Pandore – Database developer guide

This document is a developer guide for the Pandore project database

Clément LE GRUIEC
Salma CHAHMI
Nathan OLBORSKI
Hugo HOUILLON

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Table des matières

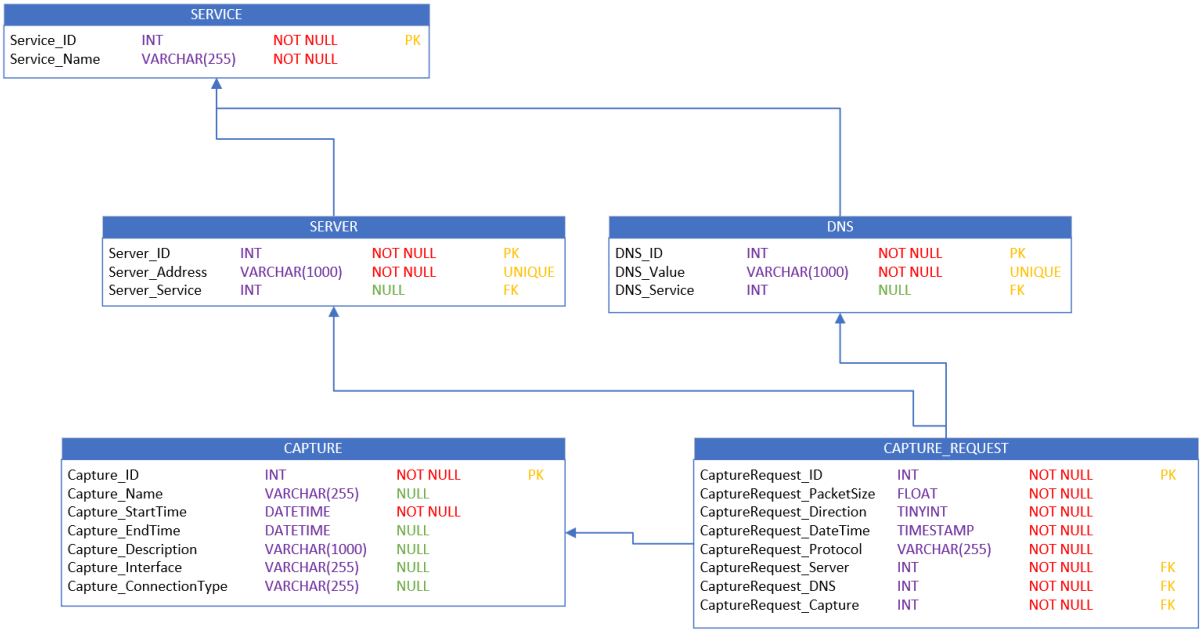
1. General informations

1.1. Server

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

1.2. Database architecture

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



2. Table Service

2.1. 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

2.2. Stored procedures

2.2.1. CreateService

2.2.1.1. Description

This procedure allows to create a new service into the database

2.2.1.2. Prototype

```
CreateService (IN Name VARCHAR(255))
```

2.2.1.3. Parameters

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

2.2.1.4. Example

```
CALL CreateService("Facebook");
```

2.2.2. ReadAllServices

2.2.2.1. Description

This procedure allows to list all services in the database

2.2.2.2. Prototype

```
ReadAllServices()
```

2.2.2.3. Returns

Service list ordered alphabetically

```
MariaDB [pandore]> CALL ReadAllServices();
+------------+--------------+
| Service_ID | Service_Name |
+------------+--------------+
|          1 | Facebook     |
|          3 | Google       |
|          2 | Youtube      |
+------------+--------------+
```

2.2.2.4. Example

```
CALL ReadAllServices();
```

2.2.3. ReadServiceByID

2.2.3.1. Description

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

2.2.3.2. Prototype

```
ReadServiceByID(IN ID INT)
```

2.2.3.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		Service ID to find

2.2.3.4. Returns

The service found with the given ID

```
MariaDB [pandore]> CALL ReadServiceByID(1);
+------------+--------------+
| Service_ID | Service_Name |
+------------+--------------+
|          1 | Facebook     |
+------------+--------------+
```

2.2.3.5. Example

```
CALL ReadServiceByID(1);
```

2.2.4. ReadServiceByName

2.2.4.1. Description

This procedure allows to find a service by its name

2.2.4.2. Prototype

```
ReadServiceByName(IN Name VARCHAR(255))
```

2.2.4.3. Parameters

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

2.2.4.4. Returns

Services found with the given service name

```
MariaDB [pandore]> CALL ReadServiceByName("Facebook");
+------------+--------------+
| Service_ID | Service_Name |
+------------+--------------+
|          1 | Facebook     |
+------------+--------------+
```

2.2.4.5. Example

```
CALL ReadServiceByName("Facebook");
```

2.2.5. UpdateService

2.2.5.1. Description

This procedure allows to update an existing service

2.2.5.2. Prototype

```
UpdateService(IN ID INT, IN Name VARCHAR(255))
```

2.2.5.3. Parameters

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

2.2.5.4. Example

```
CALL UpdateService(1, "Google");
```

2.2.6. DeleteServiceByID

2.2.6.1. Description

This procedure allows to delete a service by its ID

2.2.6.2. Prototype

```
DeleteServiceByID (IN ID INT)
```

2.2.6.3. Parameters

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

2.2.6.4. Example

```
CALL DeleteServiceByID(1);
```

2.2.7. DeleteServiceByName

2.2.7.1. Description

This procedure allows to delete a service by its name

2.2.7.2. Prototype

```
DeleteServiceByName (IN Name VARCHAR(255))
```

2.2.7.3. Parameters

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

2.2.7.4. Example

```
CALL DeleteServiceByName("Facebook");
```

3. Table Server

3.1. 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

3.2. Stored procedures

3.2.1. CreateServer

3.2.1.1. Description

This procedure allows to create a new server in the database

3.2.1.2. Prototype

```
CreateServer(IN Address VARCHAR(1000), IN Service INT)
```

3.2.1.3. Parameters

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

3.2.1.4. Example

```
CALL CreateServer("10.135.20.32", 1);
```

3.2.2. ReadAllServers

3.2.2.1. Description

This procedure allows to list all servers in the database

3.2.2.2. Prototype

```
ReadAllServers()
```

3.2.2.3. Returns

Server list ordered alphabetically

```
MariaDB [pandore]> CALL ReadAllServers();
+-----------+----------------+----------------+
| Server_ID | Server_Address | Server_Service |
+-----------+----------------+----------------+
|         2 | 10.135.20.31   |              1 |
|         3 | 10.135.20.37   |              1 |
|         4 | 125.15.02.171  |              2 |
+-----------+----------------+----------------+
```

3.2.2.4. Example

```
CALL ReadAllServices();
```

3.2.3. ReadServersByServiceID

3.2.3.1. Description

This procedure allows to list all servers for a given service

3.2.3.2. Prototype

```
ReadServersByServiceID(IN Service INT)
```

3.2.3.3. Parameters

Name	Type	Allow NULL	Description
Service	INT		ID of the service

3.2.3.4. Returns

List of servers associated to given service

```
MariaDB [pandore]> CALL ReadServersByServiceID(1);
+-----------+----------------+----------------+
| Server_ID | Server_Address | Server_Service |
+-----------+----------------+----------------+
|         2 | 10.135.20.31   |              1 |
|         3 | 10.135.20.37   |              1 |
+-----------+----------------+----------------+
```

3.2.3.5. Example

```
CALL ReadServersByServiceID(1);
```

3.2.4. ReadServerByID

3.2.4.1. Description

This procedure allows to find a server by its ID

3.2.4.2. Prototype

```
ReadServerByID(IN ID INT)
```

3.2.4.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		ID of the server

3.2.4.4. Returns

Server found with given ID

```
MariaDB [pandore]> CALL ReadServerByID(2);
+-----------+----------------+----------------+
| Server_ID | Server_Address | Server_Service |
+-----------+----------------+----------------+
|         2 | 10.135.20.31   |              1 |
+-----------+----------------+----------------+
```

3.2.4.5. Example

```
CALL ReadServerByID(1);
```

3.2.5. ReadServerByAddress

3.2.5.1. Description

This procedure allows to find a server by its address

3.2.5.2. Prototype

```
ReadServerByAddress(IN Address VARCHAR(1000))
```

3.2.5.3. Parameters

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

3.2.5.4. Returns

Server found with given address

```
MariaDB [pandore]> CALL ReadServerByAddress("10.135.20.31");
+-----------+----------------+----------------+
| Server_ID | Server_Address | Server_Service |
+-----------+----------------+----------------+
|         2 | 10.135.20.31   |              1 |
+-----------+----------------+----------------+
```

3.2.5.5. Example

```
CALL ReadServerByAddress("10.135.20.31");
```

3.2.6. UpdateServer

3.2.6.1. Description

This procedure allows to update a server

3.2.6.2. Prototype

```
UpdateServer(IN ID INT, IN Address VARCHAR(1000), IN Service INT)
```

3.2.6.3. 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

3.2.6.4. Example

```
CALL UpdateServer(1, "147.112.200.11", 2);
```

3.2.7. DeleteServerByID

3.2.7.1. Description

This procedure allows to delete a server by its ID

3.2.7.2. Prototype

```
DeleteServerByID(IN ID INT)
```

3.2.7.3. Parameters

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

3.2.7.4. Example

```
CALL DeleteServerByID(1);
```

3.2.8. DeleteServerByAddress

3.2.8.1. Description

This procedure allows to delete a server by its address

3.2.8.2. Prototype

```
DeleteServerByAddress(IN Address VARCHAR(1000))
```

3.2.8.3. Parameters

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

3.2.8.4. Example

```
CALL DeleteServerByAddress("10.135.20.31");
```

3.2.9. DeleteServerByServiceID

3.2.9.1. Description

This procedure allows to delete servers associated to given service

3.2.9.2. Prototype

```
DeleteServerByServiceID(IN Service INT)
```

3.2.9.3. Parameters

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

3.2.9.4. Example

```
CALL DeleteServerByServiceID(1);
```

4. Table DNS

4.1. 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

4.2. Stored procedures

4.2.1. CreateDNS

4.2.1.1. Description

This procedure allows to create a new DNS in the database

4.2.1.2. Prototype

```
CreateDNS(IN Value VARCHAR(1000), IN Service INT)
```

4.2.1.3. Parameters

Name	Type	Allow NULL	Description
Value	VARCHAR(1000)		Value of the DNS
Service	INT		ID of the associated service

4.2.1.4. Example

```
CALL CreateDNS('my.dns.com', 1);
```

4.2.2. ReadAllDNS

4.2.2.1. Description

This procedure allows to list all DNS in the database

4.2.2.2. Prototype

```
ReadAllDNS();
```

4.2.2.3. Returns

DNS list ordered alphabetically

```
MariaDB [pandore]> CALL ReadAllDNS();
+--------+-----------------+-------------+
| DNS_ID | DNS_Value       | DNS_Service |
+--------+-----------------+-------------+
|      1 | my.facebook.com |           1 |
|      2 | my.youtube.com  |        NULL |
+--------+-----------------+-------------+
```

4.2.2.4. Example

```
CALL ReadAllDNS();
```

4.2.3. ReadDNSByServiceID

4.2.3.1. Description

This procedure allows to list all DNS for a given service

4.2.3.2. Prototype

```
ReadDNSByServiceID(IN Service INT)
```

4.2.3.3. Parameters

Name	Type	Allow NULL	Description
Service	INT		ID of the service

4.2.3.4. Returns

List of DNS associated to given service

```
MariaDB [pandore]> CALL ReadDNSByServiceID(1);
+--------+-----------------+-------------+
| DNS_ID | DNS_Value       | DNS_Service |
+--------+-----------------+-------------+
|      1 | my.facebook.com |           1 |
+--------+-----------------+-------------+
```

4.2.3.5. Example

```
CALL ReadDNSByServiceID(1);
```

4.2.4. ReadDNSByID

4.2.4.1. Description

This procedure allows to find a DNS by its ID

4.2.4.2. Prototype

```
ReadDNSByID(IN ID INT)
```

4.2.4.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		ID of the DNS

4.2.4.4. Returns

DNS found with given ID

```
MariaDB [pandore]> CALL ReadDNSByID(1);
+--------+----------------+-------------+
| DNS_ID | DNS_Value      | DNS_Service |
+--------+----------------+-------------+
|      1 | my.facebook.com |           1 |
+--------+----------------+-------------+
```

4.2.4.5. Example

```
CALL ReadDNSByID(1);
```

4.2.5. ReadServerByAddress

4.2.5.1. Description

This procedure allows to find a DNS by its value

4.2.5.2. Prototype

```
ReadDNSByValue(IN Value VARCHAR(1000))
```

4.2.5.3. Parameters

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

4.2.5.4. Returns

DNS found with given value

```
MariaDB [pandore]> CALL ReadDNSByValue('my.facebook.com');
+--------+-----------------+-------------+
| DNS_ID | DNS_Value       | DNS_Service |
+--------+-----------------+-------------+
|      1 | my.facebook.com |           1 |
+--------+-----------------+-------------+
```

4.2.5.5. Example

```
CALL ReadDNSByValue('my.facebook.com');
```

4.2.6. UpdateDNS

4.2.6.1. Description

This procedure allows to update a DNS

4.2.6.2. Prototype

```
UpdateDNS(IN ID INT, IN Value VARCHAR(1000), IN Service INT)
```

4.2.6.3. 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

4.2.6.4. Example

```
CALL UpdateDNS(1, 'my.new.dns', 2);
```

4.2.7. DeleteDNSByID

4.2.7.1. Description

This procedure allows to delete a DNS by its ID

4.2.7.2. Prototype

```
DeleteDNSByID(IN ID INT)
```

4.2.7.3. Parameters

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

4.2.7.4. Example

```
CALL DeleteDNSByID(1);
```

4.2.8. DeleteDNSByValue

4.2.8.1. Description

This procedure allows to delete a DNS by its value

4.2.8.2. Prototype

```
DeleteDNSByValue(IN Value VARCHAR(1000))
```

4.2.8.3. Parameters

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

4.2.8.4. Example

```
CALL DeleteDNSByValue('my.dns.todelete.com');
```

4.2.9. DeleteDNSByServiceID

4.2.9.1. Description

This procedure allows to delete DNSs associated to given service

4.2.9.2. Prototype

```
DeleteDNSByServiceID(IN Service INT)
```

4.2.9.3. Parameters

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

4.2.9.4. Example

```
CALL DeleteDNSByServiceID(1);
```

5. Table Capture

5.1. 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

5.2. Stored procedures

5.2.1. CreateCapture

5.2.1.1. Description

This procedure allows to create a new capture in the database

5.2.1.2. Prototype

```
CreateCapture(IN Name VARCHAR(255), IN StartTime DATETIME, IN EndTime DATETIME,
   IN Description VARCHAR(1000), IN Interface VARCHAR(1000), IN ConnectionType VARCHAR(1000))
```

5.2.1.3. 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

5.2.1.4. Example

```
CALL CreateCapture("My new capture", "2021-10-09 14:37:12", NULL,
    "My description", "eth0", NULL);
```

5.2.2. ReadAllCaptures

5.2.2.1. Description

This procedure allows to list all captures

5.2.2.2. Prototype

```
ReadAllCaptures()
```

5.2.2.3. Returns

List of all captures

```
MariaDB [pandore]> CALL ReadAllCaptures();
+------------+--------------+-------------------+-------------------+-----------------------------------+-----------------+-----------------------+
| Capture_ID | Capture_Name | Capture_StartTime | Capture_EndTime   | Capture_Description               | Capture_Interface | Capture_ConnectionType |
+------------+--------------+-------------------+-------------------+-----------------------------------+-----------------+-----------------------+
|          1 | My new capture | 2020-06-03 14:36:21 | 2020-06-03 17:42:45 | Capture to test my application traffic | eth1            | NULL                  |
|          2 | NULL         | 2021-10-10 13:40:01 | NULL              | NULL                              | NULL            | NULL                  |
+------------+--------------+-------------------+-------------------+-----------------------------------+-----------------+-----------------------+
```

5.2.2.4. Example

```
CALL ReadAllCaptures();
```

5.2.3. ReadCaptureByID

5.2.3.1. Description

This procedure allows to find a capture by its ID

5.2.3.2. Prototype

```
ReadCaptureByID(IN ID INT)
```

5.2.3.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		Capture ID

5.2.3.4. Returns

The capture found with given ID

```
MariaDB [pandore]> CALL ReadCaptureByID(1);
+------------+--------------+---------------------+---------------------+-----------------------------------+-----------------+-------------------------+
| Capture_ID | Capture_Name | Capture_StartTime   | Capture_EndTime     | Capture_Description               | Capture_Interface | Capture_ConnectionType |
+------------+--------------+---------------------+---------------------+-----------------------------------+-----------------+-------------------------+
|          1 | My new capture | 2020-06-03 14:36:21 | 2020-06-03 17:42:45 | Capture to test my application traffic | eth1            | NULL                    |
+------------+--------------+---------------------+---------------------+-----------------------------------+-----------------+-------------------------+
```

5.2.3.5. Example

```
CALL ReadAllCaptures();
```

5.2.4. UpdateCapture

5.2.4.1. Description

This procedure allows to update a capture by its ID

5.2.4.2. Prototype

```
CREATE PROCEDURE UpdateCapture(IN ID INT, IN Name VARCHAR(255), IN StartTime DATETIME, IN EndTime DATETIME,
IN Description VARCHAR(1000), IN Interface VARCHAR(1000), IN ConnectionType VARCHAR(1000))
```

5.2.4.3. 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

5.2.4.4. Example

```
CALL UpdateCapture(1, "My new capture name", "2021-10-09 14:37:12", NULL,
    "My new description", "eth0", NULL);
```

5.2.5. DeleteCaptureByID

5.2.5.1. Description

This procedure allows to delete a capture by its ID

5.2.5.2. Prototype

```
DeleteCaptureByID(IN ID INT)
```

5.2.5.3. Parameters

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

5.2.5.4. Example

```
CALL DeleteCaptureByID(1);
```

6. Table Capture_Request

6.1. 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

6.2. Stored procedures

6.2.1. CreateRequest

6.2.1.1. Description

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

6.2.1.2. Prototype

```
CreateRequest(IN PacketSize FLOAT, IN Direction TINYINT(1), IN Protocol VARCHAR(255),
IN Server INT, IN DNS INT, IN Capture INT)
```

6.2.1.3. 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

6.2.1.4. Example

```
CALL CreateRequest(17, 0, 'TCP', 1, 1, 1);
```

6.2.2. CreateRequestString

6.2.2.1. 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.

6.2.2.2. Prototype

```
CreateRequestString(IN PacketSize FLOAT, IN Direction TINYINT(1), IN Protocol VARCHAR(255),
IN Server VARCHAR(1000), IN DNS VARCHAR(1000), IN Capture INT)
```

6.2.2.3. 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

6.2.2.4. Example

```
CALL CreateRequestString(17, 0, 'TCP', '10.135.20.32', 'dns.name.com', 1);
```

6.2.3. ReadAllRequests

6.2.3.1. Description

This procedure allows to list all capture requests

6.2.3.2. Prototype

```
ReadAllRequests()
```

6.2.3.3. Returns

List of all capture requests

```
MariaDB [pandore]> CALL ReadAllRequests();
+---------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+
| CaptureRequest_ID   | CaptureRequest_PacketSize   | CaptureRequest_Direction    | CaptureRequest_DateTime   | CaptureRequest_Protocol     | CaptureRequest_Server       | CaptureRequest_DNS        | CaptureRequest_Capture      |
+---------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+
|                   3 |                          10 |                           0 | 2021-10-21 16:40:06       | TCP                         |                           1 |                         2 |                           1 |
|                   2 |                          10 |                           0 | 2021-10-21 16:39:36       | TCP                         |                           2 |                         1 |                           1 |
|                   1 |                          10 |                           0 | 2021-10-21 16:39:10       | TCP                         |                           1 |                         1 |                           1 |
+---------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+
```

6.2.3.4. Example

```
CALL ReadAllRequests();
```

6.2.4. ReadRequestsByCaptureID

6.2.4.1. Description

This procedure allows to list all requests for a given capture

6.2.4.2. Prototype

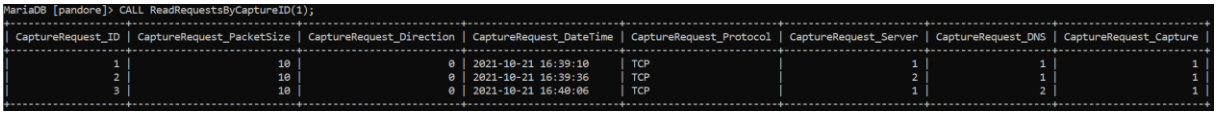
```
ReadRequestsByCaptureID(IN Capture INT)
```

6.2.4.3. Parameters

Name	Type	Allow NULL	Description
Capture	INT		ID of the capture

6.2.4.4. Returns

List of all capture requests for the given capture ID



```
MariaDB [pandore]> CALL ReadRequestsByCaptureID(1);
+---------------------+-------------------------+-----------------------------+-------------------------+-----------------------------+-----------------------------+-------------------------+-----------------------------+
| CaptureRequest_ID   | CaptureRequest_PacketSize | CaptureRequest_Direction | CaptureRequest_DateTime | CaptureRequest_Protocol | CaptureRequest_Server | CaptureRequest_DNS | CaptureRequest_Capture |
+---------------------+-------------------------+-----------------------------+-------------------------+-----------------------------+-----------------------------+-------------------------+-----------------------------+
|                   1 |                      10 |                           0 | 2021-10-21 16:39:10     | TCP                         |                           1 |                       1 |                           1 |
|                   2 |                      10 |                           0 | 2021-10-21 16:39:36     | TCP                         |                           2 |                       1 |                           1 |
|                   3 |                      10 |                           0 | 2021-10-21 16:40:06     | TCP                         |                           1 |                       2 |                           1 |
+---------------------+-------------------------+-----------------------------+-------------------------+-----------------------------+-----------------------------+-------------------------+-----------------------------+
```

6.2.4.5. Example

```
CALL ReadRequestsByCaptureID(1);
```

6.2.5. ReadRequestByID

6.2.5.1. Description

This procedure allows to find a request by its ID

6.2.5.2. Prototype

```
ReadRequestByID(IN ID INT)
```

6.2.5.3. Parameters

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

6.2.5.4. Returns

The request found for the given request ID

```
MariaDB [pandore]> CALL ReadRequestByID(1);
+--------------------+-------------------------+-------------------------+-------------------------+-------------------------+-------------------------+----------------------+-----------------------+
| CaptureRequest_ID | CaptureRequest_PacketSize | CaptureRequest_Direction | CaptureRequest_DateTime | CaptureRequest_Protocol | CaptureRequest_Server | CaptureRequest_DNS | CaptureRequest_Capture |
+--------------------+-------------------------+-------------------------+-------------------------+-------------------------+-------------------------+----------------------+-----------------------+
|                  1 |                      10 |                       0 | 2021-10-21 16:39:10     | TCP                     |                       1 |                    1 |                     1 |
+--------------------+-------------------------+-------------------------+-------------------------+-------------------------+-------------------------+----------------------+-----------------------+
```

6.2.5.5. Example

```
CALL ReadRequestByID(1);
```

6.2.6. UpdateRequest

6.2.6.1. Description

This procedure allows to update an existing capture request

6.2.6.2. Prototype

```
UpdateRequest(IN ID INT, IN PacketSize FLOAT, IN Direction TINYINT(1), IN DateTime TIMESTAMP,
IN Protocol VARCHAR(255), IN Server INT, IN DNS INT, IN Capture INT)
```

6.2.6.3. 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

6.2.6.4. Example

```
CALL CreateRequest(1, 17, 0, '2021-11-27 14:08:30', 'TCP', 1, 1, 1);
```

6.2.7. DeleteRequestByID

6.2.7.1. Description

This procedure allows to delete a capture request by its ID

6.2.7.2. Prototype

```
DeleteRequestByID(IN ID INT)
```

6.2.7.3. Parameters

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

6.2.7.4. Example

```
CALL DeleteRequestByID(1);
```

6.2.8. DeleteRequestByCaptureID

6.2.8.1. Description

This procedure allows to delete all requests for a given capture

6.2.8.2. Prototype

```
DeleteRequestByCaptureID(IN Capture INT)
```

6.2.8.3. Parameters

Name	Type	Allow NULL	Description
Capture	INT		ID of the capture

6.2.8.4. Example

```
CALL DeleteRequestByCaptureID(1);
```