

RAPPORT INTERMÉDIAIRE

InfraDonn2
HEIG-VD - 2022 - M50-2

Groupe : Flop

Léa Talon, Jovan Stojanovic, Marion de Kerchove, Ricardo Delgado
Miranda, Loïc Fontaine & Nisa Azizi

Sommaire

Question 1	02
Question 2	04
Question 3	06
Question 4	09
Question 5	10
Question 6	11
Question 7	13
CouchDB	16
Sources	01

Question 1

Depuis l'existence des ordinateurs, quelle a été l'évolution des besoins au niveau de la gestion de l'information et de la gestion des données ?

La gestion de données ou data management

Numérisation, digitalisation et augmentation des données

Avec l'apparition des ordinateurs, un nouveau moyen de stocker de l'information voit le jour. Toutes les informations peuvent désormais être numérisées. La numérisation consiste à prendre un fichier "physique" et le transformer en une version numérique. Les fichiers prennent ainsi moins de place physique, sont plus accessibles et plus sécurisés. La numérisation permet aussi un partage plus rapide.

Une fois les fichiers numérisés, il est possible de les digitaliser, c'est-à-dire les rendre accessibles via des clouds par exemple ou d'utiliser ses informations pour améliorer les performances.

Comme il est désormais plus simple de stocker de l'information, on en a beaucoup plus et de différents types. Les données à stocker deviennent plus importantes et demandent une plus grande capacité de stockage qu'auparavant. Il faut donc distinguer ces données par type et les organiser selon leur utilité et leur importance. Cette forte augmentation de données demande d'être plus rigoureux et d'avoir une organisation particulière afin d'en tirer le maximum et de rester performant. Pour trier ces données et les optimiser il existe des outils tels que les SGBD.

Data Management

Toutes ces données stockées ne sont utiles que lorsqu'elles sont correctement triées et indexées sans cela elles deviennent inutiles car introuvables.

D'où vient le problème du contrôle de la qualité des données et de leur architecture. Cela rentre dans le processus du Data Management qui ingère, stocke, organise et maintient les données collectées par une entreprise.

Selon le consortium DAMA international: "le développement et l'exécution d'architectures, de règles, de pratiques et de procédures visant à gérer les besoins en cycle de vie de l'information d'une entreprise de façon efficace".

Lorsque l'on va chercher de l'information, il existe différents formats de compression et de manière de stocker. Une image peut exister sous un format jpeg, png, raw etc. Il est important de définir et imposer un standard sur les formats. Cela permet alors de comparer les différentes informations et d'éliminer les doubles. Une fois qu'un standard est décidé, cela rend possible l'analyse de données. Vous pouvez désormais classer vos données selon leurs attributs, comme par exemple leur date de création etc.

Réglementation de la gestion de données : Le RGPD

Avec la digitalisation des données, de nouvelles règles ont vu le jour. Le RGPD, soit le Règlement Général de la Protection des Données, permet de réguler la gestion de données afin que celle-ci soit plus éthique et sécurisée. Les obligations de ce règlement sont les suivantes :

- Aviser les clients de la collecte de leurs données et dans quelles conditions elles sont collectées
- Collecter et utiliser uniquement les informations indispensables aux relations commerciales
- Assurer la confidentialité des données
- Inscrire le fichier client dans le Registre des activités de traitement afin d'en garder une trace
- Limiter la conservation des données dans le temps

Les entreprises sont soumises à ces règles mais ont aussi d'autres obligations comme :

- Prendre en compte la protection de la vie privée des utilisateurs dès la conception d'un produit data management (Privacy by design)
- Assurer la transparence et la traçabilité des données
- Garantir la sécurité du traitement des données : mesures contre les risques de piratage, discrétion des salariés impliqués dans le traitement des données, notification des clients si violation de la vie privée
- Alerter, assister et conseiller l'entreprise en cas de problème
- Détruire les données clients et leur copie à la fin du contrat avec l'entreprise

Grâce à ces règles, il est possible de rendre le data management plus sécurisé.

Question 2

Quels ont été les facteurs conduisant à l'émergence des big data ? Quelles en ont été les conséquences ?

Qu'est ce que les big data ?

"Big data" est une expression qui serait apparue en 1997 pour définir de grands ensembles de données. Sa naissance est liée à un "big bang des données" dû à la croissance des capacités de stockage et d'analyse des informations. Aujourd'hui, on se base sur les trois V pour définir ce qui entre dans cette catégorie, ou pas : la variété, la vitesse et le volume. Ces dernières nécessitent l'utilisation de technologies spécifiques, souvent sur plusieurs machines en parallèle. Il est difficile de prédire ce que deviendront les big data à l'avenir mais il est probable qu'elles soient à l'origine d'énormes bouleversements dans des secteurs comme l'économie, la sociologie ou encore dans le domaine militaire.

Les facteurs conduisant à l'émergence des big data

Le premier facteur à avoir conduit à l'émergence des big data est la démocratisation du web. Cet événement a conduit à une croissance exponentielle des données et par association, de leur stockage. Les méthodes et supports de stockage ont évolué également afin de suivre la cadence de la croissance des données et permettre d'en stocker toujours plus et d'y accéder toujours plus rapidement. Généralement, c'est la quantité de données qui augmente le plus rapidement et le hardware qui ne suit pas toujours.

L'évolution du marketing vers des campagnes toujours plus personnalisées a également eu un poids non négligeable dans le nombre de données stockées et échangées. Effectivement, le seul moyen de proposer une solution la plus personnalisée possible est de récolter une énorme quantité de données sur les différentes pratiques, comportements ou attentes des utilisateurs ou de les acheter. Aujourd'hui cette pratique représente un marché très fructueux.

Les conséquences de l'émergence des big data

Le big data n'a pas seulement impliqué une énorme augmentation du volume des données mais également de la circulation de ces dernières. Il existe aujourd'hui un marché très fructueux de données consommateurs qui sont transmises entre les différentes entreprises. Cette quantité d'informations transmises entraîne des risques de fuites plus importants.

Plusieurs fois déjà les données personnelles sensibles de milliers de personnes se sont retrouvées entre de mauvaises mains. Ces échanges de données permettent également aux entreprises de cibler de mieux en mieux leurs campagnes publicitaires, allant parfois jusqu'à des campagnes individuelles basées sur les données récoltées d'un utilisateur. Ce ciblage pose problème à certaines personnes qui y voient une pratique peu éthique. Aujourd'hui, le phénomène est si répandu qu'on parle de "mort" de l'anonymat et de la confidentialité sur internet.

L'explosion de la quantité des données à stocker induit évidemment des serveurs toujours plus performants, rapides et gros. L'énergie nécessaire pour alimenter tout ce réseau augmente donc de la même manière et dans une société où le réchauffement climatique est au centre des inquiétudes, cette quantité d'énergie utilisée pose problème.

D'un point de vue plus positif, l'avènement des big data créent de nouveaux emplois de data scientist, data analyst, etc.. La quantité et qualité des données disponibles permet également d'améliorer la précision des analyses et des prises de décisions ainsi que le développement de projets de machine learning.

Question 3

Quelles solutions ont été trouvées pour répondre à la problématique du big data ?

L'émergence des solutions de la gestion du big data est venue des acteurs qui étaient concrètement confrontés à la problématique de la quantité de données à stocker et de la possibilité d'y accéder rapidement. La taille des données ne permettait plus de tout stocker sur un seul système physique. De plus ce type de systèmes de bases de données n'était plus assez performant. Il a donc été nécessaire d'innover.

Pour régler les problèmes du Big Data, des solutions technologiques ont été trouvées. Ces solutions visent à répondre aux limitations des architectures classiques suivant trois axes, le volume, la vitesse, la variété. Ces trois axes aussi appelés les 3V.

Le premier est le volume, c'est-à-dire que la gestion d'un énorme volume de données nécessite d'imaginer des nouvelles architectures pour obtenir des bonnes performances et compenser les limites physiques des débits des disques. En deuxième, il y a la vitesse. En effet, l'arrivée des objets connectés et des capteurs intelligents qui sont capables de délivrer de la donnée en continu impose de trouver des solutions permettant de recevoir cette donnée en temps réel. Finalement le dernier V est la variété, puisqu'il existe plusieurs types d'information, notamment l'information non-structurée que beaucoup d'entreprises envisagent d'utiliser grâce aux réseaux sociaux.

Pour remédier à ces trois facteurs problématiques, plusieurs solutions technologiques ont fait surface.

Google, à cause de son grand volume de données lié à ses multiples solutions ainsi que son besoin de gérer et parcourir ces données, est une des premières entreprises à développer sa solution.

GoogleFS (GFS) est un “système de fichiers distribué”, ce système est un “environnement de stockage redondant et résilient fonctionnant sur un cluster constitué d'un grand nombre de machines de moyenne puissance, «jetables».” Ce système permet non seulement de stocker des données mais aussi de les manipuler, c'est-à-dire de les rechercher et les restituer. Pour ce faire, GFS utilise deux fonctions: Map et Reduce.

Map “[effectue] un traitement sur une liste de valeurs” et Reduce regroupe les résultats de la fonction Map.

Plus tard, Doug Cutting, implémentera ces fonctions en Java sous le nom Hadoop. Hadoop sera ensuite utilisé par de grande société telles que Microsoft, Windows Azure et Windows Server. En 2012, même Facebook annonce avoir utilisé Hadoop.

A partir de 2006, de nombreuse invention découle de l'apparition de la BigTab de Google qui “ressemble à une gigantesque table de hachage distribuée qui incorpore des mécanismes permettant de gérer la cohérence et la distribution des données sur GFS.” L'implémentation la plus notable est Hbase, développée par Michael Stack et Jim Kellerman, d'abord pour contribuer à Hadoop mais qui plus tard deviendra un projet à part entière d'Apache, rachetée ensuite par Microsoft.

Hbase est un “système de gestion des données orienté colonnes [...] destiné à manipuler de très larges volumes de données sur une architecture distribuée.” Aujourd'hui encore Hbase reste un des projets les plus important du monde NoSQL qui est/a été utilisé partout sur le Web.

Dans cette période de grandes découvertes, Amazon n'est pas en reste. En 2007, leur Chief Technical Officer présente un papier sur Dynamo. Dynamo “consiste en un entrepôt de paires clé-valeur destiné à être lui aussi totalement distribué, dans une architecture sans maître.”

Ces deux dernières technologies ont en quelque sorte bâtit le monde du NoSQL et ont permis à d'autres technologies de se développer, malgré le fait que leur code ne soit pas en openSource et appartienne encore aux deux géants des GAFA.

Quant au troisième membre des GAFA, Facebook a engagé un des développeurs de Dynamo “pour créer un entrepôt de données destiné à mettre en place la fonctionnalité de recherche dans les boîtes aux lettres des comptes Facebook, qui permet aux utilisateurs d'effectuer des recherches parmi tous les messages reçus.”

Apache Cassandra fut donc développé par deux ingénieurs qui est aujourd’hui utilisé par Twitter, Reddit et Cisco.

En résumé et de manière plus théorique les technologies suivantes furent développées pour répondre à cette problématique:

1) *Les bases de données analytiques*

Également appelées bases de colonnes, les bases de données analytiques ont permis d’innover avec un mode de stockage qui permet d’être très efficace pour le décisionnel il introduit un mode de suppression intéressant.

2) *In Memory Analytics*

Elles visent à monter toutes les données en mémoire vive au détriment du disque dont le rôle sera cantonné à assurer la persistance des données. Cette approche radicale est prometteuse et garantie des temps de réponse à la hauteur des attentes.

3) *Les appliances*

Ce sont des solutions tout en intégrant le matériel, le stockage et le logiciel pour faciliter l’administration et l’exploitation de la plateforme.

4) *Le développement du NoSQL*

Le NoSQL regroupe aujourd’hui une multitude d’initiatives et de projets, en évolution rapide, et qui s’appuient principalement sur le framework Hadoop d’Apache. Hadoop propose par son design une très forte tolérance aux pannes et une excellente scalabilité dans un contexte de traitements batchs. Mais il ne garantit pas pour l’instant de bons temps de réponse et semble donc peu adapté aux traitements interactifs.

Question 4

Quelles sont les différences et les similitudes entre les bases de données relationnelles et les bases de données NoSQL ?

Base de données relationnelle, SQL

Les bases de données relationnelles utilisent le langage SQL. Les données sont organisées par tableaux aussi appelés table. Ces tables contiennent des lignes. Le but d'une base de données relationnelle est de stocker des données et de permettre de les manipuler. Sur ces données, il est possible de faire des requêtes et de créer des vues ou schéma pour organiser les informations

Base de données non-relationnelle, NoSQL

Dans ce langage, il n'y a pas de tableau ou de table. Ces bases de données permettent de grouper des données n'ayant pas la même structure et elles peuvent être stockées sur différentes machines et serveurs. Elle est également mieux adaptée à l'analyse.

La principale similitudes entre ces deux types de base est le fait qu'elles sont utilisées pour stocker des données. Le langage SQL se retrouve aussi dans les deux mêmes s'il est parfois différent et contient ses particularités. En dehors de cela, leur fonctionnement et leur utilisation sont différents.

Question 5

Qu'est-ce que le théorème CAP ? Quels sont ses enjeux et ses implications dans la gestion de l'information et des données ?

Le théorème CAP (*Consistency, Availability & Partition Tolerance*), aussi appelé théorème de Brewer dit qu'il est impossible pour un système informatique de calcul distribué d'assurer le fonctionnement synchrone des trois contraintes suivantes :

- **Cohérence**

On voit les mêmes données en même temps peu importe le nœud auquel on se connecte. Si les données doivent être répliquées ou mises à jour fréquemment, toutes les requêtes doivent obtenir le même résultat.

- **Disponibilité**

Les données sont accessibles en tout temps et depuis n'importe quel nœud, et ce même si certains d'entre eux sont en panne.

- **Tolérance au partitionnement**

A moins d'une coupure totale du réseau, aucune panne ne doit empêcher le système d'être fonctionnel.

Ce théorème implique qu'aucun système distribué ne peut couvrir ces trois contraintes, ce qui soulève les catégories suivantes:

- *Cohérence et disponibilité* : Oracle, MySQL, SQLServer, PostgreSQL
- *Disponibilité et tolérance au partitionnement* : Spark, ElasticSearch, Neo4j
- *Cohérence et tolérance au partitionnement* : MongoDB, BigTable, CosmosDB

Les databases cohérentes doivent être utilisés dans les milieux où la valeur de l'information retournée doit être exacte. Le domaine des finances en est un bon exemple. Quant à la disponibilité, elle est essentielle dans les cas où le service est plus important que l'information communiquée. Les e-commerce sont concernés.

Question 6

Déterminer les principales familles et architectures de bases de données NoSQL

Il existe 4 principaux types de bases de données NoSQL composée chacune d'un attribut unique et de limites spécifiques. Les 4 principaux types sont: clé/valeur, orientées colonnes, orientées graph et orientées documents.

Clé/valeur

C'est le modèle le plus simple et basique des quatres disponibles. Comme son nom l'indique, les données sont stockées sous la forme de clé-valeur. La clé identifie la donnée de manière unique et permet de la gérer. La valeur contient n'importe quel type de données. La clé est généralement une séquence de chaînes de caractères, d'entiers ou de caractères, mais peut également être un type de données plus avancé. La valeur est généralement liée ou co-liée à la clé. Les bases de données de stockage de paires clé-valeur stockent généralement les données sous forme de table de hachage où chaque clé est unique. La valeur peut être de n'importe quel type (JSON, BLOB(Binary Large Object), chaînes de caractères, etc). Ce type de modèle est généralement utilisé dans les sites Web d'achat ou les applications de commerce électronique.

Orientées colonnes

Dans ce modèle, on y retrouve le principe de "table" avec des lignes et des colonnes. Cependant, ce modèle possède quelques particularités. Les colonnes sont dynamiques, c'est à dire qu'au sein d'une même table deux individus peuvent ne pas avoir le même nombre de colonnes car les valeurs nulles ne sont pas stockées. Cette propriété permet de libérer de la place de stockage et d'améliorer les performances de traitement car la volumétrie de données à traiter est plus faible. Cette architecture se rapproche le plus des bases de données relationnelles car elles ont le même principe de tables. Une table est un ensemble de clés triées qui regroupent des familles, qui elles-mêmes rassemblent des colonnes. Finalement, l'indexation des données se fait à la valeur et non à la ligne. Cela empêche de faire des doublons de stockage d'information et c'est pour cela que cette famille est surtout utilisée pour traiter des très grandes quantités de données sans problème de disponibilité.

Orientées documents

La base de données de documents récupère et accumule les données sous la forme de paires clé-valeur, mais ici, les valeurs sont appelées documents. Le document peut être défini comme une structure de données complexe. Il peut s'agir de textes, de tableaux, de chaînes de caractères, de JSON, de XML ou de tout autre format. L'utilisation de documents imbriqués est également très courante (organisation sous forme hiérarchique). C'est très efficace car la plupart des données créées sont généralement sous forme de JSON et ne sont pas structurées.

Orientées graph

Le graph va quant à lui se concentrer sur les relations entre les différentes données. Chaque élément est stocké comme un nœud. Leur relation est exprimée par des liens entre eux. Cette architecture est optimisée pour capturer et rechercher les connexions entre les données par les JOIN de plusieurs tables de données. Ce type est idéal pour des objets complexes fortement connectés, permettant rapidité dans la recherche.

Question 7

Pour chaque famille, donner les points forts et les points faibles, donner des exemples de base de données. Pour quels types d'utilisation (donner des exemples d'applications pour lesquelles elles sont utilisées) ?

Clé/valeur

Forces :

- Facilement sécable
- Temps d'écriture/lecture très bas
- Disponibilité

Faiblesses :

- Mise à jour compliquée
- Requêtes rudimentaires : interrogation sur une clé

Types d'utilisation :

- Amazon : simpleDB
- DynamoDB
- Redis (VMware)
- Azure CosmosDB (Microsoft)
- Memcached (Danga)

Orientées colonnes

Forces :

- Le système de colonne accélère les processus de lecture des données et augmente les performances
- Grosse compression

Faiblesses :

- Requêtes limitées
- Limitation du nombre de types d'objets

Types d'utilisation :

- BigTable (Google)
- HBase (Apache, Hadoop)
- Spark SQL (Apache)
- Elasticsearch (elastic)

Orientées documents

Forces :

- Aucune restriction dans la structure des données
- Les données sont faciles à modifier
- Permettent de récupérer un ensemble important de valeurs avec une seule requête
- La souplesse permet de rapidement faire progresser un prototype, sans devoir mettre à jour un modèle relationnel, le stockage de données

Faiblesses :

- Inadaptées pour les données non structurées
- Risque de duplication de l'information
- Risque de recherche d'information inexistante, potentiellement très dégradante sur les performances

Types d'utilisation :

- CouchDB
- MongoDB
- Riak
- Redis
- Cloudant
- Amazon Document DB
- Elasticsearch
- Simple DB

Orientées graph

Forces :

- Adaptées à la gestion des données relationnelles
- Architecture modulable

Faiblesses :

- Limitées à certains cas

Types d'utilisation :

- TITAN
- Neo4j
- OrientDB (Apache)

La base de données orientées graphe sont utilisées pour la modélisation des réseaux sociaux.

LinkedIn, par exemple, utilise des bases de données orientées graphe ce qui permet d'afficher facilement le degré de séparation entre chaque contact, puisqu'il s'agit simplement de la distance entre les nœuds.

CouchDB

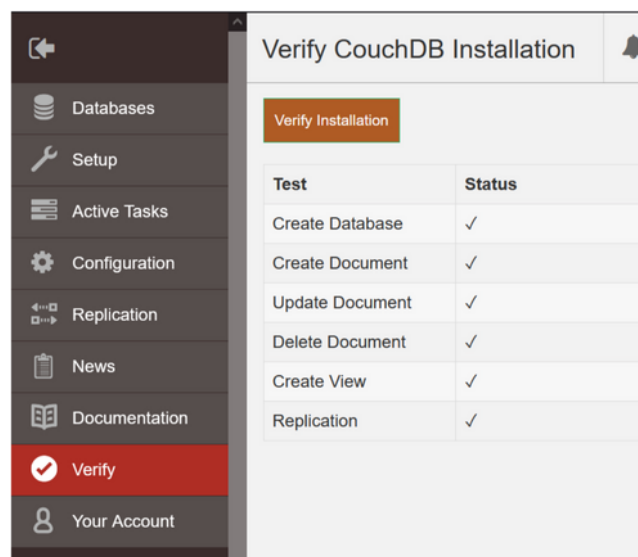
Tutoriel 1 - Lien

Partie 1 - *Créer des database*

1. Revenu dans le C: avec la commande `cd ..`
2. `curl http://127.0.0.1:5984/`
 - a. Permet de vérifier que couchDB marche
3. `curl -X GET http://admin:password@127.0.0.1:5984/_all_dbs`
 - a. Permet de voir les db qu'on a (en changeant le admin par le user choisi et pwd pareil)
4. `curl -X PUT http://admin:password@127.0.0.1:5984/baseball`
5. Permet de créer une database
6. On remet la commande 3)
 - a. On voit la db créée en 4)
7. On remet la commande qui permet de créer des database en réutilisant le même nom
 - a. On ne peut pas créer deux database avec le même nom
8. `curl -X DELETE http://admin:1234@127.0.0.1:5984/plankton`
 - a. Permet de supprimer une database

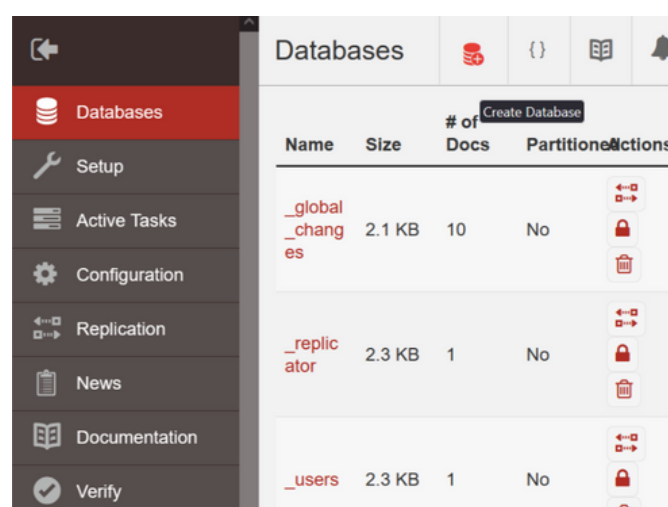
Partie 2 – Dans Fauxton

1. Log dans Fauxton avec l'admin et le password et l'on voit apparaître dans la database baseball créé précédemment avec les lignes de commande.
2. Il faut ensuite, dans fauxton, vérifier l'installation de couch db à l'aide du bouton verify à gauche.

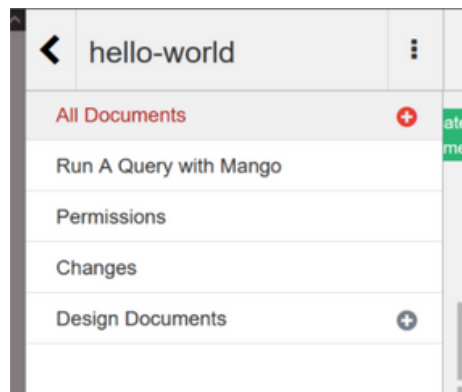


Partie 3 – Créer une database et des documents

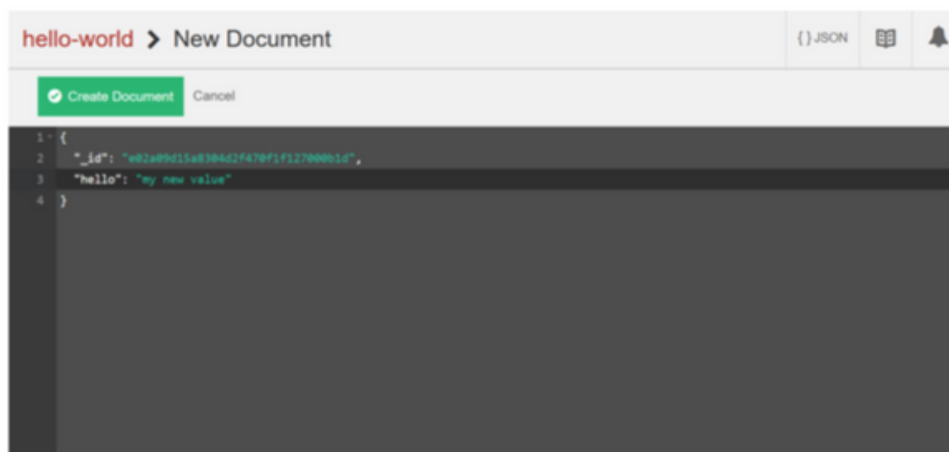
1. Dans Fauxton il est aussi possible de créer simplement une base de données à l'aide du bouton prévu à cet effet.
 - a. On crée une database appelée hello-world.



2. On crée ensuite un document dans cette database.



3. Une fois le nouveau document créé, l'éditeur JSON se crée tout seul et on y a rajouté la ligne ci-dessous. Les virgules permettent d'ajouter des lignes.

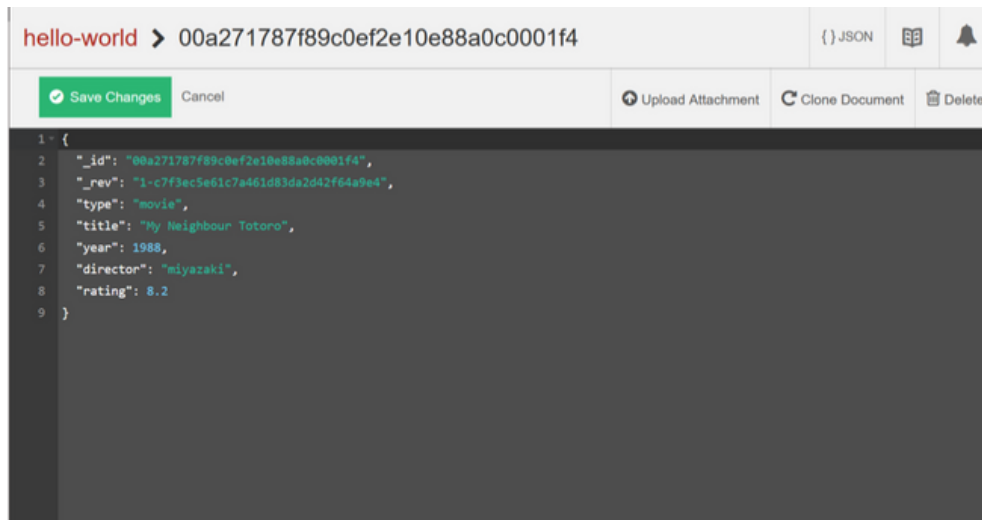


4. On essaie de rajouter des nouvelles valeurs pour voir si cela fonctionne bien.

5. Pour permettre une meilleure vision, on peut ajouter l'option d'inclure les documents options. Cela permet d'afficher le document avec les valeurs de l'id et de la révision.

Partie 4 – Mango Query

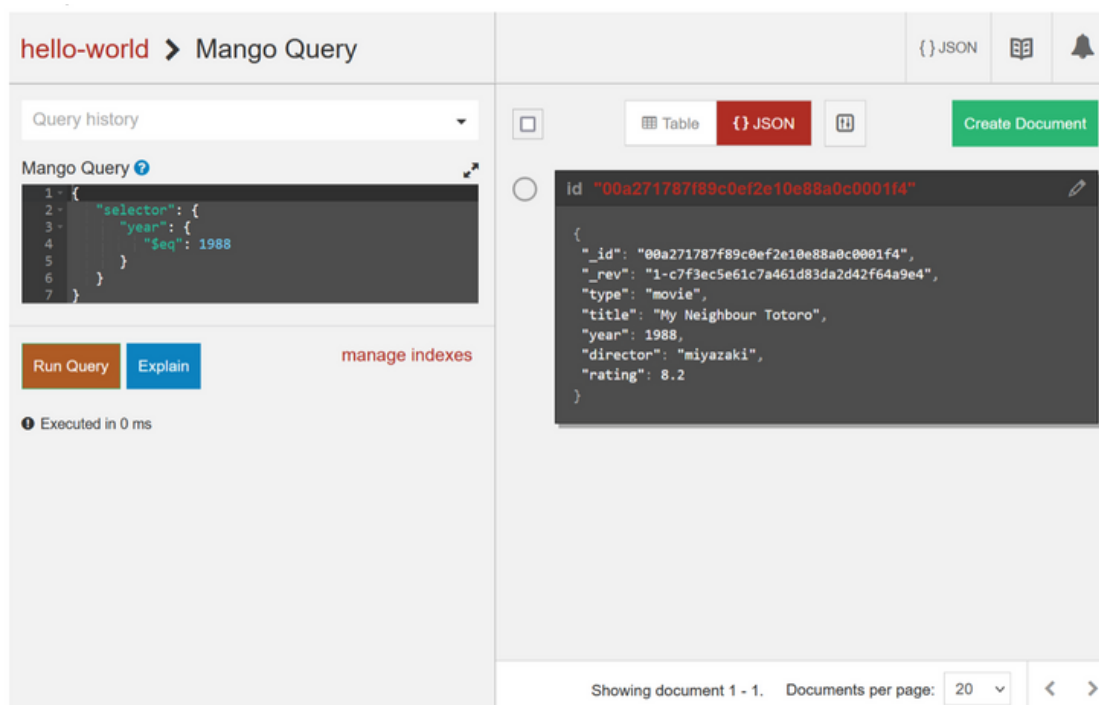
1. On crée 3 nouveaux documents dans une database. À ces éléments, on ajoute les mêmes colonnes.



The screenshot shows a document editor for a database named 'hello-world'. The document ID is '00a271787f89c0ef2e10e88a0c0001f4'. The document is in JSON format and contains the following fields:

```
1 {  
2   "_id": "00a271787f89c0ef2e10e88a0c0001f4",  
3   "_rev": "1-c7f3ec5e61c7a461d83da2d42f64a9e4",  
4   "type": "movie",  
5   "title": "My Neighbour Totoro",  
6   "year": 1988,  
7   "director": "miyazaki",  
8   "rating": 8.2  
9 }
```

2. On a créé un index mango qui permet de trouver un film par son année de sortie. Cela nous permet d'obtenir le film à l'année de l'index.



The screenshot shows the Mango Query interface. On the left, the 'Mango Query' section contains a query:

```
1 {  
2   "selector": {  
3     "year": {  
4       "$eq": 1988  
5     }  
6   }  
7 }
```

Below the query, there are buttons for 'Run Query' and 'Explain', and a status message 'Executed in 0 ms'. On the right, the 'JSON' tab is selected, showing the document with ID '00a271787f89c0ef2e10e88a0c0001f4' and its fields.

Showing document 1 - 1. Documents per page: 20

Partie 5 – Réplication d'une database

1. Le but ici est de répliquer une database. Il est possible de créer une nouvelle base de données et la désigner comme replica d'une autre.

Job Configuration

Source

Type:Local database

Name:hello-world

Authentication:None

Target

Type:Existing local database

Name:hello-duplication

Authentication:None

Options

Tutoriel 2 - Lien

1. Créer un fichier .json

```
loicf@LoicFontaine MINGW64 /c/projects/couchemployees
$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (couchemployees) app.js
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to c:\projects\couchemployees\package.json:
```

2. Créer un fichier app.js

```
Get Started index.php JS app.js X
C: > projects > couchemployees > JS app.js > app.get("/") callback
1  const express = require("express");
2  const bodyParser = require("body-parser");
3  const path = require("path");
4  const NodeCouchdb = require("node-couchdb");
5  const app = express();
6  app.set("view engine", "ejs");
7  app.set("views", path.join(__dirname, "views"));
8  app.use(bodyParser.json());
9  app.use(bodyParser.urlencoded({ extended: false }));
10 app.get("/", function (req, res) {
11   res.send("Working.....");
12 });
13 app.listen(3000, function () {
14   console.log("Server is started on Port 3000");
15 });
16
```

3. Ensuite taper: `npm install express body-parser ejs node-couchdb --save`

```
$ npm install express body-parser ejs node-couchdb --save
added 80 packages, and audited 81 packages in 6s

9 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

5. Lancer le serveur avec `node app`

```
loicf@LoicFontaine MINGW64 /c/projects/couchemployees
$ node app
Server is started on Port 3000
```

6. Le serveur est maintenant accessible via `http://localhost:3000/`

Sources

- https://fr.blog.businessdecision.com/solutions-technologiques/?doing_wp_cron=1664227771.4189929962158203125000
- <https://www.weka.ch/themes/gestion-et-management/organisation-et-gestion/gestion-dentreprise/article/management-a-lere-numerique-plus-rapide-plus-loin-et-plus-fort/>
- <https://www.lemagit.fr/definition/Quest-ce-que-la-gestion-des-donnees-Pourquoi-est-elle-fondamentale>
- <https://www.lebigdata.fr/data-management>
- <https://www.ontrack.com/fr-fr/blog/histoire-gestion-donnees>
- https://fr.wikipedia.org/wiki/Gestion_des_donn%C3%A9es
- <https://www.igualit.com/article/evolution-de-la-gestion-de-linformation/>
- <https://www.cairn.info/revue-les-cahiers-du-numerique-2018-1-page-9.htm>
- <https://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462433-choisissez-votre-famille-nosql>
- <https://www.asprom.com/dossier/NoSQL.pdf>
- <https://www.mongodb.com/scale/types-of-nosql-databases>
- https://fr.blog.businessdecision.com/solutions-technologiques/?doing_wp_cron=1664227771.4189929962158203125000