Pandore – Database developer guide

This document is a developer guide for the Pandore project database

Clément LE GRUIEC

Salma CHAHMI

Nathan OLBORSKI

Hugo HOUILLON

Table des matières

[1. General informations 4](#_Toc88754027)

[1.1. Server 4](#_Toc88754028)

[1.2. Database architecture 4](#_Toc88754029)

[2. Versions 5](#_Toc88754030)

[2.1. Version 1 5](#_Toc88754031)

[2.2. Version 2 5](#_Toc88754032)

[2.3. Version 3 5](#_Toc88754033)

[3. Table Service 6](#_Toc88754034)

[3.1. Description 6](#_Toc88754035)

[3.2. Stored procedures 7](#_Toc88754036)

[3.2.1. CreateService 7](#_Toc88754037)

[3.2.2. ReadAllServices 8](#_Toc88754038)

[3.2.3. ReadServiceByID 9](#_Toc88754039)

[3.2.4. ReadServiceByName 10](#_Toc88754040)

[3.2.5. UpdateService 11](#_Toc88754041)

[3.2.6. DeleteServiceByID 12](#_Toc88754042)

[3.2.7. DeleteServiceByName 13](#_Toc88754043)

[4. Table Server 14](#_Toc88754044)

[4.1. Description 14](#_Toc88754045)

[4.2. Stored procedures 15](#_Toc88754046)

[4.2.1. CreateServer 15](#_Toc88754047)

[4.2.2. CreateServerString 16](#_Toc88754048)

[4.2.3. ReadAllServers 17](#_Toc88754049)

[4.2.4. ReadServersByServiceID 18](#_Toc88754050)

[4.2.5. ReadServerByID 19](#_Toc88754051)

[4.2.6. ReadServerByAddress 20](#_Toc88754052)

[4.2.7. UpdateServer 21](#_Toc88754053)

[4.2.8. DeleteServerByID 22](#_Toc88754054)

[4.2.9. DeleteServerByAddress 23](#_Toc88754055)

[4.2.10. DeleteServerByServiceID 24](#_Toc88754056)

[5. Table DNS 25](#_Toc88754057)

[5.1. Description 25](#_Toc88754058)

[5.2. Stored procedures 26](#_Toc88754059)

[5.2.1. CreateDNS 26](#_Toc88754060)

[5.2.2. ReadAllDNS 27](#_Toc88754061)

[5.2.3. ReadDNSByID 28](#_Toc88754062)

[5.2.4. ReadDNSByValue 29](#_Toc88754063)

[5.2.5. UpdateDNS 30](#_Toc88754064)

[5.2.6. DeleteDNSByID 31](#_Toc88754065)

[5.2.7. DeleteDNSByValue 32](#_Toc88754066)

[6. Table Capture 33](#_Toc88754067)

[6.1. Description 33](#_Toc88754068)

[6.2. Stored procedures 34](#_Toc88754069)

[6.2.1. CreateCapture 34](#_Toc88754070)

[6.2.2. ReadAllCaptures 35](#_Toc88754071)

[6.2.3. ReadCaptureByID 36](#_Toc88754072)

[6.2.4. UpdateCapture 37](#_Toc88754073)

[6.2.5. DeleteCaptureByID 38](#_Toc88754074)

[7. Table Capture\_Request 39](#_Toc88754075)

[7.1. Description 39](#_Toc88754076)

[7.2. Stored procedures 40](#_Toc88754077)

[7.2.1. CreateRequest 40](#_Toc88754078)

[7.2.2. CreateRequestString 41](#_Toc88754079)

[7.2.3. ReadAllRequests 42](#_Toc88754080)

[7.2.4. ReadRequestsByCaptureID 43](#_Toc88754081)

[7.2.5. ReadRequestByID 44](#_Toc88754082)

[7.2.6. UpdateRequest 45](#_Toc88754083)

[7.2.7. DeleteRequestByID 46](#_Toc88754084)

[7.2.8. DeleteRequestByCaptureID 47](#_Toc88754085)

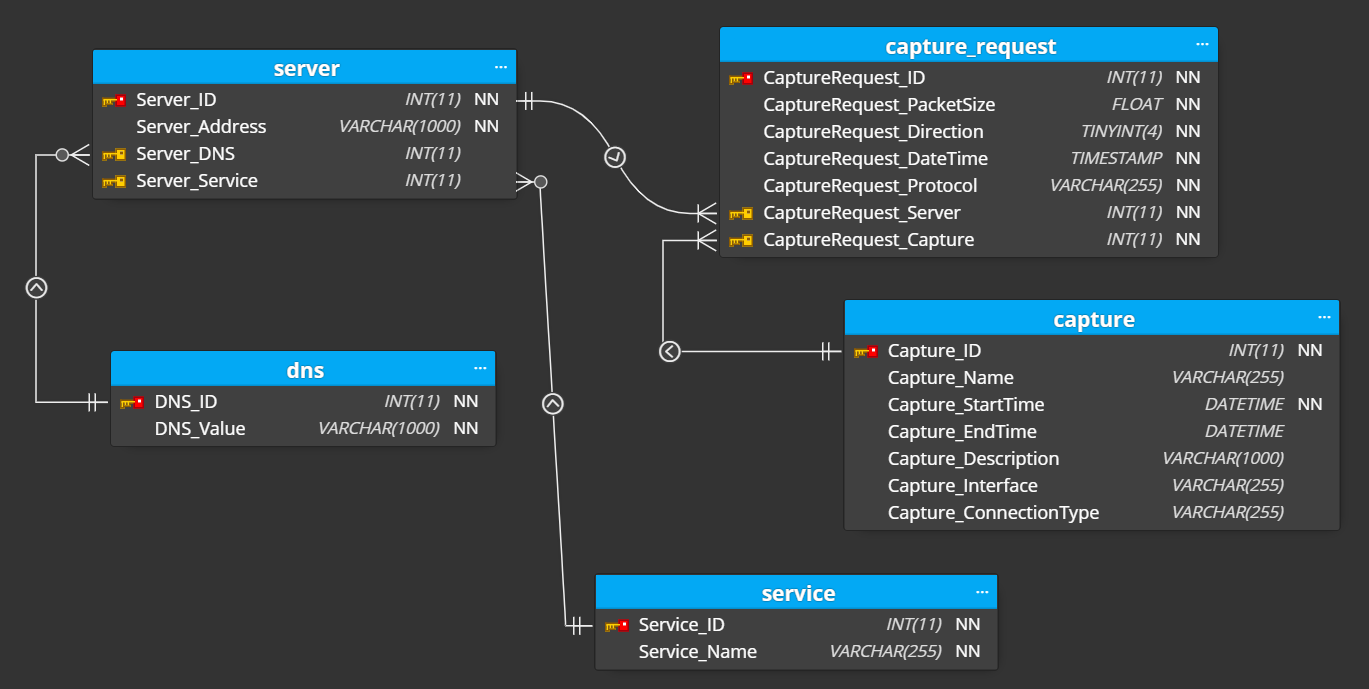
# General informations

## Server

The database server used for the Pandore project is ***MariaDB Server 10.6.4***

## Database architecture

The diagram below is a class diagram of the database architecture:



# Versions

## Version 1

* First database architecture implementation

## Version 2

* Fields *CaptureRequest\_DNS* and *CaptureRequest\_Server* now allow *NULL* values
* Stored procedure *CreateRequestString* edited. It allows now to call it with *NULL* values for *DNS* or *Server* fields
* Stored procedure *ReadAllRequests* edited. New procedure parameter *Details*. Allows now to get requests with DNS and Server details
* Stored procedure *ReadRequestsByCaptureID* edited. New procedure parameter *Details*. Allows now to get requests with DNS and Server details
* Stored procedure *ReadRequestByID* edited. New procedure parameter *Details*. Allows now to get requests with DNS and Server details

## Version 3

* Field *CaptureRequest\_DNS* deleted
* Remove field *DNS\_Server* (DNS doesn’t refer to service table anymore)
* Add field *DNS\_Service*. A DNS refers now to a server
* Multiple stored procedures edited
* Multiple stored procedures deleted
* New stored procedure *CreateServerString* which allows to add a server and a DNS from there names

# Table Service

## Description

This table is used to identify the name of different services (ex : Facebook, Youtube, …).

It consists of 2 attributes :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Allow NULL | Constraint | Description |
| Service\_ID | INT |  | PK | Service ID |
| Service\_Name | VARCHAR(255) |  | Unique | Service Name |

## Stored procedures

### CreateService

#### Description

This procedure allows to create a new service into the database

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Name | VARCHAR(255) |  | Service name |

#### Example



### ReadAllServices

#### Description

This procedure allows to list all services in the database

#### Prototype



#### Returns

Service list ordered alphabetically

Une image contenant texte

Description générée automatiquement

#### Example



### ReadServiceByID

#### Description

This procedure allows to find a service in the database by its ID

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | Service ID to find |

#### Returns

The service found with the given ID

Une image contenant texte

Description générée automatiquement

#### Example



### ReadServiceByName

#### Description

This procedure allows to find a service by its name

#### Prototype

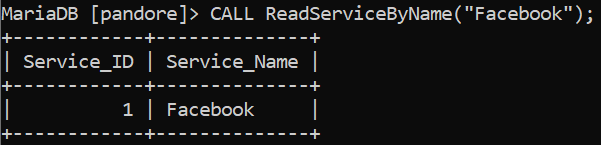


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Name | VARCHAR(255) |  | Service name to find |

#### Returns

Services found with the given service name



#### Example



### UpdateService

#### Description

This procedure allows to update an existing service

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | Service ID to update |
| Name | VARCHAR(255) |  | New service name |

#### Example



### DeleteServiceByID

#### Description

This procedure allows to delete a service by its ID

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the service to delete |

#### Example



### DeleteServiceByName

#### Description

This procedure allows to delete a service by its name

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Name | VARCHAR(255) |  | Name of the service to delete |

#### Example



# Table Server

## Description

This table is used to identify different server addresses and associate them to services

It consists of 4 attributes :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Allow NULL | Constraint | Description |
| Server\_ID | INT |  | Primary Key | Service ID |
| Server\_Address | VARCHAR(1000) |  | Unique | Service Name |
| Server\_DNS | INT | X | Foreign Key | ID of the associated DNS |
| Server\_Service | INT | X | Foreign Key | ID of the associated service |

## Stored procedures

### CreateServer

#### Description

This procedure allows to create a new server in the database

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Address | VARCHAR(1000) |  | Address of the server |
| DNS | INT | x | ID of the associated DNS |
| Service | INT | x | ID of the associated service |

#### Example



### CreateServerString

#### Description

This procedure allows to create a new server in the database and a new DNS if its value doesn’t already exist.

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Address | VARCHAR(1000) |  | Address of the server |
| DNS | VARCHAR(1000) | x | Value of the associated DNS |
| Service | INT | x | ID of the associated service |

#### Example



### ReadAllServers

#### Description

This procedure allows to list all servers in the database

#### Prototype



#### Returns

Server list ordered alphabetically

Une image contenant texte

Description générée automatiquement

#### Example



### ReadServersByServiceID

#### Description

This procedure allows to list all servers for a given service

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Service | INT |  | ID of the service |

#### Returns

List of servers associated to given service

Une image contenant texte

Description générée automatiquement

#### Example



### ReadServerByID

#### Description

This procedure allows to find a server by its ID

#### Prototype

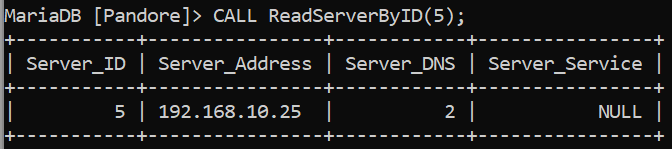


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the server |

#### Returns

Server found with given ID



#### Example



### ReadServerByAddress

#### Description

This procedure allows to find a server by its address

#### Prototype

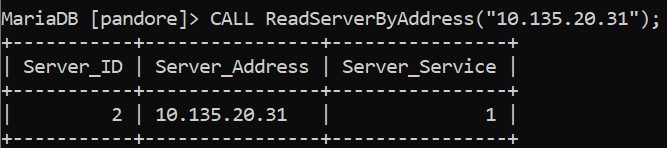


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Address | VARCHAR(1000) |  | Address of the server |

#### Returns

Server found with given address



#### Example



### UpdateServer

#### Description

This procedure allows to update a server

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the server to update |
| Address | VARCHAR(1000) |  | New server address |
| Service | INT | x | New service ID |
| DNS | INT | x | New DNS ID |

#### Example



### DeleteServerByID

#### Description

This procedure allows to delete a server by its ID

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the server to delete |

#### Example



### DeleteServerByAddress

#### Description

This procedure allows to delete a server by its address

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Address | VARCHAR(1000) |  | Address of the server to delete |

#### Example



### DeleteServerByServiceID

#### Description

This procedure allows to delete servers associated to given service

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Service | INT |  | ID of the service where servers must be deleted |

#### Example



# Table DNS

## Description

This table is used to identify different DNS and associate them to services

It consists of 2 attributes :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Allow NULL | Constraint | Description |
| DNS\_ID | INT |  | Primary Key | DNS ID |
| DNS\_Value | VARCHAR(1000) |  | Unique | DNS Name |

## Stored procedures

### CreateDNS

#### Description

This procedure allows to create a new DNS in the database

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Value | VARCHAR(1000) |  | Value of the DNS |

#### Example



### ReadAllDNS

#### Description

This procedure allows to list all DNS in the database

#### Prototype



#### Returns

DNS list ordered alphabetically

Une image contenant texte

Description générée automatiquement

#### Example



### ReadDNSByID

#### Description

This procedure allows to find a DNS by its ID

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the DNS |

#### Returns

DNS found with given ID

Une image contenant texte

Description générée automatiquement

#### Example



### ReadDNSByValue

#### Description

This procedure allows to find a DNS by its value

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Address | VARCHAR(1000) |  | Value of the DNS |

#### Returns

DNS found with given value

Une image contenant texte

Description générée automatiquement

#### Example



### UpdateDNS

#### Description

This procedure allows to update a DNS

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the DNS to update |
| Value | VARCHAR(1000) |  | New DNS address |

#### Example



### DeleteDNSByID

#### Description

This procedure allows to delete a DNS by its ID

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the DNS to delete |

#### Example



### DeleteDNSByValue

#### Description

This procedure allows to delete a DNS by its value

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Value | VARCHAR(1000) |  | Value of the DNS to delete |

#### Example



# Table Capture

## Description

This table is used to identify captures done on a device

It consists of 7 attributes :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Allow NULL | Constraint | Description |
| Capture\_ID | INT |  | Primary Key | Capture ID |
| Capture\_Name | VARCHAR(255) | x |  | Capture Name |
| Capture\_StartTime | DATETIME |  |  | Capture start time.  Format :  yyyy-MM-dd HH:mm:ss |
| Capture\_EndTime | DATETIME | x |  | Capture end time.  Format :  yyyy-MM-dd HH:mm:ss |
| Capture\_Description | VARCHAR(1000) | x |  | Capture description |
| Capture\_Interface | VARCHAR(255) | x |  | Capture interface (eth0, eth1, etc…) |
| Capture\_ConnectionType | VARCHAR(255) | x |  | Connection type description |

## Stored procedures

### CreateCapture

#### Description

This procedure allows to create a new capture in the database

#### Prototype

Une image contenant texte

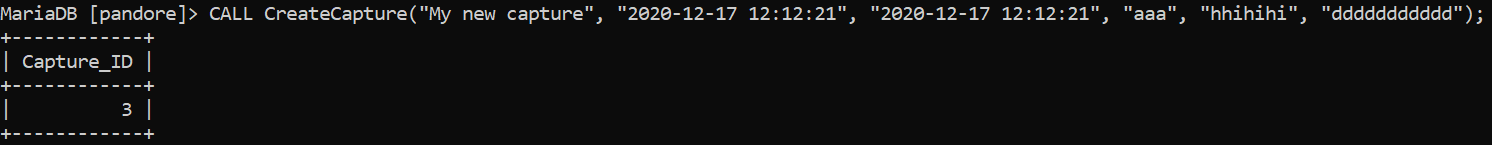
Description générée automatiquement

#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Name | VARCHAR(255) | x | Capture Name |
| StartTime | DATETIME |  | Capture start time. Format : yyyy-MM-dd HH:mm:ss |
| EndTime | DATETIME | x | Capture end time. Format : yyyy-MM-dd HH:mm:ss |
| Description | VARCHAR(1000) | x | Capture description |
| Interface | VARCHAR(255) | x | Capture interface (eth0, eth1, etc…) |
| ConnectionType | VARCHAR(255) | x | Connection type description |

#### Returns

Returns the ID of the capture created



#### Example

Une image contenant texte

Description générée automatiquement

### ReadAllCaptures

#### Description

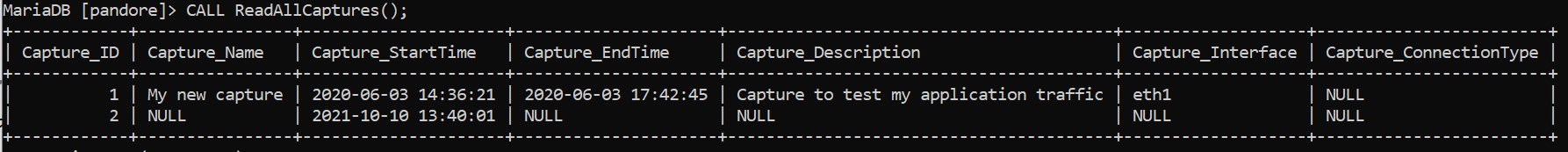
This procedure allows to list all captures

#### Prototype



#### Returns

List of all captures



#### Example



### ReadCaptureByID

#### Description

This procedure allows to find a capture by its ID

#### Prototype

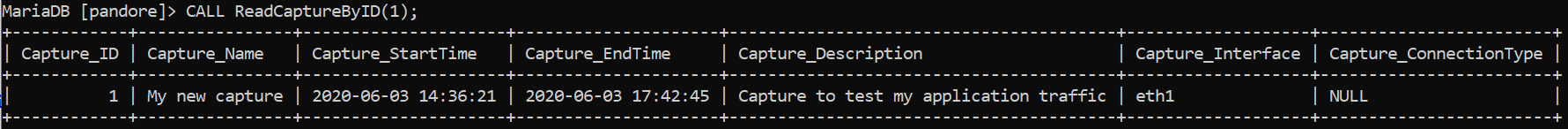


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | Capture ID |

#### Returns

The capture found with given ID



#### Example



### UpdateCapture

#### Description

This procedure allows to update a capture by its ID

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the capture to update |
| Name | VARCHAR(255) | x | New capture name |
| StartTime | DATETIME |  | New capture start time.  Format : yyyy-MM-dd HH:mm:ss |
| EndTime | DATETIME | x | New capture end time.  Format : yyyy-MM-dd HH:mm:ss |
| Description | VARCHAR(1000) | x | New capture description |
| Interface | VARCHAR(255) | x | New capture interface (eth0, eth1, etc…) |
| ConnectionType | VARCHAR(255) | x | New connection type description |

#### Example

Une image contenant texte

Description générée automatiquement

### DeleteCaptureByID

#### Description

This procedure allows to delete a capture by its ID

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the capture to delete |

#### Example



# Table Capture\_Request

## Description

This table is used to identify requests for a specific capture

It consists of 7 attributes :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Allow NULL | Constraint | Description |
| CaptureRequest\_ID | INT |  | Primary Key | Capture request ID |
| CaptureRequest\_PacketSize | FLOAT |  |  | Size of the packet |
| CaptureRequest\_Direction | TINYINT |  |  | Packet direction (0 = download, 1 = upload) |
| CaptureRequest\_DateTime | TIMESTAMP |  |  | Request date time.  Format :  yyyy-MM-dd HH:mm:ss |
| CaptureRequest\_Protocol | VARCHAR(255) |  |  | Request protocol description |
| CaptureRequest\_Server | INT |  | Foreign Key | ID of the server |
| CaptureRequest\_Capture | INT |  | Foreign Key | ID of the capture |

## Stored procedures

### CreateRequest

#### Description

This procedure allows to create a new capture request in the database

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| PacketSize | FLOAT |  | Size of the packet |
| Direction | TINYINT |  | Packet direction (0 = download, 1 = upload) |
| Protocol | VARCHAR(255) |  | Request protocol description |
| Server | INT |  | ID of the server |
| Capture | INT |  | ID of the capture |

#### Example



### CreateRequestString

#### Description

This procedure allows to create a new capture request in the database without knowing the id of the DNS and the server but only their values.

If DNS is not null and doesn’t exist in the database, new DNS is added in DNS table.

If Server is not null and doesn’t exist in the database, new server is added in Server table.

If DNS is null or empty, request DNS will also be null and nothing will be added in DNS table.

If Server is null or empty, request Server will also be null and nothing will be added in Server table.

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| PacketSize | FLOAT |  | Size of the packet |
| Direction | TINYINT |  | Packet direction (0 = download, 1 = upload) |
| Protocol | VARCHAR(255) |  | Request protocol description |
| Server | VARCHAR(1000) | x | Server address |
| DNS | VARCHAR(1000) | x | DNS value |
| Capture | INT |  | ID of the capture |

#### Example



### ReadAllRequests

#### Description

This procedure allows to list all capture requests

#### Prototype

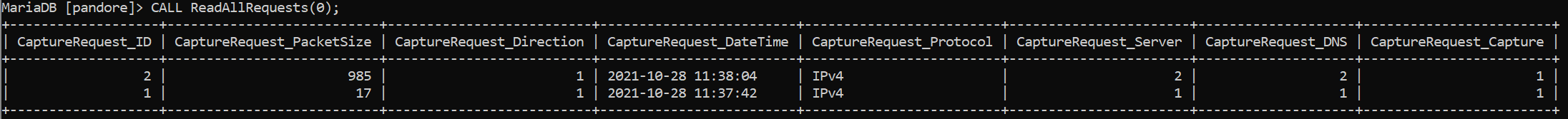


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Details | TINYINT(1) |  | 0 = returns list of requests  1 = returns list of requests with join with DNS and server tables |

#### Returns

List of all capture requests (details = 0)



List of all capture requests with DNS and server details (details = 1)



#### Example



### ReadRequestsByCaptureID

#### Description

This procedure allows to list all requests for a given capture

#### Prototype

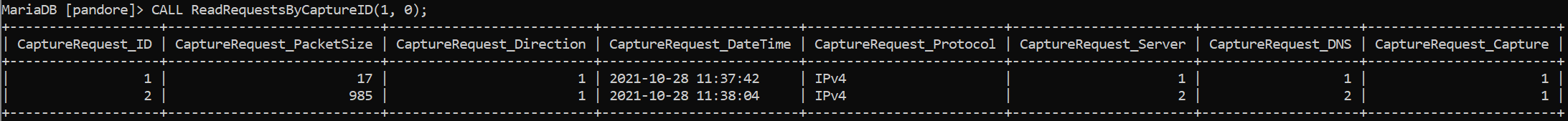


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Capture | INT |  | ID of the capture |
| Details | TINYINT(1) |  | 0 = returns list of requests  1 = returns list of requests with join with DNS and server tables |

#### Returns

List of all capture requests for the given capture ID (details = 0)



List of all capture requests for the given capture ID with DNS and server details (details = 1)



#### Example



### ReadRequestByID

#### Description

This procedure allows to find a request by its ID

#### Prototype

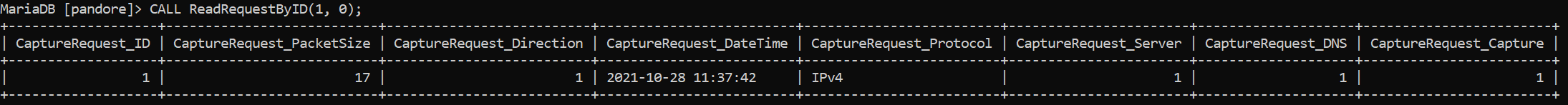


#### Parameters

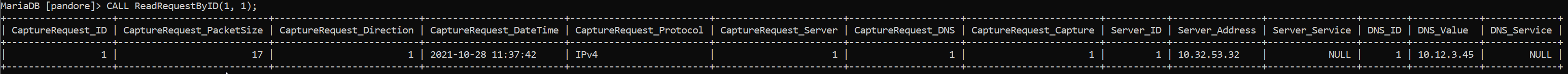
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the request to update |
| Details | TINYINT(1) |  | 0 = returns list of requests  1 = returns list of requests with join with DNS and server tables |

#### Returns

The request found for the given request ID (details = 0)



The request found for the given request ID with DNS and server details (details = 1)



#### Example



### UpdateRequest

#### Description

This procedure allows to update an existing capture request

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the request to update |
| PacketSize | FLOAT |  | New packet size |
| Direction | TINYINT |  | New packet direction (0 = download, 1 = upload) |
| DateTime | TIMESTAMP |  | New request date time.  Format :  yyyy-MM-dd HH:mm:ss |
| Protocol | VARCHAR(255) |  | New protocol description |
| Server | INT |  | ID of the new server |
| DNS | INT |  | ID of the new DNS |
| Capture | INT |  | ID of the new capture |

#### Example



### DeleteRequestByID

#### Description

This procedure allows to delete a capture request by its ID

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the capture request to delete |

#### Example



### DeleteRequestByCaptureID

#### Description

This procedure allows to delete all requests for a given capture

#### Prototype



#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| Capture | INT |  | ID of the capture |

#### Example

