

# Bases de données - NoSQL

---

## CM 2

## Les caractéristiques du NoSQL

Namrata PATEL

Namrata.Patel@univ-montp3.fr

# Le modèle relationnel

---

## Principes de base

# Le modèle relationnel

---

- **Entités-Associations :**
  - Les données sont modélisées sous forme de **tables reliées entre elles** par des relations sur des clefs
- **Formes Normales :**
  - Ces **relations respectent des propriétés** spécifiques afin de garantir la robustesse
- **Transactions :**
  - Actions liées à l'accès et la modification de la base de données
- **OLTP :**
  - Protocole de transactions **respectant les propriétés ACID**

# Formes Normales

- **8 formes normales dans les systèmes OLTP**
  - 3 premières les plus connues et utilisées
    - 1FN :
      - Atomicité, 1 seule valeur par colonne
    - 2FN :
      - les attributs non clé dépendent de toute la clé
      - et non d'une partie de la clé
    - 3FN :
      - chaque attribut de la relation ne dépend que de la clé
      - et pas d'un autre attribut de la relation

Nom	Prénom	Ville	Pays
<u>Patel</u>	<u>Namrata</u>	Montpellier	France
<u>Patel</u>	<u>Ekta</u>	Pondichéry	Inde

# Les transactions ACID

---

## Critères ACID

### – Atomicity / Indivisible

- Tout ou rien

### – Consistency / Cohérence

- D'un état valide à un autre état valide

### – Isolation / Verrouillage

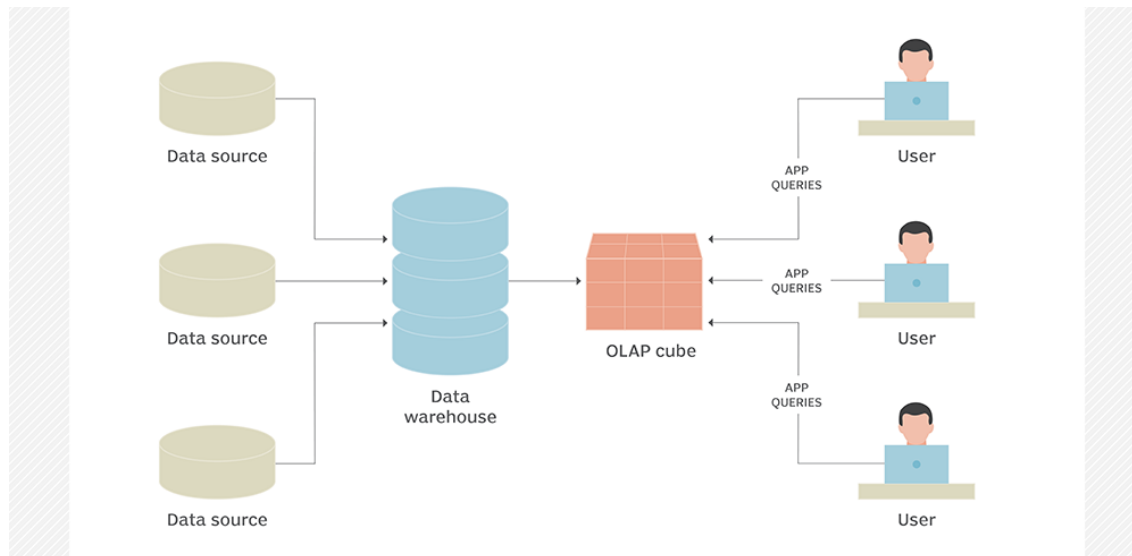
- L'exécution simultanée de transactions produit le même état que celui qui serait obtenu par l'exécution en série des transactions

### – Durability / Persistance

- les résultats sont enregistrés de façon permanente

# OLAP – Traitement des données complexes

- **OLAP**
  - Solution pour les modèles relationnels lorsque le OLTP ne suffit pas
  - Caractérisé par 12 règles



# 12 règles OLAP

---

## 1. Multidimensionnalité

- Le Modèle OLAP est multidimensionnel par nature

## 2. Transparence

- L'emplacement physique du serveur OLAP est transparent pour l'utilisateur.

## 3. Accessibilité

- L'utilisateur OLAP dispose de l'accessibilité à toutes les données nécessaires à ses analyses

## 4. Stabilité

- La performance reste stable indépendamment du nombre de dimensions.

# 12 règles OLAP

---

## 5. Client-Serveur

- Le serveur OLAP s'intègre dans une architecture client serveur.

## 6. Dimensionnement

- Le dimensionnement est générique afin de ne pas fausser les analyses.

## 7. Gestion complète

- Le serveur OLAP assure la gestion des données clairessemées.

## 8. Multi-utilisateurs

- Le serveur OLAP offre un support multi-utilisateurs (gestion des mises à jour, intégrité, sécurité).



# 12 règles OLAP

---

## 9. Inter Dimension

- Le serveur OLAP permet la réalisation d'opérations inter dimensions sans restriction.

## 10. Intuitif

- Le serveur OLAP permet une manipulation intuitive des données.

## 11. Flexibilité

- La flexibilité (ou souplesse) de l'édition des rapports est intrinsèque au modèle.

## 12. Analyse sans limites

- Le nombre de dimensions et de niveaux d'agrégation possibles est suffisant pour autoriser les analyses les plus poussées.

# Comparatif

	OLTP	OLAP
<b>Utilisateurs</b>	Lambda	Spécialisés
<b>Fonctions</b>	Journalier	Décision
<b>Design DB</b>	Orienté application	Orienté sujet
<b>Données</b>	Courantes, détaillées, plates, à jour	Historiques, résumées, multi-dimensionnelles, intégrées, consolidées
<b>Utilisation</b>	Répétitive	Ad hoc
<b>Accès</b>	Écrire/Lire Index / Hachage sur les clés primaires	Lire
<b>Transactions</b>	Transaction courte et simple	Requêtes complexes

# ...Et dans le contexte Big Data ?

---

## Difficultés, enjeux

# Normalisation

---

- **Séparation en tables**
  - Complique l'accès aux données
  - La réorganisation des données dans une nouvelle table est couteuse
- **Modification de schéma**
  - Rigidité relative du modèle relationnel
  - La séparation des données implique des jointures dans les requêtes
- **Donc :**
  - **Partitionnement**

# Les transactions ACID

---

## Critères ACID

### – Atomicity / Indivisible

- Tout ou rien
- Volumétrie trop importante pour respecter l'atomicité

### – Consistency / Cohérence

- D'un état valide à un autre état valide
- Transitions longues; il se peut que la donnée soit invalide lors de la transaction

### – Isolation / Verrouillage

- L'exécution simultanée de transactions produit le même état que celui qui serait obtenu par l'exécution en série des transactions
- Computationnellement impossible à maintenir

### – Durability / Persistence

- les résultats sont enregistrés de façon permanente

# Le partitionnement des données

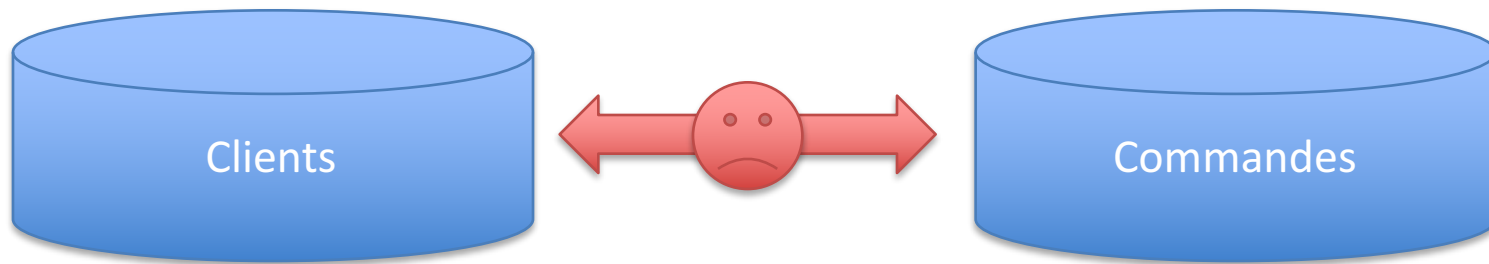
## Système distribué

- Système logiciel qui permet de coordonner plusieurs ordinateurs.
  - Généralement, cette coordination se fait par envoi de messages via un réseau auquel sont reliés ces ordinateurs.
- Pourquoi ?
  - Parce qu'on manipule un très grand volume de données.
  - Sans distribution, on n'a pas d'application "scalable".



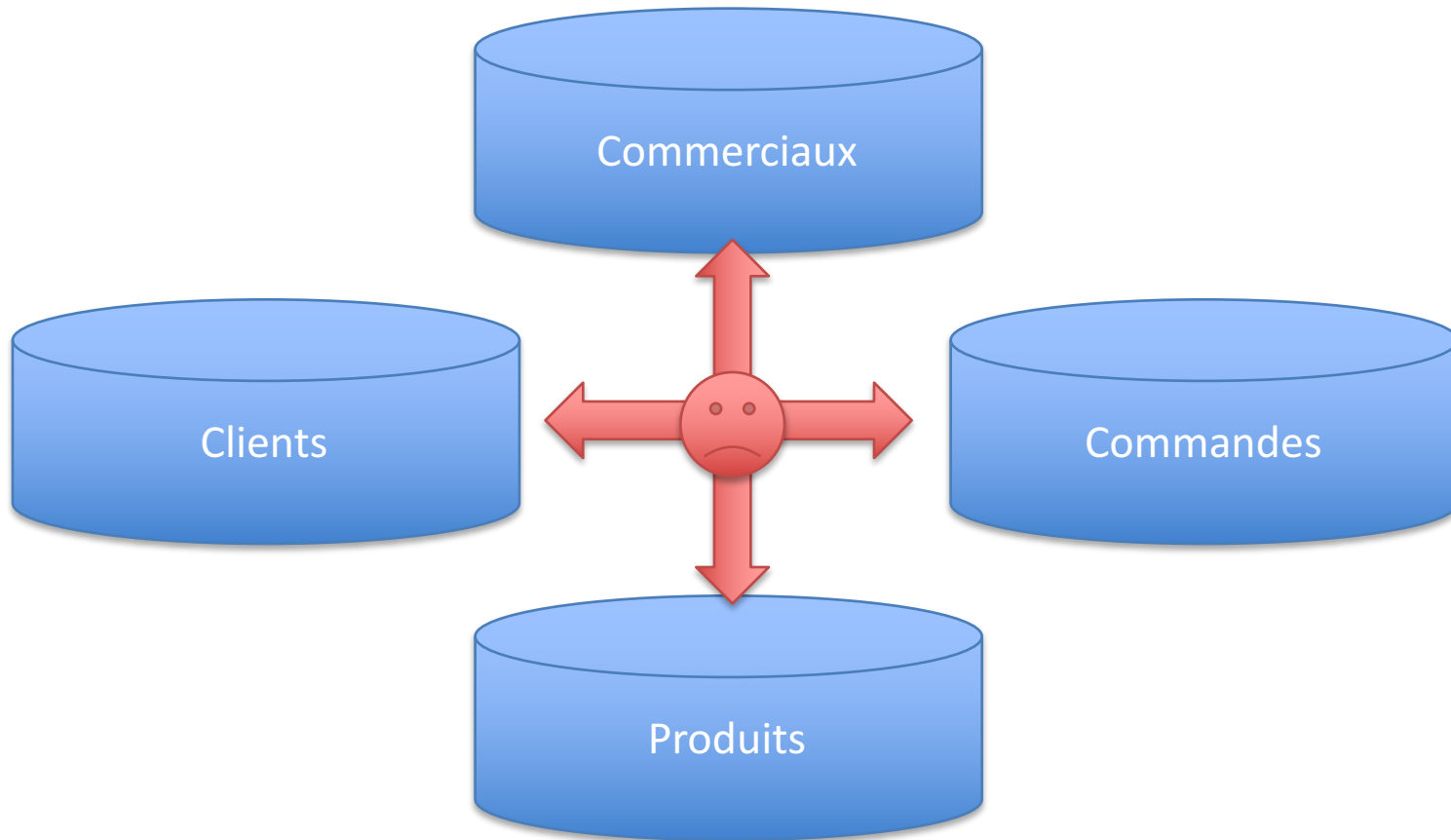
# Le partitionnement des données

- **Comment récupérer les commandes pour un client « Alpha » ?**



# Le partitionnement des données

---





# Le théorème de CAP

---

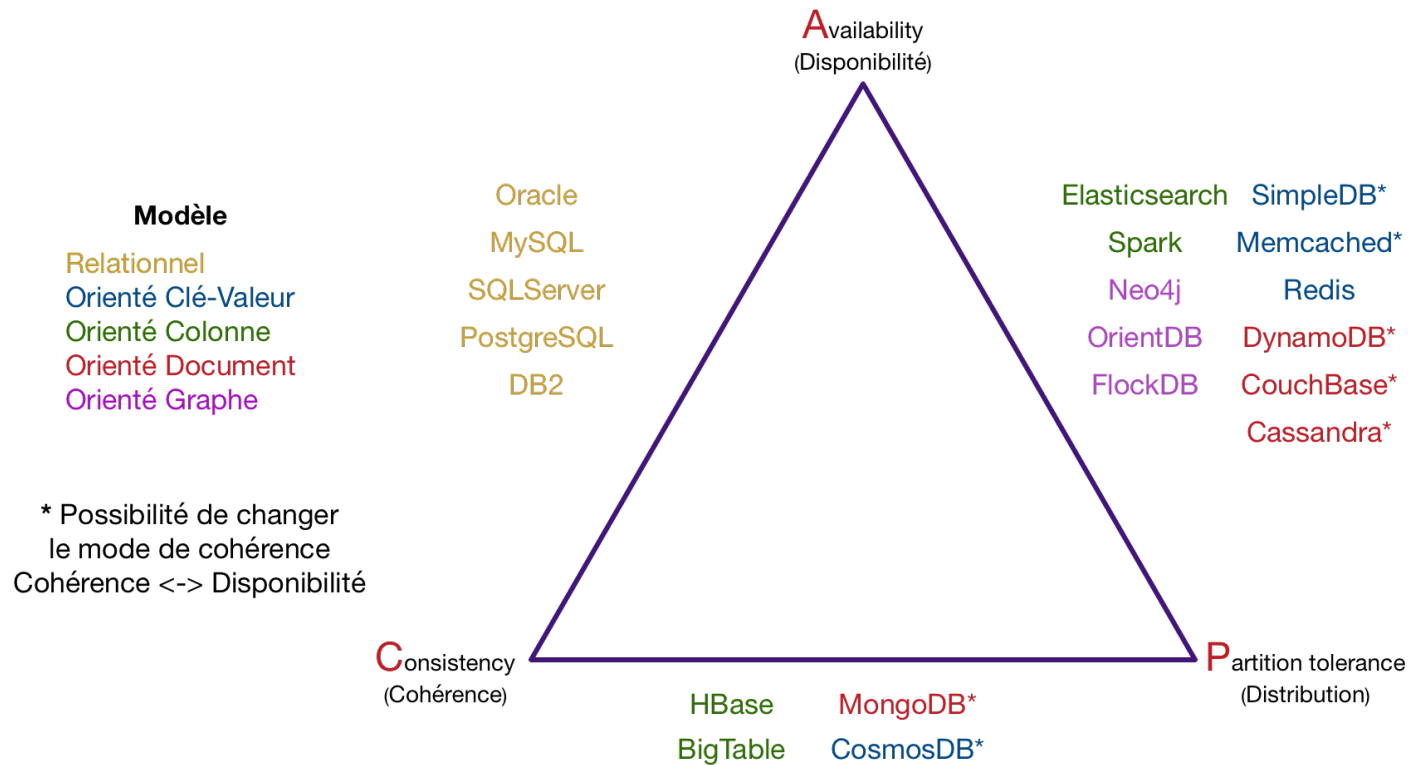
Le « 42 » des SGBD

# THÉORÈME DU CAP

---

- **C**onsistency (Cohérence)
  - Tous les noeuds sont à jour au même moment
- **A**vailability (Disponibilité)
  - La perte d'un noeud n'empêche pas le système de restituer l'intégralité des données
- **P**artition Tolerance (Résistance au morcellement)
  - Chaque noeud doit pouvoir fonctionner de manière autonome

# CAP et NoSQL



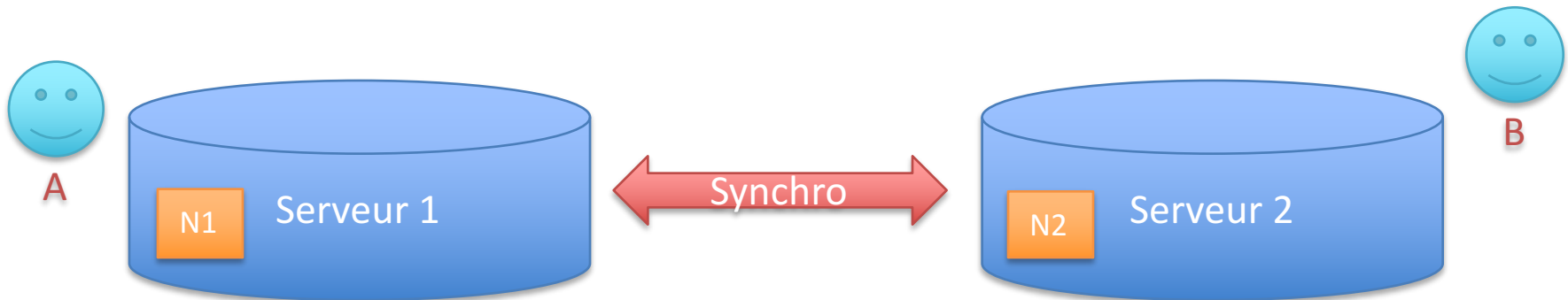
# Transactions BASE

---

- **Basically Available**
  - En général, disponibilité des données
- **Soft State**
  - L'état du système peut changer même si aucune donnée est modifiée
- **Eventually Consistent**
  - Mais éventuellement, les systèmes doivent assurer la cohérence des données

# Exemple

- Soit A et B deux utilisateurs du système, et Noeud 1 et Noeud 2, deux noeuds du système.
- Si A modifie une valeur sur Noeud 1, alors pour que B voie cette valeur sur Noeud 2, il faut attendre que Noeud 1 et Noeud 2 soient synchronisés.



# Conclusion

---

Résumé, comparatifs

# Comparatif SQL vs NoSQL

	Relationnel (SQL)	NoSQL
Capacité de stockage	Modérée (< 1 To)	Très élevée (> 1 To)
Architecture	Centralisée	Distribuée
Modèle de données	Relationnel (tabulaire)	Destructuré
Réponse à la charge	Lecture en majorité	Lecture et écriture
Scalabilité	Horizontale (nombre)	Verticale (puissance)
Moteur de requêtes	SQL	Propre au système
Principales caractéristiques	ACID	BASE
Ancienneté de la technologie	Eprouvée	Récente