Table of Contents

Custom data tables 3
NetYCE databases 3
Database accounts 3
Patches tables 3
Table requirements 4
MySQL managers 5
Using the "mysql" command 5

Custom data tables

The NetYCE database allows you to create custom data tables for use in Relations and for general purposes. These tables are accessible for data maintenance using the “Custom data” tool under the “Admin” menu.

This article describes how to create a custom table.

NetYCE databases

NetYCE supports tables in four distinct databases, each for a specific purpose.

Database	Function	Remarks
YCE	Networking device models and objects	netYCE managed
LOGS	Application logs	netYCE managed
NCCM	Network device configurations history	netYCE managed
CMDB	CMDB tables	NetYCE managed, customer accessible
NMS	Custom tables and Customer specials	Customer managed

Custom tables are permitted only in the “**NMS**” database. The tables created here will be included in the Custom data tool.

When the system creates a backup of its database, the YCE, LOGS, CMDB and NMS databases are included in the “YCEdb” archives. The NCCM is archived separately in the “NCCMdb” archives.

NetYCE uses the “MariaDB” database engine which is a derivative of “MySQL” and still uses that name for its processes and configuration.

Database accounts

To facilitate the permissions over the various databases and tables, the “mysql” database comes predefined with the necessary MySQL accounts.

User	Scope	Password
netYCE	full admin	*not public*
touchup	application r/w	*not public
justread	application r/o	gmngmhhw
replication	master/master replication	*not public*
cmdbadmin	CMDB table mgmt	cmdbadmin
nmsadmin	NMS table mgmt	nmsadmin

To create and manipulate tables in the NMS database the 'nmsadmin' account must be used.

Patches tables

NetYCE relies on an extensive system of patches that automatically update database tables and data during installation and software updates. To keep track of the patches already installed or to be executed a dedicated table is created in each of the databases. Its existence and data integrity is essential for the proper operation of NetYCE. These tables should never be modified in any sense.

In the NMS database this table is named “Nms_patches”. Likewise the CMSB database has a “Cmdb_patches” table.

Table requirements

For NetYCE to support Custom tables properly, a few requirements on the table definition apply:

- Table “**ENGINE=MyISAM DEFAULT CHARSET=utf8**”
  The table must be defined as MyISAM using the utf8 character set
- Table at least one primary key “**PRIMARY KEY (`columnA`,`columnB`)**”
  Tables without a primary key cannot be managed using the “Custom data” tools.
- Use column type “**VARCHAR(n)**” for strings where possible.
  String columns should be defined using the VARCHAR type (up yo 64KB). Integers as INT and for larger sizes the MEDIUMTEXT (up to 16MB) should be used. Binary values are best served using VARBINARY (up to 64KB) or LONGBLOB (up to 4GB).
- Indicate a default value like and empty string and if a NULL value is allowed “**varchar(20) NOT NULL DEFAULT ''**”
  Key columns should never have NULL's allowed, and using an empty string as default simplifies retrieving data that is either null of empty.
- Create secondary (MUL) keys for columns often used in JOINS “**KEY `Hosts` (`HostA`,`HostB`)**”
  To retrieve data faster, create secondary keys for column combinations that are often used in combination. Especially in larger tables or complex JOIN statements can these extra indexes speed up retrievals. Avoid creating too many secondaries as they need to be computed and maintained with every INSERT.

1. Last column definition to make database replication more reliable: “**Timestamp` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()**”“
   This statement added a column Timestamp that is updated with the current date and time each time the row is modified. Apart from being handy when examining changed rows, it also optimizes the synchronization between database servers in the master/master replication setup NetYCE deploys.

Example of a poorly defined table:

```
CREATE TABLE `NMS`.`Custom_domains` (
  `Domain` char(20) NOT NULL,
  `Domain_name` char(100)
);
```

The same table properly defined:

```
CREATE TABLE `NMS`.`Custom_domains` (
```

```
  `Domain` varchar(20) NOT NULL DEFAULT '',
  `Domain_name` varchar(100) DEFAULT '',
  `Timestamp` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
  PRIMARY KEY (`Domain`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

MySQL managers

To create or modify a Custom table, any of the available Mysql management applications can be used. They often offer many different functions for examining and modifying databases and are available for Linux, Mac and Windows. Many commercial and free tools are available on the internet.

- Mysql Workbench for all platforms
- Sequel Pro for Mac

Additionally, installed with the NetYCE distribution is the “mysql” client for the command line. It is a very basic tool that can be used to manipulate the database using SQL statements. A sample session is included below. This tool is only useful for operators familiar with SQL.

Using the "mysql" command

The sample session below shows how to login and first create a table, and then remove it.

Login as “yce” user using ssh. Then execute the “mysql” command as below. Enter the password when prompted.

```
$ mysql -u nmsadmin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8735
Server version: 10.2.41-MariaDB-log MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MariaDB [(none)]> use NMS;
Database changed
MariaDB [NMS]> show tables;
+--------------------+
| Tables_in_NMS      |
+--------------------+
| Nms_patches        |
+--------------------+
1 row in set (0.00 sec)

MariaDB [NMS]> describe Nms_patches;
```

```
+--------------+--------------+------+-----+---------------------+----------
---------------------+
| Field        | Type         | Null | Key | Default             | Extra
|
+--------------+--------------+------+-----+---------------------+----------
---------------------+
| Patch_id     | varchar(10)  | NO   | PRI |                     |
|
| Patch_base   | varchar(10)  | NO   |     |                     |
|
| Patch_status | varchar(50)  | NO   |     |                     |
|
| Patch_descr  | varchar(200) | NO   |     |                     |
|
| Patch_log    | text         | YES  |     | NULL                |
|
| Timestamp    | timestamp    | NO   |     | current_timestamp() | on update
current_timestamp() |
+--------------+--------------+------+-----+---------------------+----------
---------------------+
6 rows in set (0.00 sec)

MariaDB [NMS]> SELECT Patch_id, Patch_descr FROM Nms_patches ORDER BY
Patch_id DESC LIMIT 1;
+----------+-----------------------------------------------+
| Patch_id | Patch_descr                                   |
+----------+-----------------------------------------------+
| 21092706 | NMS cutoff patch for setting 7.2.0 to 8.0.0 |
+----------+-----------------------------------------------+
1 row in set (0.00 sec)

MariaDB [NMS]> CREATE TABLE `Custom_domains` (
    ->   `Domain` varchar(20) NOT NULL DEFAULT '',
    ->   `Domain_name` varchar(100) DEFAULT '',
    ->   `Timestamp` timestamp NOT NULL DEFAULT current_timestamp() ON
UPDATE current_timestamp(),
    ->   PRIMARY KEY (`Domain`)
    -> ) ENGINE=MyISAM DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.00 sec)

MariaDB [NMS]> describe Custom_domains;
+-------------+--------------+------+-----+---------------------+-----------
---------------------+
| Field       | Type         | Null | Key | Default             | Extra
|
+-------------+--------------+------+-----+---------------------+-----------
---------------------+
| Domain      | varchar(20)  | NO   | PRI |                     |
|
| Domain_name | varchar(100) | YES  |     |                     |
|
```

```
| Timestamp   | timestamp    | NO   |     | current_timestamp() | on update
current_timestamp() |
+-------------+--------------+------+-----+---------------------+-----------
--------------------+
3 rows in set (0.00 sec)

MariaDB [NMS]> DROP TABLE Custom_domains;
Query OK, 0 rows affected (0.00 sec)

MariaDB [NMS]> \q
Bye
```

From:
https://wiki.netyce.com/ - **Technical documentation**

Permanent link:
https://wiki.netyce.com/doku.php/guides:reference:custom_tables

Last update: **2022/01/08 14:08**