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](#_Toc85729827)

[1.1. Server 4](#_Toc85729828)

[1.2. Database architecture 4](#_Toc85729829)

[2. Table Service 5](#_Toc85729830)

[2.1. Description 5](#_Toc85729831)

[2.2. Stored procedures 6](#_Toc85729832)

[2.2.1. CreateService 6](#_Toc85729833)

[2.2.2. ReadAllServices 7](#_Toc85729834)

[2.2.3. ReadServiceByID 8](#_Toc85729835)

[2.2.4. ReadServiceByName 9](#_Toc85729836)

[2.2.5. UpdateService 10](#_Toc85729837)

[2.2.6. DeleteServiceByID 11](#_Toc85729838)

[2.2.7. DeleteServiceByName 12](#_Toc85729839)

[3. Table Server 13](#_Toc85729840)

[3.1. Description 13](#_Toc85729841)

[3.2. Stored procedures 14](#_Toc85729842)

[3.2.1. CreateServer 14](#_Toc85729843)

[3.2.2. ReadAllServers 15](#_Toc85729844)

[3.2.3. ReadServersByServiceID 16](#_Toc85729845)

[3.2.4. ReadServerByID 17](#_Toc85729846)

[3.2.5. ReadServerByAddress 18](#_Toc85729847)

[3.2.6. UpdateServer 19](#_Toc85729848)

[3.2.7. DeleteServerByID 20](#_Toc85729849)

[3.2.8. DeleteServerByAddress 21](#_Toc85729850)

[3.2.9. DeleteServerByServiceID 22](#_Toc85729851)

[4. Table DNS 23](#_Toc85729852)

[4.1. Description 23](#_Toc85729853)

[4.2. Stored procedures 24](#_Toc85729854)

[4.2.1. CreateDNS 24](#_Toc85729855)

[4.2.2. ReadAllDNS 25](#_Toc85729856)

[4.2.3. ReadDNSByServiceID 26](#_Toc85729857)

[4.2.4. ReadDNSByID 27](#_Toc85729858)

[4.2.5. ReadServerByAddress 28](#_Toc85729859)

[4.2.6. UpdateDNS 29](#_Toc85729860)

[4.2.7. DeleteDNSByID 30](#_Toc85729861)

[4.2.8. DeleteDNSByValue 31](#_Toc85729862)

[4.2.9. DeleteDNSByServiceID 32](#_Toc85729863)

[5. Table Capture 33](#_Toc85729864)

[5.1. Description 33](#_Toc85729865)

[5.2. Stored procedures 34](#_Toc85729866)

[5.2.1. CreateCapture 34](#_Toc85729867)

[5.2.2. ReadAllCaptures 35](#_Toc85729868)

[5.2.3. ReadCaptureByID 36](#_Toc85729869)

[5.2.4. UpdateCapture 37](#_Toc85729870)

[5.2.5. DeleteCaptureByID 38](#_Toc85729871)

[6. Table Capture\_Request 39](#_Toc85729872)

[6.1. Description 39](#_Toc85729873)

[6.2. Stored procedures 40](#_Toc85729874)

[6.2.1. CreateRequest 40](#_Toc85729875)

[6.2.2. CreateRequestString 41](#_Toc85729876)

[6.2.3. ReadAllRequests 42](#_Toc85729877)

[6.2.4. ReadRequestsByCaptureID 43](#_Toc85729878)

[6.2.5. ReadRequestByID 44](#_Toc85729879)

[6.2.6. UpdateRequest 45](#_Toc85729880)

[6.2.7. DeleteRequestByID 46](#_Toc85729881)

[6.2.8. DeleteRequestByCaptureID 47](#_Toc85729882)

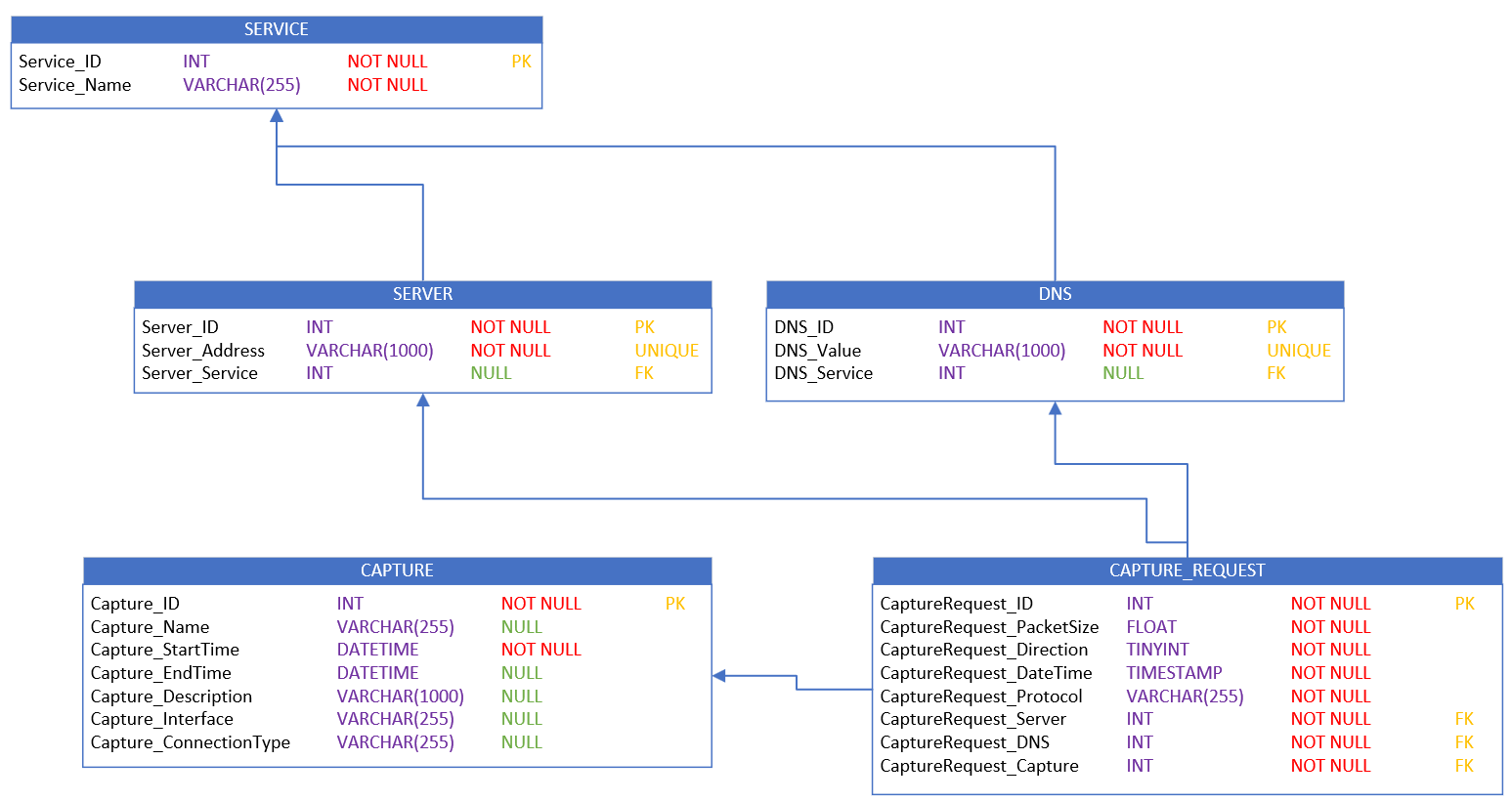
# General informations

## 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:



# 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

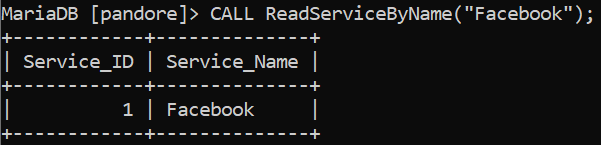


#### Parameters

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

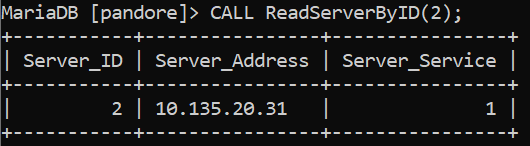


#### Parameters

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

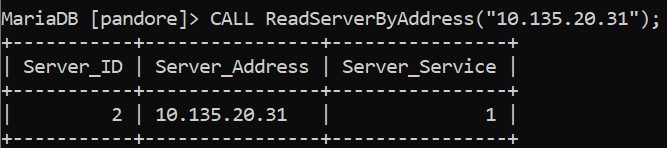


#### Parameters

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

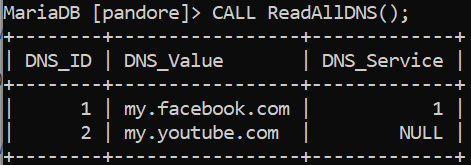
This procedure allows to list all DNS in the database

#### Prototype



#### Returns

DNS list ordered alphabetically



#### Example



### ReadDNSByServiceID

#### Description

This procedure allows to list all DNS for a given service

#### Prototype

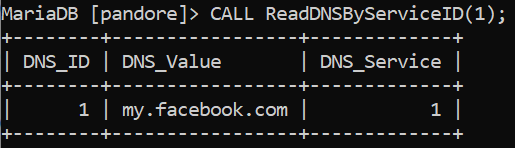


#### Parameters

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

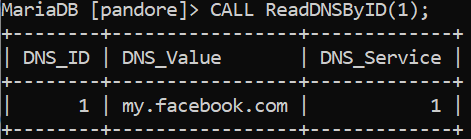


#### Parameters

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

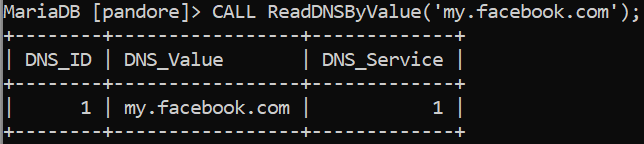


#### Parameters

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

#### Example

Une image contenant texte

Description générée automatiquement

### ReadAllCaptures

#### Description

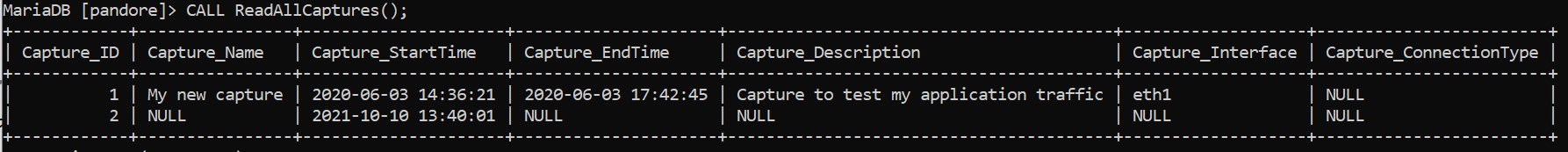
This procedure allows to list all captures

#### Prototype



#### Returns

List of all captures



#### Example



### ReadCaptureByID

#### Description

This procedure allows to find a capture by its ID

#### Prototype

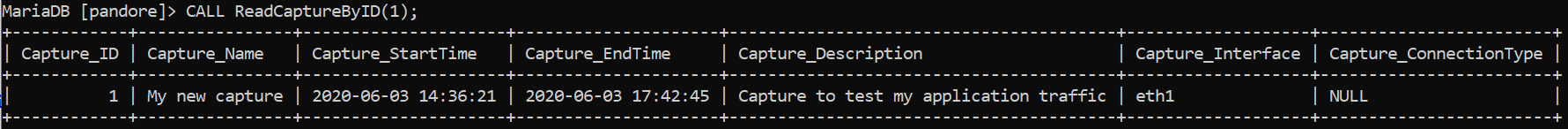


#### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |  | ID of the server |
| DNS | INT |  | 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.

#### 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) |  | Server address |
| DNS | VARCHAR(1000) |  | DNS value |
| Capture | INT |  | ID of the capture |

#### Example



### ReadAllRequests

#### Description

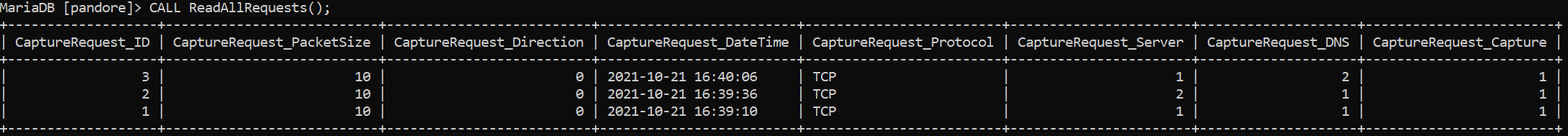
This procedure allows to list all capture requests

#### Prototype



#### Returns

List of all capture requests



#### Example



### ReadRequestsByCaptureID

#### Description

This procedure allows to list all requests for a given capture

#### Prototype

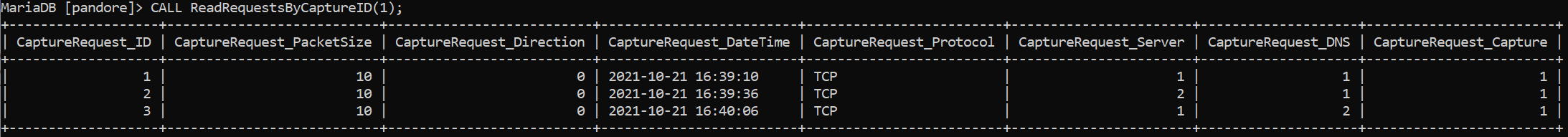


#### Parameters

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

#### Returns

List of all capture requests for the given capture ID



#### Example



### ReadRequestByID

#### Description

This procedure allows to find a request by its ID

#### Prototype

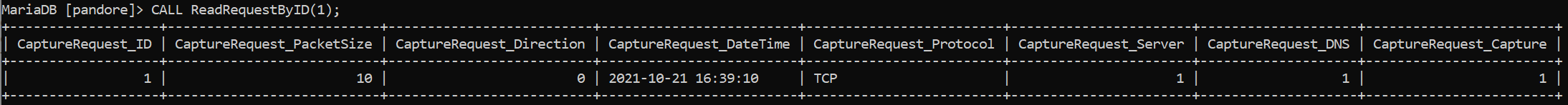


#### Parameters

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

#### Returns

The request found for the given request ID



#### 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

