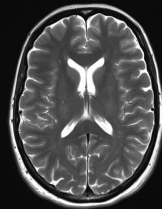
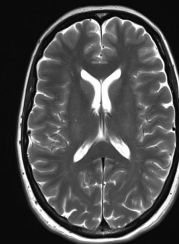


# Présentation de populse\_db, et de son utilisation dans MIA

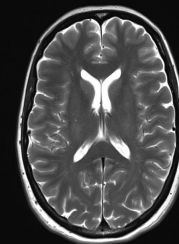


# Plan



1. Solutions
2. Filtrage
  - Rapide
  - Avancé
3. Tests
4. Base de données dans MIA
5. Transactions
6. Optimisation
7. Options de la base de données

# Base de données



- Schéma relationnel générique retenu
  - Collections (Tableau)
  - Documents (Ligne)
  - Fields (Colonne)
- A chaque Collection est associée une clé primaire (nom de la colonne permettant l'indigage des documents)
- A chaque Field sont associés une collection, un type, et une description

# Base de données

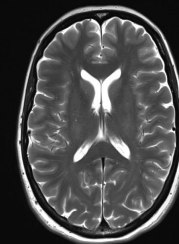
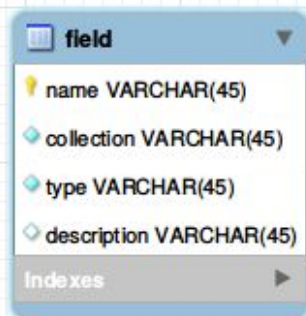


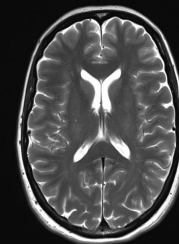
Schéma relationnel générique à la création de la base de données



Types acceptés

- chaîne de caractères
- entier
- flottant
- date
- time
- datetime
- booléen
- json
- + liste pour chaque type simple

# Base de données



## Méthodes de l'API

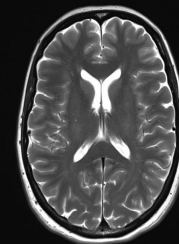
- Collections

- `add_collection(name, primary_key)`
- `remove_collection(name)`
- `get_collection(name)`
- `get_collections()`
- `get_collections_names()`

- Field

- `add_field(collection, name, field_type, description)`
- `remove_field(collection, field)`
- `get_field(collection, name)`
- `get_fields(collection)`
- `get_fields_names(collection)`

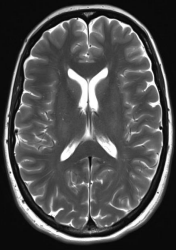
# Base de données



## Méthodes de l'API

- Document
  - `add_document(collection, document)`
  - `remove_document(collection, document)`
  - `get_document(collection, document)`
  - `get_documents(collection)`
  - `get_documents_names(collection)`
- Value
  - `new_value(collection, document, field, value)`
  - `remove_value(collection, document, field)`
  - `set_values(collection, document, values)`
  - `set_value(collection, document, field, new_value)`
  - `get_value(collection, document, field)`

# Base de données



- Etat de la base de données à sa création

| Field  |            |        |             | Collection |             |
|--------|------------|--------|-------------|------------|-------------|
| name   | collection | type   | description | name       | primary_key |
| Filter | Filter     | Filter | Filter      | Filter     | Filter      |

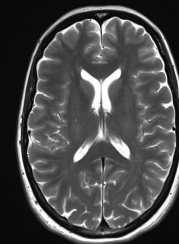
- Etat de la base de données après l'ajout d'une collection
  - `session.add_collection("collection_test", "index")`

| Field  |                 |        |  |
|--------|-----------------|--------|--|
| name   | collection      | type   | description  |
| Filter | Filter          | Filter | Filter   |
| index  | collection_test | string | Primary_key of the document collection collection_test |

| Collection      |             |
|-----------------|-------------|
| name            | primary_key |
| Filter          | Filter      |
| collection_test | index       |

| Collection_test |        |
|-----------------|--------|
| index           | Filter |

# Base de données



- Etat de la base de données après l'ajout d'un document

- `session.add_document("collection_test", "document")`

| Field  |                 |        |  |
|--------|-----------------|--------|--|
| name   | collection      | type   | description  |
| Filter | Filter          | Filter | Filter   |
| index  | collection_test | string | Primary_key of the document collection collection_test |

| Collection      |             |
|-----------------|-------------|
| name            | primary_key |
| Filter          | Filter      |
| collection_test | index       |

| Collection_test |  |
|-----------------|--|
| index           |  |
| Filter          |  |
| document        |  |

- Etat de la base de données après l'ajout d'un field

- `session.add_field("collection_test", "field", populse_db.database.FIELD_TYPE_STRING, "field test")`

| Field  |                 |        |  |
|--------|-----------------|--------|--|
| name   | collection      | type   | description  |
| Filter | Filter          | Filter | Filter   |
| index  | collection_test | string | Primary_key of the document collection collection_test |
| field  | collection_test | string | field test   |

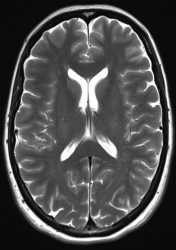
| Collection      |             |
|-----------------|-------------|
| name            | primary_key |
| Filter          | Filter      |
| collection_test | index       |

| Collection_test |                                  |
|-----------------|----------------------------------|
| index           | 06e3d36fa30cea095545139854ad1fb9 |
| Filter          | Filter                           |
| document        | NULL                             |

“field” hashé md5



# Base de données



- Etat de la base de données après l'ajout d'une nouvelle valeur
  - `session.new_value("collection_test", "document", "field", "value_test")`

Field

| name   | collection      | type   | description  |
|--------|-----------------|--------|--|
| Filter | Filter          | Filter | Filter   |
| index  | collection_test | string | Primary_key of the document collection collection_test |
| field  | collection_test | string | field test   |

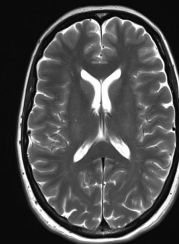
Collection

| name            | primary_key |
|-----------------|-------------|
| Filter          | Filter      |
| collection_test | index       |

Collection\_test

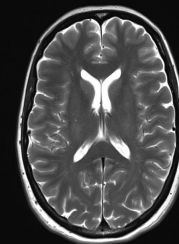
|          |                                  |
|----------|----------------------------------|
| index    | 06e3d36fa30cea095545139854ad1fb9 |
| Filter   | Filter                           |
| document | value_test                       |

# Base de données



- Choix de conception
  - Une colonne par field
    - Les valeurs doivent être typées (car  $5 < 10$  est vrai mais "5" < "10" est faux)
  - Noms de colonne des fields hashés en md5
    - Certains caractères sont interdits en SQL, et si on les retirent juste, on risque de tomber sur la même colonne avec des noms de tag très proches

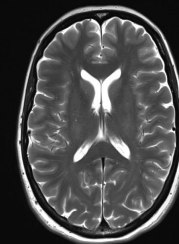
# Filtrage



## Rapide

- Recherche saisie par l'utilisateur dans la barre de recherche (chaîne de caractères)
- Recherche dans les valeurs des images et tags visibles
- Opérateur ILIKE (sensible à la casse)

# Filtrage



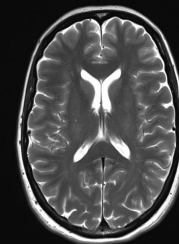
Tags ▾ Filters ▾

|   | FileName  | Bricks | Exp Type             | Type |
|---|---|--------|----------------------|------|
| 1 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-02-G1_Guerbet_Anat-RARE__pvm_-00-02-20.000.nii     |        | <b>*Not Defined*</b> | Scan |
| 2 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-04-G3_Guerbet_MDEFT-MDEFT__pvm_-00-09-40.800.nii   |        | <b>*Not Defined*</b> | Scan |
| 3 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-05-G4_Guerbet_T1SE_800-RARE__pvm_-00-01-42.400.nii |        | <b>*Not Defined*</b> | Scan |
| 4 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-06-G4_Guerbet_T1SE_800-RARE__pvm_-00-01-42.400.nii |        | <b>*Not Defined*</b> | Scan |
| 5 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-08-G4_Guerbet_T1SE_800-RARE__pvm_-00-01-42.400.nii |        | <b>*Not Defined*</b> | Scan |
| 6 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-09-G4_Guerbet_T1SE_800-RARE__pvm_-00-01-42.400.nii |        | <b>*Not Defined*</b> | Scan |
| 7 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-10-G3_Guerbet_MDEFT-MDEFT__pvm_-00-09-40.800.nii   |        | <b>*Not Defined*</b> | Scan |
| 8 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-11-G4_Guerbet_T1SE_800-RARE__pvm_-00-01-42.400.nii |        | <b>*Not Defined*</b> | Scan |

Tags ▾ Filters ▾

|   | FileName  | Bricks | Exp Type             | Type |
|---|---|--------|----------------------|------|
| 1 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-02-G1_Guerbet_Anat-RARE__pvm_-00-02-20.000.nii |        | <b>*Not Defined*</b> | Scan |

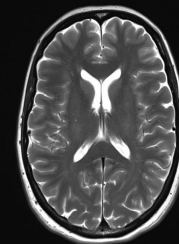
# Filtrage



## Avancé

- Ligne de filtrage
  - Colonnes (Tag ou tous les tags visibles)
  - Condition (==, !=, >, <, >=, <=, CONTAINS, IN, BETWEEN)
  - Valeur (Chaîne de caractères)
- Négation possible sur la ligne de filtrage
- Lignes de filtrage liées par AND ou OR

# Filtrage



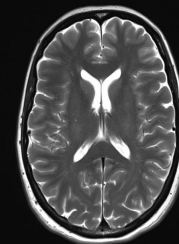
▼ FileName CONTAINS G1 Search

|   | FileName  | Bricks | Exp Type             | Type |
|---|---|--------|----------------------|------|
| 1 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-02-G1_Guerbet_Anat-RARE__pvm_-00-02-20.000.nii |        | <i>*Not Defined*</i> | Scan |

▼ FileName CONTAINS G4  
AND ▼ EchoTime == 10 Search

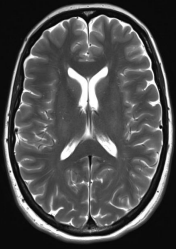
|   | FileName  | BandWidth | Bricks | ByteOrder    | EchoTime | Exp Type             | Type |
|---|---|-----------|--------|--------------|----------|----------------------|------|
| 5 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-08-G4_Guerbet_T1SE_800-RARE__pvm_-00-01-42.400.nii | 65789,48  |        | littleEndian | 10       | <i>*Not Defined*</i> | Scan |
| 6 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-09-G4_Guerbet_T1SE_800-RARE__pvm_-00-01-42.400.nii | 65789,48  |        | littleEndian | 10       | <i>*Not Defined*</i> | Scan |
| 8 | data/raw_data/Guerbet-C6-2014-Rat-K52-Tube27-2014-02-14_10-23-17-11-G4_Guerbet_T1SE_800-RARE__pvm_-00-01-42.400.nii | 65789,48  |        | littleEndian | 10       | <i>*Not Defined*</i> | Scan |

# Filtrage



- Possibilité d'enregistrer le filtrage courant, et de le recharger par la suite
  - Enregistré sous forme de fichier JSON
- Écriture d'un langage de requêtage
  - Méthode `filter_documents(collection, filter_query)`
  - Requête : chaîne de caractères
  - Filtre : `field + opérateur + valeur`
  - Filtres liées par AND ou OR
  - Intérêt : Gestion des valeurs de type liste
    - Requête SQL si possible
    - Passage au requêtage Python si SQL pas possible (listes)

# Tests

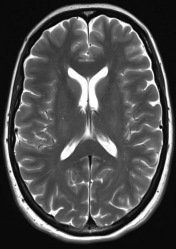


## Base de données

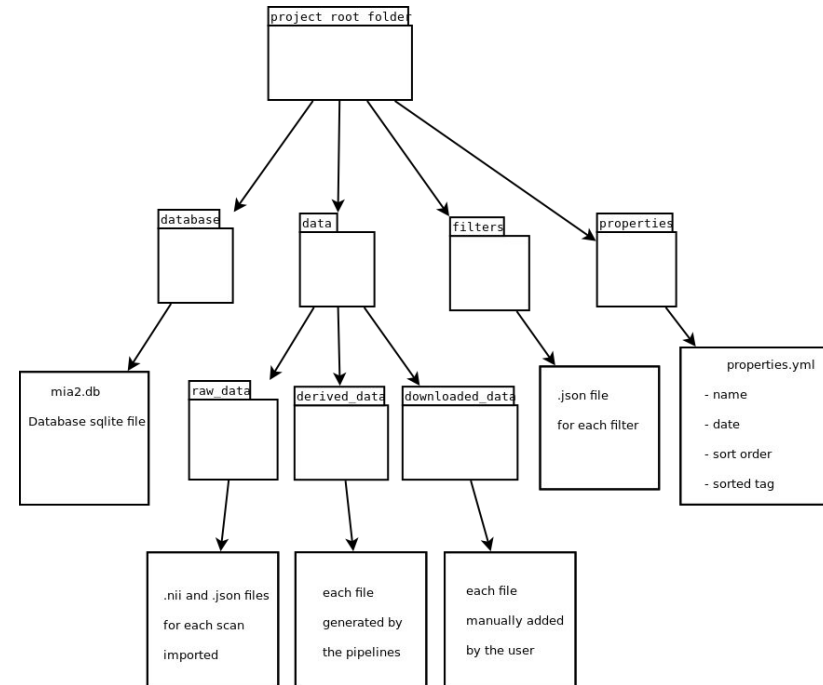
- Ecriture de tests unitaires
- module unittest
- Intégration continue avec Travis
  - Les tests sont lancés à chaque fois que le code source est modifié
- Vérifications
  - Erreur renvoyée si paramètre(s) invalides
  - Bonnes actions réalisées si paramètres valides



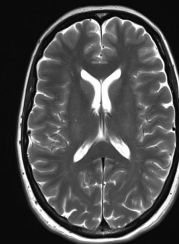
# Base de données dans MIA



- Fonctionnement en projets
  - Structure de fichiers d'un projet
  - Projet temporaire vide à l'ouverture du logiciel
- Surcharge des classes Database et DatabaseSession
  - Ajout des attributs nécessaires aux fields (visibilité, origine, unité, et valeur par défaut)



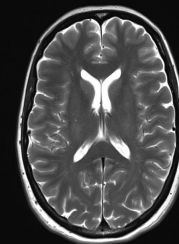
# Base de données dans MIA



## Nouveau projet

- 3 collections
  - initial : valeurs initiales
  - current : valeurs courantes
  - bricks : historique des briques
- 4 tags dans les collections initial et current
  - Checksum : Checksum du document
  - Type : Type de document (Scan, Matrix, ...)
  - Exp Type : Type de donnée
  - Bricks : Briques ayant généré le document
- 7 tags dans la collection bricks
  - Name : Nom de la brique
  - Inputs : Entrées de la brique
  - Outputs : Sorties de la brique
  - Init : Statut de l'initialisation
  - Init time : Date de l'initialisation
  - Exec : Statut de l'exécution
  - Exec time : Date de l'exécution

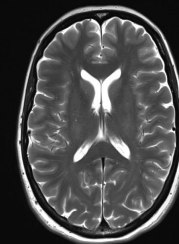
# Transactions



## Besoins dans MIA

- Ne pas enregistrer les modifications en cours tant que l'utilisateur n'enregistre pas le travail (Ctrl + S)
- Option `auto_save`, permettant l'enregistrement automatique des modifications
- Pour les pipelines, enregistrement automatique pour avoir des informations si l'exécution ne va pas jusqu'à la fin

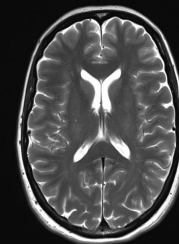
# Transactions



## Solutions

- Transaction unique à l'ouverture du projet
- Si auto\_save désactivé
  - commit quand l'utilisateur fait Ctrl + S
- Si auto\_save activé
  - commit après chaque modification dans le projet
- Pipelines
  - commit avant et après chaque brique traitée

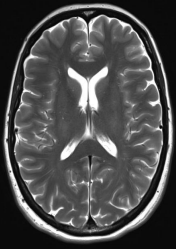
# Optimisation



Potentiellement un nombre important de données

- Optimisation du temps d'exécution
  - Séquence d'import du logiciel tiers beaucoup travaillée
    - Analyse détaillée grâce au profiling
    - Flush des objets le moins souvent possible
    - Pas de commit
    - Ajout des tags, puis des images avec toutes les valeurs lui étant associées
  - Utilisation de caches
    - Stockage en mémoire de tous les objets utilisés

# Options de la base de données



- **caches** : booléen
  - Si vrai, toutes les lignes de la base de données sont gardées en mémoire dans des dictionnaire, afin d'y accéder plus rapidement
  - Si faux, requête à chaque fois qu'on souhaite accéder à des lignes
- **list\_tables** : booléen
  - Si vrai, création de tables pour stocker les valeurs des fields de type liste, afin de permettre le requêtage SQL
  - Si faux, seule la représentation en chaîne de caractères des valeurs des fields de type liste est stockée, et requêtage en Python
- **query\_type** : [sql, python, mixed, guess]
  - Si sql, les filtres sont réalisées en requêtes SQL, erreur renvoyée si impossible
  - Si python, les filtres sont réalisés en Python, marche tout le temps
  - Si mixed, les filtres sont réalisés en SQL, avec éventuellement une partie en Python
  - Si guess, les filtres sont mixed, et si impossible, en Python