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](#_Toc86314287)

[1.1. Server 4](#_Toc86314288)

[1.2. Database architecture 4](#_Toc86314289)

[2. Versions 5](#_Toc86314290)

[2.1. Version 1 5](#_Toc86314291)

[2.2. Version 2 5](#_Toc86314292)

[3. Table Service 6](#_Toc86314293)

[3.1. Description 6](#_Toc86314294)

[3.2. Stored procedures 7](#_Toc86314295)

[3.2.1. CreateService 7](#_Toc86314296)

[3.2.2. ReadAllServices 8](#_Toc86314297)

[3.2.3. ReadServiceByID 9](#_Toc86314298)

[3.2.4. ReadServiceByName 10](#_Toc86314299)

[3.2.5. UpdateService 11](#_Toc86314300)

[3.2.6. DeleteServiceByID 12](#_Toc86314301)

[3.2.7. DeleteServiceByName 13](#_Toc86314302)

[4. Table Server 14](#_Toc86314303)

[4.1. Description 14](#_Toc86314304)

[4.2. Stored procedures 15](#_Toc86314305)

[4.2.1. CreateServer 15](#_Toc86314306)

[4.2.2. ReadAllServers 16](#_Toc86314307)

[4.2.3. ReadServersByServiceID 17](#_Toc86314308)

[4.2.4. ReadServerByID 18](#_Toc86314309)

[4.2.5. ReadServerByAddress 19](#_Toc86314310)

[4.2.6. UpdateServer 20](#_Toc86314311)

[4.2.7. DeleteServerByID 21](#_Toc86314312)

[4.2.8. DeleteServerByAddress 22](#_Toc86314313)

[4.2.9. DeleteServerByServiceID 23](#_Toc86314314)

[5. Table DNS 24](#_Toc86314315)

[5.1. Description 24](#_Toc86314316)

[5.2. Stored procedures 25](#_Toc86314317)

[5.2.1. CreateDNS 25](#_Toc86314318)

[5.2.2. ReadAllDNS 26](#_Toc86314319)

[5.2.3. ReadDNSByServiceID 27](#_Toc86314320)

[5.2.4. ReadDNSByID 28](#_Toc86314321)

[5.2.5. ReadServerByAddress 29](#_Toc86314322)

[5.2.6. UpdateDNS 30](#_Toc86314323)

[5.2.7. DeleteDNSByID 31](#_Toc86314324)

[5.2.8. DeleteDNSByValue 32](#_Toc86314325)

[5.2.9. DeleteDNSByServiceID 33](#_Toc86314326)

[6. Table Capture 34](#_Toc86314327)

[6.1. Description 34](#_Toc86314328)

[6.2. Stored procedures 35](#_Toc86314329)

[6.2.1. CreateCapture 35](#_Toc86314330)

[6.2.2. ReadAllCaptures 36](#_Toc86314331)

[6.2.3. ReadCaptureByID 37](#_Toc86314332)

[6.2.4. UpdateCapture 38](#_Toc86314333)

[6.2.5. DeleteCaptureByID 39](#_Toc86314334)

[7. Table Capture\_Request 40](#_Toc86314335)

[7.1. Description 40](#_Toc86314336)

[7.2. Stored procedures 41](#_Toc86314337)

[7.2.1. CreateRequest 41](#_Toc86314338)

[7.2.2. CreateRequestString 42](#_Toc86314339)

[7.2.3. ReadAllRequests 43](#_Toc86314340)

[7.2.4. ReadRequestsByCaptureID 44](#_Toc86314341)

[7.2.5. ReadRequestByID 45](#_Toc86314342)

[7.2.6. UpdateRequest 46](#_Toc86314343)

[7.2.7. DeleteRequestByID 47](#_Toc86314344)

[7.2.8. DeleteRequestByCaptureID 48](#_Toc86314345)

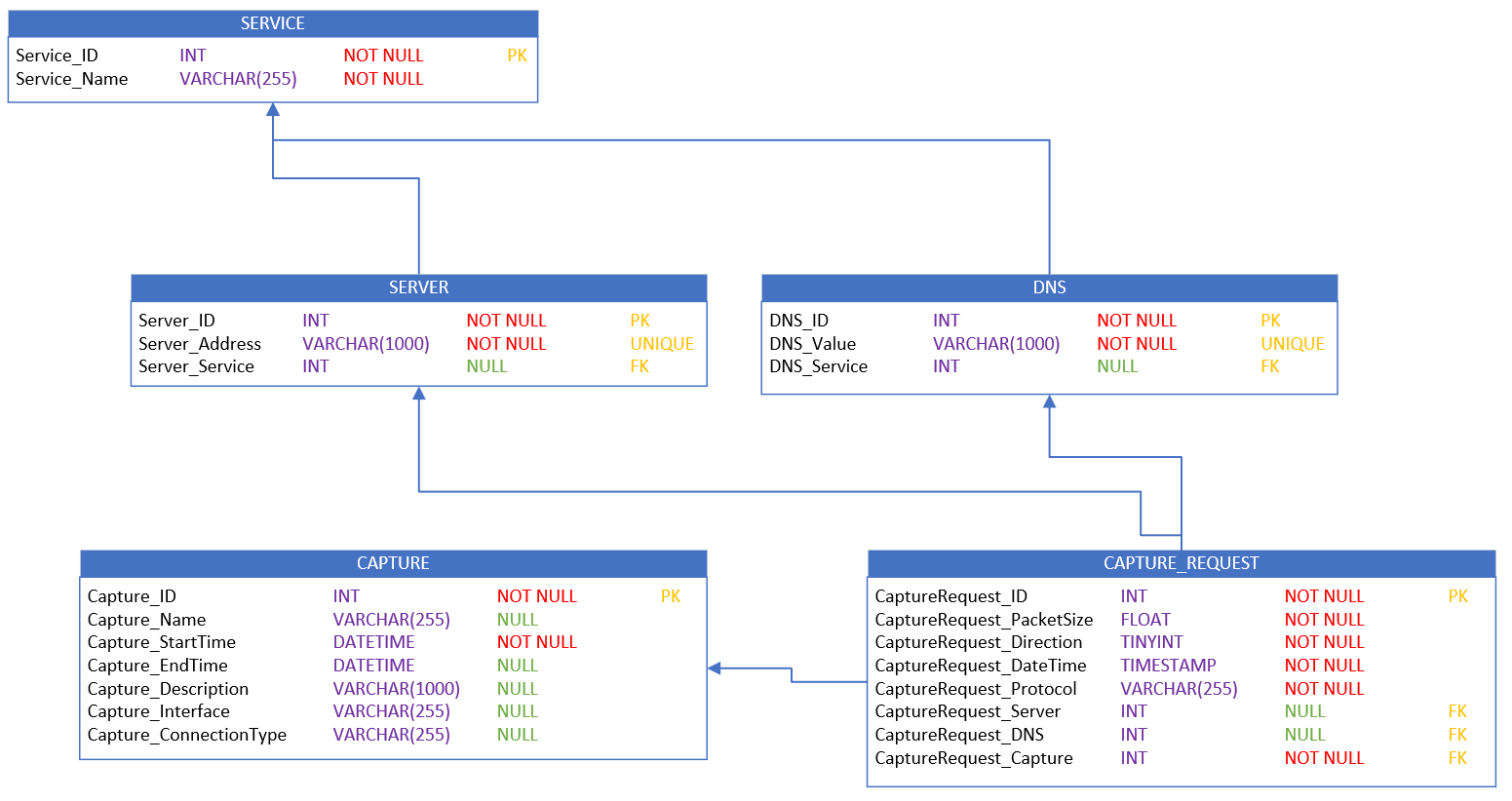
# General informations

## 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

# 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

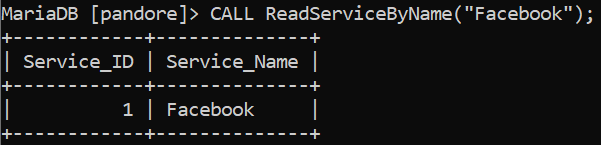


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| 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 3 attributes :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Allow NULL | Constraint | Description |
| Server\_ID | INT |  | Primary Key | Service ID |
| Server\_Address | VARCHAR(1000) |  | Unique | Service Name |
| Server\_Service | INT |  | 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 |
| Service | INT |  | 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

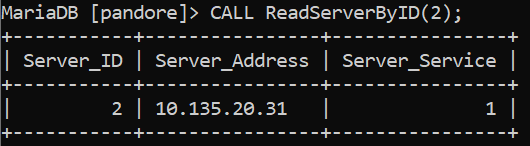


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| 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

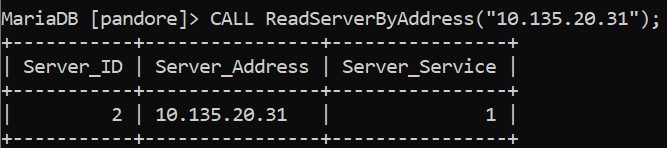


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |  | New service 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 3 attributes :

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

## 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 |
| Service | INT |  | ID of the associated service |

#### Example



### ReadAllDNS

#### Description

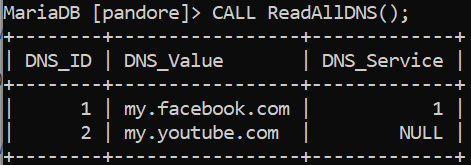
This procedure allows to list all DNS in the database

#### Prototype



#### Returns

DNS list ordered alphabetically



#### Example



### ReadDNSByServiceID

#### Description

This procedure allows to list all DNS for a given service

#### Prototype

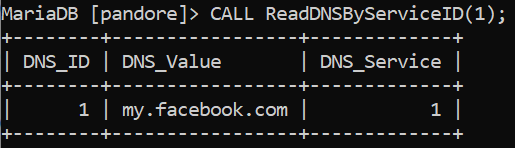


#### Parameters

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

#### Returns

List of DNS associated to given service



#### Example



### ReadDNSByID

#### Description

This procedure allows to find a DNS by its ID

#### Prototype

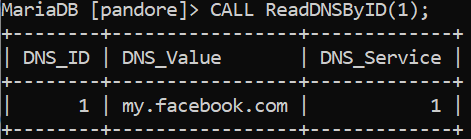


#### Parameters

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

#### Returns

DNS found with given ID



#### Example



### ReadServerByAddress

#### Description

This procedure allows to find a DNS by its value

#### Prototype

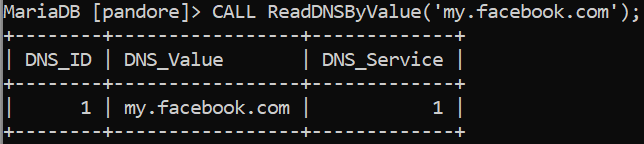


#### Parameters

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

#### Returns

DNS found with given value



#### 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 |
| Service | INT |  | New service ID |

#### 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



### DeleteDNSByServiceID

#### Description

This procedure allows to delete DNSs associated to given service

#### Prototype



#### Parameters

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

#### 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

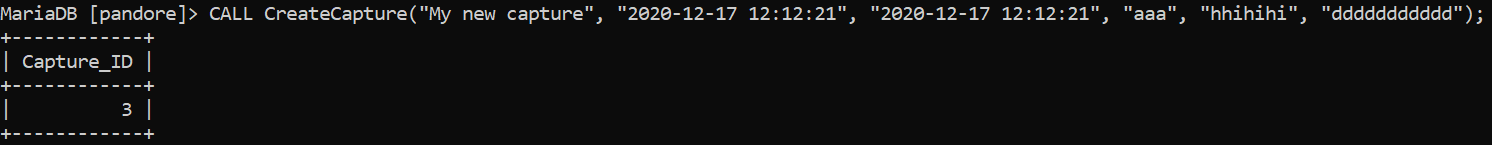
Description générée automatiquement

#### 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

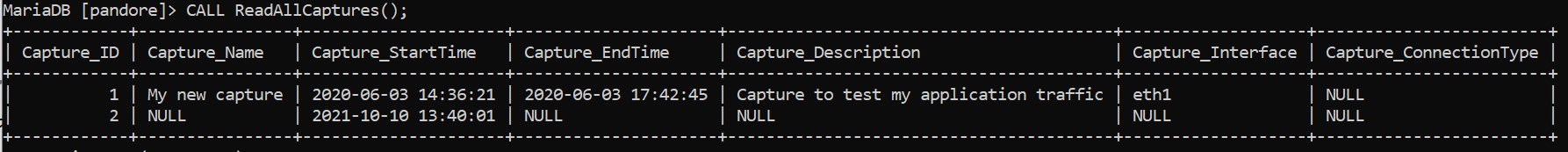
This procedure allows to list all captures

#### Prototype



#### Returns

List of all captures



#### Example



### ReadCaptureByID

#### Description

This procedure allows to find a capture by its ID

#### Prototype

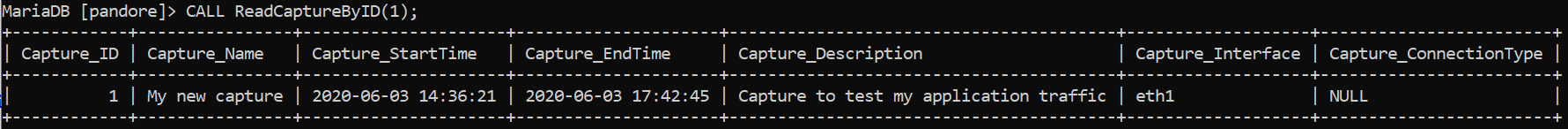


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| 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 8 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\_DNS | INT |  | Foreign Key | ID of the DNS |
| 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 | x | ID of the server |
| DNS | INT | x | ID of the DNS |
| 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

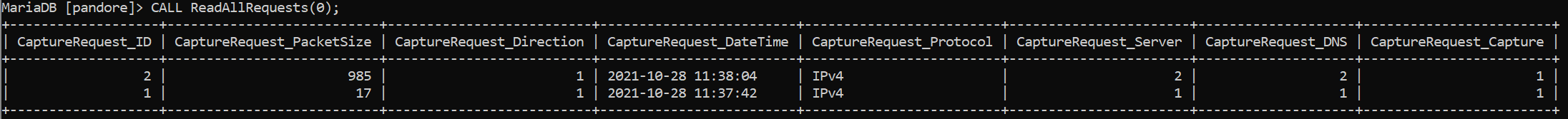


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| 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

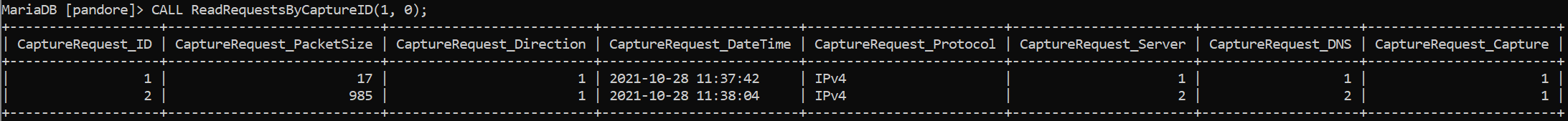


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| 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

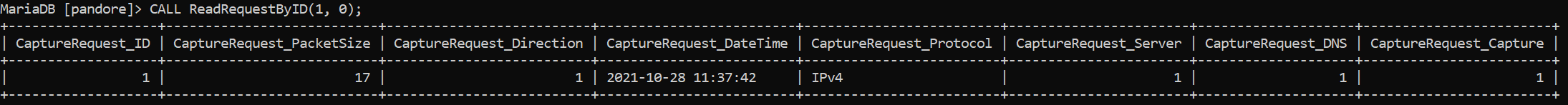


#### Parameters

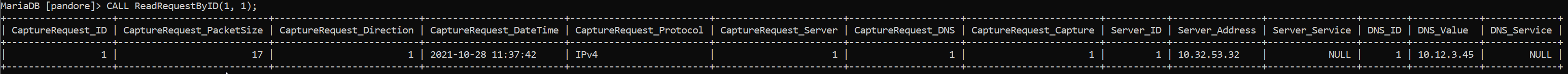
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Allow NULL | Description |
| ID | INT |  | ID of the request to update |
| Details | TINYINT(1) |  | 0 = returns list of requests  1 = returns list of requests with join with DNS and server tables |

#### 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

