



Inventario de dispositivos de Networking.



Alumno Jesus Troconiz

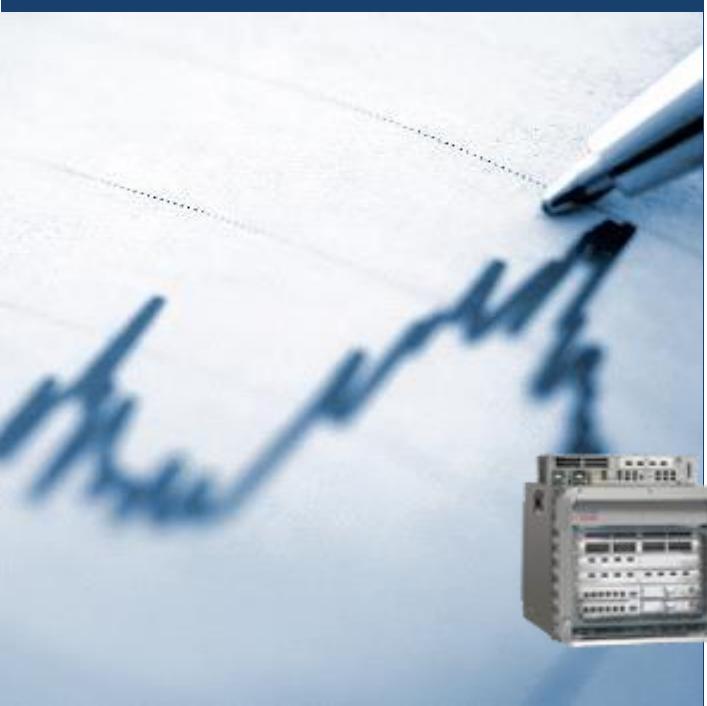
Tutor German Rodriguez

Profesor Miguel Briones

INVEST IN YOUR FUTURE

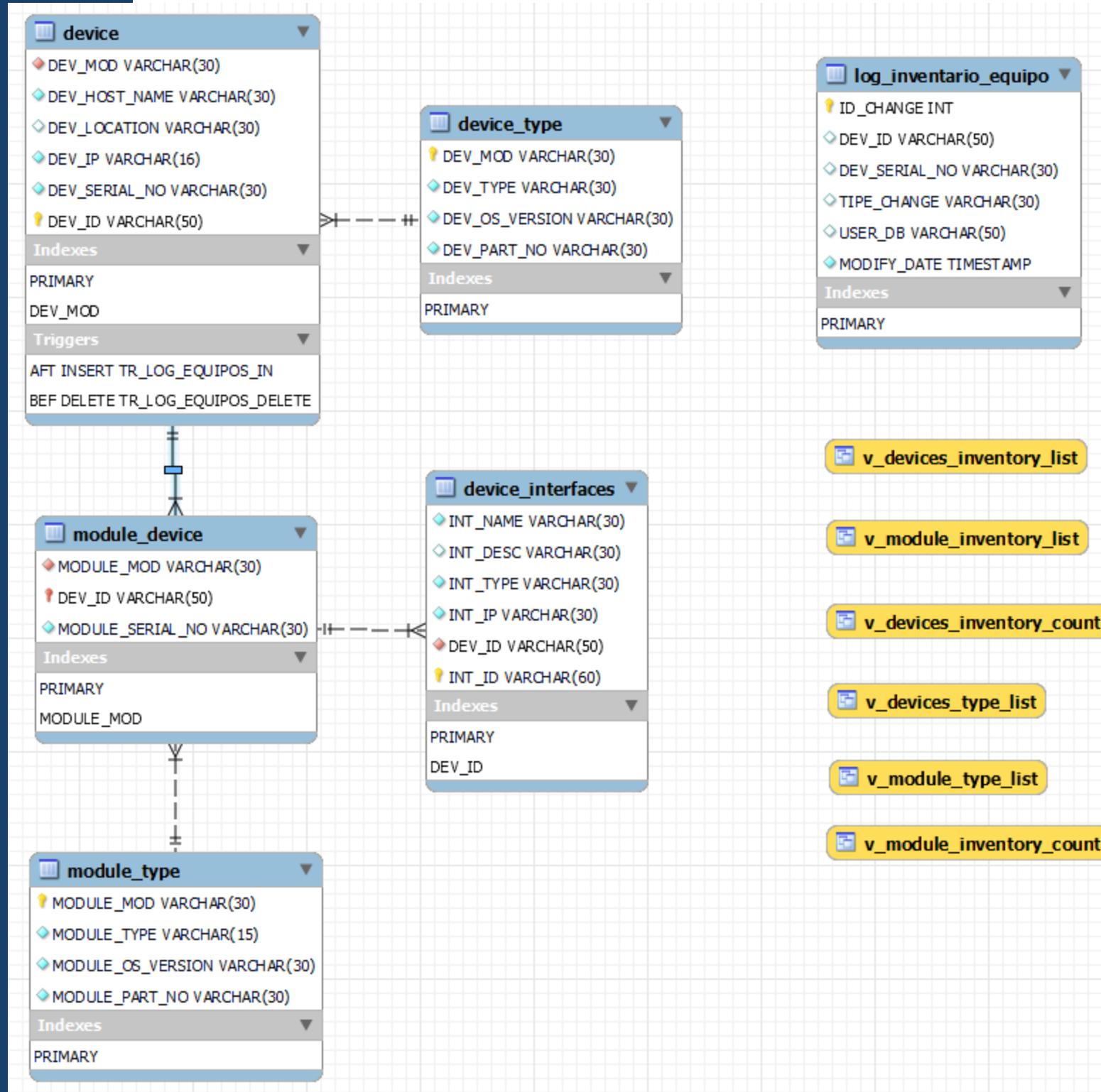
Propuesta

Se necesita realizar una base de datos sobre equipos de telecomunicaciones en donde se tengan datos como modelo, número de serie, ip de gestión, puertos físicos, dirección, cantidad y datos de configuración como servicios internos (IPs), log de modificaciones de las tablas y control de cambios o usuarios.



Entidad de relación

- Se centra en el inventario del numero total de los equipos, y sus modelos.
- Además en el inventario de los modelos que tienes los equipos y sus particularidades ejemplo como una raza.
- Cada equipo va a tener módulos.
- Cada modulo cuenta con distintas interfaces.
- Y cada interfaz tiene distintos servicios.
- Además se incluyen las distintas vistas y la tabla de control de usuarios y cambios



Tablas:

Tabla de modelos de equipos (device_type)					
PK	FK	Field name	Field database name	Data type	NULL
				LEN	NOTES
X		Modelo	DEV_MOD	VARCHAR	NOT
		Device Type	DEV_TYPE	VARCHAR	NOT
		IOS	DEV_OS_VERSION	VARCHAR	NOT
		Part number	DEV_PART_NO	VARCHAR	NOT
					30 Numero de parte del fabricante
Tabla de equipos (device)					
PK	FK	Field name	Field database name	Data type	NULL
				LEN	NOTES
		Host name	DEV_HOST_NAME	VARCHAR	NOT
		Location	DEV_LOCATION	VARCHAR	YES
		IP	DEV_IP	VARCHAR	NOT
X		Modelo	DEV_MOD	VARCHAR	NOT
		Serial	DEV_SERIAL_NO	VARCHAR	NOT
X		ID	DEV_ID	VARCHAR	NOT
					50 DEV_IP + DEV_HOST_NAME
Tabla de modelos de modulos (module_type)					
PK	FK	Field name	Field database name	Data type	NULL
				LEN	NOTES
X		Modelo	MODULE_MOD	VARCHAR	NOT
		Type	MODULE_TYPE	VARCHAR	NOT
		IOS	MODULE_OS_VERSION	VARCHAR	NOT
		Part number	MODULE_PART_NO	VARCHAR	NOT
					30 Numero de parte del fabricante
Tabla de modulos en devices (module_device)					
PK	FK	Field name	Field database name	Data type	NULL
				LEN	NOTES
X	X	ID	DEV_ID	VARCHAR	NOT
	X	Modelo	MODULE_MOD	VARCHAR	NOT
		Serial number	MODULE_SERIAL_NO	VARCHAR	NOT
					30 Numero de Serie
Tabla de Interfaz (device_interfaces)					
PK	FK	Field name	Field database name	Data type	NULL
				LEN	NOTES
X		ID	DEV_ID	VARCHAR	NOT
		Interface name	INT_NAME	VARCHAR	NOT
		Interface description	INT_DESC	VARCHAR	yes
		Interface type	INT_TYPE	VARCHAR	NOT
		Interface IP	INT_IP	VARCHAR	NOT
X		ID	INT_ID	VARCHAR	NOT
					60 INT_IP + INT_NAME

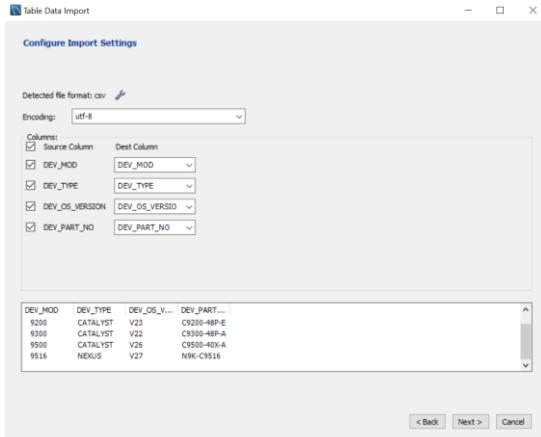
Insert device_type

The screenshot shows the MySQL Workbench interface with the following details:

- Sidebar:** Shows the database schema with the 'network_inventory' database selected. Under 'Tables', 'device_type' is expanded, showing its columns: DEV_MOD (tinyint), DEV_TYPE (varchar), DEV_OS_VERSION (varchar), and DEV_PART_NO (varchar).
- Result Grid:** A preview grid for the 'device_type' table is displayed, showing one row with all columns set to NULL.
- Table Data Import - Select File to Import:** A modal window titled 'Table Data Import' is open, prompting the user to select a file to import. The file path 'C:\Users\troc\Desktop\SQL\tables\device_type.csv' is entered in the 'File Path' field.
- Table Data Import - Select Destination:** Another modal window titled 'Table Data Import' is open, showing the 'Select Destination' step. It asks to 'Select destination table and additional options.' There are two radio button options:
 - Use existing table: network_inventory.device_type
 - Create new table: network_inventory . device_type

A checkbox for 'Truncate table before import' is also present.

Insertando datos a las tablas

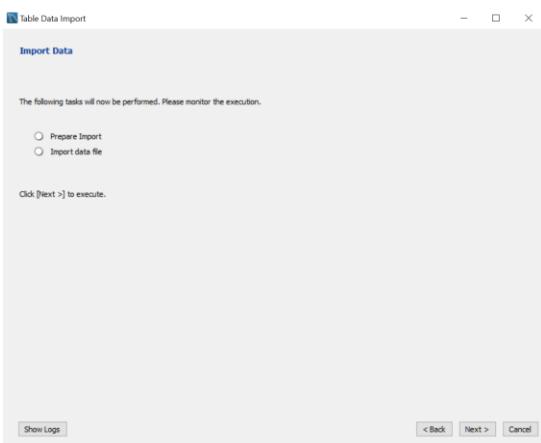


Import Results

File C:\Users\jtroc\Desktop\SQL\tables\device_type.csv was imported in 0.361 s

Table network_inventory.device_type has been used

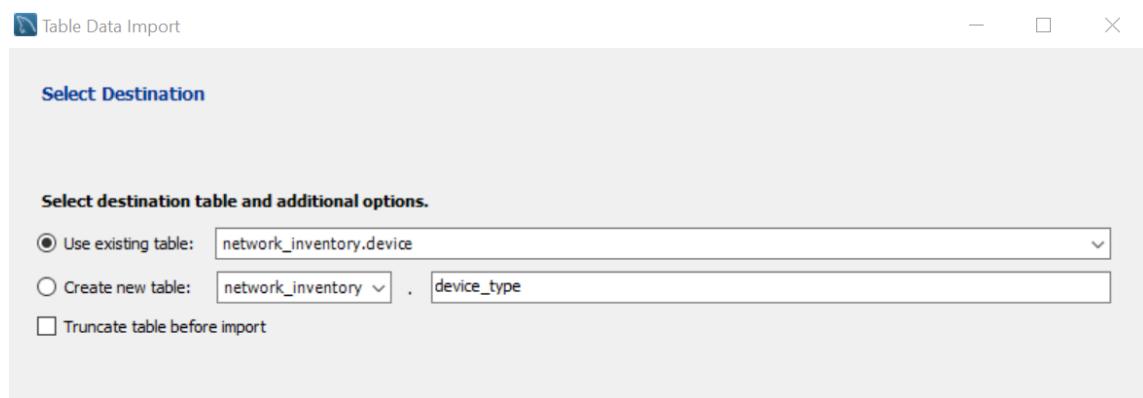
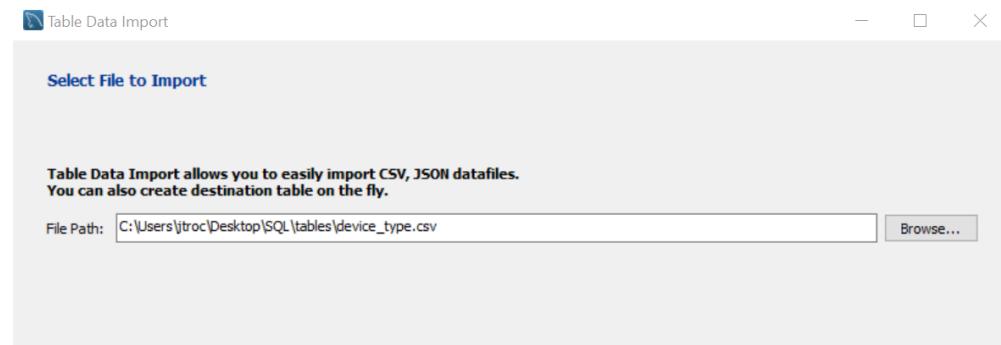
6 records imported

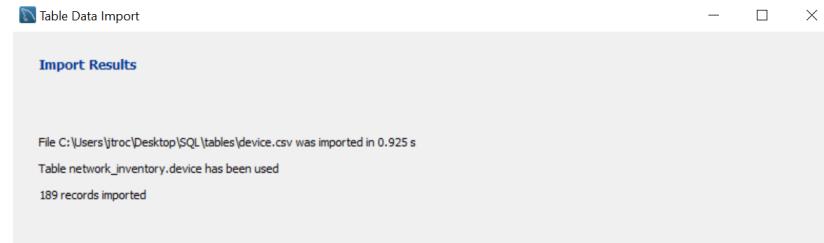
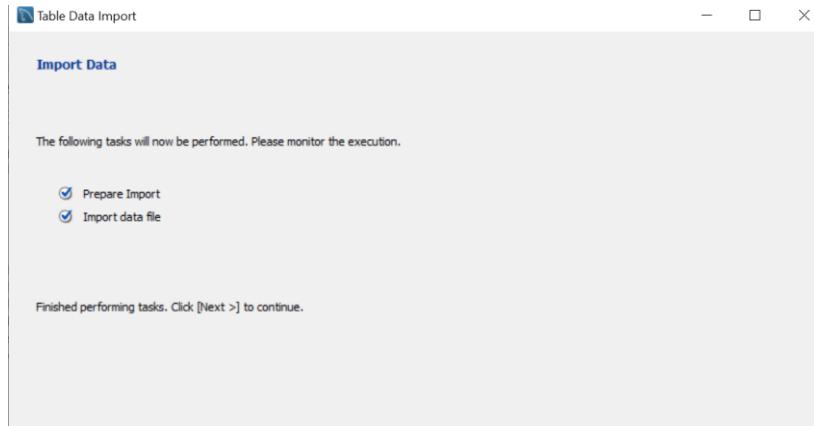


```
1 •   SELECT * FROM network_inventory.device_type;
```

result Grid			
DEV_MOD	DEV_TYPE	DEV_OS_VERSION	DEV_PART_NO
C3850	CATALYST	V23	WS-C3850-48F-S
C9200	CATALYST	V23	C9200-48P-E
C9300	CATALYST	V22	C9300-48P-A
C9500	CATALYST	V26	C9500-40X-A
N9504	NEXUS	V27	N9K-C9504
N9516	NEXUS	V27	N9K-C9516

Insert device





InventarioNetworking-SQL

File 7* SQL File 8* InventarioNetworking-SQL device device device_type device device_type device

1 • SELECT * FROM network_inventory.device;

Navigator

SCHEMAS

- gammers_model
- network_inventory
- sakila
- sys
- world

Tables

- device
- device_interfaces
- device_type
- module_device
- module_type

Views

Stored Procedures

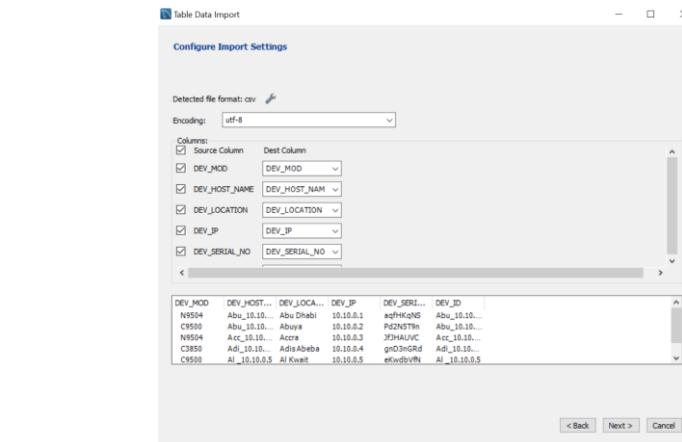
Functions

Administration Schemas Information

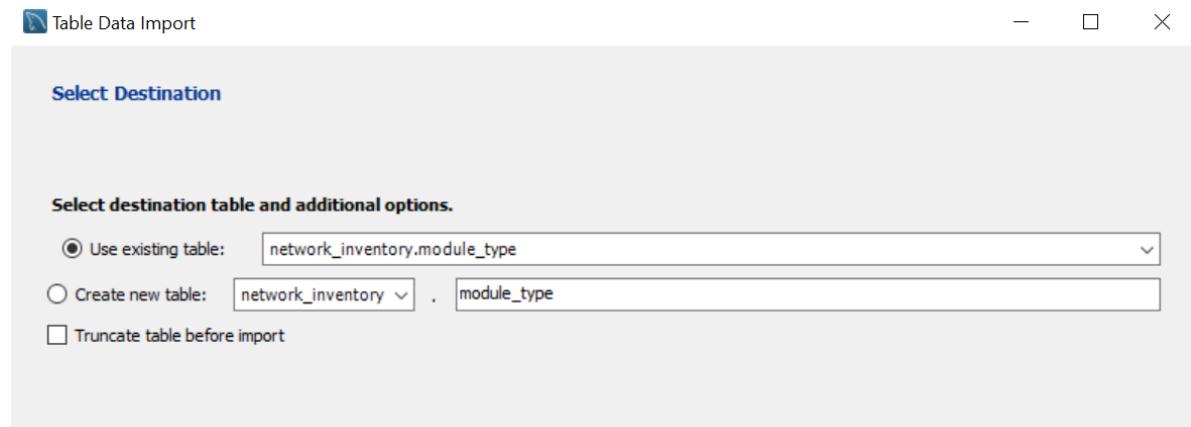
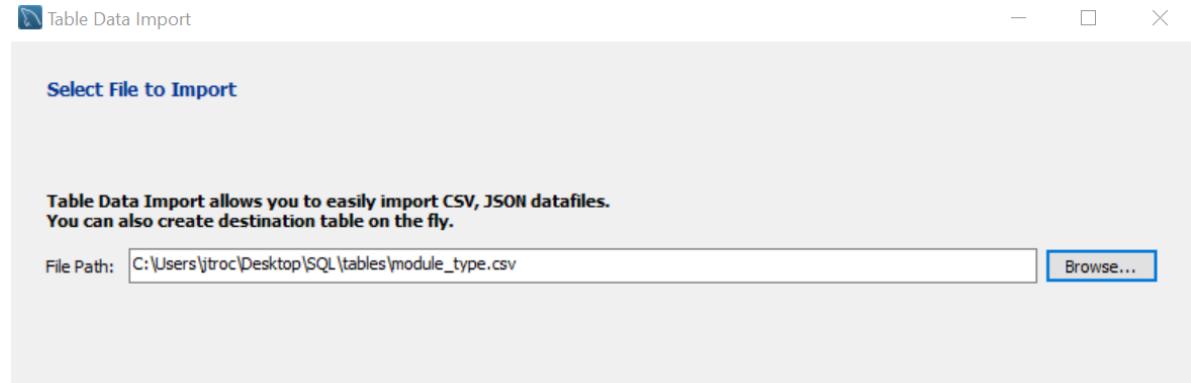
Table: device

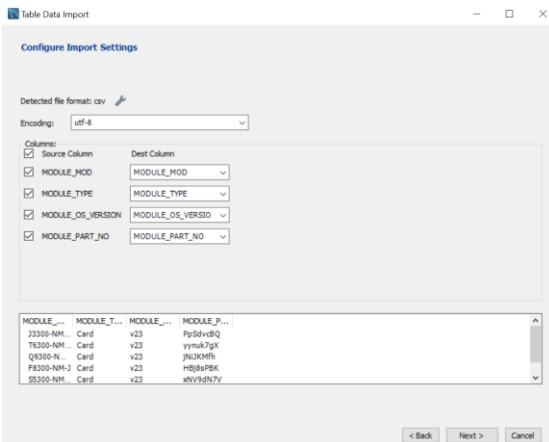
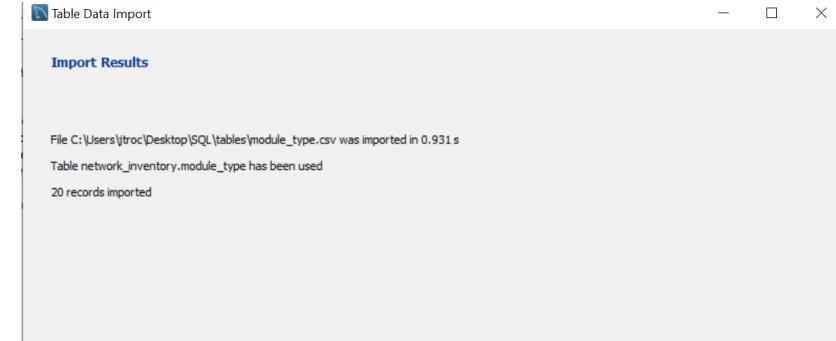
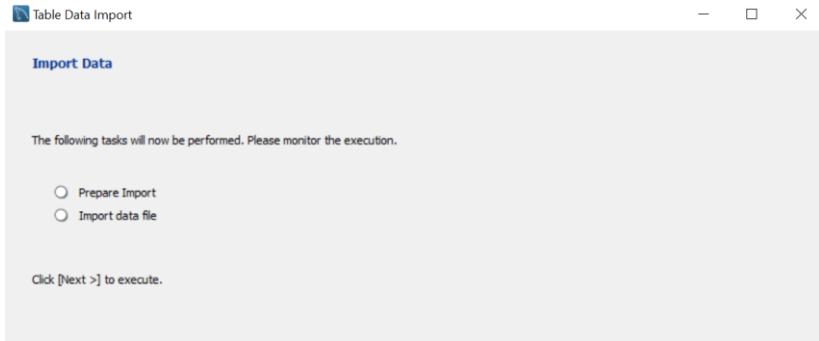
Columns:

DEV_MOD	DEV_HOST_NAME	DEV_LOCATION	DEV_IP	DEV_SERIAL_NO	DEV_ID
N9504	Abu_10.10.0.1	Abu Dhabi	10.10.0.1	aqHqKqN5	Abu_10.10.0.1
C9500	Abu_10.10.0.2	Abuya	10.10.0.2	Pd2NTSt9n	Abu_10.10.0.2
N9504	Acc_10.10.0.3	Accra	10.10.0.3	JfHAUVC	Acc_10.10.0.3
C3850	Ad_10.10.0.4	Adis Abeba	10.10.0.4	gnD3nGRd	Ad_10.10.0.4
C9500	Al_10.10.0.5	Al Kuwait	10.10.0.5	eKvdbVFn	Al_10.10.0.5
C3850	Amm_10.10.0.6	Amman	10.10.0.6	cieRr94	Amm_10.10.0.6
C3850	Ams_10.10.0.7	Ämsterdam	10.10.0.7	BectJnb	Ams_10.10.0.7
N9504	And_10.10.0.8	Andorra la Vella	10.10.0.8	NSMouJC6	And_10.10.0.8
C9200	Ant_10.10.0.10	Antananarivo	10.10.0.10	V8jpi965	Ant_10.10.0.10
C9500	Ap_10.10.0.11	Apia	10.10.0.11	QhnDyq2z	Ap_10.10.0.11
C9500	Arg_10.10.0.12	Argel	10.10.0.12	dhrQQu	Arg_10.10.0.12
C3850	Asj_10.10.0.13	Asjabad	10.10.0.13	cyhAvH5iv	Asj_10.10.0.13
C9200	Asm_10.10.0.14	Asmara	10.10.0.14	6uvvG6n	Asm_10.10.0.14
C3850	Asu_10.10.0.15	Asunción	10.10.0.15	nviWfIvlz	Asu_10.10.0.15
C9200	Ate_10.10.0.16	Atenas	10.10.0.16	ZeojcYt	Ate_10.10.0.16
C9500	Bag_10.10.0.17	Bagdad	10.10.0.17	XaiJffWSh	Bag_10.10.0.17
C9200	Bak_10.10.0.18	Baku	10.10.0.18	S7.aYyKd	Bak_10.10.0.18
C9200	Bam_10.10.0.19	Bamako	10.10.0.19	SgrBLTM	Bam_10.10.0.19



Insert module_type



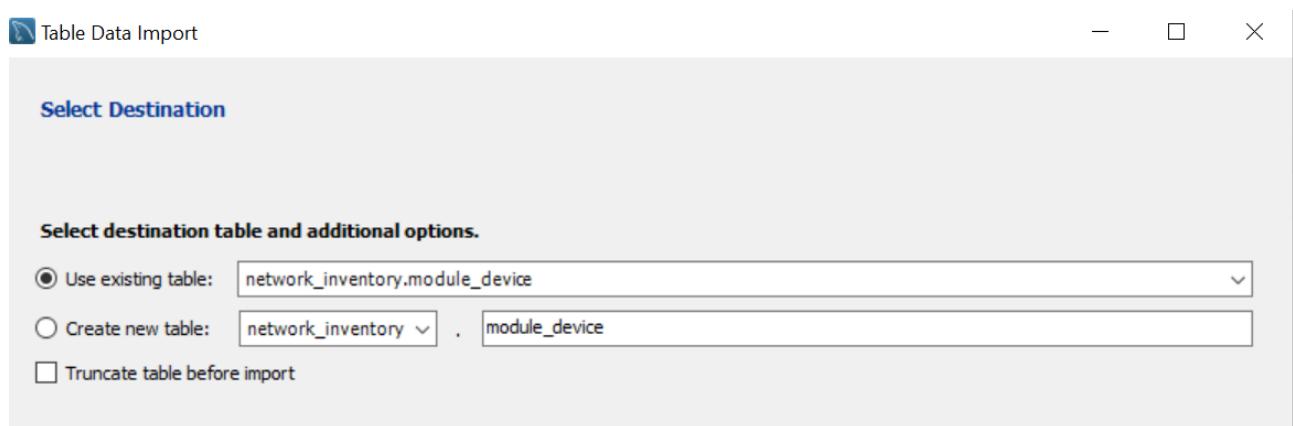
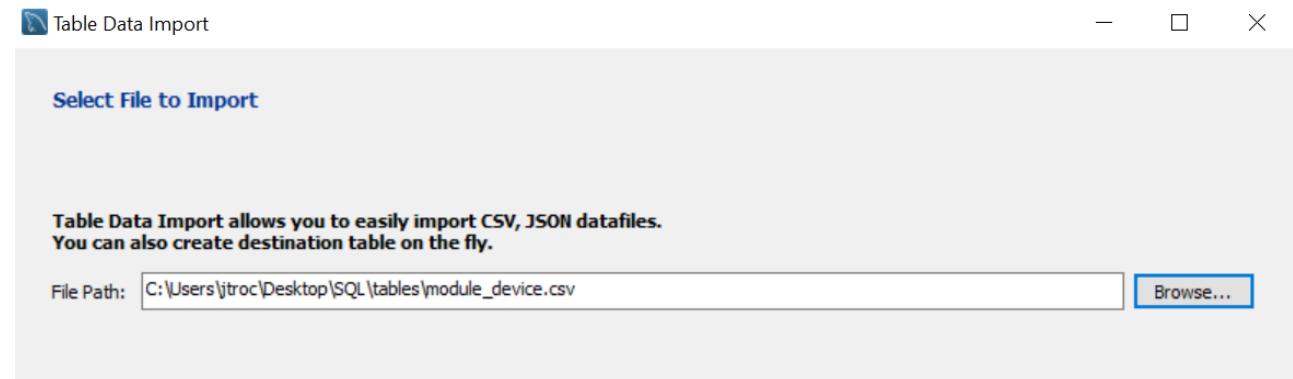


1 • `SELECT * FROM network_inventory.module_type;`

Result Grid

MODULE_MOD	MODULE_TYPE	MODULE_OS_VERSION	MODULE_PART_NO
F2300-NM-H	Card	v33	HbmKqpc
F8300-NM-J	Card	v23	HBjsPBK
F9300-NM-H	Card	v33	xbog3J0M
G7300-NM-M	Card	v23	qLEby43
G7300-NM-Q	Card	v33	3ExSoGkQ
G8300-NM-A	Card	v33	es5C6Eds
H9300-NM-W	Card	v23	7wRLU7h6
J3300-NM-M	Card	v23	PpSd7BQ
K9300-NM-F	Card	v33	T2VUmPw
K9300-NM-N	Card	v33	DzY2LPH2
M5300-NM-B	Card	v31	z457YkbQ
N3300-NM-D	Card	v23	MpMkzePz
N6300-NM-A	Card	v33	ykqW36n
Q3300-NM-C	Card	v32	fH6sqBQG
Q3300-NM-V	Card	v33	2JwT9kx3
Q9300-NM-J	Card	v23	jNjUK0fh
S5300-NM-Y	Card	v23	xN9dn7v
T6300-NM-S	Card	v23	yrruk7gk

Insert module_device



Navigator:

- SCHEMAS**
 - Filter objects
 - gammers_model
 - network_inventory**
 - Tables
 - device
 - device_interfaces
 - device_type
 - module_device**
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions
 - sakila
 - sys
 - world

Administration Schemas

Information

Table: module_device

Columns:

MODULE_MOD	DEV_ID	MODULE_SERIAL_NO
J3300-NM-M	Abu_10.10.0.1	fGHeGhWnnmeJLR
T6300-NM-S	Abu_10.10.0.2	ogkQ8P7mA63z5x
Q9300-NM-J	Acc_10.10.0.3	XZLE8JnnH4yQYt
F8300-NM-J	Adi_10.10.0.4	G9XLXDPdPc2QaQ
S5300-NM-Y	AI_10.10.0.5	zyvPZ6g8JzGjW
N3300-NM-D	Amm_10.10.0.6	iPN5Chvbhuomp
H9300-NM-W	Ans_10.10.0.7	stekuvuueUmh9
G7300-NM-M	And_10.10.0.8	PmbfugZW5UpWW
Q3300-NM-C	Ant_10.10.0.10	SKgdDqkj9BKUL
K9300-NM-N	Ap_10.10.0.11	FbX4qPUSeERMC
F2300-NM-H	Arg_10.10.0.12	72JxK7Oz28xNpV
X6300-NM-Y	Asl_10.10.0.13	VZsks#RVGmjqQ
K8300-NM-F	Asm_10.10.0.14	GEV1dpTLGvH2f
Z3300-NM-U	Asu_10.10.0.15	evCZ3HLKGvKC97
Q3300-NM-V	Ate_10.10.0.16	4TaD2u42J8L6A
F9300-NM-H	Bag_10.10.0.17	RRRNNUyFpRc9K
G8300-NM-A	Bal_10.10.0.18	Tk6FLYzTtaR4v
N6300-NM-A	Ban_10.10.0.19	9GhabJWPbHE7b

Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

<input checked="" type="checkbox"/> Source Column	Dest Column
<input checked="" type="checkbox"/> MODULE_MOD	MODULE_MOD
<input checked="" type="checkbox"/> DEV_ID	DEV_ID
<input checked="" type="checkbox"/> MODULE_SERIAL_NO	MODULE_SERIAL_N

Import Data

The following tasks will now be performed. Please monitor the execution.

Prepare Import
 Import data file

Click [Next >] to execute.

Table Data Import

Configure Import Settings

Detected file format: csv

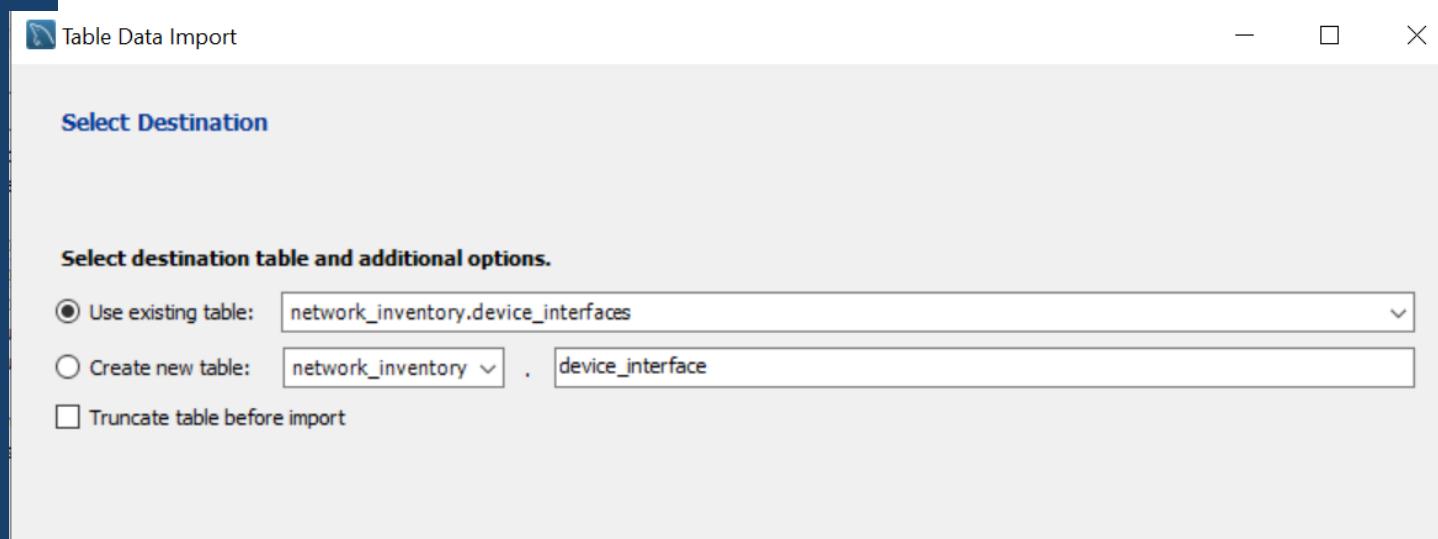
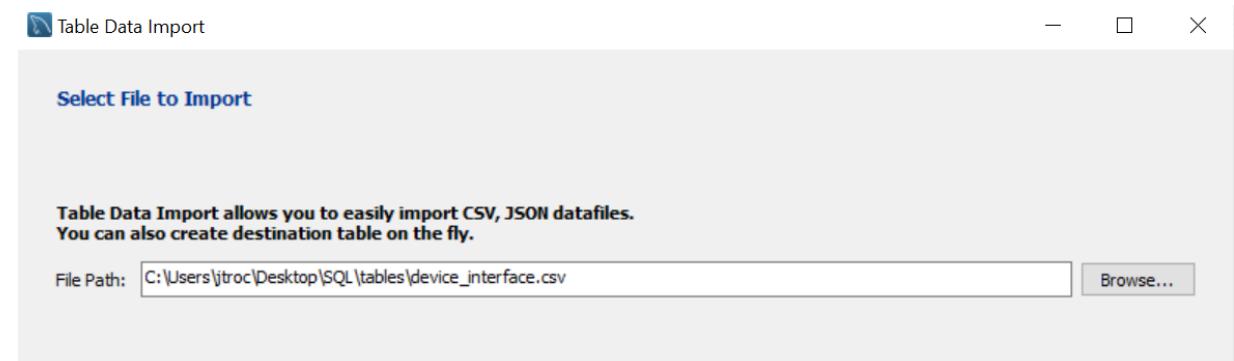
Encoding: utf-8

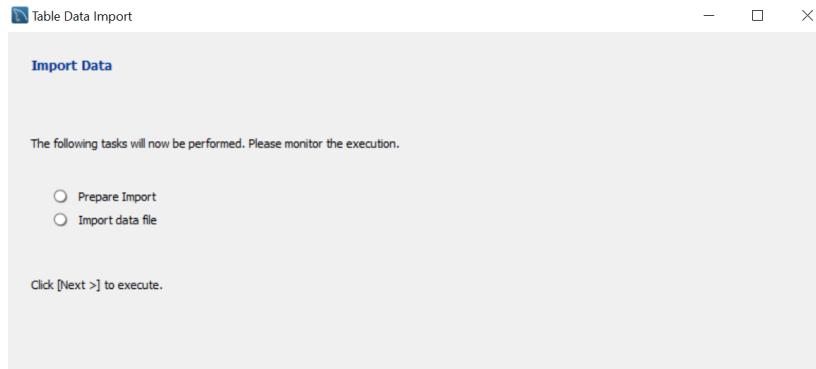
Columns:

MODULE_...	DEV_ID	MODULE_S...
J3300-NM...	Abu_10.10....	fGHeGhWn...
T6300-NM...	Abu_10.10....	ogkQ8P7m...
Q9300-N...	Acc_10.10....	XZLE8JnnH...
F8300-NM-J	Adi_10.10....	G9XLXDPd...
S5300-NM...	AI_10.10.0.5	zyvPZ6g8J...

< Back Next > Cancel

Insert device_interface





Filter objects

1 • **SELECT * FROM network_inventory.device_interfaces;**

Result Grid

INT_NAME	INT_DESC	INT_TYPE	INT_IP	DEV_ID	INT_ID
ge-1/0/1	ge	10.0.1.1	Abu_10.10.0.1	ge-1/0/1_10.0.1.1	
ge-1/0/1	ge	10.0.1.10	Ant_10.10.0.10	ge-1/0/1_10.0.1.10	
ge-1/0/1	ge	10.0.1.100	Map_10.10.0.100	ge-1/0/1_10.0.1.100	
ge-1/0/1	ge	10.0.1.101	Map_10.10.0.101	ge-1/0/1_10.0.1.101	
ge-1/0/1	ge	10.0.1.102	Mas_10.10.0.102	ge-1/0/1_10.0.1.102	
ge-1/0/1	ge	10.0.1.103	Mba_10.10.0.103	ge-1/0/1_10.0.1.103	
ge-1/0/1	ge	10.0.1.104	Mn_10.10.0.104	ge-1/0/1_10.0.1.104	
ge-1/0/1	ge	10.0.1.105	Mog_10.10.0.105	ge-1/0/1_10.0.1.105	
ge-1/0/1	ge	10.0.1.106	Món_10.10.0.106	ge-1/0/1_10.0.1.106	
ge-1/0/1	ge	10.0.1.107	Mon_10.10.0.107	ge-1/0/1_10.0.1.107	
ge-1/0/1	ge	10.0.1.108	Mer_10.10.0.108	ge-1/0/1_10.0.1.108	
ge-1/0/1	ge	10.0.1.109	Mos_10.10.0.109	ge-1/0/1_10.0.1.109	
ge-1/0/1	ge	10.0.1.11	Ap_10.10.0.11	ge-1/0/1_10.0.1.11	
ge-1/0/1	ge	10.0.1.110	Na_10.10.0.110	ge-1/0/1_10.0.1.110	
ge-1/0/1	ge	10.0.1.111	Ne_10.10.0.111	ge-1/0/1_10.0.1.111	
ge-1/0/1	ge	10.0.1.112	Nas_10.10.0.112	ge-1/0/1_10.0.1.112	
ge-1/0/1	ge	10.0.1.113	Nge_10.10.0.113	ge-1/0/1_10.0.1.113	
ge-1/0/1	ge	10.0.1.114	Nia_10.10.0.114	ge-1/0/1_10.0.1.114	

Administration **Schemas** **Information**

Table: device_interfaces

Columns: INT_NAME varchar(255)

Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

Source Column	Dest Column
<input checked="" type="checkbox"/> INT_NAME	INT_NAME
<input checked="" type="checkbox"/> INT_DESC	INT_DESC
<input checked="" type="checkbox"/> INT_TYPE	INT_TYPE
<input checked="" type="checkbox"/> INT_IP	INT_IP
<input checked="" type="checkbox"/> DEV_ID	DEV_ID
<input checked="" type="checkbox"/> INT_ID	INT_ID

Result Grid

INT_NAME	INT_DESC	INT_TYPE	INT_IP	DEV_ID	INT_ID
ge-1/0/1	ge	10.0.1.1	Abu_10.10.0.1	ge-1/0/1_10.0.1.1	
ge-1/0/1	ge	10.0.1.10	Ant_10.10.0.10	ge-1/0/1_10.0.1.10	
ge-1/0/1	ge	10.0.1.100	Map_10.10.0.100	ge-1/0/1_10.0.1.100	
ge-1/0/1	ge	10.0.1.101	Map_10.10.0.101	ge-1/0/1_10.0.1.101	
ge-1/0/1	ge	10.0.1.102	Mas_10.10.0.102	ge-1/0/1_10.0.1.102	
ge-1/0/1	ge	10.0.1.103	Mba_10.10.0.103	ge-1/0/1_10.0.1.103	
ge-1/0/1	ge	10.0.1.104	Mn_10.10.0.104	ge-1/0/1_10.0.1.104	
ge-1/0/1	ge	10.0.1.105	Mog_10.10.0.105	ge-1/0/1_10.0.1.105	
ge-1/0/1	ge	10.0.1.106	Món_10.10.0.106	ge-1/0/1_10.0.1.106	
ge-1/0/1	ge	10.0.1.107	Mon_10.10.0.107	ge-1/0/1_10.0.1.107	
ge-1/0/1	ge	10.0.1.108	Mer_10.10.0.108	ge-1/0/1_10.0.1.108	
ge-1/0/1	ge	10.0.1.109	Mos_10.10.0.109	ge-1/0/1_10.0.1.109	
ge-1/0/1	ge	10.0.1.11	Ap_10.10.0.11	ge-1/0/1_10.0.1.11	
ge-1/0/1	ge	10.0.1.110	Na_10.10.0.110	ge-1/0/1_10.0.1.110	
ge-1/0/1	ge	10.0.1.111	Ne_10.10.0.111	ge-1/0/1_10.0.1.111	
ge-1/0/1	ge	10.0.1.112	Nas_10.10.0.112	ge-1/0/1_10.0.1.112	
ge-1/0/1	ge	10.0.1.113	Nge_10.10.0.113	ge-1/0/1_10.0.1.113	
ge-1/0/1	ge	10.0.1.114	Nia_10.10.0.114	ge-1/0/1_10.0.1.114	

< Back Next > Cancel

Vistas

Se crean las distintas vistas para ser usadas por el personal con permiso de solo lectura, y así no trabajen en las tablas raw



Screenshot of a database management system (DBMS) interface showing the creation of a view named `v_devices_type_list`.

The Navigator pane shows the schema structure:

- SCHEMAS:
 - Views
 - v_devices_inventory_count
 - v_devices_inventory_list
 - v_devices_type_list
 - v_module_inventory_count
 - v_module_inventory_list
 - v_module_type_list
 - Stored Procedures
 - Functions
- sakila
- sys
- world

Administration Schemas Information

View: `v_devices_type_list`

Columns:

Modelo	Tipo	Software	Numero_de_Parte
C3850	CATALYST	V23	WS-C3850-48F-S
C9200	CATALYST	V23	C9200-48P-E
C9300	CATALYST	V22	C9300-48P-A
C9500	CATALYST	V26	C9500-40X-A
N9504	NEXUS	V27	N9K-C9504
N9516	NEXUS	V27	N9K-C9516

Result Grid Filter Rows: Export: Wrap Cell Content: Read Only

Screenshot of a database management system (DBMS) interface showing the creation of a view named `v_devices_inventory_count`.

The Navigator pane shows the schema structure:

- SCHEMAS:
 - Views
 - v_devices_inventory_count
 - v_devices_inventory_list
 - v_devices_type_list
 - v_module_inventory_count
 - v_module_inventory_list
 - v_module_type_list
 - Stored Procedures
 - Functions
- sakila
- sys
- world

Administration Schemas Information

View: `v_devices_inventory_count`

Columns:

Modelos	Numero_de_Modelos	Familia	Numero_de_parte	Software
C9300	35	CATALYST	C9300-48P-A	V22
C3850	33	CATALYST	WS-C3850-48F-S	V23
C9200	32	CATALYST	C9200-48P-E	V23
N9516	32	NEXUS	N9K-C9516	V27
N9504	31	NEXUS	N9K-C9504	V27
C9500	26	CATALYST	C9500-40X-A	V26

Result Grid Filter Rows: Export: Wrap Cell Content: Read Only

Vistas



Schemas

v_devices_type_list v_devices_inventory_count v_devices_inventory_list

1 • SELECT * FROM network_inventory.v_devices_inventory_list;

Result Grid | Filter Rows: Export: Wrap Cell Content: Read Only

Host_Name	Ubicacion	IP	Serial	Modelo	Familia	Numero_de_parte	Software
Adi_10.10.0.4	Adis Abeba	10.10.0.4	gnD3nGRd	C3850	CATALYST	WS-C3850-48F-S	V23
Amm_10.10.0.6	Ammán	10.10.0.6	deRrr94	C3850	CATALYST	WS-C3850-48F-S	V23
Áms_10.10.0.7	Ámsterdam	10.10.0.7	BscdUpNB	C3850	CATALYST	WS-C3850-48F-S	V23
Asj_10.10.0.13	Asjabad	10.10.0.13	cya8wKSV	C3850	CATALYST	WS-C3850-48F-S	V23
Asu_10.10.0.15	Asunción	10.10.0.15	nx4INVNL	C3850	CATALYST	WS-C3850-48F-S	V23
Bas_10.10.0.24	Basseterre	10.10.0.24	5HWRCYS0	C3850	CATALYST	WS-C3850-48F-S	V23
Bra_10.10.0.36	Brazzaville	10.10.0.36	wryKoZ7Y	C3850	CATALYST	WS-C3850-48F-S	V23
Bue_10.10.0.41	Buenos Aires	10.10.0.41	mmuNbRZn	C3850	CATALYST	WS-C3850-48F-S	V23
Buy_10.10.0.66	Buyumbura	10.10.0.66	ZJUEPSYR	C3850	CATALYST	WS-C3850-48F-S	V23

View: v_devices_inventory_list

Columns:

Host_Name	varchar(30)
Ubicacion	varchar(30)
IP	varchar(16)
Serial	varchar(30)
Modelo	varchar(30)
Familia	varchar(30)
Numero_de_parte	varchar(30)
Software	varchar(30)

Information

Schemas

v_devices_type_list v_devices_inventory_count v_devices_inventory_list v_module_type_list

1 • SELECT * FROM network_inventory.v_module_type_list;

Result Grid | Filter Rows: Export: Wrap Cell Content: Read Only

Modelo	Tipo	Software	Numero_de_Parte
F2300-NM-H	Card	v33	hBmkLqsc
F8300-NM-J	Card	v23	HBjsPBK
F9300-NM-H	Card	v33	xbogJ8JM
G7300-NM-M	Card	v23	qlEbvy43
G7300-NM-Q	Card	v33	3EXSoGkQ
G8300-NM-A	Card	v33	es5CcEds
H9300-NM-W	Card	v23	7wRLU7h6
J3300-NM-M	Card	v23	PpSdvcBQ
K8300-NM-F	Card	v33	T2VUm4Pw

View: v_module_type_list

Columns:

Modelo	varchar(30)
Tipo	varchar(15)
Software	varchar(30)
Numero_de_Parte	varchar(30)

Information

Vistas



Navigator

SCHEMAS

Views

- v_devices_inventory_count
- v_devices_inventory_list
- v_devices_type_list
- v_module_inventory_count
- v_module_inventory_list
- v_module_type_list

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

View: v_module_inventory_count

Columns:

Modelos	Numero_de_Modelos	Tipo	Software	Numero_de_Parte
F8300-NM-J	10	Card	v23	HBj8sPBK
G7300-NM-M	10	Card	v23	qLEbyy43
H9300-NM-W	10	Card	v23	7wRLU7h6
J3300-NM-M	10	Card	v23	Pp5dvcbQ
M5300-NM-B	10	Card	v31	z457YkbQ
N3300-NM-D	10	Card	v23	MpMkzePz
Q9300-NM-J	10	Card	v23	jNIJKMfh
S5300-NM-Y	10	Card	v23	xNV9dn7V
T6300-NM-S	10	Card	v23	yynuk7gX

antory_count 1 ×

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Modelos Numero_de_Modelos Tipo Software Numero_de_Parte

Output

Action Output

Time Action Message

1 Read Only

Navigator

SCHEMAS

Views

- v_devices_inventory_count
- v_devices_inventory_list
- v_devices_type_list
- v_module_inventory_count
- v_module_inventory_list
- v_module_type_list

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

View: v_module_inventory_list

Columns:

Modelo_del_modulo	Serial_modelo	Host_Name	Ubicacion	IP	Serial_equipo	Modelo_de
F2300-NM-H	72Jxk7DzZxwPv	Arg_10.10.0.12	Argel	10.10.0.12	dhjRQaLu	C9500
F2300-NM-H	CmGXmdWRkABhhd	Ciu_10.10.0.32	Ciudad del Cabo	10.10.0.32	yAbzwQ3n	C9200
F2300-NM-H	2D55wxajK7WMSj	Dak_10.10.0.52	Dakar	10.10.0.52	3gfk8ulf	C9500
F2300-NM-H	FepCwB8s6Mpfeh	Jar_10.10.0.72	Jartum	10.10.0.72	iGbrtwZ5	N9516
F2300-NM-H	sTJcTTNhGvSCB	Lux_10.10.0.92	Luxemburgo	10.10.0.92	ER3YzWj4	C9300
F2300-NM-H	5nShNRg9jf2C	Nas_10.10.0.112	Nasáu	10.10.0.112	FfY8LxmN	N9504
F2300-NM-H	GgcwqlKX8lNyC8	Pra_10.10.0.132	Praga	10.10.0.132	ntrqoUdu	C9300
F2300-NM-H	UzVnLyNTJw6y9	San_10.10.0.152	Santo Tomé	10.10.0.152	WnyXDnxR	N9516

inventory_list1 ×

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Modelo_del_modulo Serial_modelo Host_Name Ubicacion IP Serial_equipo Modelo_de

Output

Action Output

Time Action Message

116 15:55:42 SELECT * FROM network_invento... 6 row(s) returned

1 Read Only

Pruebas Funciones

La siguiente función cuenta la cantidad de interfaces por equipo.

The screenshot shows the SSMS interface with the following details:

- Navigator:** Shows the database schema. Under the **network_inventory** schema, there is a **Functions** folder containing two functions: **FUNC_BUSQUEDA_EQUIPO** and **FUNC CUENTA_INTERFAZ**.
- SQL Editor:** The current query window is titled **SQL File 31***. It contains the following T-SQL code:

```
7  READS SQL DATA
8  BEGIN
9      declare CUENTA INT;
10     SELECT count(device_interfaces.INT_ID)
11        INTO CUENTA
12        from device
13           inner join device_interfaces on device_interfaces.DEV_ID = device.DEV_ID
14             where device.DEV_ID = ID;
15     RETURN CUENTA;
16  END
17  //
18  DELIMITER ;
19
20 • select FUNC CUENTA_INTERFAZ ("Abu_10.10.0.1") AS ID_EQUIPO
21
```
- Result Grid:** Below the editor, the results of the last query (line 20) are displayed in a grid:

ID_EQUIPO
2
- Information Bar:** Shows the function name **Function: FUNC_BUSQUEDA_EQUIPO**.
- Toolbars and Menus:** Standard SSMS toolbar and menu items are visible at the top and right side of the window.

Pruebas Funciones

La siguiente función esta hecha para que los usuarios que solo conocen la IP del equipo puedan traerla información del equipo y ver si existe en la tabla

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **network_inventory** schema expanded, revealing tables, views, stored procedures, and functions. Functions include **FUN_BUSQUEDA_EQUIPO** and **FUN CUENTA_INTERFAZ**.
- SQL Editor:** The current tab is **SP Busqueda de equipo***. The code is as follows:

```
1  DELIMITER //
2  • CREATE PROCEDURE SP_NETWORK_INVENTORY_BY_IP (IN IP VARCHAR(60))
3  BEGIN
4      -- procedure de busqueda de equipos
5      SELECT * FROM network_inventory.device
6      WHERE DEV_IP = IP;
7  END
8  //
9  Delimiter ;
10
11 • call SP_NETWORK_INVENTORY_BY_IP ("10.10.0.1");
12
13 • SELECT * FROM network_inventory.log_inventario_equipo;
```

- Result Grid:** Shows the results of the query `call SP_NETWORK_INVENTORY_BY_IP ("10.10.0.1");`. The data is:

DEV_MOD	DEV_HOST_NAME	DEV_LOCATION	DEV_IP	DEV_SERIAL_NO	DEV_ID
N9504	Abu_10.10.0.1	Abu Dhabi	10.10.0.1	aqfIKqNS	Abu_10.10.0.1

Muestra de SP funcionando

Se realiza el SP para que los usuarios puedan ordenar por hostname y/o modelos, ya que son las dos columnas mas utilizadas.

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The "Schemas" pane shows the database structure. It includes the "device_type", "module_device", "module_type" tables under "device"; a "Views" section; a "Stored Procedures" section containing "SP_NETWORK_INVENTORY_BY_IP" and "call SP_NETWORK_INVENTORY_ORDER('DEV_HOST_NAME')"; and a "Functions" section with "FUN_BUSQUEDA_EQUIPO" and "FUN CUENTA_INTERFAZ". Other schemas listed are "sakila", "sys", and "world".
- SQL Editor:** The main editor window contains the following SQL code:

```
-- SELECT query_orden;
-- ejecutar la consulta

PREPARE ejecutar FROM @query_orden;

EXECUTE ejecutar;

DEALLOCATE PREPARE ejecutar;
END
//
DELIMITER ;
```

Line 26: `call SP_NETWORK_INVENTORY_ORDER ('DEV_HOST_NAME')`
Line 27: `call SP_NETWORK_INVENTORY_ORDER ('DEV_MOD')`
- Result Grid:** The result grid displays the output of the stored procedure. The columns are: DEV_MOD, DEV_HOST_NAME, DEV_LOCATION, DEV_IP, DEV_SERIAL_NO, and DEV_ID. The data includes various hostnames and their corresponding details.
- Output:** The "Action Output" tab shows the log of actions taken:

#	Time	Action	Message	Duration / Fetch
140	23:55:28	call SP_NETWORK_INVENTORY_ORDER (DEV_HOST_NAME)	189 row(s) returned	0.000 sec / 0.000 sec
141	23:55:55	call SP_NETWORK_INVENTORY_ORDER (DEV_MOD)	189 row(s) returned	0.000 sec / 0.000 sec

Muestra de SP funcionando

Se realiza el SP para que los usuarios puedan traer la información del equipo y su inventario para el uso de auditorias

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The left pane displays the database schema with the "SP_NETWORK_INVENTORY_BY_IP" stored procedure highlighted under the "Stored Procedures" section.
- Query Editor:** The main pane contains the SQL code for the stored procedure and a call to it.

```
DELIMITER //
CREATE PROCEDURE SP_NETWORK_INVENTORY_BY_IP (IN IP VARCHAR(50))
BEGIN
    -- procedure de busqueda de equipos
    SELECT * FROM network_inventory.device
    WHERE DEV_IP = IP;
END //
Delimiter ;
```

```
call SP_NETWORK_INVENTORY_BY_IP ("10.10.0.1");
```
- Result Grid:** Below the query editor, the result grid shows one row of data from the stored procedure execution:

DEV_MOD	DEV_HOST_NAME	DEV_LOCATION	DEV_IP	DEV_SERIAL_NO	DEV_ID
N9504	Abu_10.10.0.1	Abu Dhabi	10.10.0.1	aqfHKqNS	Abu_10.10.0.1
- Output:** The bottom pane displays the execution log for the stored procedure calls:

#	Time	Action	Message	Duration / Fetch
140	23:55:28	call SP_NETWORK_INVENTORY_ORDER (DEV_HOST_NAME)	189 row(s) returned	0.000 sec / 0.000 sec
141	23:55:55	call SP_NETWORK_INVENTORY_ORDER (DEV_MOD)	189 row(s) returned	0.000 sec / 0.000 sec

Muestra de SP funcionando

Se crea un SP para la inserción de datos

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Local instance MySQL80 contains Indexes, Foreign Keys, Triggers, device_interfaces, device_type, module_device, module_type, Views, Stored Procedures (SP_NETWORK_INVENTORY_BY_IP, SP_NETWORK_INVENTORY_ORDER), and Functions.
- Script:** SP INSERT* contains the following code:

```
6
7
8 BEGIN
9     INSERT INTO device VALUES
10        (DEV_MOD, DEV_HOST_NAME, DEV_LOCATION, DEV_IP, DEV_SERIAL_NO, DEV_ID);
11
12 END
13 //
14 DELIMITER ;
15
16 • CALL SP_insert_device("N9504", "Caracas_192.168.1.1", "caracas", "192.168.1.1", "xckdlep", "Caracas_192.168.1.1");
17 ;
18
19 • call SP_NETWORK_INVENTORY_ORDER ('DEV_HOST_NAME')
```
- Result Grid:** Shows a table with columns: DEV_MOD, DEV_HOST_NAME, DEV_LOCATION, DEV_IP, DEV_SERIAL_NO, and DEV_ID. The data includes rows for various cities and their corresponding details.
- Action Output:** Shows two log entries:

#	Time	Action	Message	Duration / Fetch
199	00:24:53	call SP_NETWORK_INVENTORY_ORDER (DEV_HOST_NAME)	1 row(s) returned	0.000 sec / 0.000 sec
200	00:24:53	call SP_NETWORK_INVENTORY_ORDER (DEV_HOST_NAME)	190 row(s) returned	- / 0.000 sec

Creación de Tabla log mas 2 triggers

Se crean los triggers para el control de log de ingreso, modificación o eliminación de cualquier dato en la tabla de equipos



Screenshot of MySQL Workbench showing the creation of a log table and two triggers.

Navigator: Local instance MySQL80 → Schemas → Tables → device → Triggers → TR_LOG_EQUIPOS_IN, TR_LOG_EQUIPOS_DELETE.

SQL Editor:

```
1 ● use network_inventory;
2 -- Crear tabla para el control de modificaciones del inventario de dispositivos
3 ● CREATE TABLE LOG_INVENTARIO_EQUIPO (
4     ID_CHANGE INT unsigned AUTO_INCREMENT PRIMARY KEY,
5     DEV_ID VARCHAR(50),
6     DEV_SERIAL_NO VARCHAR(30),
7     TIPE_CHANGE VARCHAR(30),
8     USER_DB VARCHAR(50),
9     MODIFY_DATE TIMESTAMP NOT NULL
10 );
11
12 -- Registro de log de nuevos dispositivos
13 DELIMITER //
14 ● CREATE TRIGGER TR_LOG_EQUIPOS_IN
15 AFTER INSERT ON DEVICE
16 FOR EACH ROW
17
18 BEGIN
19     INSERT INTO LOG_INVENTARIO_EQUIPO (ID_CHANGE, DEV_ID, DEV_SERIAL_NO, TIPE_CHANGE, USER_DB, MODIFY_DATE)
20     VALUES (
21         NULL , NEW.DEV_ID, NEW.DEV_SERIAL_NO, "Insert", SYSTEM_USER(), current_timestamp()
22     );
23 END //
24
25
26 -- Registro de log de eliminacion de dispositivos
27 DELIMITER //
28 ● CREATE TRIGGER TR_LOG_EQUIPOS_DELETE
29 BEFORE DELETE ON DEVICE
30 FOR EACH ROW
31
32 BEGIN
33     INSERT INTO LOG_INVENTARIO_EQUIPO (DEV_ID, DEV_SERIAL_NO, TIPE_CHANGE, USER_DB, MODIFY_DATE)
34     VALUES (
35         OLD.DEV_ID, OLD.DEV_SERIAL_NO, "DELETED", SYSTEM_USER(), current_timestamp()
36     );
37 END //
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
```

Output:

#	Time	Action	Message
70	22:06:39	INSERT INTO device ('DEV_MOD','DEV_HOST_NAME','DEV_LOCATION','DEV_IP','DEV_SERIAL_NO','DEV_ID') V...	1 row(s) affected
71	22:06:42	SELECT * FROM network_inventory.log_inventario_equipo LIMIT 0, 400	9 row(s) returned

Pruebas realizadas triggers

The screenshot shows the MySQL Workbench interface. On the left, the 'Tables' section of the object browser is expanded, showing the 'device' table and its associated triggers ('TR_LOG_EQUIPOS_IN' and 'TR_LOG_EQUIPOS_DELETE'), as well as the 'log_inventario_equipo' table and its columns ('ID_CHANGE', 'DEV_ID', 'DEV_SERIAL_NO', 'TIPE_CHANGE', 'USER_DB', 'MODIFY_DATE'). The 'Administration' tab is selected at the bottom of the browser.

In the main pane, a query window displays the results of the following SQL statement:

```
1 •  SELECT * FROM network_inventory.log_inventario_equipo;
```

The result grid shows 10 rows of data, each representing a log entry for a device change. The columns are: ID_CHANGE, DEV_ID, DEV_SERIAL_NO, TIPE_CHANGE, USER_DB, and MODIFY_DATE. The data includes various device IDs and serial numbers, along with insert and delete operations, and timestamps ranging from 2023-01-16 22:02:14 to 2023-01-16 22:06:39.

```
-- Prueba de nuevo registro
INSERT INTO device ('DEV_MOD', 'DEV_HOST_NAME', 'DEV_LOCATION', 'DEV_IP', 'DEV_SERIAL_NO', 'DEV_ID') VALUES ('C9300', 'test_192.168.20.1', 'test', '192.168.20.1', 'v5mjo2YS', 'test_192.168.20.1');
INSERT INTO device ('DEV_MOD', 'DEV_HOST_NAME', 'DEV_LOCATION', 'DEV_IP', 'DEV_SERIAL_NO', 'DEV_ID') VALUES ('C9300', 'test_192.168.20.2', 'test', '192.168.20.2', 'v5Pj02YS', 'test_192.168.20.2');
INSERT INTO device ('DEV_MOD', 'DEV_HOST_NAME', 'DEV_LOCATION', 'DEV_IP', 'DEV_SERIAL_NO', 'DEV_ID') VALUES ('C9300', 'test_192.168.20.3', 'test', '192.168.20.3', 'v5PLo2YS', 'test_192.168.20.3');

-- prueba de eliminacion de registro
DELETE FROM device where DEV_ID = "test_192.168.20.1" ;
DELETE FROM device where DEV_ID = "test_192.168.20.2" ;
DELETE FROM device where DEV_ID = "test_192.168.20.3" ;
```



Implementación de sentencias

I. Creación de usuario con acceso solo lectura a todas las tablas y muestra de error al querer editar un registro de una tabla

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the schema structure under 'SCHEMAS'. The 'network_inventory' schema is selected, showing tables like device, device_interfaces, device_type, log_inventario_equipo, module_device, and module_type. The 'Tables' section is expanded. On the right, the 'Query 1' editor contains the following SQL code:

```
1 drop user vs_devices;
2
3 • create user 'vs_devices' identified by '1234';
4 -- solo permiso de seleccion / lectura para todas las tablas
5 • grant select on *.* to 'vs_devices';
6 • show grants for 'vs_devices';
7 -- ver permisos otorgados
8 • drop user vs_devices;
9
10 -- Desde el usuario vs_devices
11 • select * from network_inventory.device;
12 • update network_inventory.device
   set DEV_HOST_NAME = 'hola mundo'
13 where DEV_ID = 'Abu_10.10.0.1'
```

The 'Output' pane at the bottom shows the execution results. A single row is listed in the 'Action Output' table:

#	Time	Action
1	21:27:51	update network_inventory.device set DEV_HOST_NAME = 'hola mundo' where DEV_ID = 'Abu_10.10.0.1'

The 'Message' column indicates an error: "Error Code: 1142. UPDATE command denied to user 'vs_devices'@localhost for table 'device'". The 'Duration / Fetch' column shows "0.000 sec".

2. Creación de usuario con acceso de AMV alta modificación y vista pero no de eliminación de datos a todas las tablas.

Se visualiza la modificación exitosa de un registro de una tabla, mas error al querer eliminar el registro.

The screenshot shows the MySQL Workbench interface. In the top left, the Navigator pane displays the Schemas: `gamers_model`, `network_inventory`, `sakila`, `sys`, and `world`. The main area is titled "Query 1" and contains the following SQL script:

```

1 • create user 'am_pepito' identified by '1234';
2 -- garantizar permisos de lectura Alta y modificacion de datos sin poder eliminar
3 • grant select, insert, update on *.* to 'am_pepito';
4 • show grants for 'am_pepito';
5 -- drop user am_devices;
6
7 • select * from network_inventory.device;
8 • update network_inventory.device
9     set DEV_HOST_NAME = 'hola mundo'
10 where DEV_ID = 'Abu_10.10.0.1';
11 • select * from network_inventory.device;

```

The "Result Grid" tab shows the results of the `SELECT` query:

DEV_MOD	DEV_HOST_NAME	DEV_LOCATION	DEV_IP	DEV_SERIAL_NO	DEV_ID
C9500	Api_10.10.0.11	Apia	10.10.0.11	QInDyqZq	Api_10.10.0.11
C9500	Arg_10.10.0.12	Argel	10.10.0.12	dhRQaLu	Arg_10.10.0.12
C3850	Asi_10.10.0.13	Ajshabad	10.10.0.13	cyABwKSV	Asi_10.10.0.13
C9200	Asm_10.10.0.14	Asmara	10.10.0.14	6tuvc6n	Asm_10.10.0.14
C3850	Asu_10.10.0.15	Asuncion	10.10.0.15	nx#HlwLs	Asu_10.10.0.15

The "Output" tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	21:37:27	select * from network_inventory.device	193 row(s) returned	0.000 sec / 0.000 sec

The screenshot shows the MySQL Workbench interface with the same schema navigation and "Query 1" window. The SQL script now includes a DELETE statement:

```

1 • create user 'am_pepito' identified by '1234';
2 -- garantizar permisos de lectura Alta y modificacion de datos sin poder eliminar
3 • grant select, insert, update on *.* to 'am_pepito';
4 • show grants for 'am_pepito';
5 -- drop user am_devices;
6
7 • select * from network_inventory.device;
8 • update network_inventory.device
9     set DEV_HOST_NAME = 'hola mundo'
10 where DEV_ID = 'Abu_10.10.0.1';
11 • select * from network_inventory.device;
12
13 • delete from network_inventory.device
14 where DEV_HOST_NAME = 'hola mundo';

```

The screenshot shows the MySQL Workbench interface with the same schema navigation and "Query 1" window. The "Output" tab shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
3	21:39:45	select * from network_inventory.device	193 row(s) returned	0.000 sec / 0.000 sec
4	21:40:59	delete from network_inventory.device where DEV_HOST_NAME = hol...	Error Code: 1142. DELETE command denied to user 'am_pepito'@local...	0.000 sec

The screenshot shows the MySQL Workbench interface with the same schema navigation and "Query 1" window. The "Output" tab shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
2	21:39:45	update network_inventory.device set DEV_HOST_NAME = hola mund...	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.000 sec
3	21:39:45	select * from network_inventory.device	193 row(s) returned	0.000 sec

Sentencias del sublenguaje TCL

I. Inserción de nuevos registros con dos savepoint.

The screenshot shows a database management interface with three result grids and an output log at the bottom.

Result Grid 1: Shows the initial state of the device_type table.

DEV_MOD	DEV_TYPE	DEV_OS_VERSION	DEV_PART_NO
C3850	CATALYST	V23	WS-C3850-48F-S
C9200	CATALYST	V23	C9200-48P-E
C9300	CATALYST	V22	C9300-48P-A
C9500	CATALYST	V26	C9500-40X-A
N9504	NEXUS	V27	N9K-C9504

Result Grid 2: Shows the state of the device_type table after the first savepoint and rollback.

DEV_MOD	DEV_TYPE	DEV_OS_VERSION	DEV_PART_NO
C3850	CATALYST	V23	WS-C3850-48F-S
C9200	CATALYST	V23	C9200-48P-E
C9300	CATALYST	V22	C9300-48P-A
C9500	CATALYST	V26	C9500-40X-A
N9504	NEXUS	V27	N9K-C9516
N9516	NEXUS	V27	N9K-C9516
T2301	WIFI	VU	VU-T2301
T2302	WIFI	VU	VU-T2302
T2303	WIFI	VU	VU-T2303
T2304	WIFI	VU	VU-T2304
T2305	WIFI	VU	VU-T2305
HULL	HULL	HULL	HULL

Result Grid 3: Shows the state of the device_type table after the second savepoint and rollback.

DEV_MOD	DEV_TYPE	DEV_OS_VERSION	DEV_PART_NO
C3850	CATALYST	V23	WS-C3850-48F-S
C9200	CATALYST	V23	C9200-48P-E
C9300	CATALYST	V22	C9300-48P-A
C9500	CATALYST	V26	C9500-40X-A
N9504	NEXUS	V27	N9K-C9504
N9516	NEXUS	V27	N9K-C9516
T2301	WIFI	VU	VU-T2301
T2302	WIFI	VU	VU-T2302
T2303	WIFI	VU	VU-T2303
T2304	WIFI	VU	VU-T2304
T2305	WIFI	VU	VU-T2305
HULL	HULL	HULL	HULL

Action Output:

#	Time	Action	Message	Duration / Fetch
47	22:19:04	INSERT INTO device_type (DEV_MOD, DEV_TYPE, DEV_OS_VERSI...)	1 row(s) affected	0.000 sec
48	22:19:04	savepoint snap3	0 row(s) affected	0.000 sec
49	22:19:17	release savepoint snap2	0 row(s) affected	0.000 sec
50	22:19:20	rollback to snap2	Error Code: 1305. SAVEPOINT snap2 does not exist	0.000 sec
51	22:19:25	rollback to snap1	0 row(s) affected	0.000 sec
52	22:19:29	SELECT * FROM network_inventory.device_type	11 row(s) returned	0.000 sec / 0.000 sec

2. Eliminación de productos con savepoint y commit para la eliminación definitiva

Screenshot of Oracle SQL Developer showing a transaction script for deleting network interfaces and then restoring them.

```

1 •   SELECT * FROM network_inventory.device_interfaces;
2 •   start transaction;
3 •   delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.1';
4 •   delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.10';
5 •   delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.103';
6 •   savepoint snapd1;
7 •   delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.100';
8 •   delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.102';
9 •   delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.101';
10 •  savepoint snapd2;
11
12 •  rollback to snapd1;
13 •  rollback;
14 •  commit;
15
16 -- insertar de vuelta lo eliminado
17 •  INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.1','Abu_10.10.0.1','ge-1/0/1_10.0.1.1');
18 •  INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.10','Ant_10.10.0.10','ge-1/0/1_10.0.1.10');
19 •  INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.103','Mba_10.10.0.103','ge-1/0/1_10.0.1.103');
20 •  INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.100','Map_10.10.0.100','ge-1/0/1_10.0.1.100');
21 •  INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.102','Mas_10.10.0.102','ge-1/0/1_10.0.1.102');
22 •  INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.101','Mas_10.10.0.101','ge-1/0/1_10.0.1.101');
23

```

Result Grid:

INT_NAME	INT_DESC	INT_TYPE	INT_IP	DEV_ID	INT_ID
ge-1/0/1	ge	10.0.1.1	Abu_10.10.0.1	ge-1/0/1_10.0.1.1	

SQL Script:

```

1 •  SELECT * FROM network_inventory.device_interfaces;
2 •  start transaction;
3 •  delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.1';
4 •  delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.10';
5 •  delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.103';
6 •  savepoint snapd1;
7 •  delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.100';
8 •  delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.102';
9 •  delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.101';
10 • savepoint snapd2;
11
12 • rollback to snapd1;
13 -- rollback;
14 • commit;
15
16 -- insertar de vuelta lo eliminado
17 • INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.1','Abu_10.10.0.1','ge-1/0/1_10.0.1.1');
18 • INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.10','Ant_10.10.0.10','ge-1/0/1_10.0.1.10');
19 • INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.103','Mba_10.10.0.103','ge-1/0/1_10.0.1.103');
20 • INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.100','Map_10.10.0.100','ge-1/0/1_10.0.1.100');
21 • INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.102','Mas_10.10.0.102','ge-1/0/1_10.0.1.102');
22 • INSERT INTO device_interfaces VALUES ('ge-1/0/1','ge','10.0.1.101','Mas_10.10.0.101','ge-1/0/1_10.0.1.101');
23

```

Result Grid:

INT_NAME	INT_DESC	INT_TYPE	INT_IP	DEV_ID	INT_ID
ge-1/0/2	ge	10.0.2.96	Mal_10.10.0.96	ge-1/0/2_10.0.2.96	
ge-1/0/2	ge	10.0.2.95	Mal_10.10.0.95	ge-1/0/2_10.0.2.95	
ge-1/0/2	ge	10.0.2.94	Maj_10.10.0.94	ge-1/0/2_10.0.2.94	
ge-1/0/2	ge	10.0.2.93	Mad_10.10.0.93	ge-1/0/2_10.0.2.93	
ge-1/0/2	ge	10.0.2.92	Lux_10.10.0.92	ge-1/0/2_10.0.2.92	
ge-1/0/2	ge	10.0.2.91	Lus_10.10.0.91	ge-1/0/2_10.0.2.91	
ge-1/0/2	ge	10.0.2.90	Lua_10.10.0.90	ge-1/0/2_10.0.2.90	
ge-1/0/2	ge	10.0.2.89	Est_10.10.0.9	ge-1/0/2_10.0.2.89	
ge-1/0/2	ge	10.0.2.88	Lon_10.10.0.89	ge-1/0/2_10.0.2.88	
ge-1/0/2	ge	10.0.2.87	Lom_10.10.0.88	ge-1/0/2_10.0.2.88	
ge-1/0/2	ge	10.0.2.87	Liu_10.10.0.87	ge-1/0/2_10.0.2.87	

Action Output:

#	Time	Action	Message	Duration / Fetch
72	22:36:27	delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.1'	1 row(s) affected	0.000 sec
73	22:36:27	delete from network_inventory.device_interfaces where INT_ID = 'ge-1/0/1_10.0.1.10'	1 row(s) affected	0.000 sec
74	22:36:27	savepoint snapd2	0 row(s) affected	0.000 sec
75	22:36:27	commit	0 row(s) affected	0.016 sec
76	22:36:29	SELECT * FROM network_inventory.device_interfaces	372 row(s) returned	0.000 sec / 0.000 sec
77	22:36:36	rollback to snapd1	Error Code: 1305. SAVEPOINT snapd1 does not exist	0.000 sec

BACKUP

The screenshot shows the MySQL Workbench Data Export interface. On the left, a sidebar navigation includes sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup). Below these are tabs for Administration and Schemas, with 'Administration' currently selected. A message 'No object selected' is displayed. The main area is titled 'Local instance MySQL80 Data Export'. Under 'Tables to Export', several databases and schemas are listed: Exp... Schema (gamblers_model, incidents, network_inventory, sakila, sys, world) and Exp... Schema Objects (device_interfaces, device_type, log_inventario_equipo, module_device, module_type, v_devices_inventory_count, v_devices_inventory_list, v_devices_type_list, v_module_inventory_count, v_module_inventory_list, v_module_type_list). A message '12 tables selected' is shown. Below this are sections for 'Objects to Export' (Dump Stored Procedures and Functions, Dump Events, Dump Triggers), 'Export Options' (Export to Dump Project Folder or Self-Contained File, Create Dump in a Single Transaction), and 'Output' (Action Output table showing a single row with ID 1, Time, and Action). A 'Start Export' button is at the bottom right.

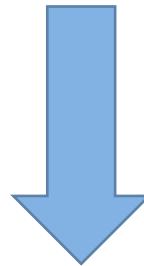
The screenshot shows the MySQL Workbench Data Export interface after the export process has completed. The title bar says 'Local instance MySQL80 Data Export'. The 'Object Selection' tab is active, showing 'Export Progress' and 'Export Completed' status. The 'Status' section indicates '12 of 12 exported'. The 'Log' section displays the command history and completion message:

```
18:41:41 Dumping network_inventory (all tables)
Running: mysqldump.exe --defaults-file="C:\Users\jtroc\AppData\Local\Temp\tmp3bz071vp.cnf" --host=localhost --port=3306 --default-character-set=utf8 --user=root --protocol=tcp --routines --events "network_inventory"
18:41:41 Export of C:\Users\jtroc\Desktop\SQL\backup\backup-jtrociz.sql has finished
```

Tecnologías utilizadas

MySQL Workbench
Excel
Draw.io
PDF
GitHub
VisualStudio Code
Word
Note ++

Data Base File



<https://github.com/jtroconiz/SQL>

