Bases de Données Avancées

TD/TP : NoSQL

– USTHB Master 01 IL–
M. AZZOUZ
Dernière mis à jour :Avril 2020

Modèle relationnel

- Le modèle relationnel, apparu dans les années 1980, est régi par des règles précises, énoncées par Codd (IBM) en 1985 :
  - séparation logique et physique,
  - langage déclaratif,
  - structuration forte des données,
  - représentation sous forme de tables,
  - contraintes définies au niveau du SGBD,
  - cohérence transactionnelle forte, etc.
- Les bases de données relationnelles font la preuve de leur efficacité depuis longtemps !

Modèle relationnel

- Ces bases relationnelles sont bien adaptées au stockage et à la manipulation d'informations bien structurées, décomposables en unités simples et représentables sous forme de tables.

- Cependant … depuis quelques années, ce modèle est remis en cause … pour des raisons que nous allons (tenter) d'expliquer à partir de deux constats.

Modèle relationnel- Constats

❑ **Constat n°1 :** le modèle relationnel présente certaines limites:

1. On ne peut pas imbriquer les informations.

► On multiplie le nombre de tables pour représenter conceptuellement le même objet.

2. La structure du schéma est très rigide.

► une base a un nombre fixé de tables, une table a un nombre fixe d'attributs, …

► Le modèle doit être défini tôt dans le processus de développement, il est souvent difficile et couteux de le faire évoluer.

Modèle relationnel- Constats

- **Constat n°2** : croissance exponentielle des données générées par le web.
- Ces masses de données apportent des opportunités d'analyses plus larges et plus fines ainsi que de nouveaux usages de l'information.
- Les données sont devenues le principal actif de certaines entreprises !
- Le phénomène « Big data » !
- Les systèmes de gestion des BD relationnelles n'ont pas été conçus pour gérer de tels volumes de données.

Modèle relationnel- Constats

- Les premiers acteurs à avoir été confrontés à ces deux constats et avoir buté sur les limites des modèles relationnels furent les fournisseurs de services en ligne les plus populaires, tels que Yahoo, Google, ou plus récemment les acteurs du web social comme Facebook, Twitter ou LinkedIn.
- Ces grands acteurs du web ont alors conçu de nouveaux systèmes leur permettant de s'affranchir de ces limites.
- C'est la naissance du phénomène NoSQL !
- NoSQL : No SQL ?
- Faut il jeter le modèle relationnel ?

Modèle relationnel- Constats

- **Réponse au constat n°1** : les modèles non-structurés ou semi-structurés, plus adaptés à de nombreux cas de collecte et de manipulation des données…

- Les bases NoSQL manipulent les données à l'aide de modèles non ou semi-structurées: par exemple XML, mais ça n'est pas le seul format de modèles semi-structurés qui existe…

Modèle relationnel- Constats

- **Réponse au constat n°2 :** gérer de grands volumes de données nécessite de mettre en place des systèmes distribués ….
- Les capacités de stockage sont démultipliées par la possibilité de répartir les données sur plusieurs nœuds.
- Les systèmes relationnels ne sont pas adaptés aux environnements distribués.
- Les solutions NoSQL implémentent naturellement des mécanismes qui permettent une distribution des données.
- **Les bases NoSQL ne remplacent pas les BD relationnelles mais en sont une alternative, un complément apportant des solutions plus intéressantes dans certains contextes.**

NoSQL

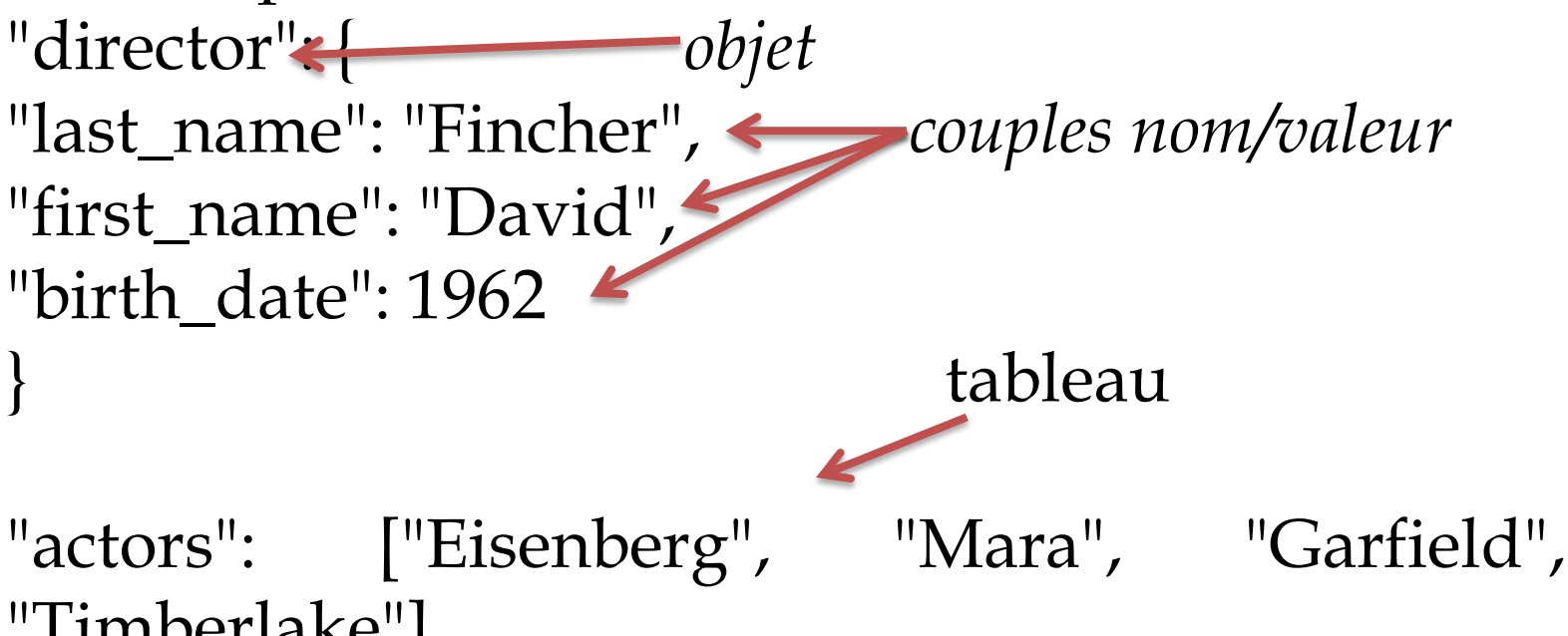
- Une proposition de définition de NoSQL: « Tout système de gestion de données sacrifiant des fonctionnalités du relationnel pour faciliter la scalabilité par distribution »[Philippe Declercq]
- Les bases NoSQL adoptent une représentation de données non relationnelle.
- Les bases NoSQL apportent une plus grande performance dans le contexte des applications Web avec des volumétries de données très importantes.
- Les bases NoSQL utilisent une très forte distribution de ces données et des traitements associés sur de nombreux serveurs.

JSON

- JSON : JavaScript Object Notation.
- Format de données textuelles dérivé de la notation des objets du langage JavaScript. Permet de représenter l'information structurée comme le permet XML.
- Dérivé de la représentation littérale d'un objet en Javascript.
- JSON est plus simple que XML, et il est très facile à associer à un langage de programmation.
- JSON est massivement utilisé dans les applications web (AJAX), les web-services (REST), et … les bases de données NoSQL.

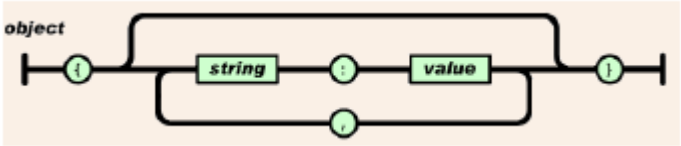
JSON

- JSON se base sur deux structures :
  - Une collection de couples nom/valeur, appelée **objet**
  - Une liste de valeurs ordonnées, appelée **tableau.**
- Exemples :

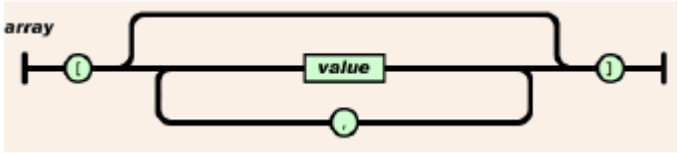


JSON

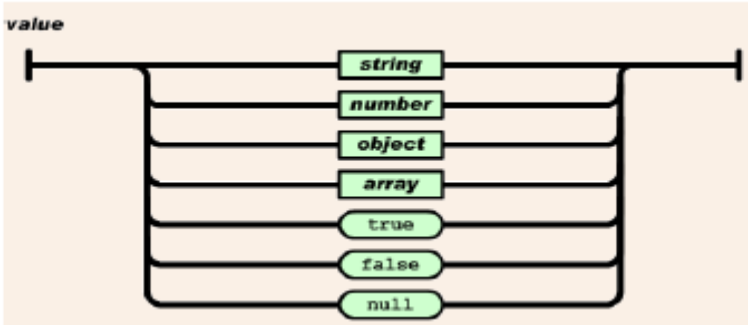
- L'objet est un ensemble de couples nom/valeur non ordonné :



- Un tableau est une collection de valeurs ordonnées :



- Une valeur peut être une chaîne de caractères, un nombre, true ou false ou null, un objet, un tableau :

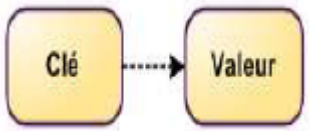


Solutions NoSQL

- Les solutions NoSQL sont généralement classées en 4 catégories selon le schéma ou la structure de données qu'ils manipulent :
  1. Bases clé/valeur,
  2. Bases orientées colonnes,
  3. Bases orientées documents,
  4. Bases orientées graphes.

Modèle orienté clé valeur

- **Les bases « clé/valeur » :**
  - ► Il s'agit de la catégorie de base de données la plus basique. Dans ce modèle chaque objet est identifié par une clé unique qui constitue la seule manière de le requêter.
  - ► La structure de l'objet est libre et le plus souvent laissé au choix du développeur de l'application (XML, JSON, ...), la base ne gérant généralement que des chaînes d'octets.
  - ► Le système de stockage ne connait pas la structure de l'information qu'il manipule.



- Exemples d'implémentation : **Voldemort, Redis, Riak**

Modèle orienté clé valeur

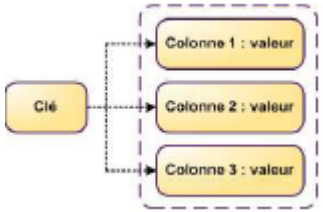
- **Intérêts :**
  - Des performances exceptionnellement élevées en lecture et en écriture.
  - Scalabilité horizontale considérable.
  - Pas de maintenance en cas d'évolution de nouveaux types d'informations.
- **Limites :**
  - Peu de possibilités de requêtage : uniquement sur les clés.
  - Le système n'ayant pas d'indice sur la structure de l'information qu'il stocke, tous les traitements (extraction du contenu, application de filtres…) doivent être effectués par le client.

Modèle orienté clé valeur

- **Usages :**
  - ► dépôt de données avec besoins de requêtage très simples,
  - ► système de stockage de cache ou de sessions distribuées,
  - ► profils, préférences d'utilisateurs,
  - ► données de panier d'achat,
  - ► données de capteur,
  - ► logs de données.
- **Exemples** :
  - ► Sur un site de réseau social, à partir d'un utilisateur (la clé), je veux obtenir une liste de ses amis (la valeur).
  - ► Dans un journal d'activités, la date d'un événement (la clé) indexe les détails de ce qui s'est passé à ce moment (la valeur).

Modèle orienté colonne

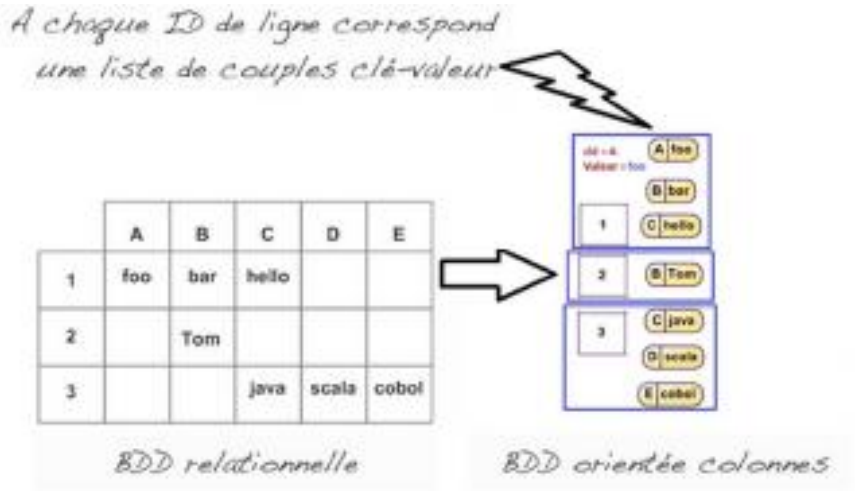
- La représentation orientée colonnes s'oppose à la représentation des tables dans les bases de données relationnelles. En effet, les SGBDR manipulent les colonnes d'une ligne d'une manière statique. Les bases de données orientées colonnes ont une vision plus flexible permettant d'avoir des colonnes différentes pour chaque ligne, de multiplier de manière conséquente le nombre de colonnes par ligne et d'optimiser le stockage des données associées.



- Exemples d'implémentation: **Hbase, Cassandra, SimpleDB.**

Modèle orienté colonne

- Optimisation du stockage par rapport à une solution relationnelle :



- Pas de place perdue lorsque des champs n'existent pas dans certaines lignes.
- L'accès aux données à partir de la clé sera beaucoup plus rapide, les données liées à une clé étant regroupées en terme de stockage.

Modèle orienté colonne

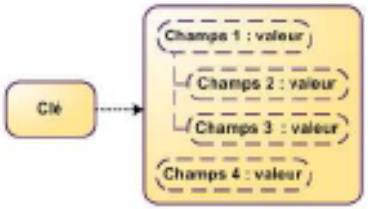
- **Intérêts :**
  - Conçues pour accueillir un grand nombre de colonnes (jusqu'à plusieurs millions) pour chaque ligne. Stockage optimisé de relations « one to many ».
  - Performances et scalabilité
- **Limites :**
  - Système de requêtes minimaliste
- **Usages** :
  - Traitements d'analyse de données et traitements massifs
  - Listes d'articles pour chaque utilisateur
  - Liste des actions effectuées par un utilisateur
  - Chronologie d'évènement maintenue et accédée en temps réel

Modèle orienté document

► Constituées de collections de documents. Un document est composé de champs et des valeurs associées, ces dernières pouvant être requêtées.

► Basées sur le modèle « clé-valeur » mais la valeur est un document en format semi-structuré hiérarchique de type JSON ou XML

► Les documents n'ont pas de schéma, mais une structure arborescente: ils contiennent une liste de champs, un champ a une valeur qui peut être une liste de champs, ...



► Exemples d'implémentation : CouchDB, RavenDB, **MongoDB.**

Modèle orienté document

❑ **Intérêts :**

► Capacité à effectuer des requêtes sur le contenu des objets

► Modèle de données simple mais puissant

► Scalabilité

►Pas de maintenance de la BD requise pour ajouter/supprimer des «colonnes»

► Permet de représenter les relations one-to-one et one-to-many. Ainsi un document pourra être sauvegardé et chargé sans aucun traitement de jointure.

❑ **Limites :**

► Inadaptée pour les données interconnectées

► Lent pour les grandes requêtes

Modèle orienté document

- **Usages:**
  - Enregistrement d'événements
  - Systèmes de gestion de contenu
  - Web analytique ou analytique temps-réel
  - Catalogue de produits
  - Systèmes d'exploitation
  - Stockage de volumes très importants de données pour lesquelles la modélisation relationnelle aurait entraînée une limitation des possibilités de partitionnement et de réplication.

Exercice 01 «Du relationnel aux modèles NoSQL »

On considère une base de données relationnelle de vente de livres comportant un ensemble de tables dont un extrait est décrit ci-dessous avec 3 tables (Vente, Livre et Auteur). On considère dans un 1er temps qu'un livre a un auteur unique et qu'une vente correspond à la vente d'un seul livre. Les valeurs des attributs id_vente, id_auteur et isbn sont uniques, et sont clés primaires des tables.

Table Auteur

id_auteur	nom	prénom
154	Gosciny	René
987	Bruchez	Rudi

Table Livre

isbn	titre	id_auteur
2154889522	Asterix et Cléopatre	154
2154889589	NoSQL	987

Table Vente

id_vente	date	isbn
10	02/06/2017	2154889522
12	02/06/2017	2154889589
20	12/09/2017	2154889589

Il est envisagé la migration de la base dans une base de données NoSQL.

1. Décrire ces informations sous forme d'un diagramme UML.
2. On a besoin de savoir pour chaque livre donné tous ses détails. Proposer une modélisation orientée clé-valeur pour ce cas d'usage.
3. On considère le cas d'usage suivant: Accéder aux livres d'un auteur donné (recherche à partir d'un nom donné). Proposer une modélisation orientée colonne pour ce cas d'usage.
4. Proposer une modélisation orientée document de ces informations centrées sur les ventes, une modélisation centrée sur les livres et une sur les auteurs.
5. Indiquer l'impact que peut avoir la modélisation retenue sur le requêtage de la base.
6. Expliquer les modifications nécessaires pour gérer plusieurs auteurs par livre dans la base de données relationnelle et dans les différents documents structurés.

Exercice 01 «Du relationnel aux modèles NoSQL »

1. Décrire ces informations sous forme d'un diagramme UML:

Exercice 01 «Du relationnel aux modèles NoSQL »

2. On a besoin de savoir pour chaque livre donné tous ses détails. Proposer une modélisation orientée clé-valeur pour ce cas d'usage:

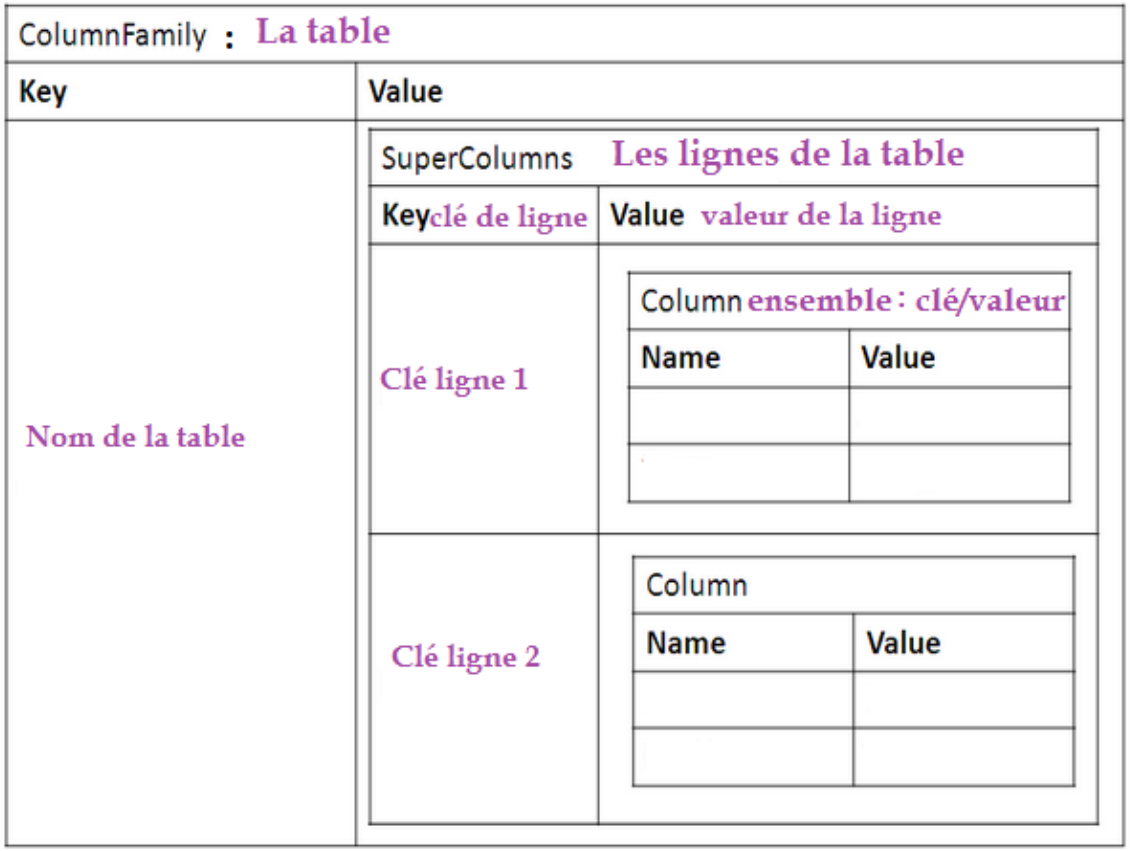
Clé(isbn)	Valeur(document JSON sauvegardé autant que BLOB)

2154889522	"{
	"**titre**":"Asterix et Cléopatre",
	"**auteur**": {
	"**id_auteur**":154,
	"**nom**":"Gosciny« ,
	"**prenom**":"René"
	},
	"**vente**": [
	{
	"**id_vente**":10,
	"**date**":02/06/2017
	}
	]
	} "

2154889589	"{
	"**titre**":"NoSQL",
	"**auteur**": {
	"**id_auteur**":987,
	"**nom**":"Bruchez",
	"**prenom**":"Rudi"
	},
	"**vente**": [
	{
	"**id_vente**":12,
	"**date**":02/06/2017
	},
	{
	"**id_vente**":20,
	"**date**":12/09/2017
	}
	]
	} "

Exercice 01 «Du relationnel aux modèles NoSQL »

3. On considère le cas d'usage suivant: Accéder aux livres d'un auteur donné (recherche à partir d'un nom donné). Proposer une modélisation orientée colonne pour ce cas d'usage :



Modélisation sous Cassandra

Exercice 01 «Du relationnel aux modèles NoSQL »

3. On considère le cas d'usage suivant: Accéder aux livres d'un auteur donné (recherche à partir d'un nom donné). Proposer une modélisation orientée colonne pour ce cas d'usage :

Le nom et le prénom de l'auteur seront stockés dans la table livre. On évite ainsi les jointures!

Livre
-isbn -Titre **-nom** **-prénom**

Exercice 01 «Du relationnel aux modèles NoSQL »

ColumnFamily			
Key	**Value**		
Livre	**SuperColumns**		
	Key (isbn)	**Value (de la ligne)**	
	2154889522	**Column**	
		Name	**Value**
		titre	Asterix et Cléopatre
		nom	Gosciny
		prénom	René
	2154889589	**Column**	
		Name	**Value**
		titre	NoSQL
		nom	Bruchez
		prénom	Rudi

Exercice 01 «Du relationnel aux modèles NoSQL »

4. Proposer une modélisation orientée document de ces informations centrées sur les ventes, une modélisation centrée sur les livres et une sur les auteurs:

- Dans une modélisation relationnelle, nous avons dû séparer les livres, les auteurs et les ventes dans trois tables distinctes, et lier chaque livre à son auteur par une clé étrangère et aussi chaque vente à son livre par clé étrangère. Grâce à l'imbrication des structures, il est possible avec un document structuré de représenter l'information. On a imbriqué un objet dans un autre, ce qui ouvre la voie à la représentation d'une entité par un unique document complet.

- Nous n'avons plus besoin du système de référencement par clés primaires / clés étrangères, remplacé par l'imbrication qui associe physiquement les entités livre et auteur ainsi que livre et vente

Exercice 01 «Du relationnel aux modèles NoSQL »

a. Une modélisation centrée sur les ventes:

- Tout peut être représenté par un unique document structuré, en tirant parti de l'imbrication d'objets dans des tableaux. Nous obtenons une unité d'information autonome représentant l'ensemble des informations relatives à une vente.

- Prenons l'exemple de **la vente de id_vente=12**:

```
{
"id_vente":12,
"date"      :02/06/2017,
"livre"     : {  "isbn"     : 2154889589
                 "titre"    :"NoSQL",
                  "auteur": {
                              "id_auteur"  :987,
                              "nom"        :"Bruchez",
                              "prénom"     :"Rudi"
                             }
              }
}
```

Exercice 01 «Du relationnel aux modèles NoSQL »

b. Une modélisation centrée sur les livres:

- Prenons l'exemple de **de livre d'isbn=**2154889589 :

```
{
"isbn"     :2154889589
"titre"    :"NoSQL",
"auteur": {
              "id_auteur" :987,
              "nom"       :"Bruchez",
              "prénom"    :"Rudi"
            },
"vente":  [
            {
             "id_vente":12,
             "date"       :02/06/2017
            },
            {
             "id_vente":20,
             "date"       :12/09/2017
            }
          ]
}
```

Exercice 01 «Du relationnel aux modèles NoSQL »

c. Une modélisation centrée sur les auteurs:

- Prenons l'exemple de **de l'auteur de l'id_auteur=987** :

```
{
  "id"        :987,
  "nom"       :"Bruchez",
  "prénom":"Rudi"
  "livre"     : [
                 {
                   "isbn"   :2154889589
                   "titre"   :"NoSQL",
                   "vente":  [
                              {
                                "id_vente":12,
                                "date"       :02/06/2017
                               },
                              {
                                "id_vente":20,
                                "date"       :12/09/2017
                               }
                             ]
                  }
                ]
}
```

Exercice 01 «Du relationnel aux modèles NoSQL »

5. Indiquer l'impact que peut avoir la modélisation retenue sur le requêtage de la base:

- Efficace pour les interrogations par clé mais peut être limité pour les interrogations par le contenu des documents, limité aux données hiérarchiques.
- Si on prend l'exemple de modélisation centrée sur les ventes et on veut afficher pour chaque auteur la liste de livres qu'il les a écrit.
  - Il faut parcourir tous les documents,
  - Pour chaque document lu, il faut descendre dans la structure jusqu'à la clé livre,
  - Puis entrer dans la structure de de livre et accéder à la clé auteur s'il s'agit de même auteur garder le livre sinon passer au prochain document.

Exercice 01 «Du relationnel aux modèles NoSQL »

6. Expliquer les modifications nécessaires pour gérer plusieurs auteurs par livre dans la base de données relationnelle et dans les différents documents structurés:

a. Sur le schéma relationnel

Auteur(id_auteur, nom, prénom)
Livre(isbn, titre)
Vente(id_vente, date, isbn)
Ecrit(isbn, id_auteur)

b. Sur la modélisation orientée document

- Centrée sur les auteurs: pas de changement.
- Centrée sur les ventes ou bien livres: la clé auteur aura type tableau des auteurs.

Exercice 02 «Du document structuré au relationnel»

Exercice 2 : Du document structuré au relationnel

Le service informatique de l'USTHB a décidé de représenter ses données sous forme de documents structurés. Voici un exemple de documents centrés sur les étudiant(e)s et incluant les Unités d'Enseignement (UE) suivies par chacun(e).


```
{
 "_id": 978,
 "nom": "ADIMI Meriem",
 "UE": [{"id": "ue:11", "titre": "Java", "note": 12},
      {"id": "ue:27", "titre": "Bases de données", "note": 17},
      {"id": "ue:37", "titre": "Réseaux", "note": 14}
      ]
}
{
 "_id": 476,
  "nom": "BELABDI Ahmed",
  "UE": [{"id": "ue:13", "titre": "Méthodologie", "note": 17,
       {"id": "ue:27", "titre": "Bases de données", "note": 10},
       {"id": "ue:76", "titre": "Conduite projet", "note": 11}
       ]
}
```


1. Sachant que ces documents sont produits à partir d'une base relationnelle, reconstruire le schéma de cette base et indiquer le contenu des tables correspondant aux documents ci-dessus.
2. Proposer une autre représentation des mêmes données, centrée cette fois, non plus sur les étudiants, mais sur les UEs.

Exercice 02 «Du document structuré au relationnel»

1.Sachant que ces documents sont produits à partir d'une base relationnelle, reconstruire le schéma de cette base et indiquer le contenu des tables correspondant aux documents ci-dessus:

Table Etudiant	
id	**nom**
978	ADIMI Meriem
476	BELABDI Ahmed

Table UE	
id	**titre**
11	Java
13	Méthodologie
27	Bases de données
37	Réseaux
76	Conduite de projets

Table Evaluation		
idEtudiant	**idUE**	**note**
978	11	12
978	27	17
978	37	14
476	13	17
476	27	10
476	76	11

Exercice 02 «Du document structuré au relationnel»

Proposer une autre représentation des mêmes données, centrée cette fois, non plus sur les étudiants, mais sur les UEs:

```
{
  "_id": 11,
  "titre": "Java",
  "etudiants": [
              {"id": 978, "nom": "ADIMI Meriem", "note": 12}
             ]
}
```

```
{
  "id":13,
  "titre": "Méthodologie",
  "etudiants": [
         {"id": 476, "nom": "BELABDI Ahmed", "note": 17}
        ]
}
```

```
{
  "id": 27,
  "titre": « Base de données",
  "etudiants": [
        {"id": 978, "nom": "ADIMI Meriem", "note": 17},
        {"id": 476, "nom": "BELABDI Ahmed", "note": 10}
     ]
}
```

```
{
  "id":37,
  "titre": "Réseaux",
  "etudiants": [
        {"id": 978, "nom": "ADIMI Meriem", "note": 14}
      ]
}
```

```
{
   "id":76,
   "titre": "Conduite projet",
   "etudiants": [
         {"id": 476, "nom": "BELABDI Ahmed", "note": 11}
       ]
}
```

Exercice 03 «Modélisation orientée documents »

Exercice 3 : Modélisation orientée documents

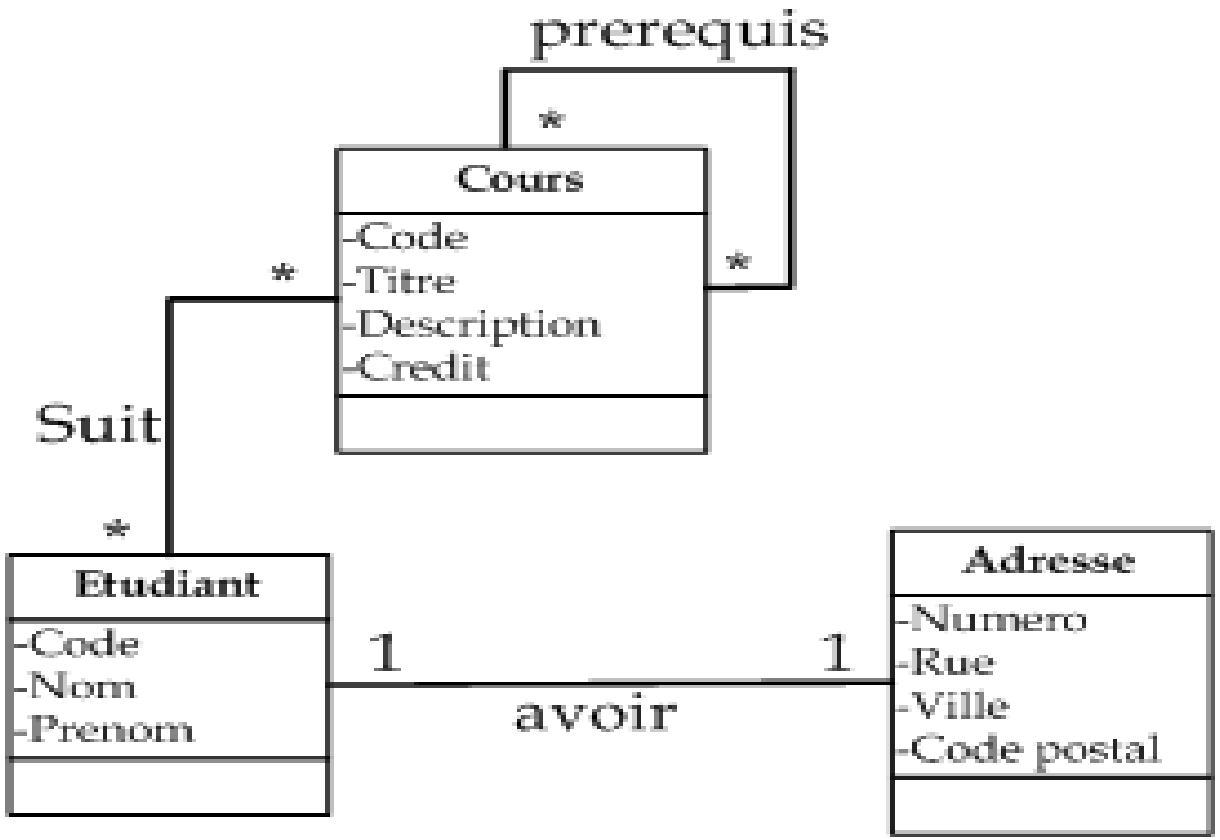
On souhaite réaliser une base de données orientée documents pour gérer des cours et des étudiants, étant données les informations suivantes :

- Un cours est décrit par les attributs code, titre, description, crédits et pré requis.
- Les prés requis sont d'autres cours.
- Un étudiant est décrit par les attributs nom, prénom, adresse.
- Les adresses sont composées d'un numéro de rue, d'une rue, d'une ville et d'un code postal.
- Un étudiant suit plusieurs cours et les cours sont suivis par plusieurs étudiants.

1. Décrire ces informations sous forme d'un diagramme UML.
2. Discuter des différentes modélisations possibles orientées documents (par exemple centrées sur les étudiants ou centrées sur les cours) en donnant des documents JSON correspondant.
3. Si l'objectif de l'application est de visualiser une liste des étudiants avec les cours que chacun suit, et d'accéder aux détails des cours uniquement lorsque l'on sélectionne son code ou son titre, proposer une solution adaptée à ce problème.

Exercice 03 «Modélisation orientée documents »

1. Décrire ces informations sous forme d'un diagramme UML.



Exercice 03 «Modélisation orientée documents »

2. Discuter des différentes modélisations possibles orientées documents (par exemple centrées étudiants ou centrées sur les cours) en donnant des documents JSON correspondant.
   a. Solution favorisant Cours (en se basant sur l'imbrication)

```
{
"code":"BDA",
"titre":"Bases de Données Avancées",
"description":".....",
"credits":5
"prerequis":
                    [
                        {
                            "code":" GDC",
                            "titre":" Gestion de données dans le Cloud ",
                            "description":".....",
                            "credits":5
                            "prerequis": [...]
                        },
                        ...
                    ]
"etudiants":
                    [
                        {
                            "nom":"LADIMI",
                            "prenom":"Manel",
                            "adresse":
                                        {
                                            "num":8,
                                            "rue":"Didouche Mourad"
                                        }
                        },
                        ...
                    ]
}
```

Exercice 03 «Modélisation orientée documents »

b. Solution favorisant les étudiants (en se basant sur l'imbrication)

```
{
"nom":"LADIMI",
"prenom":"Manel",
"adresse":
                {
                        "num":8,
                        "rue":"Didouche Mourad"
                }
"suit":
   [
                {
                        "code":"BDA",
                        "titre":"Bases de Données Avancées",
                        "description":".....",
                        "credits":5
                        "prerequis":
                                                [
                                                        {
                                                        "code":"GDC",
                                                        "titre":" Gestion de données dans le Cloud ",
                                                        "description":".....",
                                                        "credits":5
                                                        "prerequis": [...]
                                                        },
                                                        ...
                                                ]
                },
                ...
        ]
}
```

Exercice 03 «Modélisation orientée documents »

Le défaut principal de cette solution est sa redondance :

- toutes les données de tous les étudiants sont copiées dans chaque cours ;
- tous les cours sont copiés à chaque fois qu'ils sont pré-requis d'un autre cours.

Si l'on avait un cycle de référence, par exemple deux cours pré-requis l'un de l'autre, alors on aurait nécessairement besoin d'identification et de références.

Un exemple JSON basé sur les références.

Cours


```
{
"_id":"30ae9e51-f5c8-4022-a68d-3f3948dbdcb1",
"code":"BDA",
"titre":"Base de Données Avancées",
"description":".....",
"credits":5,
"prerequis":["a1449020-9b24-44a1-b12d-84ef592f8853","cf936817-19fa-4635-b009-a383c90ab6d7"]
}
```


Étudiant


```
{
"_id":"850b1657-a070-4a25-ab63-6f61b436cf9d",
"nom":"LADIMI",
"prenom":"Manel",
"cours": ["30ae9e51-f5c8-4022-a68d-3f3948dbdcb1","cf936817-19fa-4635-b009-a383c90ab6d7"]
}
```


Adresse


```
{
"_id":"7e93be68-7d6e-4506-9be5-ffd78e5afb5d",
"etudiant":"850b1657-a070-4a25-ab63-6f61b436cf9d",
"num":8,
"rue":"Didouche Mourad"
}
```

Exercice 03 «Modélisation orientée documents »

3. Si l'objectif de l'application est de visualiser une liste des étudiants avec les cours que chacun suit, et d'accéder aux détails des cours uniquement lorsque l'on sélectionne son code ou son titre, proposer une solution adaptée à ce problème.

```
Étudiant
{
"nom":"LADIMI",
"prenom":"Manel",
"adresse":
                {
                "num":8,
                "rue":"Didouche Mourad"
                },
"cours":
                [
                        {
                                "_id":"30ae9e51-f5c8-4022-a68d-3f3948dbdcb1",
                                "code":"BDA",
                                "titre":"Base de données Avancées"
                                },
                                {
                                "_id":"cf936817-19fa-4635-b009-a383c90ab6d",
                                "code":"GDC",
                                "titre":"Gestion de données dans le Cloud"
                                }
                ]
}
```

```
Cours
{
"_id":"30ae9e51-f5c8-4022-a68d-3f3948dbdcb1",
"code":"BDA",
"titre":"Bases de Données Avancées",
"description":".....",
"credits":5,
"prerequis":
                        [
                        {
                                "_id":"a1449020-9b24-44a1-b12d-84ef592f8853",
                                "code":"GDC",
                                "titre":"Gestion de données dans le Cloud"
                        }
                        ]
}
```

Références

Pour les diapositives rappel sur NoSQL sont tirées de cours de Philippe Declercq "Bases de Données Approfondies – 2ème partie- Théorie et pratique de NoSQL".