
Aranda Device Management

The Aranda DEVICE MANAGEMENT views of version 9.6.1 allow the client to validate in detail the information generated in the company's device and IT asset management processes, through queries of the ADM views.

When are the views consulted?

If the user or administrator needs to obtain relevant information about their assets such as knowing the information of a device according to certain conditions, generating reports associated with the use of the software or the number of licenses or policies installed, they can make queries of Aranda DEVICE MANAGEMENT ADM views.

How do I query the views?

1. The user, according to their business objective, will be able to perform ADM view queries in three ways:

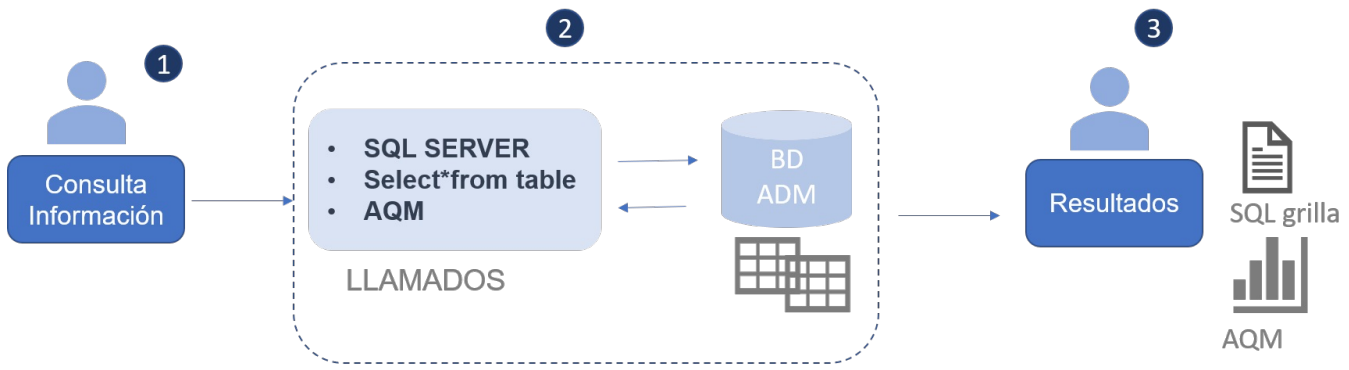
- A query through SQL Server
- A query through Select*fromTable
- Consultation with Aranda QUERY MANAGER

Each consultation varies according to the required criteria. The user sets the query conditions by selecting the existing customer views and establishing the required relationships.

> Example: The administrator can consult general information about their company's devices such as the total number of devices in the organization and additionally know the software they have installed on the device or the use of this software at a time of the day.

2. The result of the query can generate a list in the form of a table in SQL, a grid with the Select and a graph in Aranda QUERY MANAGER. 3. The generated reports can be exported in various formats.

If the query was generated in SQL Server, you can export the information in CSV and TXT formats. If the query was generated in AQM, you can export the information in XLS and CSV formats.

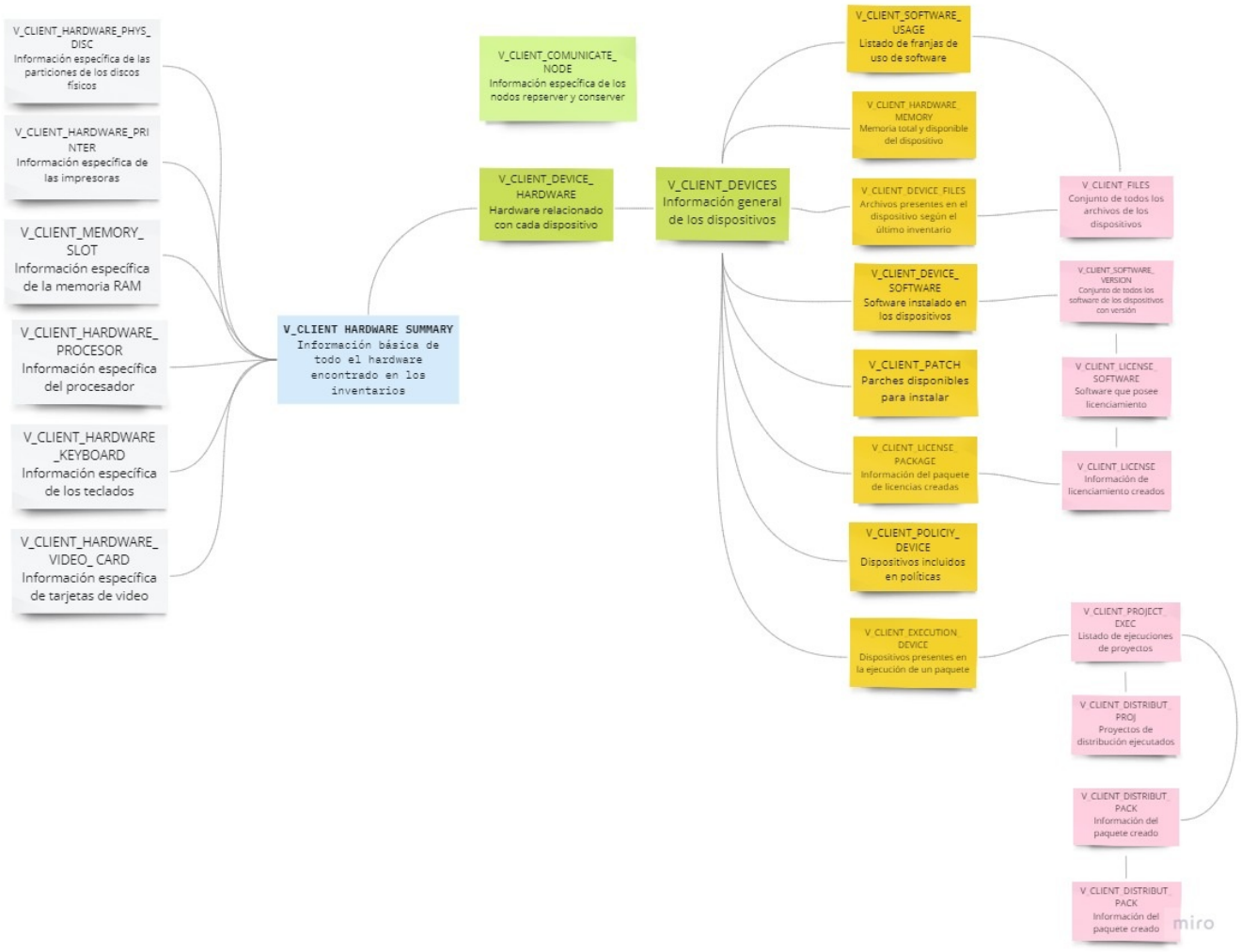


Customer Views

ADM KB Views

Entity Relationship Diagram

This diagram presents the master views of Aranda DEVICE MANAGEMENT and the relationships they have with other views to obtain the information required by the organization for the management of its resources.



Client Views Dictionary

1. ↪ V_CLIENT_DEVICES
2. ↪ V_CLIENT_SOFTWARWE_USAGE
3. V_CLIENT_HARDWARE\↪ _MEMORY
4. V_CLIENT_DEVICE_SOFTWARE ↪
5. V_CLIENT_SOFTWARE\↪ _VERSION
6. V_CLIENT_DEVICE FILES ↪
7. V_CLIENT\↪ _FILES
8. V_CLIENT\↪ _PATCH
9. V_CLIENT_PATCH\↪ _DEVICE
10. V_CLIENT_EXECUTION DEVICE ↪
11. V_CLIENT_PROJECT_EXEC ↪
12. V_CLIENT_DISTRIBUT\↪ _PROJ
13. V_CLIENT_EXEC\↪ _PACKAGE
14. V_CLIENT_DISTRIBUT_PACK ↪
15. V_CLIENT\↪ _LICENSE
16. V_CLIENT_POLICY\↪ _DEVICE
17. V_CLIENT\↪ _POLICIES
18. V_CLIENT_COMUNICAT\↪ _NODE
19. V_CLIENT_DEVICE_HARDWARE ↪
20. V_CLIENT_HARDWARE\↪ _SUMARY
21. V_CLIENT_HARD_PHYS\↪ _DISKS
22. V_CLIENT_HARDWARE\↪ _PRINTER
23. V_CLIENT_MEMORY\↪ _SLOT
24. V_CLIENT_HARD\↪ _PROCESSOR
25. V_CLIENT_HARD\↪ _KEYBOARD
26. V_CLIENT_HARD_VIDEO\↪ _CARD
27. V_CLIENT_LICENSE_SOFTWARE ↪

Customer Views1

V_CLIENT_DEVICES

Device overview

FIELD	GUY	NULLABLE	DESCRIPTION
device_id	Int	No	Device code.
device_name	Narchar	No	Device name.
device_type	Varchar	Yes	Defines the type of device (Unknown, Desktop, Laptop, Server, Printer, Switch, Router).
Description	Narchar	Yes	Description of the device.
Domain	Narchar	Yes	The domain to which the device belongs.
creation_date	datetime	No	Date of the agent's first communication with the Conserver.
device_user_name	Narchar	Yes	Device username.
responsible_user_name	Narchar	Yes	User responsible for the device.
responsible_user_email	Narchar	Yes	Email from the user responsible for the device.
Platform	Varchar	Yes	Device platform.
operating_system	Narchar	Yes	Device operating system.
operating_system_version	Narchar	Yes	Operating system version.
is_active	Varchar	Yes	Set whether the device is active or not.
model	Narchar	Yes	Device model.
serial	Narchar	Yes	Serial of the device.
manufacturer	Narchar	Yes	Device manufacturer.
agent_version	Narchar	No	Agent version installed on the device.
last_inventory	datetime	Yes	Date and time of the last inventory generated.
agent_profile	Narchar	Yes	Agent profile.
Keep	Narchar	Yes	Url of the Restaurant with which it communicates.
last_communication_update	datetime	Yes	Time and date of the agent's last communication with the Conserver.
communication_address	Narchar	Yes	IP of the device with which it communicates with the conserver.
days_since_last_inventory	Int	Yes	Days elapsed without inventory.
min_since_last_communication	Int	Yes	Minutes elapsed since the last communication.
is_discovery			Determines whether the device was discovered or if the agent was installed manually.

Existing Relationships	Primary Key
Table V_CLIENT_DEVICES with TablaV_CLIENT_DEVICE_HARDWARE	device Id
Table V_CLIENT_DEVICES with Table V_CLIENT_SOFTWARWE_USAGE	
Table V_CLIENT_DEVICES with Table V_CLIENT_HARDWARE_MEMORY	
Table V_CLIENT_DEVICES with Table V_CLIENT_DEVICE_SOFTWARE	
Table V_CLIENT_DEVICES with Table V_CLIENT_DEVICE_FILES	
Table V_CLIENT_DEVICES with Table V_CLIENT_EXECUTION_DEVICE	
Table V_CLIENT_DEVICES with Table V_CLIENT_POLICY_DEVICE	

V_CLIENT_SOFTWARWE_USAGE

List of software usage slots

FIELD	GUY	NULL ABLE	DESCRIPTION
View_device_Id	Int	No	Device code.
View_file_Id	Int	No	File code.
Use yourself	Date	No	Date the software was used.
Duration_seconds	bigint	No	Duration in seconds that the software was used.
usage_user	Narchar	Yes	User who used the software.

Existing Relationships	Foreign Keys
Table V_CLIENT_SOFTWARWE_USAGE with Table V_CLIENT_DEVICES	View_device_Id
Table V_CLIENT_SOFTWARWE_USAGE with Table V_CLIENT_FILES	View_file_Id

V_CLIENT_HARDWARE_MEMORY Total and available memory of the device

FIELD	GUY	NULL ABLE	DESCRIPTION
View_device_Id	Int	No	Device code.
total_physical_RAM	bigint	No	Total physical memory.
available_physical_RAM	bigint	No	Available physical memory.
total_vistual_memory	bigint	No	Total virtual memory.
available_virtual_RAM	bigint	No	Virtual memory available.
Max_RAM	bigint	No	Maximum memory capacity.
page_memory	bigint	No	Paging memory.
available_page_memory	bigint	No	Available paging memory.
Slot_count	Int	No	Number of occupied slots.

Existing Relationships	Foreign Key
Table V_CLIENT_HARDWARE_MEMORY with Table V_CLIENT_DEVICES	

V_CLIENT_DEVICE_SOFTWARE Breakout table indicating the software installed on the devices

FIELD	GUY	NULL ABLE	DESCRIPTION
View_software_Id	Int	No	Software code.
View_device_Id	bigint	No	Device code.

Existing Relationships	Foreign Key
Table V_CLIENT_HARDWARE_MEMORY with Table V_CLIENT_DEVICES	View_software_id view_device_Id

Client Views2

V_CLIENT_SOFTWARE_VERSION

Set of all the software of the devices with their version.

FIELD	GUY	NULLABLE	DESCRIPTION
software_Id	Int	No	Software code.
product_name	Narchar	No	Software name.
software_type	Varchar	Yes	Type of software (Operating System, Application, Update).
manufacturer	Narchar	Yes	Software developer.
Platform	Varchar	Yes	Software platform.
Is_inventory	Varchar	No	Software inventory status.
Version	Narchar	Yes	Software version.
Is_permitted	Varchar	No	Sets whether software is allowed - no.

Existing Relationships	Foreign Key
Table V_CLIENT_SOFTWARE_VERSION with Table V_CLIENT_DEVICE_SOFTWARE	software_Id

V_CLIENT_DEVICE FILES Breakage table indicating the files present on the device according to the last inventory.

FIELD	GUY	NULLABLE	DESCRIPTION
View_device_Id	Int	No	Device code.
View_file_version_Id	Int	No	File code.

Existing Relationships	Foreign Key
Table V_CLIENT_DEVICE_FILES with Table V_CLIENT_DEVICES	View_file_version_Id

V_CLIENT_FILES A set of all files on the devices.

FIELD	GUY	NULLABLE	DESCRIPTION
file_version_Id	Narchar	No	File version code.
file_Id	Narchar	No	File code.
file_extension	bigint	No	File extension.
file_name	Narchar	Yes	File name.
default_file_size	Narchar	Yes	File size in bits.
file_size_Convert	Narchar	Yes	Size of the converted file.
file_version	datetime	Yes	File version.
file_description	Narchar	Yes	File description.
file_modification_date	Narchar	No	Date of file modification.
file_Path	Narchar	No	Path where the file is located.

Existing Relationships	Foreign Key
Table V_CLIENT_FILES with Table V_CLIENT_DEVICE_FILES Table V_CLIENT_FILES with Table V_CLIENT_SOFTWARWE_USAGE	file_version_Id

V_CLIENT_PATCH

Patches available to install.

FIELD	GUY	NULLABLE	DESCRIPTION
patch_id	Int	No	Patch code.
class_type	Int	No	Patch class.
guid	Varchar	Yes	Patch guid.
bulletin_id	Varchar	Yes	Bulletin code.
Summary	Varchar	Yes	Patch summary.
sqnumber	Varchar	Yes	Patch revision number.
Name	Varchar	Yes	Name of the executable.
bulletin_url	Varchar	Yes	Patch URL.
cve_ids	Varchar	Yes	Patch vulnerabilities.
severity_patch	Int	Yes	Patch severity.
bulletin_title	Varchar	Yes	Name of the patch.
posted_date	datetime2	Yes	Date of release of the patch.
Deployable	Int	No	The patch is installable.
is_installed	Int	No	The patch is installed.
patch_type	Int	Yes	Identifier of the patch type.
download_id	Int	No	Download code.
download_status	Int	No	Download status identifier.
download_date	datetime	Yes	Download date.
scheduled_date	datetime	Yes	Programming date.
URL	Varchar	No	Patch download URL.
size	bigint	No	Patch size.
language_id	Int	No	Id of language.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_PATCH with Table V_CLIENT_PATCH_DEVICE		patch_id, language_id

V_CLIENT_PATCH_DEVICE

Patch ratio to the number of installed and uninstalled devices.

FIELD	GUY	NULLABLE	DESCRIPTION
patch_id	Int	No	Patch code.
language_id	Int	No	Patch language code.
installed_dev_count	Int	No	The number of devices that have the patch installed.
not_installed_dev_count	Int	No	The number of devices that do not have the patch installed.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_PATCH with Table V_CLIENT_PATCH_DEVICE		patch_id, language_id

Customer Views3

V_CLIENT_EXECUTION DEVICE

Devices present in the execution of a package

FIELD	GUY	NULLABLE	DESCRIPTION
View_execution_projects_ex equation_Id	Int	No	Execution Code.
View_device_Id	Int	No	Device class.
execution_status	Varchar	Yes	Status of execution.
execution_Start_date	datetime	Yes	Date of start of execution.
execution_finish_date	datetime	Yes	Execution completion date.
update_date	datetime	Yes	Date of update.

Existing Relationships	Foreign Key
Table V_CLIENT_EXECUTION_DEVICE with Table V_CLIENT_DEVICES	View_execution_projects_execution _id view_device_Id
Table V_CLIENT_EXECUTION_DEVICE with Table V_CLIENT_PROJECT_EXEC	

V_CLIENT_PROJECT_EXEC List of distribution project executions .

FIELD	GUY	NULLABLE	DESCRIPTION
execution_Id	Int	No	Code of execution.
View_distribution_project_Id	Int	No	Distribution project code.
execution_name	Narchar	No	Name of the execution.
execution_status	Varchar	Yes	Status of execution.
execution_date	datetime	Yes	Date of execution.
device_platform	Varchar	Yes	Platform where the execution was carried out.
execution_type	Varchar	Yes	Execution type (Install, Uninstall).

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_PROJECT_EXEC with Table V_CLIENT_EXECUTION_DEVICE Table V_CLIENT_PROJECT_EXEC with Table V_CLIENT_DISTRIBUT_PROJ Table V_CLIENT_PROJECT_EXEC with Table V_CLIENT_EXEC_PACKAGE	execution_Id	View_distribution _project_Id

V_CLIENT_DISTRIBUT_PROJ
Proyectos de distribución ejecutados.

FIELD	GUY	NULLABLE	DESCRIPTION
Project_Id	Int	No	Project code.
Project_name	Narchar	No	Name of the distribution project.
device_platform	Varchar	Yes	Device platform.
package_type	Varchar	Yes	Package type (Applications, Updates, Agent).
executions_failure	Int	Yes	Number of failed executions.
executions_successfull	Int	Yes	Number of executions completed successfully.
executions_count	Int	Yes	Total executions.
Last_execution_date	datetime	Yes	Date of last execution.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_DISTRIBUT_PROJ with Table _CLIENT_PROJECT_EXEC	Project_Id	

V_CLIENT_EXEC_PACKAGE Packages present per run.

FIELD	GUY	NULLABLE	DESCRIPTION
View_execution_execution_Id	Int	Yes	Code of execution.
View_distribution_package_Id	Int	No	Execution package code.
creation_date	datetime	Yes	Date of creation of the package.
execution_date	datetime	Yes	Package execution date.
creator_user	Narchar	No	User who created the package.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_EXEC_PACKAGE with Table _CLIENT_PROJECT_EXEC	view\execution _package_Id	
Table V_CLIENT_EXEC_PACKAGE with Table V_CLIENT_DISTRIBUT_PACK	View_distribution _package_Id	

V_CLIENT_DISTRIBUT_PACK Created Packages Information.

FIELD	GUY	NULLABLE	DESCRIPTION
package_Id	Int	No	Package code.
package_name	Narchar	No	Package name.
Version	Narchar	Yes	Package version.
package_type	Varchar	Yes	Package type (Applications, Updates, Agent).
device_platform	Varchar	Yes	Device platform.
architecture_package	Varchar	Yes	Package architecture.
Description_package	Narchar	Yes	Package Description

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_DISTRIBUT_PACK with Table V_CLIENT_EXEC_PACKAGE	package_Id	

Customer Views4

V_CLIENT_LICENSE

Information on licenses created.

FIELD	GUY	NULLABLE	DESCRIPTION
license_id	Int	No	License code.
license_name	Varchar	Yes	Name of the license.
creator_user	Narchar	Yes	User who created the license.
user_modifier	Narchar	Yes	User who modified the license.
creation_package_date	datetime	Yes	Date of creation of the license package.
updating_license_date	datetime	Yes	Date the license was updated.

Note	Additional licensing information in the following views

Existing Relationships	Foreign Key
Table V_CLIENT_LICENSE with Table V_CLIENT_LICENSE_SOFTWARE	license_id
Table V_CLIENT_LICENSE with Table V_CLIENT_LICENSE_PACKAGE	license_id
Table V_CLIENT_LICENSE with Table V_CLIENT_LICENSE_EVENT	license_id

V_CLIENT_POLICY_DEVICE Devices included in policies.

FIELD	GUY	NULLABLE	DESCRIPTION
View_device_Id	Int	No	Device code.
View_policy_Id	Int	No	Code of the policy.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_POLICY_DEVICE with Table V_CLIENT_DEVICES Table V_CLIENT_POLICY_DEVICE with Table V_CLIENT_POLICIES		View_device_id view_policy_Id

V_CLIENT_POLICIES Information on policies created.

FIELD	GUY	NULLABLE	DESCRIPTION
Policy_Id	Int	No	Code of the policy.
Policy_name	Narchar	No	Name of the policy.
user_name	Narchar	No	User who created the policy.
creation_date	datetime	No	Date the policy was created.
edit_date	datetime	Yes	Date of publication of the policy.
device_platform	Varchar	Yes	Device platform.
Policy_type	Varchar	Yes	Policy Type (Energy, ForbiddenPrograms, Restriction).

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_POLICIES with Table V_CLIENT_POLICY_DEVICE	Policy_Id	

V_CLIENT_COMUNICAT_NODE Node-specific information: Conservers and Repserver.

FIELD	GUY	NULLABLE	DESCRIPTION
Node_Id	Int	No	Node code.
Node_name	Narchar	No	Node name.
URL_node	Narchar	Yes	Node URL.
address	Narchar	Yes	IP address of the node.
parent_node_Id	Int	Yes	Establishes the inheritance of the node.
Node_enbled_statud	Varchar	Yes	Sets whether the node is enabled or not.
Last_communication	datetime	No	Date of last communication.

Existing Relationships	Primary Key	Foreign Key
	Node_Id	

Customer Views5

V_CLIENT_DEVICE_HARDWARE

Hardware related to each device.

FIELD	GUY	NULLABLE	DESCRIPTION
View_device_Id	Int	No	Device code.
View_hardware_Id	bigint	Yes	Hardware code.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_DEVICE_HARDWARE with Table V_CLIENT_DEVICES Table V_CLIENT_DEVICE_HARDWARE with Table V_CLIENT_HARDWARE_SUMMARY		View_device_id view_hardware_Id

V_CLIENT_HARDWARE_SUMARY Basic information on all hardware found in inventories.

FIELD	GUY	NULLABLE	DESCRIPTION
hardware_Id	bigint	Yes	HARDWARE code.
type_hardware_Id	Int	No	Hardware code.
hardware_type	Int	Yes	Hardware type code.
type	Varchar	No	Type of hardware.
Name	Narchar	Yes	Hardware name.
device_count	Int	Yes	Number of devices that have that hardware.
manufacturer	Narchar	Yes	Hardware manufacturer.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_HARDWARE_SUMMARY with Table V_CLIENT_DEVICE_HARDWARE Table V_CLIENT_HARDWARE_SUMMARY with Table V_CLIENT_HARD_PHYS_DISKS Table V_CLIENT_HARDWARE_SUMMARY with Table V_CLIENT_HARDWARE_PRINTER Table V_CLIENT_HARDWARE_SUMMARY with Table V_CLIENT_MEMORY_SLOT Table V_CLIENT_HARDWARE_SUMMARY with Table V_CLIENT_HARD_PROCESSOR Table V_CLIENT_HARDWARE_SUMMARY with Table V_CLIENT_HARD_KEYBOARD Table V_CLIENT_HARDWARE_SUMMARY with Table V_CLIENT_HARD_VIDEO_CARD	hardware_Id	type_hardware_Id

V_CLIENT_HARD_PHYS_DISKS Partition-specific information on physical disks.

FIELD	GUY	NULLABLE	DESCRIPTION
Disk_Id	Int	No	Disk code.
Disk_name	Narchar	No	Name of the album.
bus_type	Narchar	Yes	Type of bus.
Disk_size	bigint	No	Disk size.
logical_unit	Narchar	No	Disc manufacturer.
unit_type	Narchar	Yes	Type of logical unit.
unit_size	bigint	Yes	Size of the logical drive.
unit_size_free	bigint	Yes	Free space in the logical drive.
unit_file_system	Narchar	Yes	Logical drive file system.
manufacturer	Narchar	Yes	Disc manufacturer.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_HARD_PHYS_DISKS with Table V_CLIENT_HARDWARE_SUMMARY	Disk_Id	

V_CLIENT_HARDWARE_PRINTER Printer-specific information.

FIELD	GUY	NULLABLE	DESCRIPTION
printer_Id	Int	No	Printer code.
printer_name	Narchar	Yes	Printer name.
printer_port	Narchar	Yes	Printer port.
manufacturer	Narchar	Yes	Printer manufacturer.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_HARDWARE_PRINTER with Table V_CLIENT_HARDWARE_SUMMARY	printer_Id	

Customer Views6

V_CLIENT_MEMORY_SLOT

RAM-specific information.

FIELD	GUY	NULLABLE	DESCRIPTION
memory_Id	Int	No	Memory code.
memory_name	Narchar	No	Memory name.
memory_type	Narchar	Yes	Type of memory (DIMM, DDR, among others).
Socket_name	Narchar	No	Name of the socket.
Socket_size	bigint	No	Socket size.
Socket_speed	Int	No	Socket speed.
memory_bank_name	Narchar	Yes	Name of the memory bank.
Form_factor	Narchar	Yes	Get or set Form factor
menufacturer	Narchar	Yes	Memory manufacturer.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_MEMORY_SLOT with Table V_CLIENT_HARDWARE_SUMMARY	memory_Id	

V_CLIENT_HARD_PROCESSOR Processor-specific information.

FIELD	GUY	NULLABLE	DESCRIPTION
processor_Id	Int	No	Processor code.
processor_name	Narchar	Yes	Name of the processor.
Features	Narchar	Yes	Processor features.
speed_processor	bigint	Yes	Processor speed.
Socket_type	Narchar	Yes	Socket type.
Core_count	Int	Yes	Number of processor cores.
status_processor	Varchar	Yes	Processor Status (Unknow, Enalbe, DisabledByUser, DisabledByBios, Idle, Reserved1, Reserverd2, Other).
Socket_use	Varchar	Yes	Obtains or sets whether the socket is in use - no.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_HARD_PROCESSOR with Table V_CLIENT_HARDWARE_SUMMARY	processor_Id	

V_CLIENT_HARD_KEYBOARD Keyboard-specific information

FIELD	GUY	NULLABLE	DESCRIPTION
keyboard_Id	Int	No	Keypad code.
keyboard_name	Narchar	No	Keyboard name.
manufacturer	Narchar	Yes	Keyboard manufacturer.
device_identificator	Narchar	No	Unique identifier of the keyboard on the machine.
keyboard_type	Narchar	Yes	Set the keyboard type.
keyboard_language	Narchar	Yes	Sets the language in which the keyboard is configured.
keyboard_Country	Narchar	Yes	Set the country for which the keyboard is configured.
keyboard_layout	Narchar	Yes	Sets the keyboard layout for which it is configured.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_HARD_KEYBOARD with Table V_CLIENT_HARDWARE_SUMMARY	keyboard_Id	

V_CLIENT_HARD_VIDEO_CARD Video card-specific information.

FIELD	GUY	NULLABLE	DESCRIPTION
video_Id	Int	No	Video card code.
video_name	Narchar	No	Name of the video card.
manufacturer	Narchar	Yes	Video card manufacturer.
video_driver_name	Narchar	No	Driver name for the video card.
video_chipset	Narchar	Yes	Set your own chipset for the video card
video_Dac	Narchar	Yes	Set the built-in Digital Analog Converter on the video card.
video_memory_size	bigint	Yes	Sets the amount of memory built into the video card.
video_version	Narchar	Yes	Set the video card version.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_HARD_VIDEO_CARD with Table V_CLIENT_HARDWARE_SUMMARY	video_Id	

V_CLIENT_LICENSE_SOFTWARE

Displays the software associated with the license.

FIELD	GUY	NULLABLE	DESCRIPTION
Id	Int	No	Identifier of the relationship between the software and the license.
software_id	Int	No	Software code.
license_id	Int	No	Licensing code.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_LICENSE_SOFTWARE with Table V_CLIENT_LICENSE Table V_CLIENT_LICENSE_SOFTWARE with Table V_CLIENT_SOFTWARE_VERSION	Id	software_id license_id

V_CLIENT_LICENSE_PACKAGE

Displays license packages.

FIELD	GUY	NULLABLE	DESCRIPTION
Id	Int	No	License package identifier.
package_type	Narchar	No	Package type.
Quantity	Int	No	Number of licenses purchased.
unit_cost	float	Yes	Unit cost of licenses.
creation_date	datetime	No	Date of creation of the license.
expiration_date	datetime	No	License expiration date.
activation_date	datetime	No	License activation date.
package_name	Varchar	No	Name of the license package.
Currency	Varchar	Yes	Currency with which the license was purchased.
license_id	Int	No	License pool code.
Description	Varchar	Yes	Description of the license.

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_LICENSE_PACKAGE with Table V_CLIENT_LICENSE	Id	license_package_id license_id

V_CLIENT_LICENSE_EVENT

It allows you to identify which alerts have been generated from the licenses.

FIELD	GUY	NULLABLE	DESCRIPTION
licensing_id	Int	No	Identifier of the license that is related to the event
Description	Varchar	Yes	Description of the alert at the time of setting it up.
Destination	Varchar	Yes	Alert Mail Message
Message	Varchar	Yes	Content generated by the alert
severity	Varchar	Yes	Alert severity text (low, medium, high, critical)
subject	Varchar	Yes	Subject of the alert
title	Varchar	Yes	Title of the alert (when it has a subject this field has no content)
type	Varchar	Yes	Alert Type (Alert, Notification, Assign, Remove)
unit	Varchar	Yes	Unit of time to report the time the license expires (days, months, years)
value	Int	Yes	Value of the unit of time to notify when a license is going to expire (Ex: 5, if the unit is Days, then it will notify 5 days before the license becomes expired)

Existing Relationships	Primary Key	Foreign Key
Table V_CLIENT_LICENSE_EVENT with Table V_CLIENT_LICENSE		licensing_id

Database maintenance

Database maintenance

1. Recommendations on database operations ↪
2. Defragment Indexes and Update Database ↪ Statistics
3. Recommendations on index ↪ maintenance
4. What are the most trafficked tables in ADM, where deleting records is not recommended? ↪
5. What are the most trafficked tables in ADM, where records can be deleted? ↪

Recommendations on database operations

If you are running a query or direct operation against the Aranda Device Management (ADM) tables, you should consider the following recommendations that ensure that database performance is not compromised in the operation:

- You should not manually modify the information stored in the databases, avoiding unexpected behavior in the application.
- Use replicas of the data for historical reporting.
- When building reports, avoid querying all fields in a table using the wildcard character (*).
- When building reports, avoid running SQL commands that query tables in their entirety, instead define WHERE clauses that allow you to get a tight number of rows.
- Don't delete data from tables unless you're clear about the implications. Deleting data may affect the operation of the software. This task may require approval from the audit area and/or the company's management.
- Use the TRUNCATE statement when you need to delete all records from a large table, making the process more efficient since it does not generate entries in the transaction log.

Defragment Indexes and Update Database Statistics

Index defragmentation and statistics update tasks help improve database performance by improving access to data on disk. It is advisable to perform these tasks on a regular basis to ensure that the database is functioning smoothly by ensuring that the execution plans are compiled with the latest statistics from the database.

Perform the following procedure:

1. Confirm the edition of SQL Server that you are running. If your edition is enterprise or you're using Azure SQL Database, skip straight to step 4. For more information on online index trading, please see: Perform Index Operations Online
2. Make sure that specialists are not editing, querying or entering data into the tools of the Aranda suite. To do this, you can stop the Internet Information Services (IIS) where the Aranda applications are hosted.
3. Stop the Aranda Suite services on the application server.
4. Connect to the SQL Server using SQL Server Management Studio and open a new .sql file
5. Change the context to the Aranda database, e.g.

```
USE Aranda;
GO
```

1. Rebuild indexes with fragmentation greater than 30% and page usage less than 60% by running the script:

```
SET NOCOUNT ON;

DECLARE
  @v_command NVARCHAR(MAX),
  @v_init DATETIME,
  @v_fragmentation FLOAT,
  @v_page_space_used FLOAT,
  @v_table_name SYSNAME,
  @v_index_name SYSNAME;

DECLARE cur_index CURSOR LOCAL STATIC FOR
SELECT
  CONCAT (
    'ALTER INDEX [',
    i.name,
    '] ON [',
    OBJECT_NAME(s.object_id),
    IIF(s.avg_fragmentation_in_percent BETWEEN 10 AND 30, '] REORGANIZE', '] REBUILD')
  ) statement,
  s.avg_fragmentation_in_percent,
  s.avg_page_space_used_in_percent,
  OBJECT_NAME(s.object_id) table_name,
  i.name index_name
FROM sys.dm_db_index_physical_stats (
  DB_ID(),
  DEFAULT,
  DEFAULT,
  DEFAULT,
  'LIMITED'
) s
JOIN sys.indexes i ON i.index_id = s.index_id
  AND i.object_id = s.object_id
WHERE s.page_count > 50
  AND (
    s.avg_fragmentation_in_percent > 10
    OR s.avg_page_space_used_in_percent < 60
  ) AND s.index_type_desc <> 'HEAP'
  AND s.page_count < 262144;

OPEN cur_index;

FETCH NEXT FROM cur_index INTO
  @v_command,
  @v_fragmentation,
  @v_page_space_used,
  @v_table_name,
  @v_index_name;

WHILE @@FETCH_STATUS = 0
BEGIN
  BEGIN TRY
    SET @v_init = CURRENT_TIMESTAMP;
```

```
    EXEC sp_executesql @v_command;

    PRINT CONCAT(
      'Ending: ',
      IIF(@v_fragmentation BETWEEN 10 AND 30, 'reorganize ', 'rebuild '),
      @v_table_name,
      '.',
      @v_index_name,
      ' execution time ',
      CONVERT(VARCHAR(10), DATEDIFF(MILLISECOND, @v_init, GETDATE())),
      'ms ',
      'fragmentation ',
      @v_fragmentation,
      '%'
    );

  END TRY
  BEGIN CATCH
    PRINT ERROR_MESSAGE();

  END CATCH;

  FETCH NEXT FROM cur_index INTO
    @v_command,
    @v_fragmentation,
    @v_page_space_used,
    @v_table_name,
    @v_index_name;

END;
GO
```

1. Restart all previously stopped services from the Aranda Suite and Internet Information Services (IIS).

Recommendations on index maintenance

- Operations on indexes in SQL Server editions other than Enterprise or Azure SQL Database can cause table crashes causing unexpected behavior in Aranda applications, schedule these tasks to run automatically at least once every week at times of low transactional load.
- Track the physical statistics of your indexes by using SQL Server management views to determine whether you should adjust the periodicity of index defragmentation operations. If you determine that the fragmentation remains high > 50%, reduce the periodicity, if you do not want to do so, you can maintain the recommended one or increase it.

What are the most trafficked tables in ADM, where deleting records is not recommended?

- AFW_ADDITIONAL_FIELD_VALUE: This table contains the values of the additional fields of all the concepts in Aranda Device Management.
- AFW_USERS: This is the main table of users, this table concentrates customers, field specialists and web specialists. This table can be fed new users through the web console and synchronization from the LDAP active directory. This table should not be purged and its records should not be deleted.

What are the most trafficked tables in ADM, where records can be deleted?

PRC_ASM_CLEAN_SOFT_USE: Cleaning up the history of software usage can be scheduled from the ADM application, in the Configuration - Measurement option. The frequency to clear the history must be configured by inventory type (Detail, Daily, and Monthly), this task is automatically scheduled to run daily from the time the configuration is made.

- We recommend that you run this cleanup from the database before performing the configuration from the ADM application in the following cases:
- That the database contains a large history of software usage data
- Failure to run the scheduled task from the ADM console
- Procedure PRC_ASM_CLEAN_SOFT_USE removes software usage information from tables [ASM_SOFTWARE_USAGE], [ASM_SOFTWARE_USAGE_DAY] and [ASM_SOFTWARE_USAGE_MONTH]

Parameters:

- @i_detail parameter of type DATE, is used by the procedure to delete the detailed software usage history of the date specified in the backward parameter. Delete information from the table [ASM_SOFTWARE_USAGE]

  Example: @i_detail ='2022-01-01 00:00:00'

- @i_day parameter of type DATE, is used by the procedure to remove the daily rollback from the software usage of the date specified in the backward parameter. Delete information from the table [ASM_SOFTWARE_USAGE_DAY]

  Example: @i_day ='2022-01-01'

- @i_month parameter of type DATE, is used by the procedure to delete the monthly software usage rollup from the date specified in the backward parameter. Delete information from the table [ASM_SOFTWARE_USAGE_MONTH]

```
Example: @i_month ='2022-01-01'
```

Implementation of the Procedure

```
EXECUTE [dbo].[PRC_ASM_CLEAN_SOFT_USE]
@i_detail ='2020-01-01 00:00:00',
@i_day ='2020-01-01',
@i_month ='2020-01-01'
```

- AAM_AGENT_TASK - AAM_AGENT_TASK_DETAIL: These tables save the actions that are scheduled from the console to the device. For example: Request an inventory, Request remote control, Send a message, Restart device, etc...; these actions have statuses (Registered = 1, Reported = 1, Running = 2, Executed = 7, Executed with Errors = 8).
Depending on the management of the tool, it can generate several actions in turn several records. You can make a decision to execute a physical deletion script for the actions that have been performed correctly and with an effective date.

```
DELETE TASKDETAIL
FROM dbo.AAM_AGENT_TASK_DETAIL TASKDETAIL
INNER JOIN dbo.AAM_AGENT_TASK TASK ON TASKDETAIL.task_id=TASK.id
WHERE TASKDETAIL.status in (7,8) and TASK.creation_date < '2020-01-01 00:00:00.000'

DEIETE TASK
FROM dbo.AAM_AGENT_TASK TASK
LEFT JOIN dbo.AAM_AGENT_TASK_DETAIL TASKDETAIL ON TASK.id = TASKDETAIL.task_id
WHERE TASKDETAIL.task_id IS NULL
```

AFW_ALERT: This table saves the alerts that the system generates. For example: discovered devices, devices with specific software/hardware, devices with a disk occupancy percentage; these alerts have 3 statuses (New, Reviewed, and Closed). Depending on the management of the tool, it can generate several alerts and in turn several records.
A decision can be made to execute a physical deletion script for alerts that have been closed by dispatchers with an effective date since it has an index with the following keys (creation_date, status_id, category_id):

```
DELETE
FROM [BD_NAME].[dbo].[AFW_ALERT]
WHERE status_id = 3 and creation_date < '2020-01-01 00:00:00'
```

AFW_MAIL_HISTORY: The ADM System sends emails for different reasons. These emails are sent by a scheduled task and once they are sent satisfactorily they go to the AFW_MAIL_HISTORY table; This board can grow quite quickly; This history remains intact and a decision can be made under which criteria to delete the information. For example, if you want to delete information, you can use the following script:

```
DELETE
FROM [BD_NAME].[dbo].[AFW_MAIL_HISTORY]
WHERE mahi_created < '2020-01-01 00:00:00'
```

AFW_WORKER_LOG: This table has the task execution log that the Aranda worker service executes. This log is used to identify that the tasks have been executed in the expected time and to validate their status by means of the work_success field (0: Failed, 1: Successful, 2: Pending execution and 3: In process). You can make a decision to delete with a certain periodicity the tasks that were successfully executed and leave those that present some type of error in case any support or adjustment is required.

```
DELETE
FROM [BD_NAME].[dbo].[AFW_WORKER_LOG]
WHERE work_execution_date < '2020-01-01 00:00:00' AND work_success = 1
```