dbgen\bm_utils.c(546): warning C4996: '_strdup': The POSIX name for this ite
1>  c:\program files (x86)\windows kits\10\include\10.0.10150.0\ucrt\string.h(148): note: see declaratic
1>e:\msc\tpc-h_tools_v3.0.0\dbgen\bm_utils.c(563): warning C4101: 'remainder': unreferenced local variat
1>  Generating Code...
1>bm_utils.obj : warning LNK4075: ignoring '/EDITANDCONTINUE' due to '/SAFESEH' specification
1>  tpch.vcxproj -> E:\MSc\TPC-H_Tools_v3.0.0\dbgen\Debug\dbgen.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

This process creates a debug folder under DBGen directory. In that folder there is a file called dbgen.exe. Copy it and move one level up to the DBGen folder. From here. Open a command prompt ensuring the dbgen folder is the root of the command prompt.

```
dbgen -vf -s 1
```

The number entered at the end of the command determines the size of the database we want to generate in Gigabytes. The command above will generate a DB of 1 Gb. I

2.2 data generation

First task within mysql is to create the database.

```

create database tpcd;
```

In the DB Gen folder there is file called dss.ddl. This has all of the sql scripts to create tables for mysql. Edit file and run against Mysql editor of your choice (I used Navicat for mysql version 15 but Heidisql is a free utility that can also be used)

```
CREATE TABLE NATION ( N_NATIONKEY INTEGER NOT NULL,
                        N_NAME      CHAR(25) NOT NULL,
                        N_REGIONKEY INTEGER NOT NULL,
                        N_COMMENT   VARCHAR(152));

CREATE TABLE REGION ( R_REGIONKEY INTEGER NOT NULL,
                        R_NAME      CHAR(25) NOT NULL,
                        R_COMMENT   VARCHAR(152));

CREATE TABLE PART ( P_PARTKEY   INTEGER NOT NULL,
                      P_NAME       VARCHAR(55) NOT NULL,
                      P_MFGR       CHAR(25) NOT NULL,
                      P_BRAND      CHAR(10) NOT NULL,
                      P_TYPE       VARCHAR(25) NOT NULL,
                      P_SIZE       INTEGER NOT NULL,
                      P_CONTAINER  CHAR(10) NOT NULL,
                      P_RETAILPRICE DECIMAL(15,2) NOT NULL,
                      P_COMMENT    VARCHAR(23) NOT NULL );

CREATE TABLE SUPPLIER ( S_SUPPKEY INTEGER NOT NULL,
                          S_NAME      CHAR(25) NOT NULL,
                          S_ADDRESS   VARCHAR(40) NOT NULL,
                          S_NATIONKEY INTEGER NOT NULL,
                          S_PHONE     CHAR(15) NOT NULL,
                          S_ACCTBAL   DECIMAL(15,2) NOT NULL,
                          S_COMMENT   VARCHAR(101) NOT NULL);

CREATE TABLE PARTSUPP ( PS_PARTKEY INTEGER NOT NULL,
                          PS_SUPPKEY   INTEGER NOT NULL,
                          PS_AVAILQTY  INTEGER NOT NULL,
                          PS_SUPPLYCOST DECIMAL(15,2) NOT NULL,
                          PS_COMMENT   VARCHAR(199) NOT NULL );

CREATE TABLE CUSTOMER ( C_CUSTKEY INTEGER NOT NULL,
                          C_NAME      VARCHAR(25) NOT NULL,
                          C_ADDRESS   VARCHAR(40) NOT NULL,
                          C_NATIONKEY INTEGER NOT NULL,
                          C_PHONE     CHAR(15) NOT NULL,
                          C_ACCTBAL   DECIMAL(15,2)  NOT NULL,
                          C_MKTSEGMENT CHAR(10) NOT NULL,
                          C_COMMENT   VARCHAR(117) NOT NULL);

CREATE TABLE ORDERS ( O_ORDERKEY    INTEGER NOT NULL,
                        O_CUSTKEY      INTEGER NOT NULL,
                        O_ORDERSTATUS  CHAR(1) NOT NULL,
                        O_TOTALPRICE   DECIMAL(15,2) NOT NULL,
                        O_ORDERDATE    DATE NOT NULL,
                        O_ORDERPRIORITY CHAR(15) NOT NULL,
                        O_CLERK        CHAR(15) NOT NULL,
                        O_SHIPPRIORITY INTEGER NOT NULL,
                        O_COMMENT      VARCHAR(79) NOT NULL);

CREATE TABLE LINEITEM ( L_ORDERKEY INTEGER NOT NULL,
                          L_PARTKEY   INTEGER NOT NULL,
                          L_SUPPKEY   INTEGER NOT NULL,
                          L_LINENUMBER INTEGER NOT NULL,
                          L_QUANTITY  DECIMAL(15,2) NOT NULL,
                          L_EXTENDEDPRICE DECIMAL(15,2) NOT NULL,
                          L_DISCOUNT  DECIMAL(15,2) NOT NULL,
                          L_TAX       DECIMAL(15,2) NOT NULL,
                          L_RETURNFLAG CHAR(1) NOT NULL,
                          L_LINESTATUS CHAR(1) NOT NULL,
                          L_SHIPDATE  DATE NOT NULL,
                          L_COMMITDATE DATE NOT NULL,
                          L_RECEIPTDATE DATE NOT NULL,
                          L_SHIPINSTRUCT CHAR(25) NOT NULL,
                          L_SHIPMODE   CHAR(10) NOT NULL,
                          L_COMMENT    VARCHAR(44) NOT NULL);
```

Once run, these statements create tables with the exact columns the next step is to populate the Database.

The next script available allows to enable and disable primary and foreign keys in the mysql database.

Note@ there were some issues with loading this script as is in mysql. so the supplied file dss.ri needs some work. In this example provided in the file the foreign key steps does not work. See below for contents of Dss.ri.

```
ALTER TABLE TPCD.REGION DROP PRIMARY KEY;
ALTER TABLE TPCD.NATION DROP PRIMARY KEY;
ALTER TABLE TPCD.PART DROP PRIMARY KEY;
ALTER TABLE TPCD.SUPPLIER DROP PRIMARY KEY;
ALTER TABLE TPCD.PARTSUPP DROP PRIMARY KEY;
ALTER TABLE TPCD.ORDERS DROP PRIMARY KEY;
ALTER TABLE TPCD.LINEITEM DROP PRIMARY KEY;
ALTER TABLE TPCD.CUSTOMER DROP PRIMARY KEY;



-- For table REGION
ALTER TABLE TPCD.REGION
ADD PRIMARY KEY (R_REGIONKEY);

-- For table NATION
ALTER TABLE TPCD.NATION
ADD PRIMARY KEY (N_NATIONKEY);

-- ALTER TABLE TPCD.NATION
-- ADD FOREIGN KEY NATION_FK1 (N_REGIONKEY) references TPCD.REGION;

COMMIT WORK;

-- For table PART
ALTER TABLE TPCD.PART
ADD PRIMARY KEY (P_PARTKEY);

COMMIT WORK;

-- For table SUPPLIER
ALTER TABLE TPCD.SUPPLIER
ADD PRIMARY KEY (S_SUPPKEY);

--ALTER TABLE TPCD.SUPPLIER
--ADD FOREIGN KEY SUPPLIER_FK1 (S_NATIONKEY) references TPCD.NATION;

COMMIT WORK;

-- For table PARTSUPP
ALTER TABLE TPCD.PARTSUPP
ADD PRIMARY KEY (PS_PARTKEY,PS_SUPPKEY);

COMMIT WORK;

-- For table CUSTOMER
```

```
ALTER TABLE TPCD.CUSTOMER
ADD PRIMARY KEY (C_CUSTKEY);

-- ALTER TABLE TPCD.CUSTOMER
-- ADD FOREIGN KEY CUSTOMER_FK1 (C_NATIONKEY) references TPCD.NATION;

COMMIT WORK;

-- For table LINEITEM
ALTER TABLE TPCD.LINEITEM
ADD PRIMARY KEY (L_ORDERKEY,L_LINENUMBER);

COMMIT WORK;

-- For table ORDERS
ALTER TABLE TPCD.ORDERS
ADD PRIMARY KEY (O_ORDERKEY);

COMMIT WORK;

-- For table PARTSUPP
-- ALTER TABLE TPCD.PARTSUPP
-- ADD FOREIGN KEY PARTSUPP_FK1 (PS_SUPPKEY) references TPCD.SUPPLIER;

COMMIT WORK;

-- ALTER TABLE TPCD.PARTSUPP
-- ADD FOREIGN KEY PARTSUPP_FK2 (PS_PARTKEY) references TPCD.PART;

COMMIT WORK;

-- For table ORDERS
-- ALTER TABLE TPCD.ORDERS
-- ADD FOREIGN KEY ORDERS_FK1 (O_CUSTKEY) references TPCD.CUSTOMER;

COMMIT WORK;

-- For table LINEITEM
-- ALTER TABLE TPCD.LINEITEM
-- ADD FOREIGN KEY LINEITEM_FK1 (L_ORDERKEY) references TPCD.ORDERS;

COMMIT WORK;

-- ALTER TABLE TPCD.LINEITEM
-- ADD FOREIGN KEY LINEITEM_FK2 (L_PARTKEY,L_SUPPKEY) references
       TPCD.PARTSUPP;

COMMIT WORK;
```

After some research on the internet I have found an example of the key generation steps. This works fully creating primary and foreign keys and was implemented on all 4 instances.[3]

```
- Modified by Petr Haloun 18.11.2011
-- Sccsid:   @(#)dss.ri 2.1.8.1
-- TPCD Benchmark Version 8.0

-- ALTER TABLE REGION DROP PRIMARY KEY;
-- ALTER TABLE NATION DROP PRIMARY KEY;
-- ALTER TABLE PART DROP PRIMARY KEY;
-- ALTER TABLE SUPPLIER DROP PRIMARY KEY;
-- ALTER TABLE PARTSUPP DROP PRIMARY KEY;
-- ALTER TABLE ORDERS DROP PRIMARY KEY;
-- ALTER TABLE LINEITEM DROP PRIMARY KEY;
-- ALTER TABLE CUSTOMER DROP PRIMARY KEY;


-- For table REGION
ALTER TABLE REGION
ADD PRIMARY KEY (R_REGIONKEY);

-- For table NATION
ALTER TABLE NATION
ADD PRIMARY KEY (N_NATIONKEY);

ALTER TABLE NATION
ADD FOREIGN KEY NATION_FK1 (N_REGIONKEY) references REGION(R_REGIONKEY);



-- For table PART
ALTER TABLE PART
ADD PRIMARY KEY (P_PARTKEY);



-- For table SUPPLIER
ALTER TABLE SUPPLIER
ADD PRIMARY KEY (S_SUPPKEY);

ALTER TABLE SUPPLIER
ADD FOREIGN KEY SUPPLIER_FK1 (S_NATIONKEY) references NATION(N_NATIONKEY);



-- For table PARTSUPP
ALTER TABLE PARTSUPP
ADD PRIMARY KEY (PS_PARTKEY,PS_SUPPKEY);
```

[3] TPC-H key https://sites.google.com/site/halitsch88/Implementation-TPC-H-schema-into-MySQL-DBMS

```
-- For table CUSTOMER
ALTER TABLE CUSTOMER
ADD PRIMARY KEY (C_CUSTKEY);

ALTER TABLE CUSTOMER
ADD FOREIGN KEY CUSTOMER_FK1 (C_NATIONKEY) references NATION(N_NATIONKEY);



-- For table LINEITEM
ALTER TABLE LINEITEM
ADD PRIMARY KEY (L_ORDERKEY,L_LINENUMBER);



-- For table ORDERS
ALTER TABLE ORDERS
ADD PRIMARY KEY (O_ORDERKEY);



-- For table PARTSUPP
ALTER TABLE PARTSUPP
ADD FOREIGN KEY PARTSUPP_FK1 (PS_SUPPKEY) references SUPPLIER(S_SUPPKEY);



ALTER TABLE PARTSUPP
ADD FOREIGN KEY PARTSUPP_FK2 (PS_PARTKEY) references PART(P_PARTKEY);



-- For table ORDERS
ALTER TABLE ORDERS
ADD FOREIGN KEY ORDERS_FK1 (O_CUSTKEY) references CUSTOMER(C_CUSTKEY);



-- For table LINEITEM
ALTER TABLE LINEITEM
ADD FOREIGN KEY LINEITEM_FK1 (L_ORDERKEY) references ORDERS(O_ORDERKEY);



ALTER TABLE LINEITEM
ADD FOREIGN KEY LINEITEM_FK2 (L_PARTKEY,L_SUPPKEY) references
       PARTSUPP(PS_PARTKEY, PS_SUPPKEY);
```

2.3 Data Population

the following MySql commands will populate the tables using data generated from DBgen process. The first step we need to complete is to enable local data to be loaded into a database table.

this is a simple command.

```
SET GLOBAL local_infile=1;
```

we are now able to load the data with the following commands.

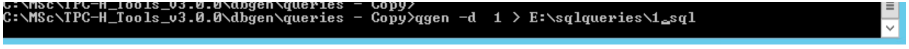
```
LOAD DATA LOCAL INFILE 'e:\\Msc\\TPC-H_Tools_v3.0.0\\dbgen\\customer.tbl'
    INTO TABLE CUSTOMER FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';
LOAD DATA LOCAL INFILE 'e:\\Msc\\TPC-H_Tools_v3.0.0\\dbgen\\orders.tbl' INTO
    TABLE ORDERS FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';
LOAD DATA LOCAL INFILE 'e:\\Msc\\TPC-H_Tools_v3.0.0\\dbgen\\lineitem.tbl'
    INTO TABLE LINEITEM FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';
LOAD DATA LOCAL INFILE 'e:\\Msc\\TPC-H_Tools_v3.0.0\\dbgen\\nation.tbl' INTO
    TABLE NATION FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';
LOAD DATA LOCAL INFILE 'e:\\Msc\\TPC-H_Tools_v3.0.0\\dbgen\\partsupp.tbl'
    INTO TABLE PARTSUPP FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';
LOAD DATA LOCAL INFILE 'e:\\Msc\\TPC-H_Tools_v3.0.0\\dbgen\\part.tbl' INTO
    TABLE PART FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';
LOAD DATA LOCAL INFILE 'e:\\Msc\\TPC-H_Tools_v3.0.0\\dbgen\\region.tbl' INTO
    TABLE REGION FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';
LOAD DATA LOCAL INFILE 'e:\\Msc\\TPC-H_Tools_v3.0.0\\dbgen\\supplier.tbl'
    INTO TABLE SUPPLIER FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';
```

2.4 Qgen Query generation

The Qgen utility takes the template queries and populates them with row and date counts to run the queries needed to generate the workload. You need to copy the QGEN.exe file and the DISTS.DSS file from the dbgen/debug folder into the queries folder.

From there run the command Qgen.exe -d filename number ¿ export folder and file path see Fig5. In my screenshot as an example. Change number of file in source and destination path to clean and populate variables on the DB template.

Figure 5: QGEN script example for first TPCH query



```
C:\MSc\TPC-H_Tools_v3.0.0\dbgen\queries - Copy>
C:\MSc\TPC-H_Tools_v3.0.0\dbgen\queries - Copy>qgen -d  1 > E:\sqlqueries\1.sql
```

3 Sql queries runtime and schedule

3.1 Install MySql

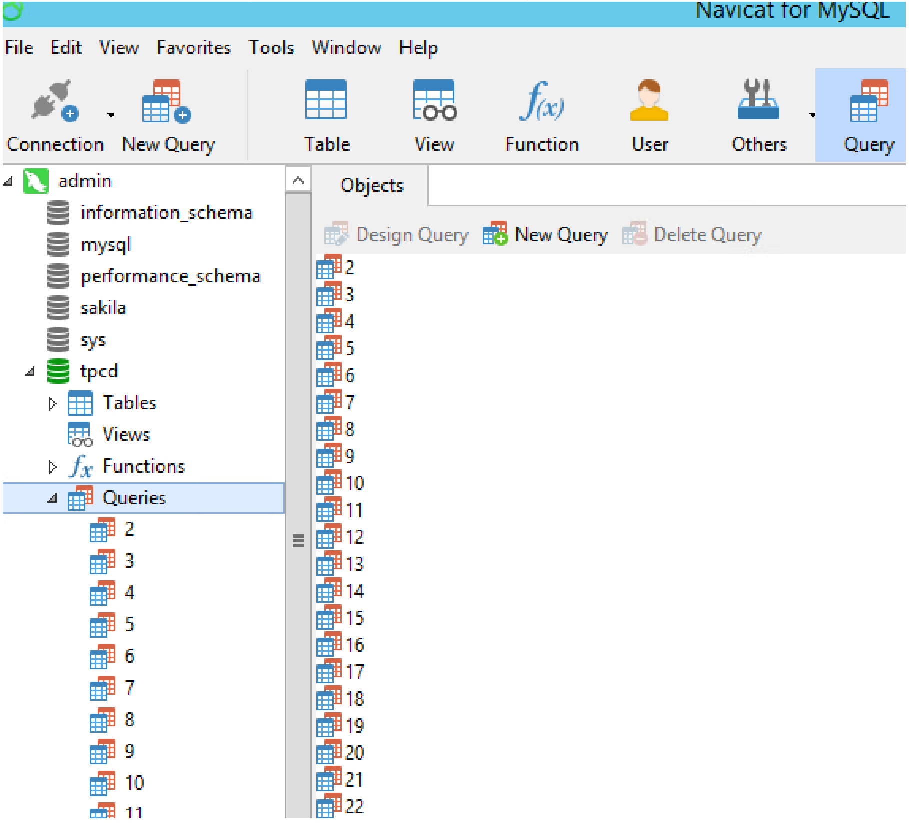
The version used to install MySql on the server is community version 8.0.25.0.

The option we use to install is Developer default.

3.2 Query automation

There are several ways to generate automation of queries in MySql. This can be achieved with a small piece of Python code, By PowerShell scripting or by using a third party utility. I have attached an example the query automation that's available with Navicat. See Fig 6.

Figure 6: Navicat query automation



3.3 Query execution logging

The command in mysql to enable query logging. NOTE: Log file named individually per VM

```
SET GLOBAL log_output= 'FILE';
SET GLOBAL general_log_file = "E:\\logs\\winvm1logs.csv";
SET GLOBAL general_log = "ON";
```

4 Powershell Configuration

The decision was taken to extract all logs from the hypervisor and VM's through a desktop VM running on the host. The workstation that was built was built with 1 CPU and 4Gb of RAM

To enable PowerCli on windows powershell run the following command begin

```
Install-module -Name Vmware.PowerCli -scope CurrentUser
```

this allows you to run all PowerCli scripts from the Windows powershell interface.

4.1 PowerCli scripts

The main PowerCli script that extracts information from each installed vm on the host server. This script uses the format of the script presented in this website. It has been modified to add more variables to extract for any extra counters that maybe required. The date format variable has been created to timestamp files and counters also. The original example of this file is linked here by way of comparison.[4]

NOTE: The file VMscriptlive.ps1 is the version of the file used to extract metrics from this environment. This is the version of the file used for this project and a modified version of the example listed in footnote

code is also listed here.

```


connect-viserver ***.***.***.** -User root -Password ************



$allvms = @()



$vms = Get-Vm




foreach($vm in $vms){
  $vmstat = "" | Select date, VmName, MemMax, MemAvg, MemMin, Activemem,
     Conmem, Grantmem, CPUMax, CPUAvg, CPUMin, CpuUse, Diskavg, Diskmax,
     DiskMin
  $vmstat.VmName = $vm.name

  $statcpu = Get-Stat -Entity ($vm)-start (get-date).AddDays(-1) -Finish
     (Get-Date) -stat cpu.usagemhz.average
  $statcpuuse = Get-Stat -Entity ($vm)-start (get-date).AddDays(-1) -Finish
     (Get-Date) -stat cpu.used.summation
  $statmem = Get-Stat -Entity ($vm)-start (get-date).AddDays(-1) -Finish
     (Get-Date) -stat mem.usage.average
  $actualmem = Get-Stat -Entity ($vm)-start (get-date).AddDays(-1) -Finish
     (Get-Date) -stat mem.active.average
  $consumemem = Get-Stat -Entity ($vm)-start (get-date).AddDays(-1) -Finish
     (Get-Date) -stat mem.consumed.average
  $statgrantmem = Get-Stat -Entity ($vm)-start (get-date).AddDays(-1) -Finish
     (Get-Date) -stat mem.granted.average
  $statdisk = Get-Stat -Entity ($vm)-start (get-date).AddDays(-1) -Finish
     (Get-Date) -stat disk.usage.average
```

[4] PavanNet PowerCli example for extracting metrics. https://https://pchawda.wordpress.com/2021/05/28/powercli-script-to-capture-cpu-memory-usage-stats-of-vms-from-vcenter/.

```
  $cpu = $statcpu | Measure-Object -Property value -Average -Maximum -Minimum
  $cpuuse = $statcpuuse | Measure-Object -Property value -Maximum
  $mem = $statmem | Measure-Object -Property value -Average -Maximum -Minimum
  $actmem = $actualmem | Measure-Object -Property value -Average -Maximum
     -Minimum
  $Conmem = $consumemem | Measure-Object -Property value -Maximum
  $Grantmem = $statgrantmem | Measure-Object -Property value -Average
  $disk = $statdisk | Measure-Object -Property value -Average -Maximum -Minimum


  $DateString = Get-Date -uFormat %Y-%m-%d-%H%M%S



  $vmstat.date = $datestring
  $vmstat.CPUMax = $cpu.Maximum
  $vmstat.CPUAvg = $cpu.Average
  $vmstat.CPUMin = $cpu.Minimum
  $vmstat.cpuuse = $cpuuse.Maximum
  <# $vmstat.MemGB = $MemGB.Maximum #>
  $vmstat.MemMax = $mem.Maximum
  $vmstat.MemAvg = $mem.Average
  $vmstat.MemMin = $mem.Minimum
  $vmstat.ActiveMem = $actmem.Maximum
  $vmstat.conmem = $conmem.Maximum
  $vmstat.grantmem = $Grantmem.Average
  $vmstat.Diskavg = $disk.average
  $vmstat.DiskMax = $disk.Maximum
  $vmstat.DiskMin = $disk.Minimum

  $allvms += $vmstat
}



$Outfile = "live-VMT-Report-" + $DateString + ".csv"
$allvms | Select date, VmName, MemMax, MemAvg, MemMin, ActiveMem, Conmem,
   Grantmem, CPUMax, CPUAvg, CPUMin, CpuUse, Diskavg, Diskmax, DiskMin |
   Export-Csv $Outfile -noTypeInformation
```

5 Log feed to Mysql Database.

All log files are saved to one drive and are held there as primary store of scarped data. To build the dataset Db all files are copied to directly to my pc and fed into a sql Database called VMT with a table called vmtdata.

to create the table the following sql command is run. To create database

```
CREATE DATABASE vmt;
```

to create table structure

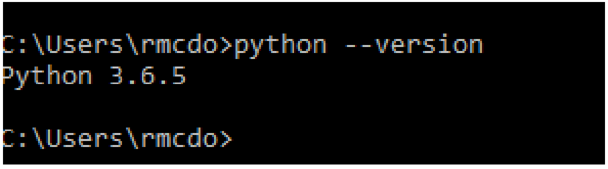
```

-- Dumping structure for table vmt.vmtdata
CREATE TABLE IF NOT EXISTS ‘vmtdata‘ (
  ‘VmDate‘ datetime DEFAULT NULL,
  ‘VmName‘ varchar(50) DEFAULT NULL,
  ‘MemMax‘ double DEFAULT NULL,
  ‘MemAvg‘ double DEFAULT NULL,
  ‘MemMin‘ double DEFAULT NULL,
  ‘Activemem‘ double DEFAULT NULL,
  ‘Conmem‘ double DEFAULT NULL,
  ‘Grantmem‘ double DEFAULT NULL,
  ‘CPUMax‘ double DEFAULT NULL,
  ‘CPUAvg‘ double DEFAULT NULL,
  ‘CPUMin‘ double DEFAULT NULL,
  ‘CpuUse‘ double DEFAULT NULL,
  ‘Diskavg‘ double DEFAULT NULL,
  ‘Diskmax‘ double DEFAULT NULL,
  ‘DiskMin‘ double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

This SQL statement has been added to the list of artefact files called ’vmttablecreate.txt’. This statement executes the columns needed to feed information to the database.

5.1 Python setup and script execution.

Software needed. MySql 8.0, Navicat for MySql for loading statements and creating tables. Please also ensure a fully functional version of python. Version running on my workstation available in figure 7

Figure 7: Python Version



```
C:\Users\rmcdo>python --version
Python 3.6.5

C:\Users\rmcdo>
```

the first step in enabling your machine to load data into mysql database by installing libraries. In a command prompt with administrative privileges run the following commands

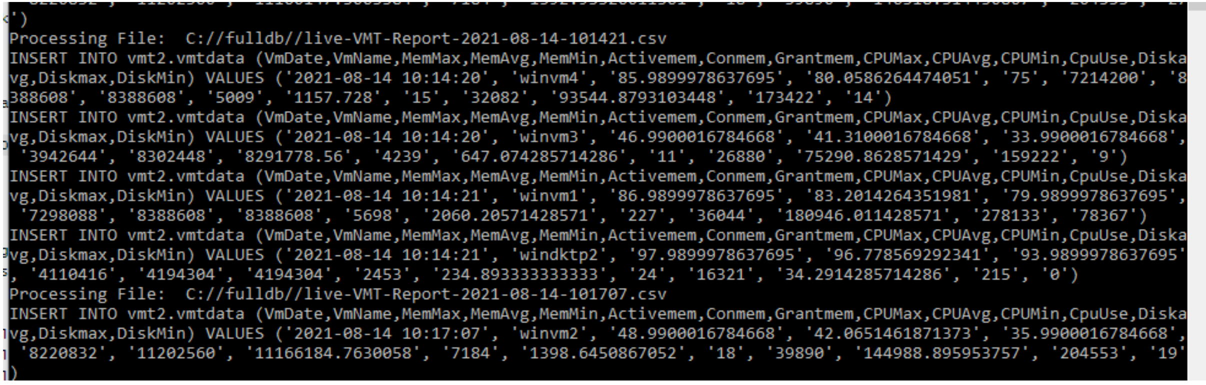
```
C:\Users\rmcdo>pip3 install --trusted-host=pypi.org
    --trusted-host=files.pythonhosted.org --user pandas
C:\Users\rmcdo>pip3 install --trusted-host=pypi.org
    --trusted-host=files.pythonhosted.org --user cx_oracle
C:\Users\rmcdo>pip3 install --trusted-host=pypi.org
    --trusted-host=files.pythonhosted.org --user psycopg2
```

```
C:\Users\rmcdo>pip3 install --trusted-host=pypi.org
    --trusted-host=files.pythonhosted.org --user matplotlib
C:\Users\rmcdo>pip3 install --trusted-host=pypi.org
    --trusted-host=files.pythonhosted.org --user mysql-connector-python
```

This file contains the python script to feed the SQL files into the database is called 'csvload.py'. This script has been added to the list of artefact files. from the command prompt where the python script resides run the following command. In Fig 8 aa screenshot of the process is shown

```
python csvload.py
```

Figure 8: DB Feed



```
')
Processing File:  C://fulldb//live-VMT-Report-2021-08-14-101421.csv
INSERT INTO vmt2.vmtdata (VmDate,VmName,MemMax,MemAvg,MemMin,Activemem,Conmem,Grantmem,CPUMax,CPUAvg,CPUMin,CpuUse,Diska
vg,Diskmax,DiskMin) VALUES ('2021-08-14 10:14:20', 'winvm4', '85.9899978637695', '80.0586264474051', '75', '7214200', '8
388608', '8388608', '5009', '1157.728', '15', '32082', '93544.8793103448', '173422', '14')
INSERT INTO vmt2.vmtdata (VmDate,VmName,MemMax,MemAvg,MemMin,Activemem,Conmem,Grantmem,CPUMax,CPUAvg,CPUMin,CpuUse,Diska
vg,Diskmax,DiskMin) VALUES ('2021-08-14 10:14:20', 'winvm3', '46.9900016784668', '41.3100016784668', '33.9900016784668',
 '3942644', '8302448', '8291778.56', '4239', '647.074285714286', '11', '26880', '75290.8628571429', '159222', '9')
INSERT INTO vmt2.vmtdata (VmDate,VmName,MemMax,MemAvg,MemMin,Activemem,Conmem,Grantmem,CPUMax,CPUAvg,CPUMin,CpuUse,Diska
vg,Diskmax,DiskMin) VALUES ('2021-08-14 10:14:21', 'winvm1', '86.9899978637695', '83.2014264351981', '79.9899978637695',
 '7298088', '8388608', '8388608', '5698', '2060.20571428571', '227', '36044', '180946.011428571', '278133', '78367')
INSERT INTO vmt2.vmtdata (VmDate,VmName,MemMax,MemAvg,MemMin,Activemem,Conmem,Grantmem,CPUMax,CPUAvg,CPUMin,CpuUse,Diska
vg,Diskmax,DiskMin) VALUES ('2021-08-14 10:14:21', 'windktp2', '97.9899978637695', '96.778569292341', '93.9899978637695'
, '4110416', '4194304', '4194304', '2453', '234.893333333333', '24', '16321', '34.2914285714286', '215', '0')
Processing File:  C://fulldb//live-VMT-Report-2021-08-14-101707.csv
INSERT INTO vmt2.vmtdata (VmDate,VmName,MemMax,MemAvg,MemMin,Activemem,Conmem,Grantmem,CPUMax,CPUAvg,CPUMin,CpuUse,Diska
vg,Diskmax,DiskMin) VALUES ('2021-08-14 10:17:07', 'winvm2', '48.9900016784668', '42.0651461871373', '35.9900016784668',
 '8220832', '11202560', '11166184.7630058', '7184', '1398.6450867052', '18', '39890', '144988.895953757', '204553', '19'
)
```

6 Machine specific extraction SQL queries for experiments

The list of individual SQL queries from VMT database to extract relevant utilisation data for each VM.

Winvm1 Memory information extraction

```
SELECT VmDate,MemMax FROM vmtdata WHERE VmName LIKE 'winvm1'

and VmDate BETWEEN CAST('2021-01-01' AS DATE) AND CAST('2021-12-31' AS DATE)

INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL Server
    8.0\\Uploads\\winvm1maxmemexp1.csv'
```

Winvm1 CPU utilisation extraction

```
SELECT VmDate,CPUMax FROM vmtdata WHERE VmName LIKE 'winvm1'

and VmDate BETWEEN CAST('2021-01-01' AS DATE) AND CAST('2021-012-31' AS DATE)

 INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL Server
     8.0\\Uploads\\winvm1maxcpuexp1.csv'
```

Winvm3 Max memory extraction

```
SELECT VmDate,MemMax FROM vmtdata WHERE VmName LIKE 'winvm3'

and VmDate BETWEEN CAST('2021-01-01' AS DATE) AND CAST('2021-12-31' AS DATE)

INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL Server
    8.0\\Uploads\\winvm3maxmemexp2.csv'
```

winvm3 CPU extraction

```
SELECT VmDate,CPUMax FROM vmtdata WHERE VmName LIKE 'winvm3'

and VmDate BETWEEN CAST('2021-01-01' AS DATE) AND CAST('2021-12-31' AS DATE)

INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL Server
    8.0\\Uploads\\winvm3maxcpuexp2.csv'
```

Winvm4 Memory information extraction

```
SELECT VmDate,MemMax FROM vmtdata WHERE VmName LIKE 'winvm4'

and VmDate BETWEEN CAST('2021-01-01' AS DATE) AND CAST('2021-12-31' AS DATE)

INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL Server
    8.0\\Uploads\\winvm4maxmemexp3.csv'
```

Winvm4 CPU information extraction

```
SELECT VmDate,CPUMax FROM vmtdata WHERE VmName LIKE 'winvm4'

and VmDate BETWEEN CAST('2021-01-01' AS DATE) AND CAST('2021-12-31' AS DATE)

INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL Server
    8.0\\Uploads\\winvm4maxcpuexp3.csv'
```

Winvm2 Memory information extraction

```
SELECT VmDate,MemMax FROM vmtdata WHERE VmName LIKE 'winvm2'

and VmDate BETWEEN CAST('2021-01-01' AS DATE) AND CAST('2021-12-31' AS DATE)

INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL Server
    8.0\\Uploads\\winvm2maxmemexp4.csv'
```

Winvm2 CPU information extraction

```
SELECT VmDate,CPUMax FROM vmtdata WHERE VmName LIKE 'winvm2'

and VmDate BETWEEN CAST('2021-01-01' AS DATE) AND CAST('2021-12-31' AS DATE)

INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL Server
    8.0\\Uploads\\winvm2maxcpuexp4.csv'
```

These statements have been added to the Artefact Repository as file 'SqlExtraction.txt'.

7 Section 7

7.1 Time-Series forecasting using R

A Number of the plots, graphs and features that were added to the Forecasting code have been found in the following text By RJ Hyndman Hyndman and Athanasopoulos (2018).

The file 'Rforecastfinalexample.R' is a breakdown of the steps run to use for forecasting. Also included in the Artefact folder is a list of the RMarkdown files and reports extracted from each report.

The two packages required to install to run all code are

```
install.packages('forecast')
install.packages('tseries')
```

References

Hyndman, R. and Athanasopoulos, G. (2018). *Forecasting: principles and practice, 2nd edition*, Otexts, Melbourne, Australia.
URL: *www.OTexts.com/fpp2*