NoSQL

NOSQL

Pourquoi ?

- NoSQL signifie “Not Only SQL”, “pas seulement SQL”
- ce terme désigne l’ensemble des bases de données qui s’opposent à la notion relationnelle des SGBDR
- NoSQL ne vient pas remplacer les BD relationnelles mais proposer une alternative ou compléter les fonctionnalités des SGBDR pour donner des solutions plus intéressantes dans certains contextes
- NoSQL répond à une nécessité de performance de traitement de données en très gros volumes
- avec NoSQL, on parlera souvent de scalabilité

NoSQL

- NoSQL ne se substitue pas aux bases de données traditionnelles
- il est un complément et un palliatif à leurs lacunes, et repose sur des systèmes distribués, permettant de gérer et coordonner plusieurs ordinateurs reliés en un réseau et communiquant par envoi de messages
- le matériel n'est pas spécialisé, il est même considéré comme standard et peut ainsi être remplacé facilement
- dans ce type d'architecture, le nombre fait la force

Les contraintes ACID

- les SGBD relationnels sont généralement transactionnels, les transactions respectent les contraintes ACID (en anglais : Atomicity, Consistency, Isolation, Durability, en français : Atomicité, Cohérence, Isolation, Durabilité) qui garantissent qu'une transaction est fiable
- **Atomicité**
  - une transaction se fait complètement … ou pas du tout
  - si une partie de la transaction ne peut pas être réalisée, on annule la transaction et on restaure les données à leur état initial
- **Cohérence**
  - chaque transaction respecte les contraintes d'intégrités définies dans la base de données par le concepteur de la base de données
  - après chaque transaction complétée, le système est laissé dans un état valide
  - la cohérence s'appuie sur le respect des contraintes d'intégrité

Les contraintes ACID

- **Isolation**
  - l'isolation assure qu'une transaction doit s'exécuter comme si elle était seule sur le système
  - cela implique que les transactions soient indépendantes
  - c'est une contrainte difficile à respecter, notamment lors de mises à jour concurrentielles
- on parle de transactions optimistes ou pessimistes
  - optimistes : le moteur de base de données considère que les autres transactions sont systématiquement réussies (pas de lock d'enregistrement ou de lock de table / fichier)
  - pessimistes : les ressources (enregistrement ou table) sont verrouillées jusqu'à la fin d'une transaction pour empêcher toute interférence entre deux transactions sur la même donnée ou sur la même table
  - exemple : modification d'une fiche client impossible si un autre utilisateur est en cours de modification soit sur la même fiche, soit sur une autre fiche client
  - c'est ce que fait le moteur ISAM en relationnel qui verrouille une table complète à chaque modification d'un enregistrement

Tableau comparatif de deux moteurs relationnels

Critères	InnoDB	MyISAM
Transactions	Oui	Non
Relations	Oui	Non
Cascade	Oui	Non
Intégrité référentielle	Oui	Non
Verrouillage	Enregistrement	Table
Utilisation des ressources	Correcte	Peu gourmand
Vitesse	Correcte	Maximale
Gestion des textes	Correcte	Optimisée

Les contraintes ACID

- **Durabilité**
  - la durabilité assure que lorsqu'une transaction a été confirmée, elle demeure enregistrée même à la suite d'un problème (coupure, panne ...)
  - on parle de persistance des données : toute donnée enregistrée doit l'être de façon durable
  - l'écriture des données sur le disque ne suffit pas si par exemple le support est endommagé
  - le système de base de données peut palier à cela grâce à des logs qui enregistrent toute l'activité de la base
  - en cas de problème sur la base, on va relire les logs pour reconstituer la base de données

NOSQL

Les 4 modèles

Clé-Valeur (key value)

- collection de couples constitués d'une clé et d'une valeur
- produits : Redis, AmazonDB, Microsoft Azure Cosmos DB, Memcached

Colonne (wide column)

- modèle clé-valeur enrichi permettant de stocker plusieurs valeurs et associations one-to-many
- produits : Cassandra, Hbase, Microsoft Azure Cosmos DB

NOSQL

Les 4 modèles

Document (document)

- modèle clé-valeur enrichi permettant de stocker une valeur complexe, un document
- chaque document est composé de champs et de valeurs associées
- produits : MongoDB, AmazonDB, Microsoft Azure Cosmos DB, Couchbase, Firebase, RavenDB

Graphe (graph)

- gestion de relations multiples entre les objets
- basé sur la théorie des graphes
- produits : Neo4J, Microsoft Azure Cosmos DB, Arango DB, OrientDB

NOSQL

1
le modèle clé - valeur

Caractéristiques

- la base de données se présente comme un tableau associatif unidimensionnel
- chaque objet de la base est un couple clé - valeur
- il est identifié par une clé unique qui est le seul moyen d'accès à l'objet
- la clé est le seul moyen de requêtage
- la valeur peut être une simple chaîne de caractères, ou un objet sérialisé
- la structure de l'objet est libre (XML, JSON, ...)

NOSQL

1
le modèle clé - valeur

- absence de structure ou de typage
- impact important sur le requêtage : toute l'intelligence portée auparavant par les requêtes SQL devra être portée par l'applicatif qui interroge la base
- il faut donc de meilleurs développeurs

Format

- enregistrement 1 : clé 1 – valeur 1
- enregistrement 2 : clé 2 – valeur 2
- …
- enregistrement n : clé n – valeur n

NOSQL

1
le modèle clé - valeur

Opérations : les 4 opérations du CRUD

- Create (clé,valeur) : crée un couple (clé,valeur)
- Read (clé) : lit une valeur à partir de sa clé
- Update (clé,valeur) : modifie une valeur à partir de sa clé
- Delete (clé) : supprime un couple (clé,valeur) à partir de sa clé

Cas d'utilisation

- requêtage simple pour des résultats rapides et en temps réel
- sessions web
- fichiers de log
- gestion de profils utilisateurs
- données de panier d'achat …

NOSQL

1
le modèle clé - valeur

Les plus

- simple, très performant en lecture et écriture
- bonne mise à l'échelle horizontale pour les lectures et écritures
- pas ou peu de maintenance du fait de la simplicité

Les moins

- interrogation seulement sur la clé
- modèle de données trop simple donc pauvre pour les données complexes

NOSQL

2
le modèle colonne

Caractéristiques

- les données sont stockées en colonnes, et pas en lignes
- la colonne est l'entité de base représentant un champ de données
- chaque colonne est identifiée par un identifiant unique
- une colonne peut contenir d'autres colonnes
- une colonne contenant d'autres colonnes est nommée super-colonne
- les super-colonnes correspondent à une table de jointure dans le modèle relationnel

NOSQL

2
le modèle colonne

- ce modèle ressemble à une table dans un SGBD relationnel
- mais ici le nombre de colonnes est dynamique
- le nombre de colonnes peut varier d'un enregistrement à un autre ce qui évite de retrouver des colonnes avec des valeurs à NULL

Format (exemple)

- enregistrement 1 : clé 1 – nom – prénom – code postal – ville
- enregistrement 2 : clé 2 – nom – email
- enregistrement 3 : clé 3 – nom – prénom – email

NOSQL

2
le modèle colonne

Opérations : les 4 opérations du CRUD

- Create : crée un enregistrement (la clé et les valeurs de la colonne)
- Read : lit une ou plusieurs valeurs à partir de la clé
- Update : modifie une ou plusieurs valeurs à partir de la clé
- Delete : supprime un enregistrement
- on peut exécuter des requêtes utilisant colonnes et super-colonnes

NOSQL

2
le modèle colonne

Cas d'utilisation

- traitements analytiques en ligne (OnLine Analytical Processing- OLAP)
- exploration de données (data mining)
- entrepôt de données (Data Warehouse)
- journalisation d'événements
- stockage de listes (messages, posts, commentaires, ...)

NOSQL

2
le modèle colonne

Les plus

- bonne mise à l'échelle horizontale
- un nombre de colonnes dynamique : variable d'un enregistrement à un autre

Les moins

- ne supporte pas les données structurées complexes
- maintenance nécessaire pour la modification de la structure en colonne
- ajout de ligne couteux en temps de réponse

NOSQL

3
le modèle document

Caractéristiques

- la base de données stocke une collection de documents sur le modèle clé-valeur
- la valeur est un document lisible par un humain dans un format semi-structuré hiérarchique (JSON, XML …)
- l'avantage est de pouvoir récupérer via une seule clé un ensemble d'informations structurées de manière hiérarchique

NOSQL

3
le modèle document

- la même opération dans un contexte relationnel impliquerait plusieurs jointures
- un document n'a pas de schéma (schemaless)
- c'est une structure arborescente qui contient une liste de champs
- un champ possède une valeur qui est soit de type simple (entier, caractère, date …) ou peut être composé de plusieurs couples clés / valeurs

NOSQL

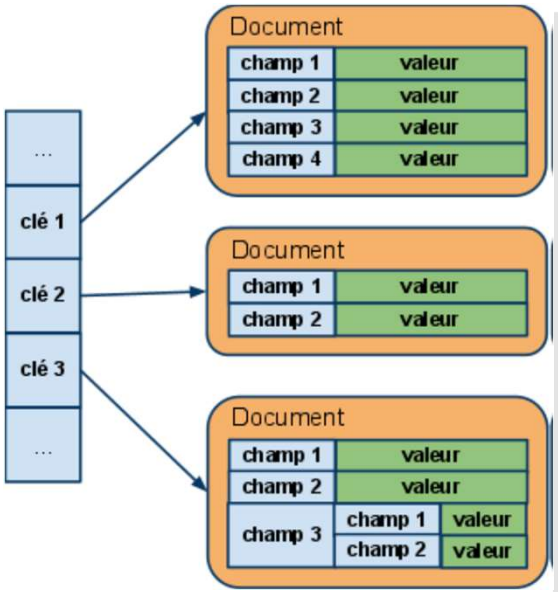
3
le modèle document

Format

Clé 1 → document avec 3 valeurs
Champ 1 → Valeur 1
Champ 2 → Valeur 2
Champ 3 → Valeur 3
Champ 4 → Valeur 4

Clé 2 → document avec seulement 2 valeurs
Champ 1 → Valeur 1
Champ 2 → Valeur 2

Clé 3 → document avec 3 champs dont un (le champ 3) qui contient 2 autres champs (les champs 1 et 2 qui contiennent eux aussi des valeurs)
Champ 1 → Valeur 1
Champ 2 → Valeur 2
Champ 3 → Champ 1 → Valeur
→ Champ 2 → Valeur



Film


```
{
 "titre": "Pulp fiction", "année": "1994", "genre": "Action", "pays": "USA",
        "Réalisateur": {
                "nomReal": "Tarantino", "prénomReal": "Quentin", "anneeNaissReal": "1963"
        },
        "Acteurs": {
                {"prénom": "John", "nom": "Travolta", "anneeNaiss": "1954", "rôle": "Vincent Vega"},
                {"prénom": "Bruce", "nom": "Willis", " anneeNaiss": "1955", "rôle": "Butch Coolidge" },
                {" prénom": "Quentin", "nom": "Tarantino", "anneeNaiss": "1963", "rôle": "Jimmy Dimmick"}
        },
        "Genres": ["Action", "Policier", "Comédie"]
}
```

NOSQL

3
le modèle document

Opérations : les 4 opérations du CRUD

- on peut exécuter des requêtes ou mettre en place des API sur les valeurs contenues dans les documents

Cas d'utilisation

- catalogues de produits
- web analytique
- enregistrement d'événements / logs
- stockage de profils utilisateur ...

NOSQL

3
le modèle document

Les plus

- performances élevées
- simplicité du modèle
- possibilité de requêtes plus complexes que dans les autres modèles
- utilisation de structures imbriquées
- forte communauté

Les moins

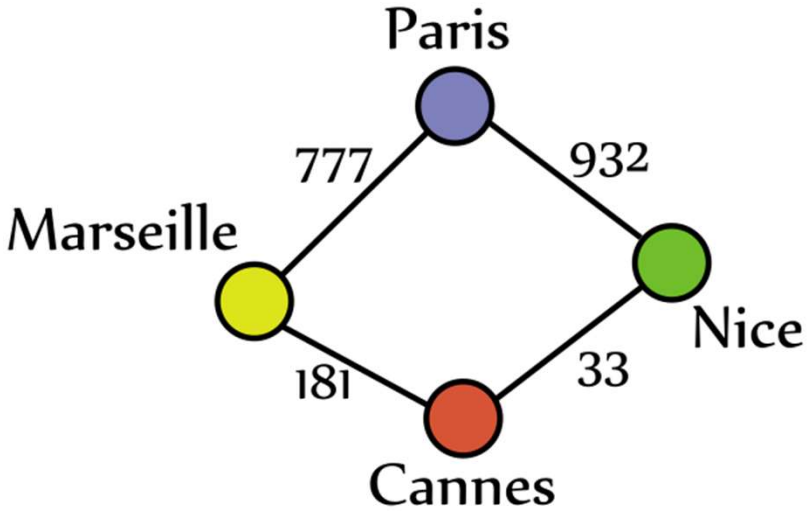
- limité pour les interrogations autres que par clé
- redondances de données

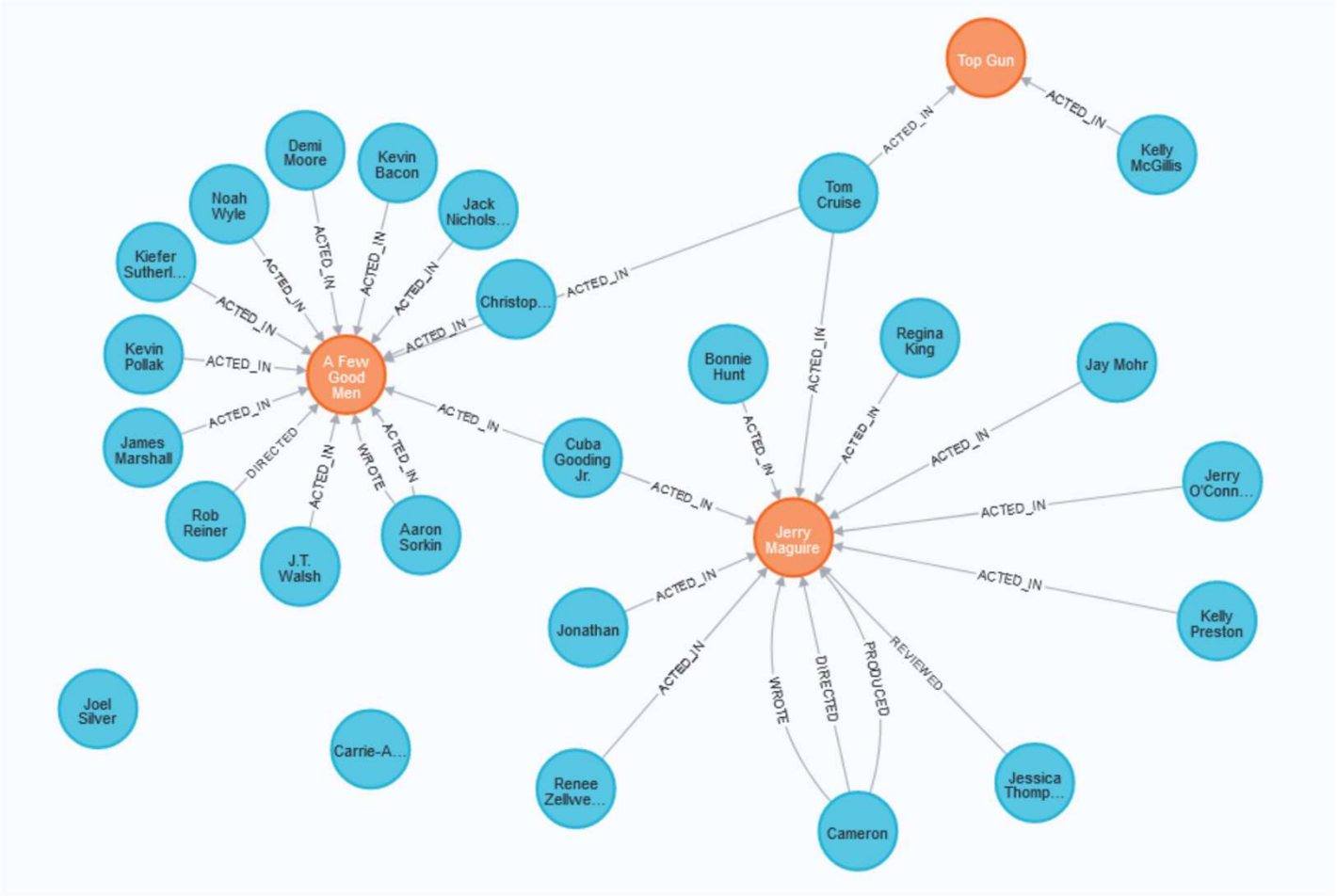
NOSQL

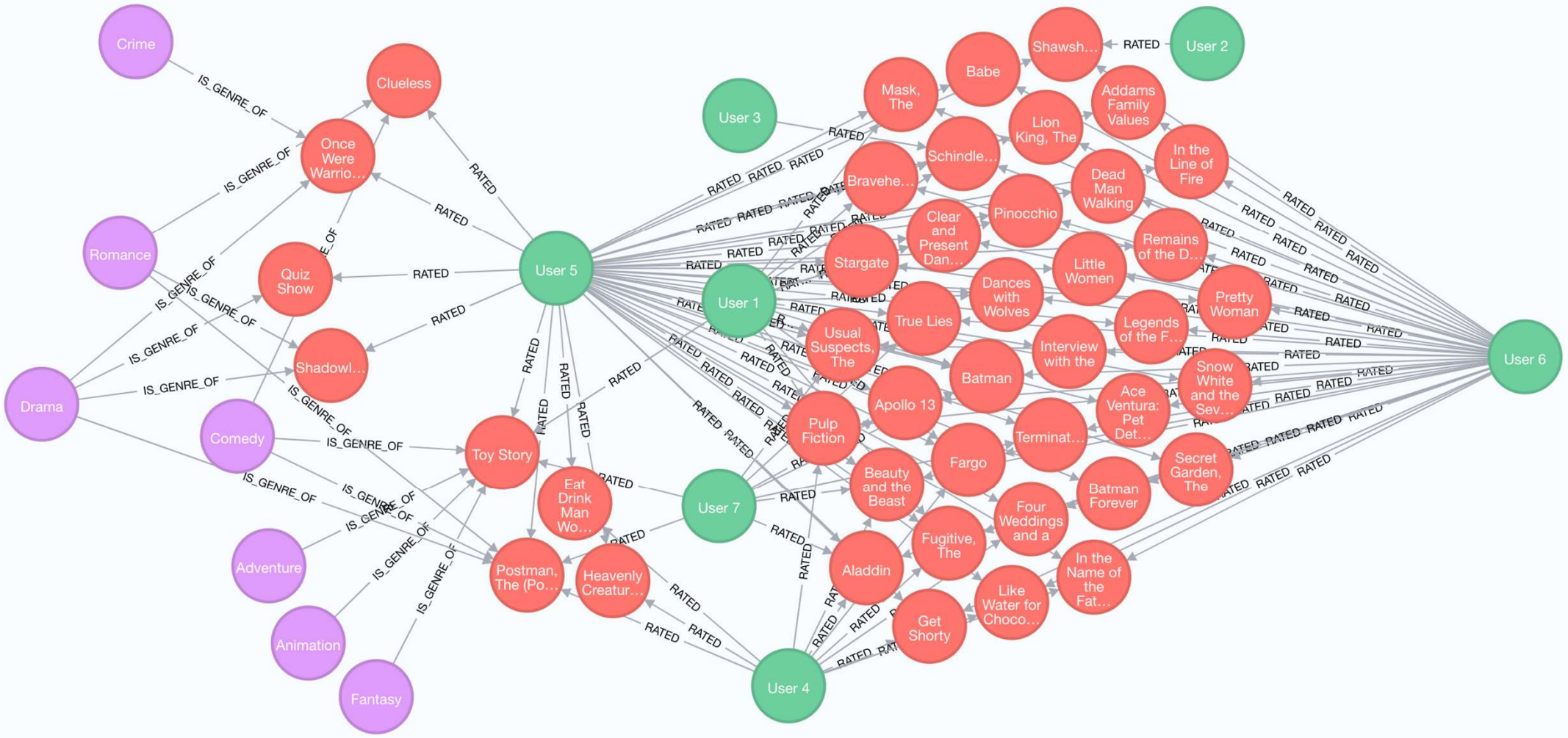
4
le modèle graphe

Caractéristiques

- modèle basé sur la théorie des graphes
- permet la gestion d'un graphe (orienté ou non)
- modélisation, stockage et manipulation de données complexes liées par des relations
- s'appuie sur les notions de nœuds, relations et propriétés




Top Gun
ACTED_IN
Kelly McGillis
Tom Cruise
Demi Moore
Kevin Bacon
Noah Wyle
Jack Nichols...
Kiefer Sutherl...
Christop...
Kevin Pollak
A Few Good Men
James Marshall
Rob Reiner
J.T. Walsh
Aaron Sorkin
DIRECTED
WROTE
Cuba Gooding Jr.
Bonnie Hunt
Regina King
Jay Mohr
Jerry Maguire
Jerry O'Conn...
Kelly Preston
Jonathan
Joel Silver
Carrie-A...
Renee Zellwe...
Cameron
Jessica Thomp...
REVIEWED
PRODUCED


Crime
Romance
Drama
Comedy
Adventure
Animation
Fantasy
IS_GENRE_OF
Clueless
Once Were Warrio…
Quiz Show
Shadowl…
Toy Story
Postman, The (Po…
Eat Drink Man Wo…
Heavenly Creatur…
RATED
User 1
User 2
User 3
User 4
User 5
User 6
User 7
Shawsh…
Babe
Mask, The
Addams Family Values
Lion King, The
Schindle…
In the Line of Fire
Bravehe…
Dead Man Walking
Pinocchio
Clear and Present Dan…
Stargate
Remains of the D…
Little Women
Dances with Wolves
Pretty Woman
True Lies
Usual Suspects, The
Legends of the F…
Interview with the
Batman
Snow White and the Sev…
Apollo 13
Ace Ventura: Pet Det…
Pulp Fiction
Terminat…
Fargo
Secret Garden, The
Beauty and the Beast
Batman Forever
Four Weddings and a
Fugitive, The
Aladdin
In the Name of the Fat…
Like Water for Choco…
Get Shorty

NOSQL : le modèle graphe

Cas d'utilisation

- informatique décisionnelle
- internet des objets (Internet of things (IoT))
- données hiérarchiques (catalogue des produits, généalogie, ...)
- réseaux sociaux
- réseaux de transport
- cartographie, GPS
- services de routage et d'expédition ...

NOSQL : le modèle graphe

Les plus

- modèle adapté aux situations où il faut modéliser beaucoup de relations
- forte communauté : nombreux langages et API existants
- très adapté aux objets organisés en réseau : cartes, réseaux sociaux …

Les moins

- modèle beaucoup trop typé car trop basé sur la théorie des graphes
- donc à ne pas utiliser pour les cas simples