

*Tutoriel :*

Source WEB

“MySQL”

« br-automation.com » / Help

Tutorial **EtudeDB AGAG**

Tutoriel mappDB transtockeur



**Communication DB**

**/**

**mappDataBase »**

MS 2022/07

**Table des matières :**

[I. Généralité sur les bases de données : Concepts de base 3](#_Toc108535144)

[II. mappBDD 3](#_Toc108535145)

[1. Liaison BDD / BR : 3](#_Toc108535146)

[2. Tutoriel\_1 : Com AR / OSbdd 3](#_Toc108535147)

[1) Objectifs : 3](#_Toc108535148)

[2) BDD : Ouvrir MySQL Workbench 8.0 (ici) 3](#_Toc108535149)

[3) Se connecter à la base de données : Principe 4](#_Toc108535150)

[4) Python installé ? 5](#_Toc108535151)

[5) Utilisateur de connexion 5](#_Toc108535152)

[6) Avec l’aide d’AS de B&R : suivre les instructions 7](#_Toc108535153)

[7) Télécharger le script python nécessaire pour se connecter à la BDD 7](#_Toc108535154)

[8) Exécuter le fichier « mysql-connector-pyton… .msi télécharger ci-dessus : 7](#_Toc108535155)

[9) Le script python à définir : 8](#_Toc108535156)

[10) Lancer le script : Ouvrir le terminale Windows (dans notre cas) avec cmd 8](#_Toc108535157)

[11) Automation Studio : Projet\_AfagP70.apj 10](#_Toc108535158)

[12) En résumé 10](#_Toc108535159)

[3. Tutoriel\_2 : MpDatabase 11](#_Toc108535160)

[1) Objectifs : 11](#_Toc108535161)

[2) mappDatabase : principe 11](#_Toc108535162)

[3) Ajouter une configuration mappDatabase 12](#_Toc108535163)

[4) Créer un programme sous la tâche TDatbase 13](#_Toc108535164)

[5) Ajouter une fonction mapp : MpDatabase 13](#_Toc108535165)

[6) Tester la fonction : 14](#_Toc108535166)

[4. Tutoriel\_ 3 : MpDatabase 15](#_Toc108535167)

[1) Objectifs : 15](#_Toc108535168)

[2) BDD : Workbench 15](#_Toc108535169)

[3) Paramétrer gMpDatabaseCore pour le système AFAG : 15](#_Toc108535170)

[4) Avec l’aide AS de B&R : Paramétrer le BF 17](#_Toc108535171)

[5. Tutoriel\_ 4 : Afficher des données de la base de données sur une IHM avec l'application mappView 18](#_Toc108535172)

[1) Objectifs : 18](#_Toc108535173)

[2) Ajouter et configurer un widget "Base de données" 18](#_Toc108535174)

[III. Annexes « BDD » 19](#_Toc108535175)

[1. Si nous avons 2 mode de connexion possible à la base de donnée : 19](#_Toc108535176)

[2. Configuration exemple du BF ‘MpDatabaseCore\_0’ 20](#_Toc108535177)

# Généralité sur les bases de données : Concepts de base

**Une Base de données :**

Une base de données informatique est un ensemble de données qui ont été stockées sur un support informatique, et organisées et structurées de manière à pouvoir facilement consulter et modifier leur contenu.

**Une base de données seule ne suffit pas, il est nécessaire d'avoir également** :

* **Un système permettant de gérer cette base ;**
* **Un langage pour transmettre des instructions à la base de données (par l'intermédiaire du système de gestion).**
* Sql
* mappBDD

# mappBDD

## Liaison BDD / BR :

**Principe :**

Pour la base de données, une table Jobs sera le point d'échange entre B&R et la base de données grâce à MpDataBase coté B&R.

Pour faire fonctionner MpDataBase, un script Python doit tourner sur la machine avec la base de donnée.

Pour plus d'information, se reporter à l'aide d'automation studio pour récupérer le script Python 3.8 et le faire tourner sous Windows ou linux avec MariaDB

## Tutoriel\_1 : Com AR / OSbdd

### Objectifs :

* BDD : MySQL Workbench
* Principe : Se connecter à une base de données
* Créer une base de données
* Activer un serveur http
* Etablir la communication

### BDD : Ouvrir MySQL Workbench 8.0 (ici)

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

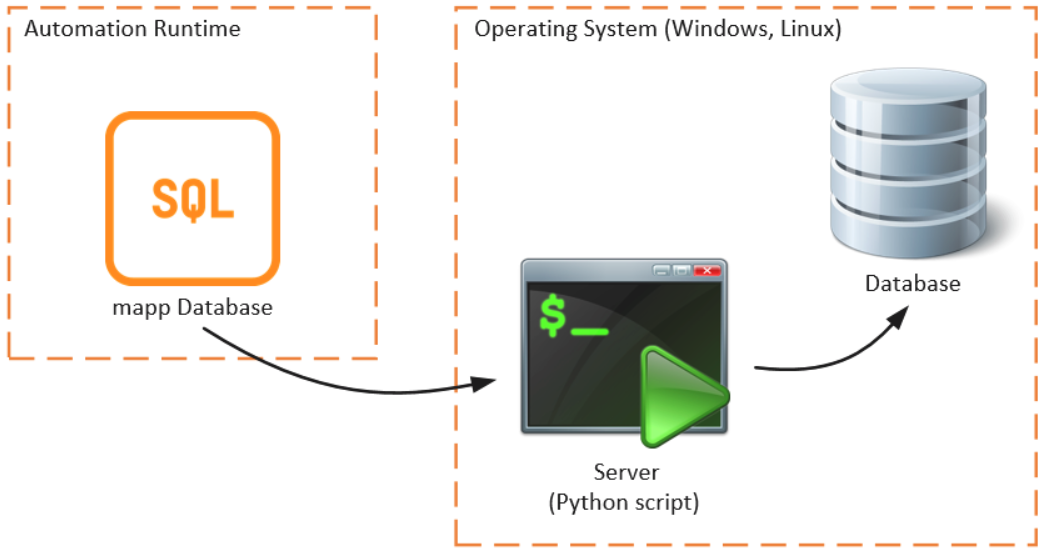
### Se connecter à la base de données : Principe

mapp Database peut être utilisé pour se connecter à une base de données. De grandes quantités de données peuvent être stockées efficacement dans une base de données.

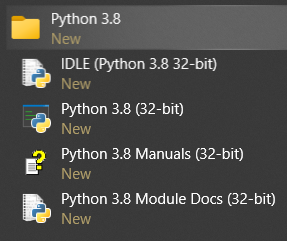
Avec mapp Database, les requêtes peuvent être utilisées dans une base de données pour créer une table et ajouter, supprimer ou modifier des entrées. L'utilisateur peut créer les requêtes à l'aide d'une interface de configuration simplifiée ou travailler directement avec des requêtes SQL. Pour plus d'informations sur le sujet, consultez la section Types de requêtes dans la configuration de MpDatabaseCore.

La communication entre le contrôleur et la base de données s'effectue via un serveur HTTP intermédiaire. Le contrôleur envoie des requêtes à un serveur HTTP. Le serveur HTTP est géré à l'aide d'un script Python. Ce script traite les données reçues du contrôleur et les transfère à la base de données sous forme de requête SQL. La base de données renvoie les données correspondantes ou exécute l'action souhaitée. L'avantage de cette solution est qu'il existe une large gamme de connecteurs de base de données différents disponibles pour les systèmes d'exploitation courants pour le PC sur lequel le script est exécuté. Pour utiliser une base de données spécifique, il suffit d'installer le connecteur approprié et éventuellement de modifier le script.

Pour une explication étape par étape de la personnalisation du script et de l'établissement de la communication, reportez-vous à la section Connexion à la base de données ci-dessous.



### Python installé ?

Vérifier que le logiciel Python version 3.8 (32 bits) est installé sur votre PC de développement. (Comme précisé dans l’aide AS

Sinon télécharger l’application sur le site : (choisir la version 32 bit)





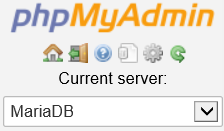
* Lors de l’exécution de l’exécutable bien vérifier avec l’ option que **🗹** « add python ……en variables » est coché.

### Utilisateur de connexion

* Créer un utilisateur de connexion BR sous PhpMyadmin entre BR et la BDD (Icon

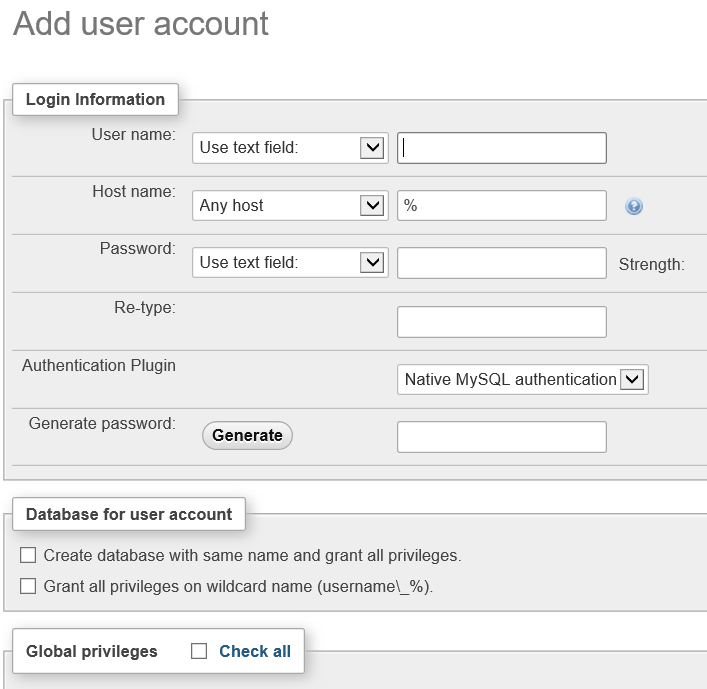
  Description automatically generatedComme sous AS)

**/ Home**



**/ User accounts / Add user account**





**BR**

**Local** (avec le menu déroulant)

123456

123456

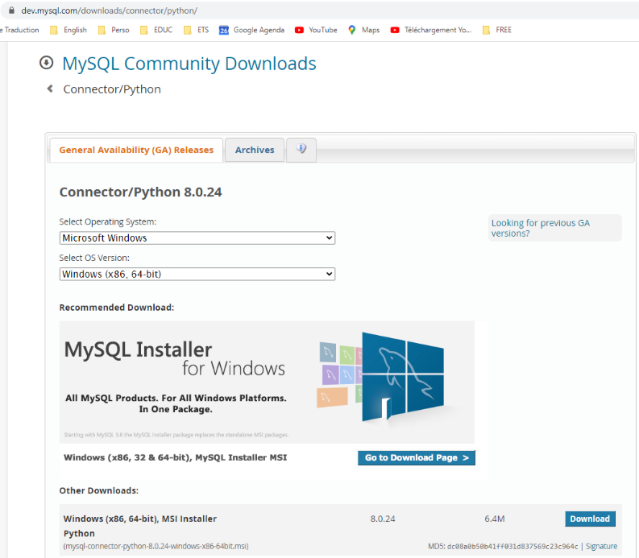
🗹 Check all

Résultat :

### Avec l’aide d’AS de B&R : suivre les instructions

Services/ mapp Services / **mapp Database**

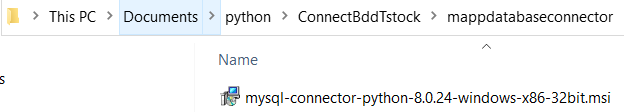




### Télécharger le script python nécessaire pour se connecter à la BDD

https://dev.mysql.com/downloads/connector/python/

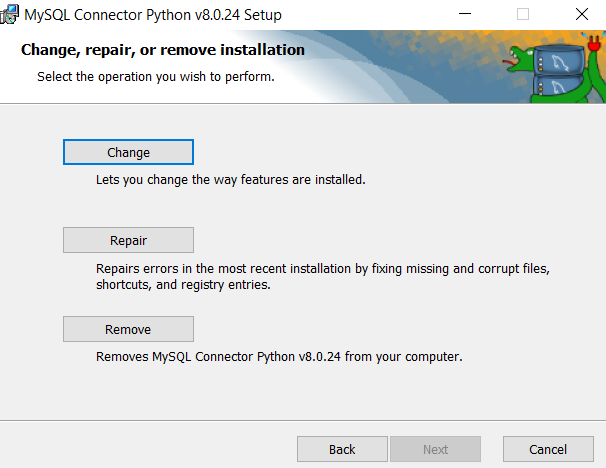
Icon

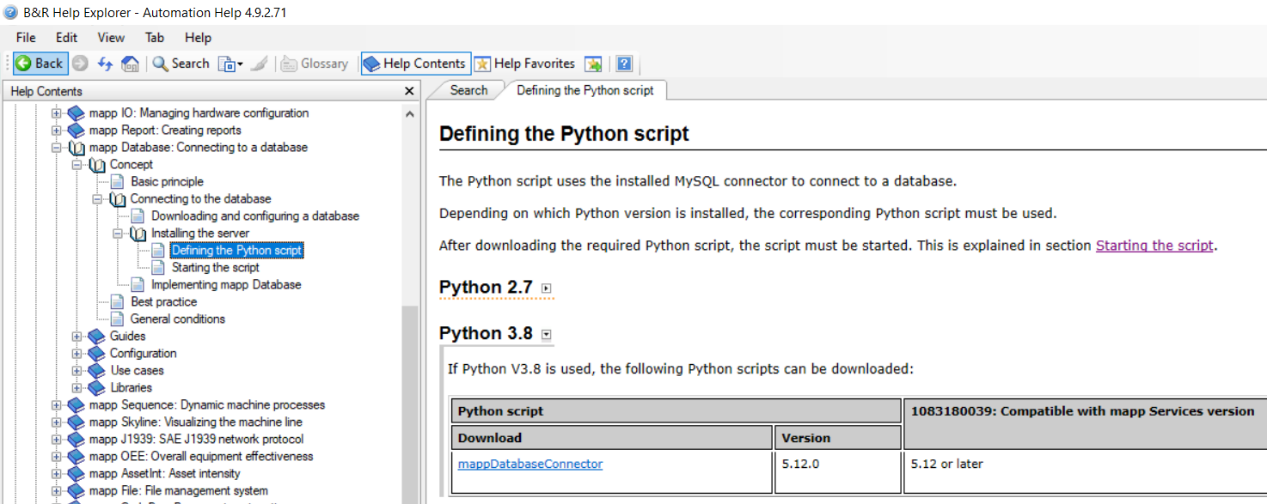
Description automatically generated Choisir la version 32 bits comme indiqué dans l’aide

### Exécuter le fichier « mysql-connector-pyton… .msi télécharger ci-dessus :

/ Next / Change / Icon

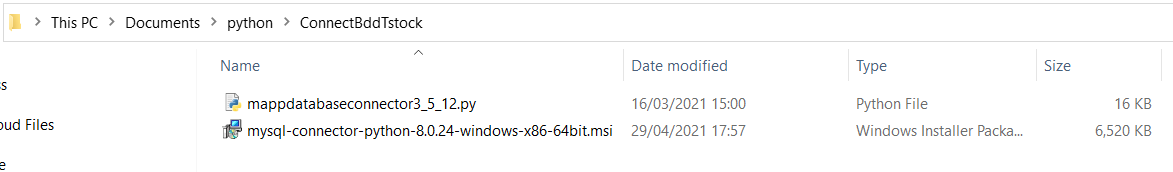
Description automatically generated choisir le connecteur pour la version 3.8/ …/ Finish





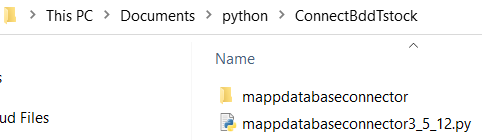
### Le script python à définir :





### Lancer le script : Ouvrir le terminale Windows (dans notre cas) avec cmd





Sachant que le script de connexion est enregistré suivant le chemin ci-dessous :

Taper : cd Documents

cd python

cd ConnectBdd

Ecrire le nom du fichier script pyton : **mappdataconnector3\_5\_12.py**

Préciser le port : **86**

L’adresse IP : **127.0.0.1**

(ou 10.16.xxx.xxx (PCbdd) ! autoriser port 86 et 3306 dans le parfeu)

Préciser le port MariaDB **3306** (ici)

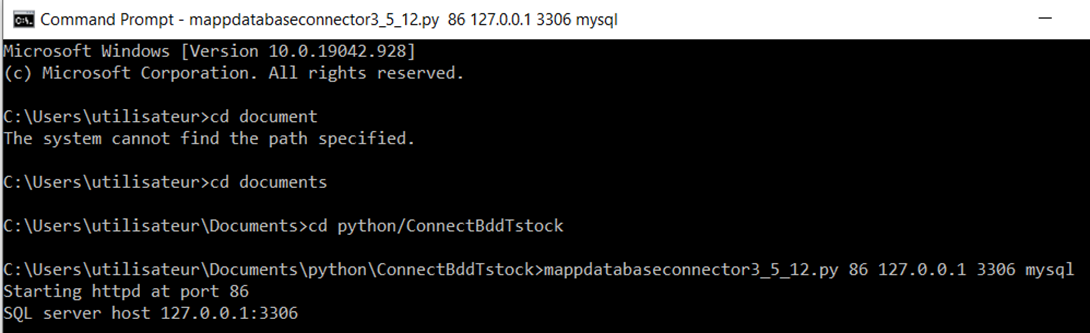
Le langage **mysql**

(⮱ voir fichier WordPad : **TstockScriptPython.txt** dans C://Documents/python/ConnectBdd)



**/Enter**

NB en réel si mon PC prof sert de pcBDD, préciser dans le script 127.0.0.1.et non 10.16.7.100 l’adresse de mon PC.

Résultat attendue :

**! Si le script ne se lance pas :**

* Sur le PC qui loge la BDD, Vérifier que le perfeu ne bloque pas le port 86, et l’autoriser dans le cas contraire

Sous windows 10 : Control panel /All control items / windows defender Firewall/ advanced setting/ inbound Rules : PYTHON DATABASE (fençetre centrale, fichier à chercher avec avec la glissière) / Action : properties (fen^tre de gauche) / onglet Protocols and ports / Local port : **86** à préciser (voir à faire de même avec le port **3306**)

Graphical user interface

Description automatically generated

* Enlever le par feu du PC BDD

### Automation Studio : Projet\_AfagP70.apj

Projet AFAG avec AS en **RUN**

* Mode simulé :

Graphical user interface, application

Description automatically generated

Icon

Description automatically generated Effectuer un transfert du projet afin d’obtenir :



### En résumé

Pour le mode simulation :

Lancer

* Graphical user interface, application

  Description automatically generated**AS** ⇨ ARsim en RUN : 
* **Wampserver**64 : 
* **Php MyAdmin** : <http://localhost:90/phpmyadmin/> (Icon

  Description automatically generated ici : port=90 au lieu de 80)

avec log = **BR**/ pwd = **123456** / BDD = AFAG

* **Le** **Terminal** avec le script python : **C:\\Users\utilisateur**\Documents\python

\ ConnectBdd>**mappdatabaseconnector3\_5\_12.py 86 127.0.0.1 3306 mysql**

Pour le mode réel :

NB en  « réel » si mon PC prof sert de pcBDD, préciser dans le script 127.0.0.1.et non 10.16.7.100 l’adresse de mon PC. (voir picture explicatif)

## Tutoriel\_2 : MpDatabase

### Objectifs :

* mappDatabase : principe
* Ajouter une configuration
* Ajouter et configurer MpDatabaseCore
* Tester la fonction

### mappDatabase : principe

⮱ source AShelp : Services / mapp Services / mapp Database / **Use case 1**

**Cas d'utilisation 1** : **Ajouter des données de production à la base de données et les interroger**

Exigence :

L'objectif est de collecter des informations statistiques sur les produits dans une base de données MySQL. Les données de la base de données devraient alors être réutilisables pour diverses analyses et calculs.

Chart, treemap chart, PowerPoint

Description automatically generatedLa solution :

Liste des composants :

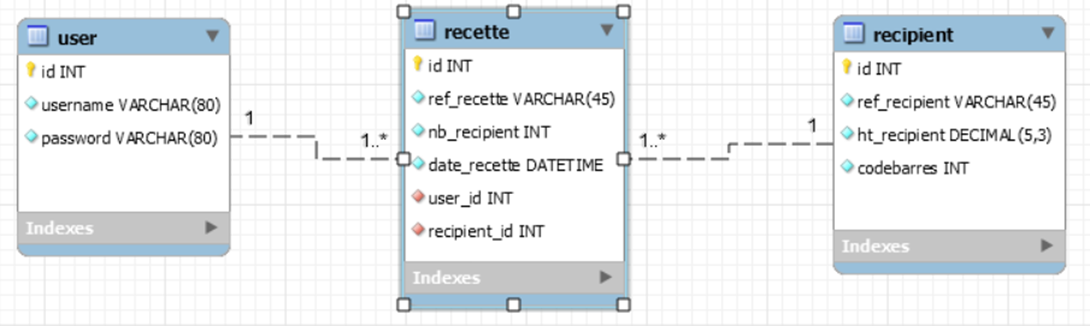
• MpDatabaseCore (MpLink personnalisé) : établit une connexion à la base de données

• MpDatabaseQuery (MpLink personnalisé) : démarre les requêtes spécifiées

Diagramme de connexion :

Base de données

Le système de base de données MySQL est utilisé. La base de données "afagdb" est créée.



La base de données comprend une ou des tables nommée(s) "user, « recette », recipient »" qui a des colonnes "id", …

Il doit être possible d'enregistrer et de demander des informations sur les produits dans le tableau.

Configuration

La configuration MpDatabaseCore est ajoutée afin d'établir une connexion à une base de données. Les paramètres de connexion doivent être spécifiés.

Une connexion à la base de données "afagdb" doit être établie. Lors de la configuration de la base de données, le nom d'utilisateur, le mot de passe, l'adresse IP et le numéro de port ont été spécifiés. Ils doivent être spécifiés dans la configuration. (Voir ci-dessous)

### Ajouter une configuration mappDatabase

Graphical user interface, text, application

Description automatically generated

Configuration View :

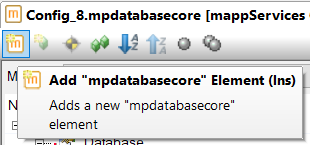
/ Toolbox

/ Search : **MpDatabase**

/ ClicG “**MpDatabaseCore**” et faire glisser sous l’arborescence mappServices

/ DclicG : “**Config**n**.mpdatabasecore**”

/ Compléter les paramètres repérer par ! sur la fonction gMpDatabaseCore pour le mode réel suivant les informations sur le screen ci-dessous

/ Ajouter un mode Simulation avec la fonction mapp : gMpDatabaseCoreSimulation

⮱ ClicG sur l’onglet Add « mpdatabasecore » Element

et compléter les paramètres repérer par ! sur la fonction suivant les informations sur le screen ci-dessous

Graphical user interface, text, application

Description automatically generated

Modifier les champs pour **gMpDatabaseCoreSimulation** et **gMpDatabaseCore** :

* Database name : **afagdb**
* User name : **BR**
* Password : **123456** (crypté par AS)
* IP address : **127.0.0.1** (mode simulation)
* Port number : **86**

Text

Description automatically generated with low confidence

### Créer un programme sous la tâche TDatbase

Logical View :

Transtockeur / Infrastructure / Toolbox / cocher : 🗹 Program

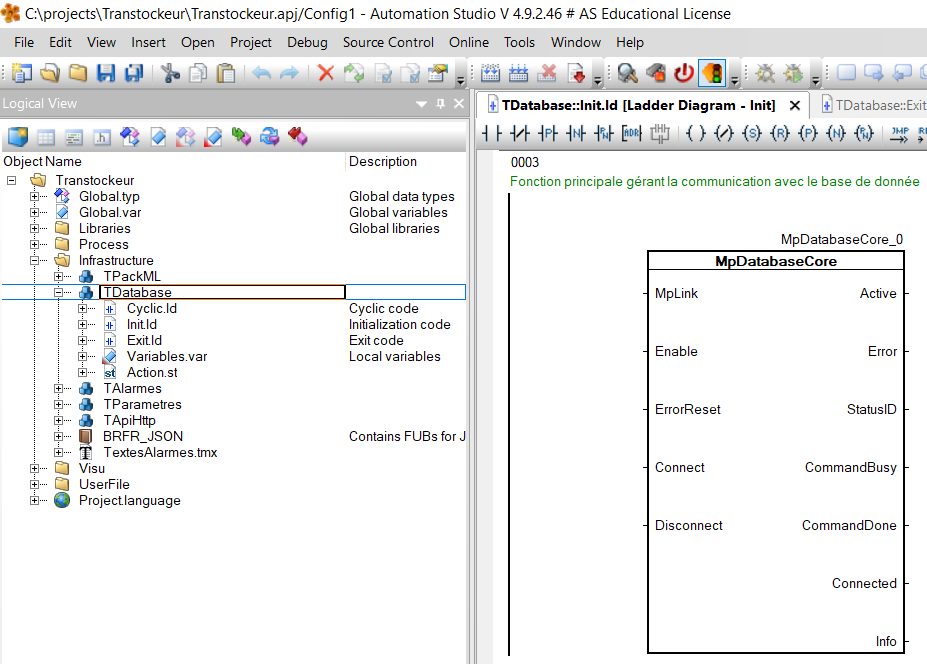
/ Dclic sur LD Program / renommer la tâche : **TDatabase**

### Ajouter une fonction mapp : MpDatabase

/ toolbox : search = **MpDatabase**

/ Sélectionner = **MpDatabaseCore**

/ Dclic sur **MpDatabaseCore**

/ Déclarer la variable **MpDatabaseCore\_0**

avec le type MpDatabaseCore et cocher **🗹 replicable** / **OK**

Icon

Description automatically generated Programme à effectuer en **Init**, **Cyclic** et **Exit** avec le même nom de variables MpDatabaseCore**\_0,…**

Graphical user interface, text, application

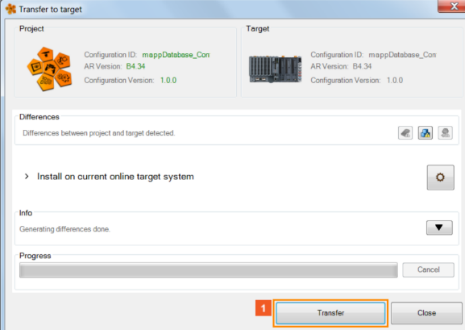
Description automatically generated/ définir l’entrée **Mplink** du bloc fonction **MpDatabaseCore**



**gDatabaseCore** utiliser sur l’entrée MpLink doit porter le même nom que la variable déclarée dans la configurationView de AS :

/ définir l’entrée EN du BF MpDatabaseCore

Ici, pas de validation de l’entrée **EN** sur le **BF MpDatabaseCore ?** (comme dans l’aide avec 1 sur EN / NB : utiliser la touche space pour le trait en lader)

/ ClicG sur l’onglet « **Transfer** »  d’AS

/ ClicG sur le bouton Transfert de la pop up

/ ClicG sur le bouton Close de la pop up

### Tester la fonction :

Une fois les modifications téléchargées, le programme peut être testé. Le mode moniteur est activé. Le bloc fonctionnel MpDatabaseCore a déjà été ajouté à la fenêtre Watch.



/ ClicG sur l’onglet Monitor d’AS

La commande "Connect" est utilisée pour établir une connexion à la base de données.

Graphical user interface

Description automatically generated with medium confidence/ClicG sur la valeur du champ « **Connect** » dans le tableau Watch

Taper la valeur **1** dans le champ « Connect » puis **Enter -> Connect** passe à **TRUE**

Graphical user interface

Description automatically generated with medium confidence

Le paramètre de sortie "**Connected**" indique que la connexion à la base de données a été établie avec succès. Cela peut également être vérifié dans la fenêtre de commande du script Python (une entrée apparaît avec l'heure actuelle).

eg : afagdb ⮷

Graphical user interface, application

Description automatically generated

Ensuite, des informations peuvent être demandées ou ajoutées à la base de données à l'aide de **MpDatabaseQuery**.

(Cas d'utilisation Les valeurs de requête de la base de données sont également disponibles à cette fin.)

## Tutoriel\_ 3 : MpDatabase

### Objectifs :

* Configurer MpDatabaseCore
* Ajout et configuration du bloc fonction MpDatabaseQuery
* Test de la fonction

### Graphical user interface, application Description automatically generatedBDD : Workbench

### Paramétrer gMpDatabaseCore pour le système AFAG :

Les requêtes doivent ensuite être configurées. Une requête est utilisée pour stocker des informations dans la table "user", « recette », « recipient » et une autre est utilisée pour lire toutes les informations de la table.

Ci-dessous le programme pour la DB : Tdatabase :

Graphical user interface, text, application

Description automatically generatedNB nous utiliserons la variable gDatabaseRecipient ce qui nous impose de l’utiliser dans notre programme :

Diagram

Description automatically generated Diagram

Description automatically generated

**Requêtes**

Graphical user interface, text, application

Description automatically generatedRequête pour demander des données de la table

La requête "GetDataRecipient" de type "SELECT" est créée. "SELECT" peut être utilisé pour demander des données à partir de la table de la base de données.

La variable spécifiée "gGetRecipientData" (au lieu de gDatabaseRecipient) est du type de données "gDatabaseRecipient\_typ". Cette variable affiche les informations de production demandées.

Graphical user interface, application

Description automatically generatedRequête pour ajouter des données à la table

La requête "SetDataRecipient" de type "INSERT" est créée. "INSERT" peut être utilisé pour ajouter des données à la table de base de données spécifiée.

Graphical user interface

Description automatically generated with medium confidenceLa variable spécifiée "gRecipientData" (au lieu de gDatabaseRecipient) est du type de données "gDatabaseRecipient\_typ". Ce type de données contient les mêmes variables que celles de la table de base de données. Il est important que les types de données soient également identiques.

**Script**

Avant que la base de données mapp puisse se connecter à la base de données, **un script doit être activé**. Il est chargé de s'assurer que les données peuvent être échangées entre la base de données et la base de données mapp. Le script peut être téléchargé dans la section Installer le serveur.

**Utilisation des composants mapp**

Les blocs fonctionnels MpDatabaseCore et MpDatabaseQuery sont ajoutés. Les blocs fonctionnels sont reliés les uns aux autres comme représenté dans le "Schéma de raccordement".

Les blocs fonctionnels sont ensuite configurés.

Les requêtes configurées peuvent être lancées à l'aide de MpDatabaseQuery. Le nom de la requête défini dans la configuration doit être spécifié dans le paramètre d'entrée "**Name**" de MpDatabaseQuery.

**"Connect = TRUE" sur MpDatabaseCore** établit une connexion à la base de données.

La requête est lancée en utilisant **"Execute = TRUE" sur MpDatabaseQuery.**

Tout d'abord, les informations de production peuvent être enregistrées dans la table de la base de données à l'aide de la requête "**SetDataRecipent**".

Ensuite, vous pouvez demander les informations de la table à l'aide de la requête "**GetDataRecipient**". Les informations se retrouvent dans la variable de processus "gGetRecipientData". D'autres analyses peuvent ensuite être effectuées à partir des données obtenues.

Graphical user interface, text, application

Description automatically generatedExécution de la requête SELECT avec l’exécution du BF « **MpDatabaseQuery\_0 »** et le nom ‘GetDataRecipient’

Résutat :

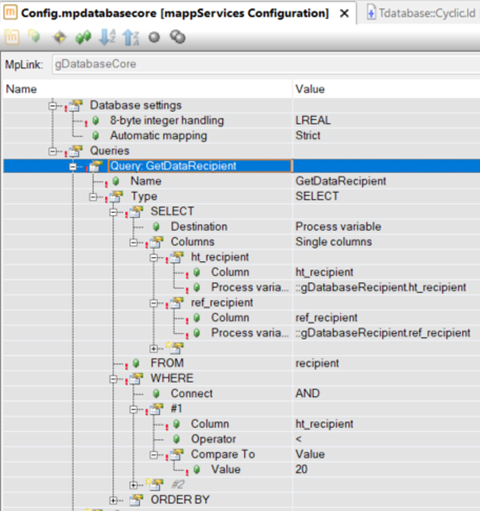
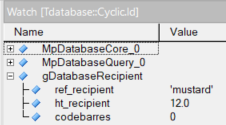
Timeline

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence

A computer screen capture

Description automatically generated with medium confidence



Résultat :



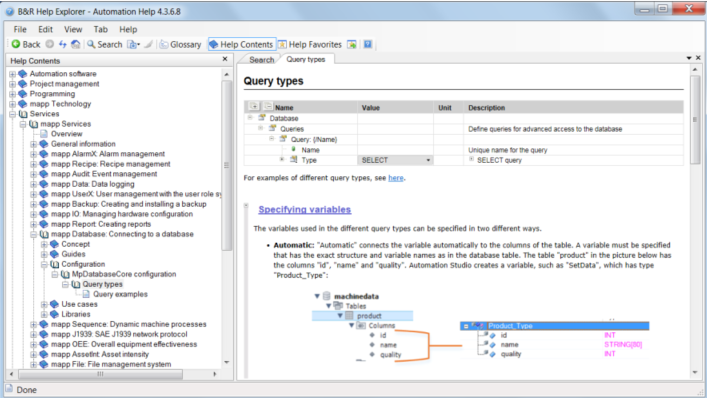
Exécution de la requête INSERT avec l’exécution du BF « **MpDatabaseQuery\_1 »** et le nom ‘SetDataRecipient’

Résultat :

A picture containing table

Description automatically generated

### Avec l’aide AS de B&R : Paramétrer le BF



Ajouter et configurer un MpDatabaseQuery

Double clicG sur le pg

ClicG sur le bouton “Append Network”

Double ClicG MpDatabaseQuery / Déclarer MpDatabaseQuery avec le type MpDatabaseQuery

Nommer le BF : MpDatabaseQuery\_0

Relier MpLink l’ « Address Contact » : ADR avec la variable gDatabaseCore

## Tutoriel\_ 4 : Afficher des données de la base de données sur une IHM avec l'application mappView

### Objectifs :

* Ajouter et configurer un widget "Base de données"
* Test de l'application IHM

### Ajouter et configurer un widget "Base de données"Graphical user interface, text, chat or text message Description automatically generated

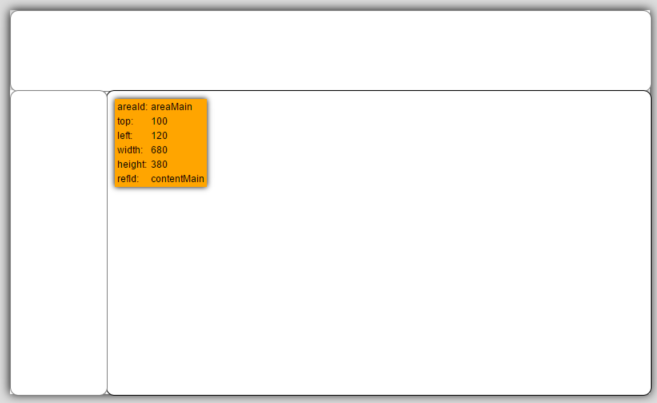
Objective :

Diagram

Description automatically generated with medium confidence

### Configurer mappView avec des pages et des contents dans la ConfigurationView et la LogicalView d’AS

Se reporter au Tutoriel « mappView »



Voit tuto 4 « mpDatabase » …

# Annexes « BDD »

## Si nous avons 2 mode de connexion possible à la base de donnée :

* en Réel
* en Simulation

(Icon

Description automatically generated ici nous avons changé le nom de la variable gDatabaseCore en gMpDatabaseCore cela engendred donc un chagement de nom aussi du nom de la variable dans la configurationView)

⮱ **Insérer la fonction MOVE :**

⮱ **Insérer la Fonction DiagCpulsSimulated**

Diagram

Description automatically generated

/ ClicG sur l’onglet « Address Contact »

et le placer devant l’entrée d’une fonction MOVE et nommer cette entrée **gMpDatabaseCore** avec l’instance 001 et **gMpDatabaseCoreSimulation** avec l’instance 002

## Configuration exemple du BF ‘MpDatabaseCore\_0’

**Application : système transtockeur :**

Construire des requêtes en SQL

/ Modifier les champs pour **gMpDatabaseCoreSimulation** et **gMpDatabaseCore** :

* Database name : afagdb
* User name : BR
* Password : 123456 (crypté par AS)
* IP address : 127.0.0.1 (mode simulation)
* Port number : 86
* Queries :

Query : **Next5Jobs**

Name : **Next5Jobs**

Type : **Custom** *(menu déroulant)*

Query : **SELECT…** (VOIR BDD)

Columns **Automatic**

Process variable : **:gDatabase.Statut.Next5Jobs**

Query : **UpdateJob**

Name : **UpdateJob**

Type : **UPDATE** *(menu déroulant)*

Table Name: **job**

Columns **Single Columns**

etat :

column : **etat**

Process variable : **:gDatabase.Param.Update.Job.Etat**

quantite :

column : **quantite**

Process variable : **:gDatabase.Param.Update.Job.Nombre**

WHERE

Connect : **AND**

#1 :

Column : **id**

Operator : **=**

Compare To : **PV**

PV : **:gDatabase.Param.Update.Job.id**