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

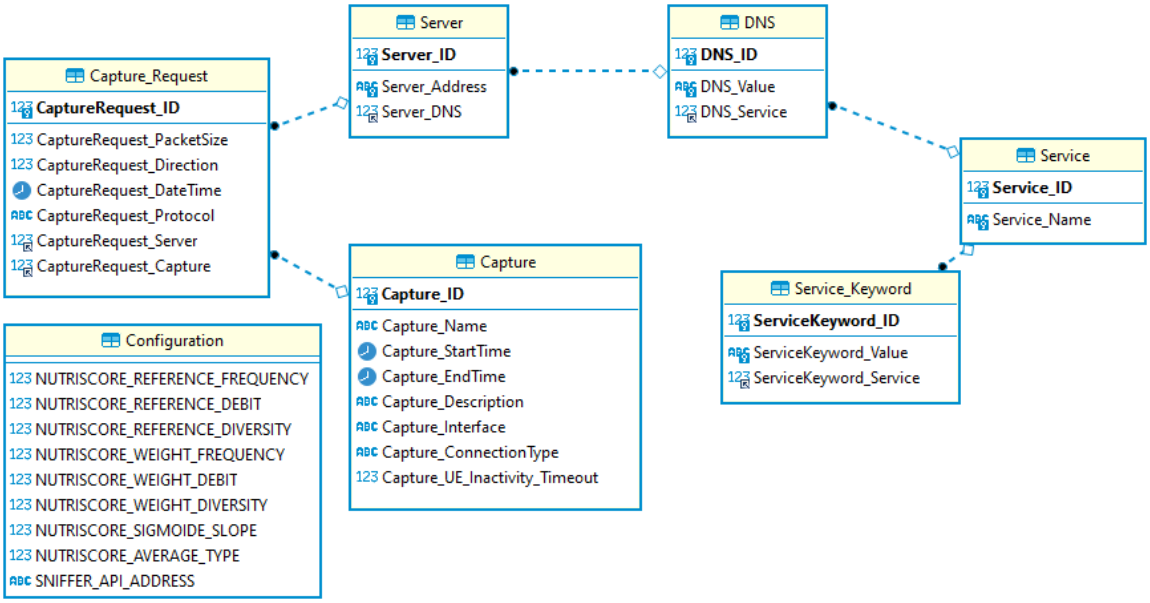
1. General informations

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. Versions

2.1. Version 1

- First database architecture implementation

2.2. 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

2.3. 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

2.4. Version 4

Too much modification to list all modifications

3. Table Configuration

3.1. Description

This table is used to store the web server configuration

It consists of 9 attributes :

Name	Type	Allow NULL	Constraint	Description
NUTRISCORE_REFERENCE_FREQUENCY	INT			Reference value for frequency for score calculation
NUTRISCORE_REFERENCE_DEBIT	FLOAT			Reference value for debit for score calculation (MB/s)
NUTRISCORE_REFERENCE_DIVERSITY	INT			Reference value for diversity for score calculation
NUTRISCORE_WEIGHT_FREQUENCY	INT			Weight of frequency parameter in the global score calculation
NUTRISCORE_WEIGHT_DEBIT	INT			Weight of debit parameter in the global score calculation
NUTRISCORE_WEIGHT_DIVERSITY	INT			Weight of diversity parameter in the global score calculation
NUTRISCORE_SIGMOIDE_SLOPE	FLOAT			Slope of the sigmoid for the score calculation
NUTRISCORE_AVERAGE_TYPE	INT			Score average type (0 = arithmetic, 1 = harmonic)
SNIFFER_API_ADDRESS	VARCHAR(1000)			Address of the probe (sniffer) API

3.2. Stored procedures

3.2.1. ReadConfiguration

3.2.1.1. Description

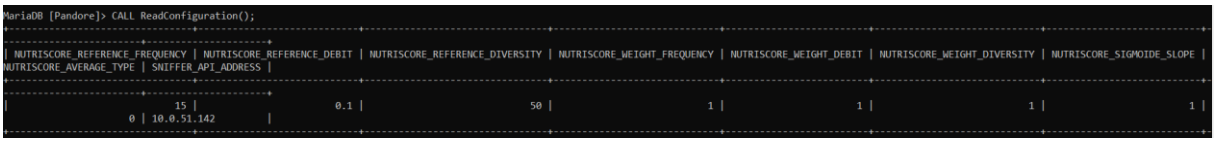
This procedure allows to get web server configuration

3.2.1.2. Prototype

```
ReadConfiguration()
```

3.2.1.3. Returns

Web server configuration



3.2.1.4. Example

```
CALL ReadConfiguration();
```

3.2.2. UpdateConfiguration

3.2.2.1. Description

This procedure allows to update the current web server configuration

3.2.2.2. Prototype

```
UpdateConfiguration(IN NUTRISCORE_REFERENCE_FREQUENCY INT, IN NUTRISCORE_REFERENCE_DEBIT FLOAT, IN NUTRISCORE_REFERENCE_DIVERSITY INT,
IN NUTRISCORE_WEIGHT_FREQUENCY INT, IN NUTRISCORE_WEIGHT_DEBIT INT, IN NUTRISCORE_WEIGHT_DIVERSITY INT, IN NUTRISCORE_SIGMOIDE_SLOPE FLOAT,
IN NUTRISCORE_AVERAGE_TYPE INT, IN SNIFFER_API_ADDRESS VARCHAR(1000))
```

3.2.2.3. Parameters

Name	Type	Allow NULL	Constraint	Description
NUTRISCORE_REFERENCE_FREQUENCY	INT			Reference value for frequency for score calculation
NUTRISCORE_REFERENCE_DEBIT	FLOAT			Reference value for debit for score calculation (MB/s)
NUTRISCORE_REFERENCE_DIVERSITY	INT			Reference value for diversity for score calculation
NUTRISCORE_WEIGHT_FREQUENCY	INT			Weight of frequency parameter in the global score calculation
NUTRISCORE_WEIGHT_DEBIT	INT			Weight of debit parameter in the global score calculation
NUTRISCORE_WEIGHT_DIVERSITY	INT			Weight of diversity parameter in the global score calculation
NUTRISCORE_SIGMOIDE_SLOPE	FLOAT			Slope of the sigmoid for the score calculation
NUTRISCORE_AVERAGE_TYPE	INT			Score average type (0 = arithmetic, 1 = harmonic)
SNIFFER_API_ADDRESS	VARCHAR(1000)			Address of the probe (sniffer) API

3.2.2.4. Example

```
CALL UpdateConfiguration(17, 0.5, 42, 1, 1, 1, 0.75, 0, "http://192.168.0.1/api");
```

4. Table Service

4.1. 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

4.2. Stored procedures

4.2.1. CreateService

4.2.1.1. Description

This procedure allows to create a new service into the database

4.2.1.2. Prototype

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

4.2.1.3. Parameters

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

4.2.1.4. Example

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

4.2.2. ReadAllServices

4.2.2.1. Description

This procedure allows to list all services in the database

4.2.2.2. Prototype

```
ReadAllServices()
```

4.2.2.3. Returns

Service list ordered alphabetically

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

4.2.2.4. Example

```
CALL ReadAllServices();
```

4.2.3. ReadServiceByID

4.2.3.1. Description

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

4.2.3.2. Prototype

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

4.2.3.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		Service ID to find

4.2.3.4. Returns

The service found with the given ID

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

4.2.3.5. Example

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

4.2.4. ReadServiceByName

4.2.4.1. Description

This procedure allows to find a service by its name

4.2.4.2. Prototype

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

4.2.4.3. Parameters

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

4.2.4.4. Returns

Services found with the given service name

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

4.2.4.5. Example

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

4.2.5. UpdateService

4.2.5.1. Description

This procedure allows to update an existing service

4.2.5.2. Prototype

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

4.2.5.3. Parameters

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

4.2.5.4. Example

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

4.2.6. DeleteServiceByID

4.2.6.1. Description

This procedure allows to delete a service by its ID

4.2.6.2. Prototype

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

4.2.6.3. Parameters

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

4.2.6.4. Example

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

4.2.7. DeleteServiceByName

4.2.7.1. Description

This procedure allows to delete a service by its name

4.2.7.2. Prototype

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

4.2.7.3. Parameters

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

4.2.7.4. Example

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

5. Table Server

5.1. 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_DNS	INT	X	Foreign Key	ID of the associated domain name

5.2. Stored procedures

5.2.1. CreateServer

5.2.1.1. Description

This procedure allows to create a new server in the database

5.2.1.2. Prototype

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

5.2.1.3. Parameters

Name	Type	Allow NULL	Description
Address	VARCHAR(1000)		Address of the server
DNS	INT	x	ID of the associated domain name

5.2.1.4. Example

```
CALL CreateServer("192.168.1.12", 1);
```

5.2.2. CreateServerString

5.2.2.1. Description

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

5.2.2.2. Prototype

```
CreateServerString(IN Address VARCHAR(1000), IN DNS VARCHAR(1000))
```

5.2.2.3. Parameters

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

5.2.2.4. Example

```
CALL CreaetServerString("10.32.65.23", "my.domain_name.com");
```

5.2.3. ReadAllServers

5.2.3.1. Description

This procedure allows to list all servers in the database

5.2.3.2. Prototype

```
ReadAllServers()
```

5.2.3.3. Returns

Server list ordered alphabetically

```
MariaDB [Pandore]> CALL ReadAllServers();
+-----------+----------------+------------+
| Server_ID | Server_Address | Server_DNS |
+-----------+----------------+------------+
|         2 | 10.43.53.233   |          2 |
|         3 | 10.43.53.234   |          2 |
|         4 | 10.43.53.235   |          2 |
|         5 | 10.43.53.237   |          2 |
|         6 | 10.9.45.78     |          3 |
|         1 | 192.168.1.12   |          1 |
+-----------+----------------+------------+
```

5.2.3.4. Example

```
CALL ReadAllServers();
```

5.2.4. ReadServerByDNSID

5.2.4.1. Description

This procedure allows to list all servers for a given domain name

5.2.4.2. Prototype

```
ReadServerByDNSID(IN DNS INT)
```

5.2.4.3. Parameters

Name	Type	Allow NULL	Description
DNS	INT		ID of the domain name

5.2.4.4. Returns

List of servers associated to given service

```
MariaDB [Pandore]> CALL ReadServerByDNSID(2);
+-----------+----------------+------------+
| Server_ID | Server_Address | Server_DNS |
+-----------+----------------+------------+
|         2 | 10.43.53.233   |          2 |
|         3 | 10.43.53.234   |          2 |
|         4 | 10.43.53.235   |          2 |
|         5 | 10.43.53.237   |          2 |
+-----------+----------------+------------+
```

5.2.4.5. Example

```
CALL ReadServerByDNSID(2);
```

5.2.5. ReadServerByID

5.2.5.1. Description

This procedure allows to find a server by its ID

5.2.5.2. Prototype

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

5.2.5.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		ID of the server

5.2.5.4. Returns

Server found with given ID

```
MariaDB [Pandore]> CALL ReadServerByID(2);
+-----------+----------------+------------+
| Server_ID | Server_Address | Server_DNS |
+-----------+----------------+------------+
|         2 | 10.43.53.233   |          2 |
+-----------+----------------+------------+
```

5.2.5.5. Example

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

5.2.6. ReadServerByAddress

5.2.6.1. Description

This procedure allows to find a server by its address

5.2.6.2. Prototype

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

5.2.6.3. Parameters

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

5.2.6.4. Returns

Server found with given address

```
MariaDB [Pandore]> CALL ReadServerByAddress("10.43.53.233");
+-----------+----------------+------------+
| Server_ID | Server_Address | Server_DNS |
+-----------+----------------+------------+
|         2 | 10.43.53.233   |          2 |
+-----------+----------------+------------+
```

5.2.6.5. Example

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

5.2.7. UpdateServer

5.2.7.1. Description

This procedure allows to update a server

5.2.7.2. Prototype

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

5.2.7.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		ID of the server to update
Address	VARCHAR(1000)		New server address
DNS	INT	x	New domain name ID

5.2.7.4. Example

```
CALL UpdateServer(1, "192.168.0.2", NULL);
```

5.2.8. DeleteServerByID

5.2.8.1. Description

This procedure allows to delete a server by its ID

5.2.8.2. Prototype

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

5.2.8.3. Parameters

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

5.2.8.4. Example

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

5.2.9. DeleteServerByAddress

5.2.9.1. Description

This procedure allows to delete a server by its address

5.2.9.2. Prototype

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

5.2.9.3. Parameters

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

5.2.9.4. Example

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

6. Table DNS

6.1. 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	x	Foreign Key	ID of the associated service

6.2. Stored procedures

6.2.1. CreateDNS

6.2.1.1. Description

This procedure allows to create a new DNS in the database

6.2.1.2. Prototype

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

6.2.1.3. Parameters

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

6.2.1.4. Example

```
CALL CreateDNS("facebook.com", 2);
```

6.2.2. ReadAllDNS

6.2.2.1. Description

This procedure allows to list all DNS in the database

6.2.2.2. Prototype

```
ReadAllDNS();
```

6.2.2.3. Returns

DNS list ordered alphabetically

```
MariaDB [Pandore]> CALL ReadAllDNS();
+--------+---------------+-------------+
| DNS_ID | DNS_Value     | DNS_Service |
+--------+---------------+-------------+
|      2 | facebook.com  |        NULL |
|      4 | instagram.com |        NULL |
|      1 | localhost     |           1 |
|      5 | tiktok.com    |        NULL |
|      3 | twitter.com   |        NULL |
+--------+---------------+-------------+
```

6.2.2.4. Example

```
CALL ReadAllDNS();
```

6.2.3. ReadDNSByID

6.2.3.1. Description

This procedure allows to find a DNS by its ID

6.2.3.2. Prototype

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

6.2.3.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		ID of the DNS

6.2.3.4. Returns

DNS found with given ID

```
MariaDB [Pandore]> CALL ReadDNSByID(1);
+--------+-----------+-------------+
| DNS_ID | DNS_Value | DNS_Service |
+--------+-----------+-------------+
|      1 | localhost |           1 |
+--------+-----------+-------------+
```

6.2.3.5. Example

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

6.2.4. ReadDNSByValue

6.2.4.1. Description

This procedure allows to find a DNS by its value

6.2.4.2. Prototype

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

6.2.4.3. Parameters

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

6.2.4.4. Returns

DNS found with given value

```
MariaDB [Pandore]> CALL ReadDNSByValue("facebook.com");
+--------+--------------+-------------+
| DNS_ID | DNS_Value    | DNS_Service |
+--------+--------------+-------------+
|      2 | facebook.com |        NULL |
+--------+--------------+-------------+
```

6.2.4.5. Example

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

6.2.5. ReadDNSByServiceID

6.2.5.1. Description

This procedure allows to list all domain names for a given service

6.2.5.2. Prototype

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

6.2.5.3. Parameters

Name	Type	Allow NULL	Description
Service	INT		Value of the service

6.2.5.4. Returns

DNS found with given service ID

```
MariaDB [Pandore]> CALL ReadDNSByServiceID(589);
+--------+-------------+-------------+
| DNS_ID | DNS_Value   | DNS_Service |
+--------+-------------+-------------+
|      2 | facebook.com |         589 |
|      6 | fbcdn.com    |         589 |
+--------+-------------+-------------+
```

6.2.5.5. Example

```
CALL ReadDNSByServiceID(589);
```

6.2.6. ReadIncompleteDNS

6.2.6.1. Description

This procedure allows to list all domain names which are not associated to any service

6.2.6.2. Prototype

```
ReadIncompleteDNS()
```

6.2.6.3. Returns

DNS found who are not associated to any service

```
MariaDB [Pandore]> CALL ReadIncompleteDNS();
+--------+---------------+-------------+
| DNS_ID | DNS_Value     | DNS_Service |
+--------+---------------+-------------+
|      3 | twitter.com   |        NULL |
|      4 | instagram.com |        NULL |
|      5 | tiktok.com    |        NULL |
+--------+---------------+-------------+
```

6.2.6.4. Example

```
CALL ReadIncompleteDNS();
```

6.2.7. UpdateDNS

6.2.7.1. Description

This procedure allows to update a DNS

6.2.7.2. Prototype

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

6.2.7.3. Parameters

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

6.2.7.4. Example

```
CALL UpdateDNS(2, "facebook.com", 4);
```

6.2.8. DeleteDNSByID

6.2.8.1. Description

This procedure allows to delete a DNS by its ID

6.2.8.2. Prototype

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

6.2.8.3. Parameters

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

6.2.8.4. Example

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

6.2.9. DeleteDNSByValue

6.2.9.1. Description

This procedure allows to delete a DNS by its value

6.2.9.2. Prototype

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

6.2.9.3. Parameters

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

6.2.9.4. Example

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

7. Table Capture

7.1. 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
Capture_UE_Inactivity_Timeout	INT			Time in seconds after which the UE is disconnected from the network

7.2. Stored procedures

7.2.1. CreateCapture

7.2.1.1. Description

This procedure allows to create a new capture in the database and returns the ID of the created capture

7.2.1.2. Prototype

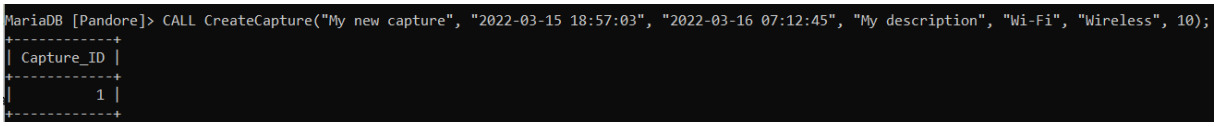
```
CreateCapture(IN Name VARCHAR(255), IN StartTime DATETIME, IN EndTime DATETIME, IN Description VARCHAR(1000),
IN Interface VARCHAR(1000), IN ConnectionType VARCHAR(1000), IN InactivityTimeout INT)
```

7.2.1.3. 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
InactivityTimeout	INT		Time in second after which UE is disconnected from the network

7.2.1.4. Returns

Returns the ID of the capture created



7.2.1.5. Example

```
CALL CreateCapture("My new capture", "2022-03-15 18:57:03",
"2022-03-16 07:12:45", "My description", "Wi-Fi", "Wireless", 10);
```

7.2.2. ReadAllCaptures

7.2.2.1. Description

This procedure allows to list all captures

7.2.2.2. Prototype

```
ReadAllCaptures()
```

7.2.2.3. Returns

List of all captures

```
MariaDB [Pandore]> CALL ReadAllCaptures();
+------------+-------------------------+---------------------+---------------------+-----------------------------------------+-----------------+----------------------+-----------------------------+
| Capture_ID | Capture_Name            | Capture_StartTime   | Capture_EndTime     | Capture_Description                     | Capture_Interface | Capture_ConnectionType | Capture_UE_Inactivity_Timeout |
+------------+-------------------------+---------------------+---------------------+-----------------------------------------+-----------------+----------------------+-----------------------------+
|          1 | My new capture          | 2022-03-15 18:57:03 | 2022-03-16 07:12:45 | My description                          | Wi-Fi           | Wireless             |                          10 |
|          2 | Capture for demonstration | 2022-06-15 04:35:12 | 2022-07-03 07:12:45 | My description for this demonstration capture | tun0            | 4G                   |                          15 |
+------------+-------------------------+---------------------+---------------------+-----------------------------------------+-----------------+----------------------+-----------------------------+
```

7.2.2.4. Example

```
CALL ReadAllCaptures();
```

7.2.3. ReadCaptureByID

7.2.3.1. Description

This procedure allows to find a capture by its ID

7.2.3.2. Prototype

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

7.2.3.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		Capture ID

7.2.3.4. 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 | Capture_UE_Inactivity_Timeout |
+------------+--------------+---------------------+---------------------+--------------------+-----------------+----------------------+-------------------------------+
|          1 | My new capture | 2022-03-15 18:57:03 | 2022-03-16 07:12:45 | My description      | Wi-Fi           | Wireless             |                            10 |
+------------+--------------+---------------------+---------------------+--------------------+-----------------+----------------------+-------------------------------+
```

7.2.3.5. Example

```
CALL ReadAllCaptures();
```

7.2.4. ReadSavedCaptures

7.2.4.1. Description

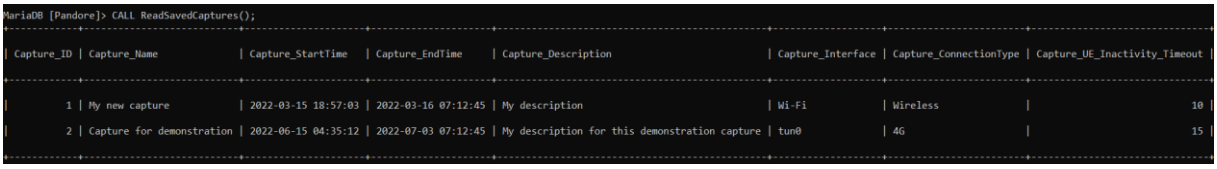
This procedure allows to list all captures who are finished (where capture end time is not null)

7.2.4.2. Prototype

```
ReadSavedCaptures()
```

7.2.4.3. Returns

All capture where Capture_EndTime is not null



```
MariaDB [Pandore]> CALL ReadSavedCaptures();
+------------+-------------------------+---------------------+---------------------+-----------------------------------------+-----------------+-----------------------+-----------------------------+
| Capture_ID | Capture_Name            | Capture_StartTime   | Capture_EndTime     | Capture_Description                     | Capture_Interface | Capture_ConnectionType | Capture_UE_Inactivity_Timeout |
+------------+-------------------------+---------------------+---------------------+-----------------------------------------+-----------------+-----------------------+-----------------------------+
|          1 | My new capture          | 2022-03-15 18:57:03 | 2022-03-16 07:12:45 | My description                          | Wi-Fi           | Wireless              |                          10 |
|          2 | Capture for demonstration | 2022-06-15 04:35:12 | 2022-07-03 07:12:45 | My description for this demonstration capture | tun0            | 4G                    |                          15 |
+------------+-------------------------+---------------------+---------------------+-----------------------------------------+-----------------+-----------------------+-----------------------------+
```

7.2.4.4. Example

```
CALL ReadSavedCaptures();
```

7.2.5. ReadCaptureServicesStats

7.2.5.1. Description

This procedure allows to find the list of services name contacted in a capture and the total download and upload traffic captures for each service. If no service name found, returns domain name instead. If no domain name found, returns server address instead.

7.2.5.2. Prototype

```
ReadCaptureServicesStats(IN ID INT)
```

7.2.5.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		Capture ID

7.2.5.4. Returns

List of all services contacted in a capture and their associated amount of downloaded and uploaded traffic

```
MariaDB [Pandore]> CALL ReadCaptureServicesStats(1);
+-------------+------------+----------+
| Name        | DownTrafic | UpTrafic |
+-------------+------------+----------+
| Facebook    |      34567 |    56789 |
| twitter.com |        679 |      567 |
+-------------+------------+----------+
```

7.2.5.5. Example

```
CALL ReadCaptureServicesStats(1);
```

7.2.6. ReadRunningCapture

7.2.6.1. Description

This procedure allows to list all captures who are not finished (where capture end time is null)

7.2.6.2. Prototype

```
ReadRunningCapture()
```

7.2.6.3. Returns

All capture where Capture_EndTime is null

```
MariaDB [Pandore]> CALL ReadRunningCapture();
+------------+--------------+---------------------+---------------+---------------------+-----------------+-----------------------+-----------------------------+
| Capture_ID | Capture_Name | Capture_StartTime   | Capture_EndTime | Capture_Description | Capture_Interface | Capture_ConnectionType | Capture_UE_Inactivity_Timeout |
+------------+--------------+---------------------+---------------+---------------------+-----------------+-----------------------+-----------------------------+
|          1 | My new capture | 2022-03-15 18:57:03 | NULL          | My description      | Wi-Fi           | Wireless              |                          10 |
+------------+--------------+---------------------+---------------+---------------------+-----------------+-----------------------+-----------------------------+
```

7.2.6.4. Example

```
CALL ReadRunningCapture();
```

7.2.7. ReadCaptureTotalTrafic

7.2.7.1. Description

This procedure allows to get the total amount of data downloaded and uploaded in bytes

7.2.7.2. Prototype

```
ReadCaptureTotalTrafic(IN Capture INT)
```

7.2.7.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		Capture ID

7.2.7.4. Returns

The total amount of data downloaded and uploaded in bytes for a given capture ID

```
MariaDB [Pandore]> CALL ReadCaptureTotalTrafic(1);
+-------+-------+
| DOWN  | UP    |
+-------+-------+
| 35246 | 57356 |
+-------+-------+
```

7.2.7.5. Example

```
CALL ReadCaptureTotalTrafic(1);
```

7.2.8. UpdateCapture

7.2.8.1. Description

This procedure allows to update a capture by its ID

7.2.8.2. Prototype

```
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), IN InactivityTimeout INT)
```

7.2.8.3. 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
InactivityTimeout	INT		New UE inactivity timeout (in second)

7.2.8.4. Example

```
CALL UpdateCapture(1, "New capture name", "2022-03-10 09:12:45",
"2022-03-11 12:34:12", "My new capture description", "Wi-Fi", "Wireless", 15);
```

7.2.9. UpdateCaptureEndTime

7.2.9.1. Description

This procedure allows to update a capture end time by its ID

7.2.9.2. Prototype

```
UpdateCaptureEndTime(IN ID INT, IN EndTime DATETIME)
```

7.2.9.3. Parameters

Name	Type	Allow NULL	Description
ID	INT		ID of the capture to update
EndTime	DATETIME	x	New capture end time. Format : yyyy-MM-dd HH:mm:ss

7.2.9.4. Example

```
CALL UpdateCaptureEndTime(1, "2022-03-17 16:12:45");
```

7.2.10. DeleteCaptureByID

7.2.10.1. Description

This procedure allows to delete a capture by its ID

7.2.10.2. Prototype

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

7.2.10.3. Parameters

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

7.2.10.4. Example

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

8. Table Capture_Request

8.1. 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

8.2. Stored procedures

8.2.1. CreateRequest

8.2.1.1. Description

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

8.2.1.2. Prototype

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

8.2.1.3. 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
Capture	INT		ID of the capture

8.2.1.4. Example

```
CALL CreateRequest(17, 0, 'TCP', 1, 3);
```

8.2.2. CreateRequestString

8.2.2.1. 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.

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.

8.2.2.2. Prototype

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

8.2.2.3. 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)	x	Server address
DNS	VARCHAR(1000)	x	DNS value
Capture	INT		ID of the capture

8.2.2.4. Example

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

8.2.3. ReadAllRequests

8.2.3.1. Description

This procedure allows to list all capture requests

8.2.3.2. Prototype

```
ReadAllRequests(IN Details TINYINT(1))
```

8.2.3.3. 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

8.2.3.4. Returns

List of all capture requests (details = 0)

```
MariaDB [pandore]> CALL ReadAllRequests(0);
```

CaptureRequest_ID	CaptureRequest_PacketSize	CaptureRequest_Direction	CaptureRequest_DateTime	CaptureRequest_Protocol	CaptureRequest_Server	CaptureRequest_DNS	CaptureRequest_Capture
2	985	1	2021-10-28 11:38:04	IPv4	2	2	1
1	17	1	2021-10-28 11:37:42	IPv4	1	1	1

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

```
MariaDB [pandore]> CALL ReadAllRequests(1);
```

CaptureRequest_ID	CaptureRequest_PacketSize	CaptureRequest_Direction	CaptureRequest_DateTime	CaptureRequest_Protocol	CaptureRequest_Server	CaptureRequest_DNS	CaptureRequest_Capture	Server_ID	Server_Address	Server_Service	DNS_ID	DNS_Value	DNS_Service
2	985	1	2021-10-28 11:38:04	IPv4	2	2	1	2	123.12.53.32	NULL	2	10.157.3.65	NULL
1	17	1	2021-10-28 11:37:42	IPv4	1	1	1	1	10.32.53.32	NULL	1	10.12.3.45	NULL

8.2.3.5. Example

```
CALL ReadAllRequests(1);
```

8.2.4. ReadRequestsByCaptureID

8.2.4.1. Description

This procedure allows to list all requests for a given capture

8.2.4.2. Prototype

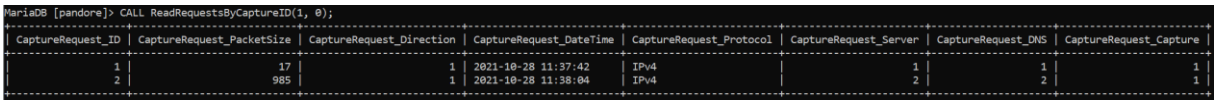
```
ReadRequestsByCaptureID(IN Capture INT, IN Details TINYINT(1))
```

8.2.4.3. 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

8.2.4.4. Returns

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



```
MariaDB [pandore]> CALL ReadRequestsByCaptureID(1, 0);
+---------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+
| CaptureRequest_ID   | CaptureRequest_PacketSize   | CaptureRequest_Direction    | CaptureRequest_DateTime     | CaptureRequest_Protocol     | CaptureRequest_Server       | CaptureRequest_DNS        | CaptureRequest_Capture      |
+---------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+
|                   1 |                          17 |                           1 | 2021-10-28 11:37:42         | IPv4                        |                           1 |                         1 |                           1 |
|                   2 |                         985 |                           1 | 2021-10-28 11:38:04         | IPv4                        |                           2 |                         2 |                           1 |
+---------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+
```

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

```
MariaDB [pandore]> CALL ReadRequestsByCaptureID(1, 1);
+---------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+-------------+-----------------+---------------+-----------+-----------------+
| CaptureRequest_ID   | CaptureRequest_PacketSize   | CaptureRequest_Direction    | CaptureRequest_DateTime     | CaptureRequest_Protocol     | CaptureRequest_Server       | CaptureRequest_DNS        | CaptureRequest_Capture      | Server_ID   | Server_Address  | Server_Service | DNS_ID    | DNS_Value       | DNS_Service     |
+---------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+-------------+-----------------+---------------+-----------+-----------------+-----------------+
|                   1 |                          17 |                           1 | 2021-10-28 11:37:42         | IPv4                        |                           1 |                         1 |                           1 |           1 | 10.32.53.32     | NULL          |         1 | 10.12.3.45      | NULL            |
|                   2 |                         985 |                           1 | 2021-10-28 11:38:04         | IPv4                        |                           2 |                         2 |                           1 |           2 | 123.12.53.32    | NULL          |         2 | 10.157.3.65     | NULL            |
+---------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+---------------------------+-----------------------------+-------------+-----------------+---------------+-----------+-----------------+-----------------+
```

8.2.4.5. Example

```
CALL ReadRequestsByCaptureID(2, 1);
```

8.2.5. ReadRequestByID

8.2.5.1. Description

This procedure allows to find a request by its ID

8.2.5.2. Prototype

```
ReadRequestByID(IN ID INT, IN Details TINYINT(1))
```

8.2.5.3. 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

8.2.5.4. Returns

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

```
MariaDB [pandore]> CALL ReadRequestByID(1, 0);
+-------------------+---------------------------+--------------------------+-------------------------+-------------------------+-----------------------+--------------------+------------------------+
| CaptureRequest_ID | CaptureRequest_PacketSize | CaptureRequest_Direction | CaptureRequest_DateTime | CaptureRequest_Protocol | CaptureRequest_Server | CaptureRequest_DNS | CaptureRequest_Capture |
+-------------------+---------------------------+--------------------------+-------------------------+-------------------------+-----------------------+--------------------+------------------------+
|                 1 |                        17 |                        1 | 2021-10-28 11:37:42     | IPv4                    |                     1 |                  1 |                      1 |
+-------------------+---------------------------+--------------------------+-------------------------+-------------------------+-----------------------+--------------------+------------------------+
```

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

```
MariaDB [pandore]> CALL ReadRequestByID(1, 1);
+-------------------+---------------------------+--------------------------+-------------------------+-------------------------+-----------------------+--------------------+------------------------+-----------+----------------+----------------+--------+-----------+-------------+
| CaptureRequest_ID | CaptureRequest_PacketSize | CaptureRequest_Direction | CaptureRequest_DateTime | CaptureRequest_Protocol | CaptureRequest_Server | CaptureRequest_DNS | CaptureRequest_Capture | Server_ID | Server_Address | Server_Service | DNS_ID | DNS_Value | DNS_Service |
+-------------------+---------------------------+--------------------------+-------------------------+-------------------------+-----------------------+--------------------+------------------------+-----------+----------------+----------------+--------+-----------+-------------+
|                 1 |                        17 |                        1 | 2021-10-28 11:37:42     | IPv4                    |                     1 |                  1 |                      1 |         1 | 10.32.53.32    |           NULL |      1 | 10.12.3.45 |        NULL |
+-------------------+---------------------------+--------------------------+-------------------------+-------------------------+-----------------------+--------------------+------------------------+-----------+----------------+----------------+--------+-----------+-------------+
```

8.2.5.5. Example

```
CALL ReadAllRequestByID(1, 1);
```

8.2.6. UpdateRequest

8.2.6.1. Description

This procedure allows to update an existing capture request

8.2.6.2. 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)
```

8.2.6.3. 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

8.2.6.4. Example

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

8.2.7. DeleteRequestByID

8.2.7.1. Description

This procedure allows to delete a capture request by its ID

8.2.7.2. Prototype

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

8.2.7.3. Parameters

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

8.2.7.4. Example

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

8.2.8. DeleteRequestByCaptureID

8.2.8.1. Description

This procedure allows to delete all requests for a given capture

8.2.8.2. Prototype

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

8.2.8.3. Parameters

Name	Type	Allow NULL	Description
Capture	INT		ID of the capture

8.2.8.4. Example

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