

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

généralisation /
spécialisation

Spécification des
diagrammes EA

Contraintes
d'intégrité

NoSQL

Normalisation
d'une relation

- 1 Les concepts généraux
- 2 Notions essentielles pour le modèle relationnel
- 3 Algèbre Relationnelle et SQL
- 4 Le modèle conceptuel Entités-Associations
 - Concepts de base et Diagramme EA
 - Représentations multiples : généralisation/spécialisation
 - La spécification des diagrammes EA
 - Contraintes d'intégrité
- 5 NoSQL
- 6 Normalisation d'une relation

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

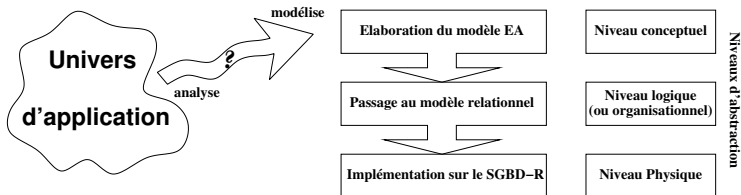
le modèle Entités- Associations

Concepts de base et
Diagramme EA
généralisation /
spécialisation
Spécification des
diagrammes EA
Contraintes
d'intégrité

NoSQL

Normalisation d'une relation

- Les BD constituent le cœur du système d'information.
- La conception de ces bases de données est la tâche la plus ardue du processus de développement du système d'information. :
 - conception bien gérée $\Rightarrow$ **développement facile.**
 - si des difficultés non révélées $\Rightarrow$ **la phase de développement sera obligée de reprendre des points non éclaircis lors de la phase de conception. De nombreux bugs verront le jour.**
- Méthodes de conception pour assister la démarche de conception.
3 étapes principales $\sim$ 3 niveaux d'abstraction différents :



Introduction aux BDR

Jean-Paul Comet,
Nadia Abchiche-Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités-Associations

Concepts de base et Diagramme EA

généralisation / spécialisation

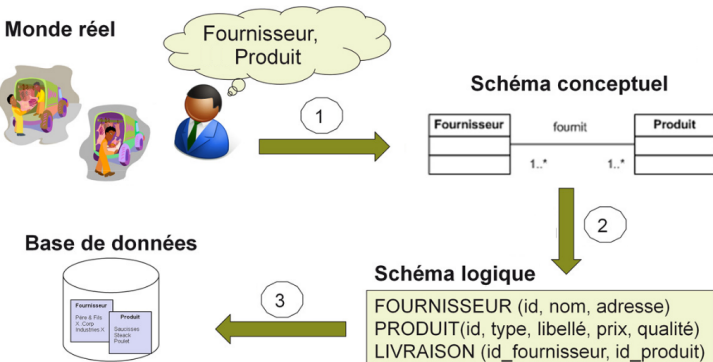
Spécification des diagrammes EA

Contraintes d'intégrité

NoSQL

Normalisation d'une relation

- Les BD constituent le cœur du système d'information.
- La conception de ces bases de données est la tâche la plus ardue du processus de développement du système d'information. :
 - conception bien gérée $\Rightarrow$ **développement facile.**
 - si des difficultés non révélées $\Rightarrow$ **la phase de développement sera obligée de reprendre des points non éclaircis lors de la phase de conception. De nombreux bugs verront le jour.**
- Méthodes de conception pour assister la démarche de conception.
3 étapes principales $\sim$ 3 niveaux d'abstraction différents :



Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

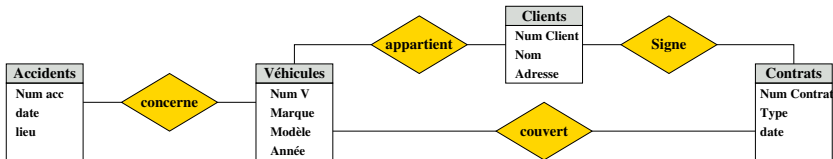
généralisation /
spécialisation

Spécification des
diagrammes EA

Contraintes
d'intégrité

NoSQL

Normalisation
d'une relation



- Le modèle **entités-associations** (EA), aussi appelé **entités-relations** (ER) est un modèle de données de type conceptuel.
- But : **exprimer une partie du monde réel indépendamment de toute contingence informatique.**
Utilisé dans des méthodes et outils d'aide à la conception (Merise, par ex.).
Se limite à une description statique : son but est d'aider à la conception des structures de données d'une application.
- Idée sous-jacente : représentation des **entités** du monde réel que l'observateur peut identifier de manière distincte.

On distingue alors

- les **entités**
(personne, voitures, contracts, accidents...),
- les propriétés observables de ces entités, appelées **attributs**
(taille, couleur, date de signature, lieu de l'accident...),
- et les **associations** qui relient les entités
(une personne est propriétaire d'une voiture).

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

Concepts de base et Diagramme EA

généralisation / spécialisation

Spécification des diagrammes EA

Contraintes d'intégrité

NoSQL

Normalisation d'une relation

- Les **entités** représentent des objets du monde réel, (concrets ou abstraits)
- Les **types d'entités** représentent les ensembles d'entités perçues comme similaires et ayant des caractéristiques comparables.

Clients	Véhicules	Contrats	Accidents
Num Client Nom Adresse	Num V Marque Modèle Année	Num Contrat Type date	Num acc date lieu

- **occurrence** d'un type d'entités / **population** d'un type d'entités.
- Une occurrence d'un type d'entités est vue par l'utilisateur comme un ensemble de valeurs : une valeur est associée à chacun des attributs
- Les **attributs** : représentations des propriétés associées à un type d'entités.
- **attribut composé** $\equiv$ attribut décomposé en attributs plus simples.
Exemple : date $\equiv$ jour + mois + année

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Les **associations** représentent les liens existants entre les entités. Les associations sont caractérisées, comme les entités, par un nom et éventuellement des attributs.

- **Types d'associations** : ensembles d'associations perçues comme similaires et ayant des caractéristiques qui peuvent être comparées.
- **Occurrences** d'un type d'associations : instances d'un type d'associations.
- **Population** d'un type d'associations : ensemble d'occurrences d'un type d'associations.

Cardinalité : la description complète d'une association nécessite la définition précise de la participation des entités. La cardinalité est le **nombre de participation d'une entité à une association.**

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

généralisation /
spécialisation

Spécification des
diagrammes EA

Contraintes
d'intégrité

NoSQL

Normalisation
d'une relation

- **Cardinalité un à un :**

- **Cardinalité un à plusieurs :**

- **Cardinalité plusieurs à plusieurs :**

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

Concepts de base et Diagramme EA

généralisation / spécialisation

Spécification des diagrammes EA

Contraintes d'intégrité

NoSQL

Normalisation d'une relation

- **Cardinalité un à un** : un employé ne peut être directeur que dans un seul département et un département n'a qu'un seul employé comme directeur.
- **Cardinalité un à plusieurs** :
- **Cardinalité plusieurs à plusieurs** :

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentiels

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

généralisation /
spécialisation

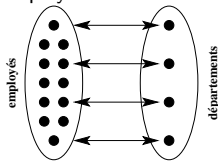
Spécification des
diagrammes EA

Contraintes
d'intégrité

NoSQL

Normalisation
d'une relation

- **Cardinalité un à un** : un employé ne peut être directeur que dans un seul département et un département n'a qu'un seul employé comme directeur.



- **Cardinalité un à plusieurs** :
- **Cardinalité plusieurs à plusieurs** :

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

généralisation /
spécialisation

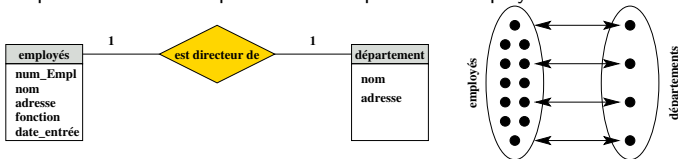
Spécification des
diagrammes EA

Contraintes
d'intégrité

NoSQL

Normalisation
d'une relation

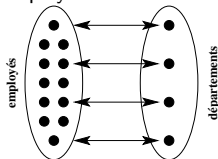
- **Cardinalité un à un** : un employé ne peut être directeur que dans un seul département et un département n'a qu'un seul employé comme directeur.



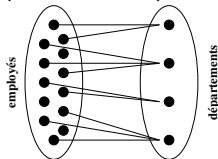
- **Cardinalité un à plusieurs** : un département emploie plusieurs personnes (employés) mais chaque employé ne fait partie que d'un seul département.

- **Cardinalité plusieurs à plusieurs** :

- **Cardinalité un à un** : un employé ne peut être directeur que dans un seul département et un département n'a qu'un seul employé comme directeur.



- **Cardinalité un à plusieurs** : un département emploie plusieurs personnes (employés) mais chaque employé ne fait partie que d'un seul département.



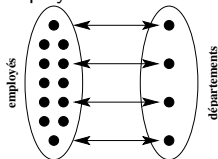
- Cardinalité plusieurs à plusieurs :

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

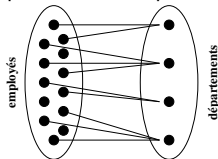
NoSQL

Normalisation
d'une relation

- **Cardinalité un à un** : un employé ne peut être directeur que dans un seul département et un département n'a qu'un seul employé comme directeur.

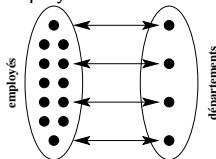


- **Cardinalité un à plusieurs** : un département emploie plusieurs personnes (employés) mais chaque employé ne fait partie que d'un seul département.

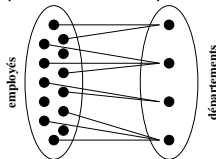


- **Cardinalité plusieurs à plusieurs** : un type de produit peut être fabriqué dans plusieurs usines et une usine donnée peut fabriquer plusieurs produits.

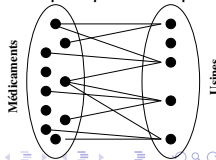
- **Cardinalité un à un** : un employé ne peut être directeur que dans un seul département et un département n'a qu'un seul employé comme directeur.



- **Cardinalité un à plusieurs** : un département emploie plusieurs personnes (employés) mais chaque employé ne fait partie que d'un seul département.



- **Cardinalité plusieurs à plusieurs** : un type de produit peut être fabriqué dans plusieurs usines et une usine donnée peut fabriquer plusieurs produits.



Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

Concepts de base et Diagramme EA

généralisation / spécialisation

Spécification des diagrammes EA

Contraintes d'intégrité

NoSQL

Normalisation d'une relation

Cardinalités maximales : nb max de participations d'une entité à une association.

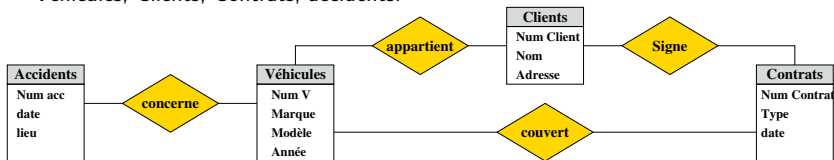
Cardinalités minimales : nb min de participations d'une entité à une association (souvent 0 ou 1).

Les cardinalités max et min traduisent les contraintes propres aux entités et associations.

Notations :

0-1	aucune ou une seule
1-1	une et une seule
0-N	aucune ou plusieurs
1-N	une ou plusieurs

Exercice : Faire le schéma Entités-Associations correspondant aux entités Véhicules, Clients, Contrats, accidents.



Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

Concepts de base et Diagramme EA

généralisation / spécialisation

Spécification des diagrammes EA

Contraintes d'intégrité

NoSQL

Normalisation d'une relation

Cardinalités maximales : nb max de participations d'une entité à une association.

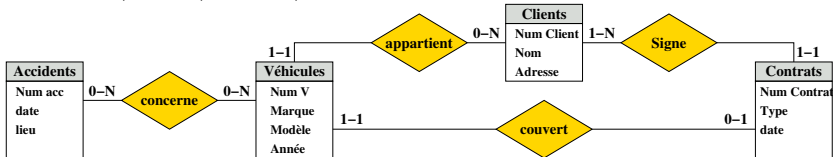
Cardinalités minimales : nb min de participations d'une entité à une association (souvent 0 ou 1).

Les cardinalités max et min traduisent les contraintes propres aux entités et associations.

Notations :

0-1	aucune ou une seule
1-1	une et une seule
0-N	aucune ou plusieurs
1-N	une ou plusieurs

Exercice : Faire le schéma Entités-Associations correspondant aux entités Véhicules, Clients, Contrats, accidents.



Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Notion de rôles.

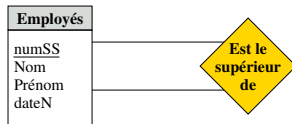
- Une association permet de lier plusieurs entités entre elles.
- Chaque entité va donc jouer un **rôle** vis à vis de la sémantique de l'association considérée.
- Si un type d'associations est relié à un même type d'entités par plusieurs liens différents, le rôle de chacun des types d'entités est primordial (indispensable), car c'est lui qui va distinguer les différents cas possibles.

Exemple : les liens hiérarchiques dans une entreprise
2 rôles :

- un rôle pour exprimer le chef.
- un rôle pour exprimer le ou les subordonnés,

Les cardinalités :

- 0-N pour le rôle « *chef* »,
- 0-1 pour le rôle « *subordonné* ».



Une **occurrence d'un type d'associations** est vue par l'utilisateur comme

- un ensemble de valeurs : une valeur pour chaque attribut du type d'associations (s'il en existe)
- un ensemble d'occurrences de types d'entités : pour chaque rôle associé au type d'associations, une occurrence du type d'entités qui joue ce rôle.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Notion de rôles.

- Une association permet de lier plusieurs entités entre elles.
- Chaque entité va donc jouer un **rôle** vis à vis de la sémantique de l'association considérée.
- Si un type d'associations est relié à un même type d'entités par plusieurs liens différents, le rôle de chacun des types d'entités est primordial (indispensable), car c'est lui qui va distinguer les différents cas possibles.

Exemple : les liens hiérarchiques dans une entreprise
2 rôles :

- un rôle pour exprimer le chef.
- un rôle pour exprimer le ou les subordonnés,

Les cardinalités :

- 0-N pour le rôle « *chef* »,
- 0-1 pour le rôle « *subordonné* ».



Une **occurrence d'un type d'associations** est vue par l'utilisateur comme

- un ensemble de valeurs : une valeur pour chaque attribut du type d'associations (s'il en existe)
- un ensemble d'occurrences de types d'entités : pour chaque rôle associé au type d'associations, une occurrence du type d'entités qui joue ce rôle.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Notion de rôles.

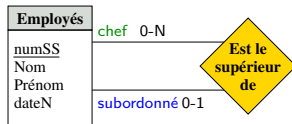
- Une association permet de lier plusieurs entités entre elles.
- Chaque entité va donc jouer un **rôle** vis à vis de la sémantique de l'association considérée.
- Si un type d'associations est relié à un même type d'entités par plusieurs liens différents, le rôle de chacun des types d'entités est primordial (indispensable), car c'est lui qui va distinguer les différents cas possibles.

Exemple : les liens hiérarchiques dans une entreprise
2 rôles :

- un rôle pour exprimer le chef.
- un rôle pour exprimer le ou les subordonnés,

Les cardinalités :

- 0-N pour le rôle « *chef* »,
- 0-1 pour le rôle « *subordonné* ».



Une **occurrence d'un type d'associations** est vue par l'utilisateur comme

- un ensemble de valeurs : une valeur pour chaque attribut du type d'associations (s'il en existe)
- un ensemble d'occurrences de types d'entités : pour chaque rôle associé au type d'associations, une occurrence du type d'entités qui joue ce rôle.

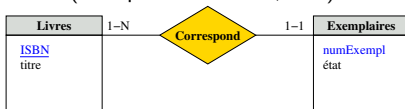
Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

- Un **identifiant** est un attribut ou un *ensemble d'attributs* permettant de déterminer une et une seule entité à l'intérieur de la population.
 - Un **identifiant** est un ensemble minimum d'attributs tel qu'il n'existe pas deux occurrences du TE qui ont la même valeur pour ces attributs.
- ⇒ s'il y a un identifiant composé d'au moins 2 attributs, bien le distinguer de deux identifiants simples.

Exemple. Dans une bibliothèque, on distingue le livre (œuvre protégée) de l'exemplaire de cet œuvre (exemplaire numéro 1, 2 ...).



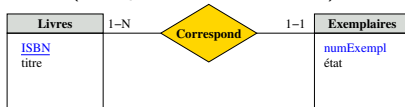
Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

- Un **identifiant** est un attribut ou un *ensemble d'attributs* permettant de déterminer une et une seule entité à l'intérieur de la population.
 - Un **identifiant** est un ensemble minimum d'attributs tel qu'il n'existe pas deux occurrences du TE qui ont la même valeur pour ces attributs.
- ⇒ s'il y a un identifiant composé d'au moins 2 attributs, bien le distinguer de deux identifiants simples.

Exemple. Dans une bibliothèque, on distingue le livre (œuvre protégée) de l'exemplaire de cet œuvre (exemplaire numéro 1, 2 ...).



- on dit que le TE *Exemplaires* est **faible**.
- Un TE est dit **faible** lorsque :
 - aucun sous-ensemble de ses attributs constitue un identifiant, et
 - un identifiant peut être défini en intégrant un identifiant d'un autre TE' qui lui est lié par un TA binaire de cardinalité (1,1) pour TE.
 On dit dans ce cas que TE **dépend** de TE'.
- le TE *Exemplaires* dépend du TE *Livres*, car tout exemplaire est rattaché à un livre.
L'identifiant est (**Livres.ISBN**, numExempl).

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

Concepts de base et Diagramme EA

généralisation / spécialisation

Spécification des diagrammes EA

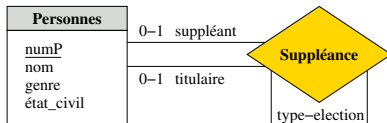
Contraintes d'intégrité

NoSQL

Normalisation d'une relation

La définition des identifiants est la même que pour les types d'entités.

Exemples.



- Une occurrence du TA Suppléance est un triplet.
`numP` est un identifiant de `Personnes`, la cardinalité des deux rôles est 0-1

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

Concepts de base et Diagramme EA

généralisation / spécialisation

Spécification des diagrammes EA

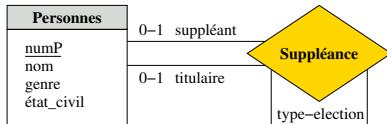
Contraintes d'intégrité

NoSQL

Normalisation d'une relation

La définition des identifiants est la même que pour les types d'entités.

Exemples.



- Une occurrence du TA Suppléance est un triplet.
numP est un identifiant de Personnes, la cardinalité des deux rôles est 0-1
⇒ Suppléance possède 2 identifiants :
(suppléant.numP, type_election) et
(titulaire.numP, type_election).

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

généralisation /
spécialisation

Spécification des
diagrammes EA

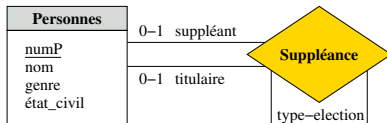
Contraintes
d'intégrité

NoSQL

Normalisation
d'une relation

La définition des identifiants est la même que pour les types d'entités.

Exemples.



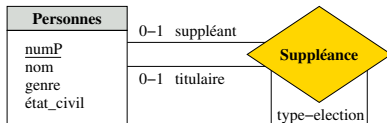
- Une occurrence du TA Suppléance est un triplet.
`numP` est un identifiant de `Personnes`, la cardinalité des deux rôles est 0-1
⇒ `Suppléance` possède 2 identifiants :
(`suppléant.numP`, `type_election`) et
(`titulaire.numP`, `type_election`).
- Ce raisonnement est bon uniquement si la base de données ne contient que les suppléances en cours. Si on mémorisait toutes les suppléances, les cardinalités des rôles `suppléant` et `titulaire` seraient 0-N, et les identifiants de `Suppléance` seraient :

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

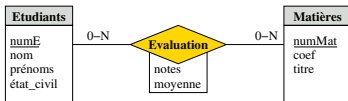
Normalisation
d'une relation

La définition des identifiants est la même que pour les types d'entités.

Exemples.

- Une occurrence du TA Suppléance est un triplet.
numP est un identifiant de Personnes, la cardinalité des deux rôles est 0-1
⇒ Suppléance possède 2 identifiants :
(suppléant.numP, type_election) et
(titulaire.numP, type_election).
- Ce raisonnement est bon uniquement si la base de données ne contient que les suppléances en cours. Si on mémorisait toutes les suppléances, les cardinalités des rôles suppléant et titulaire seraient 0-N, et les identifiants de Suppléance seraient :
(suppléant.numP, type_election, date) et
(titulaire.numP, type_election, date).

Exercice : Donner les identifiants des types d'associations suivants.



Correction :

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

généralisation /
spécialisation

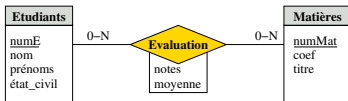
Spécification des
diagrammes EA

Contraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Exercice : Donner les identifiants des types d'associations suivants.



Correction :

Dans le 1er cas, l'identifiant est (**Etudiant.numE, Matière.numMat**).

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

généralisation /
spécialisation

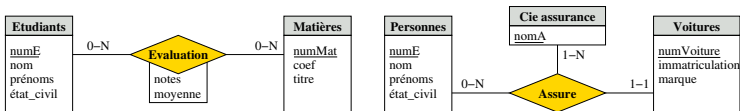
Spécification des
diagrammes EA

Contraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Exercice : Donner les identifiants des types d'associations suivants.



Correction :

Dans le 1er cas, l'identifiant est `(Etudiant.numE,Matière.numMat)`.

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

Concepts de base et Diagramme EA

généralisation / spécialisation

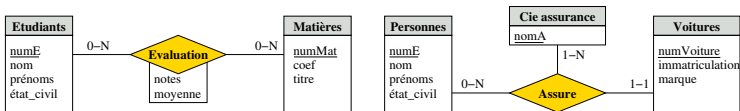
Spécification des diagrammes EA

Contraintes d'intégrité

NoSQL

Normalisation d'une relation

Exercice : Donner les identifiants des types d'associations suivants.



Correction :

Dans le 1er cas, l'identifiant est **(Etudiant.numE,Matière.numMat)**.

Dans le 2ème cas, l'identifiant est **Voitures.numVoiture**.

Règle 1. Lorsqu'un type associations est lié à un TE par un lien de cardinalité maximum égale à 1, alors tout identifiant de ce TE fait partie de l'identifiant du TA considéré.

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

Concepts de base et Diagramme EA

généralisation / spécialisation

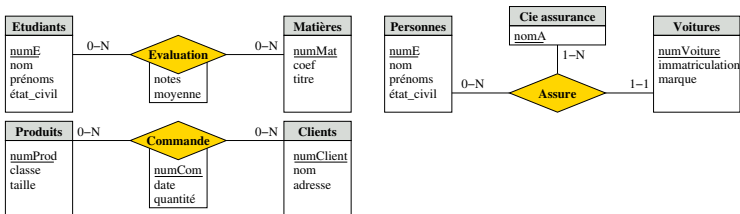
Spécification des diagrammes EA

Contraintes d'intégrité

NoSQL

Normalisation d'une relation

Exercice : Donner les identifiants des types d'associations suivants.



Correction :

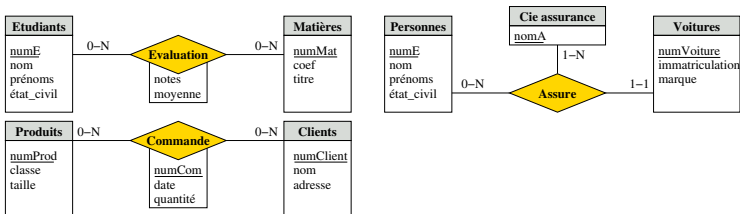
Dans le 1er cas, l'identifiant est **(Etudiant.numE,Matière.numMat)**.

Dans le 2ème cas, l'identifiant est **Voitures.numVoiture**.

Règle 1. Lorsqu'un type associations est lié à un TE par un lien de cardinalité maximum égale à 1, alors tout identifiant de ce TE fait partie de l'identifiant du TA considéré.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation**Exercice :** Donner les identifiants des types d'associations suivants.**Correction :**

Dans le 1er cas, l'identifiant est (Etudiant.numE,Matière.numMat).

Dans le 2ème cas, l'identifiant est Voitures.numVoiture.

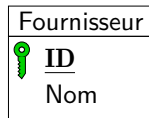
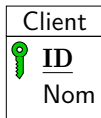
Dans le 3ème cas, l'identifiant est
(Produit.numProd,numCom,Client.numClient).

Règle 1. Lorsqu'un type associations est lié à un TE par un lien de cardinalité maximum égale à 1, alors tout identifiant de ce TE fait partie de l'identifiant du TA considéré.

Règle 2. Lorsque plusieurs occurrences d'un TA peuvent lier les mêmes occurrences des TE, alors l'identifiant du TA contient au moins un attribut du TA.

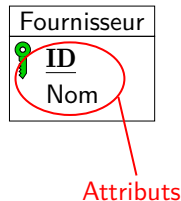
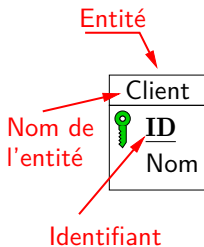
Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

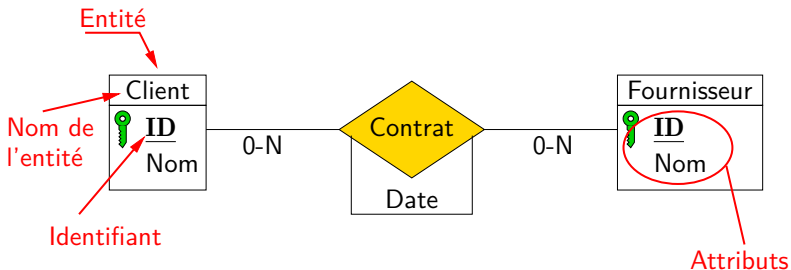
Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

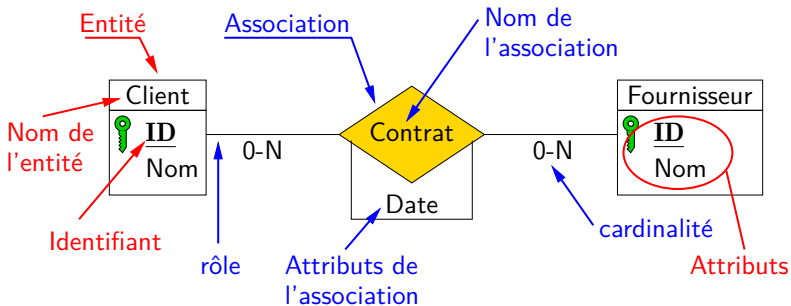
généralisation /
spécialisation

Spécification des
diagrammes EA

Contraintes
d'intégrité

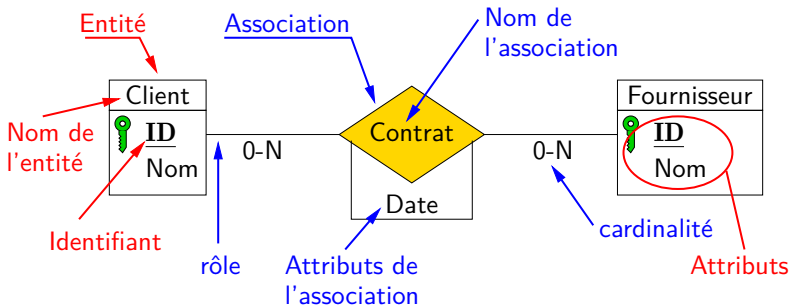
NoSQL

Normalisation
d'une relation



Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Association : Contrat
Dimension (arité) = 2

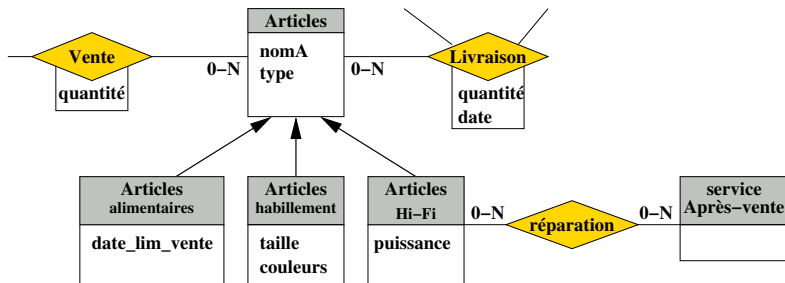
Les entités Client et Fournisseur
sont les participants de
l'association Contrat

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

- Un type d'entités représente une classe d'objets du monde réel.
 - Ce même ensemble peut être perçu d'un autre point de vue comme plusieurs classes. **Exemple** : articles d'un supermarché
 - pour certains, tous les articles ont le même statut,
 - pour d'autres, articles d'alimentation, articles d'habillement... (traitements différents)
 - Comment définir un TE générique *Article*, et des TE spécialisés représentant des sous-classes spécialisées ?
Les classes d'objets *ne sont donc plus distinctes*.
- ❶ **Généralisation** : TE spécialisé (*spécifique*) $\longrightarrow$ TE générique
Spécialisation : TE générique $\longrightarrow$ TE spécialisé (*spécifique*)
« à toute entité d'un TE spécialisé correspond une et une seule entité du TE générique »
« à toute entité du TE générique correspond au plus une entité d'un TE spécialisé »
Lien « est un » (ou « IS A »)
le TE spécialisé est appelé **sous-type** du TE générique
le TE générique est appelé **sur-type** du TE spécialisé.
Lorsqu'il y a plusieurs niveaux, **hiérarchie de généralisations**.
- ❷ les attributs qui sont communs à tous les sous-types, sont rattachés au type d'entités générique : les sous-types **héritent** de ces attributs.



- Il n'est pas nécessaire que l'ensemble des sous-types constituent une partition du sur-type.
- Les TE spécifiques peuvent représenter des populations qui ne sont pas disjointes.
- **généralisation multiple** lorsqu'un TE spécialisé est sous-type de plusieurs autres TE. (Cf exemple suivant)
- **Identifiant d'un TE sous-type**. Soit E un TE sous-type d'un TE E'. Tout identifiant de E' est aussi identifiant de E. E n'a pas nécessairement d'identifiant qui lui soit propre.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Exercice : Décrire par un diagramme EA l'ensemble des personnes travaillant dans une université. Chaque individu est une personne, la population des personnes est partagée en étudiants et employés. La population des personnels est découpée en trois : les enseignants (MCF, Pr, et autres), les techniciens et les administratifs. Parmi les étudiants, on particularise les doctorants, parmi lesquels certains sont moniteurs, c'est-à-dire héritent de la population des enseignants.

Les attributs associés aux personnes sont : n^o d'identifiant, nom, prénom. L'étudiant a comme attribut le dépt, le doctorant a un sujet de thèse, et l'employé a pour attribut sa classe (A,B,C).

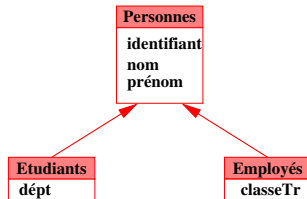
Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Exercice : Décrire par un diagramme EA l'ensemble des personnes travaillant dans une université. **Chaque individu est une personne, la population des personnes est partagée en étudiants et employés.** La population des personnels est découpée en trois : les enseignants (MCF, Pr, et autres), les techniciens et les administratifs. Parmi les étudiants, on particularise les doctorants, parmi lesquels certains sont moniteurs, c'est-à-dire héritent de la population des enseignants.

Les attributs associés aux personnes sont : n^o d'identifiant, nom, prénom. L'étudiant a comme attribut le dépt, le doctorant a un sujet de thèse, et l'employé a pour attribut sa classe (A,B,C).



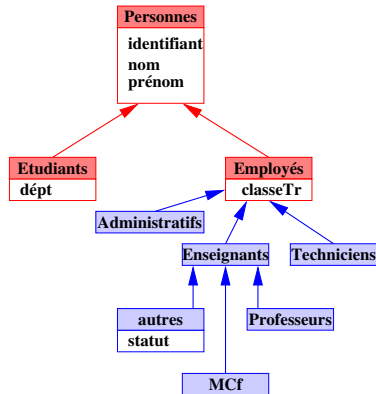
Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Exercice : Décrire par un diagramme EA l'ensemble des personnes travaillant dans une université. **Chaque individu est une personne, la population des personnes est partagée en étudiants et employés. La population des personnels est découpée en trois : les enseignants (MCF, Pr, et autres), les techniciens et les administratifs.** Parmi les étudiants, on particularise les doctorants, parmi lesquels certains sont moniteurs, c'est-à-dire héritent de la population des enseignants.

Les attributs associés aux personnes sont : n^o d'identifiant, nom, prénom. L'étudiant a comme attribut le dépt, le doctorant a un sujet de thèse, et l'employé a pour attribut sa classe (A,B,C).



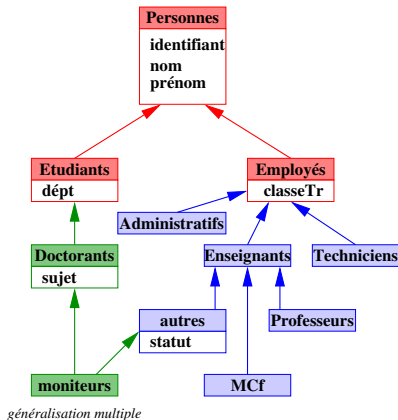
Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Exercice : Décrire par un diagramme EA l'ensemble des personnes travaillant dans une université. **Chaque individu est une personne, la population des personnes est partagée en étudiants et employés. La population des personnels est découpée en trois : les enseignants (MCF, Pr, et autres), les techniciens et les administratifs. Parmi les étudiants, on particularise les doctorants, parmi lesquels certains sont moniteurs, c'est-à-dire héritent de la population des enseignants.**

Les attributs associés aux personnes sont : n^o d'identifiant, nom, prénom. L'étudiant a comme attribut le dépt, le doctorant a un sujet de thèse, et l'employé a pour attribut sa classe (A,B,C).



généralisation multiple

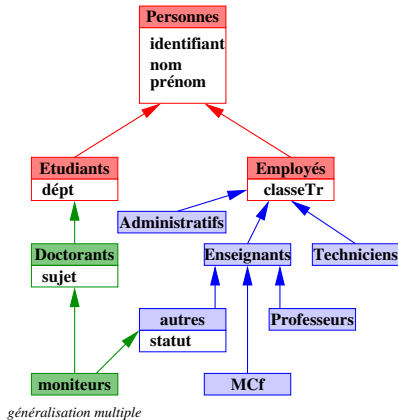
Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Exercice : Décrire par un diagramme EA l'ensemble des personnes travaillant dans une université. **Chaque individu est une personne, la population des personnes est partagée en étudiants et employés. La population des personnels est découpée en trois : les enseignants (MCF, Pr, et autres), les techniciens et les administratifs. Parmi les étudiants, on particularise les doctorants, parmi lesquels certains sont moniteurs, c'est-à-dire héritent de la population des enseignants.**

Les attributs associés aux personnes sont : n^o d'identifiant, nom, prénom. L'étudiant a comme attribut le dépt, le doctorant a un sujet de thèse, et l'employé a pour attribut sa classe (A,B,C).



Problèmes liés à l'héritage

- les sous-types peuvent hériter deux fois d'un ancêtre commun
- conflit d'héritage lorsqu'un sous-type hérite de 2 TE ayant un nom d'attribut en commun.
- Ces problèmes doivent être résolus par le concepteur : préférence d'héritage par exemple.

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

Concepts de base et
Diagramme EA

généralisation /
spécialisation

Spécification des
diagrammes EA

Contraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Les Types d'Entités (TE)

- nom du type,
- nom des sur-types s'il en existe,
- la définition libre (commentaire) précisant la sémantique du TE,
- la description des attributs,
- les identifiants.

Deux TE différents ne peuvent pas avoir le même nom.

Les Types d'Associations (TA)

- nom du type,
- la définition libre (commentaire) précisant la sémantique du TA,
- le nom des TE avec le nom du rôle les associant au TA (en pratique, ce n'est nécessaire que pour les associations cycliques),
- pour chacun des rôles, ses cardinalités ; si la cardinalité max est supérieure à 1, on précise si les rôles liant une entité constitue une liste ou un ensemble (valeur par défaut),
- la description des attributs,
- la composition des identifiants, s'il en existe.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Les **attributs** : plusieurs types :

- **attributs simples** (attributs non composés),
- **attributs complexes** (attributs composés),
- **attributs monovalués** (cardinalité $\max=1$),
- **attributs multivalués** (cardinalité $\max>1$),
- **attributs obligatoires** (cardinalité $\min=1$),
- **attributs facultatifs** (cardinalité $\min=0$).

Un attribut est décrit par les spécifications suivantes :

- nom de l'attribut,
- définition libre,
- cardinalité (min :max)
Si la cardinalité est supérieure à 1, on précise si les valeurs de l'attribut constituent une liste ou un ensemble (valeur par défaut)
- si l'attribut n'est pas composé d'autres attributs : domaine de valeurs définissant les valeurs possibles,
- si l'attribut est composé d'autres attributs : description des attributs composants.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Exemple : On considère un type d'entités *employés* qui doit mémoriser en plus des identifiants, noms et prénoms de la personne, les formations validées, ainsi que l'ensemble des postes occupés par l'employé en question.

employés	cardinalité	type d'attribut
identifiantE	1:1	simple, monovalué, obligatoire
Nom	1:1	simple, monovalué, obligatoire
prénoms	1:N	simple, multivalué, obligatoire
formationValidée	0:N	complexe, multivalué, facultatif
diplome	1:1	
année	1:1	
poste	1:N	complexe, multivalué, obligatoire
intitule	1:1	
date-debut	1:1	
date_fin	1:1	
salaire	1:N	
montant	1:1	
date	1:1	
année	1:1	
mois	1:1	

Considérons le type d'associations Mariages et le type d'entités Personnes.

- si une personne participe à une association du type Mariages son état civil doit être marié.

$$\forall x, y \in \text{Personnes}, (x, y) \in \text{Mariages} \implies \begin{cases} x.\text{état} = \text{"Marié"} \wedge \\ y.\text{état} = \text{"Marié"} \end{cases}$$

- si l'état civil d'une personne est "Marié", il ne peut être changé par la suite en "Célibataire"

$$\forall x \in \text{Personnes}, \forall t_1, t_2 \in \text{Temps}, \\ (t_1 < t_2) \wedge x(t_1).\text{état} = \text{"Marié"} \implies x(t_2).\text{état} \neq \text{"Célibataire"}$$

Autre exemple : tournois de doubles mixtes.

- seuls les hommes (resp. femmes) participent à l'association Equipe dans le rôle d'homme (resp. de femme) :

$$\forall x, y \in \text{Personnes}, (\text{homme} : x, y) \in \text{Equipe} \implies x.\text{genre} = \text{"M"}$$

$$\forall x, y \in \text{Personnes}, (\text{femme} : x, y) \in \text{Equipe} \implies x.\text{genre} = \text{"F"}$$

Ces règles sont décrites à l'aide de **règles d'intégrité**. Si ces règles ne sont pas satisfaites par la BD, la base est dite **incohérente**.

L'implémentation se fait par des prédicats de contraintes (**CHECK**), par des procédures déclenchées automatiquement (voir les **TRIGGERS**) ou par des procédures associées aux schémas (voir les stored procedure).

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

Sur les attributs : Les contraintes les plus fréquentes cherchent à réduire les valeurs possibles d'un attribut à certaines valeurs du domaine sous-jacent.

Exemple :

- un mois dans une date sera un entier compris entre 1 et 12 :
 $\text{mois} \in [1 : 12]$.

Ces contraintes peuvent aussi dépendre d'un contexte :

- si $\text{mois} \in \{4, 6, 9, 11\}$ alors $\text{jour} \in [1 : 30]$, sinon si $\text{mois} = 2$ alors $\text{jour} \in [1 : 29]$, sinon $\text{jour} \in [1 : 31]$.
- si une personne participe à l'association Mariages, alors son état civil est marié.

Sur les cardinalités : Prenons un exemple :

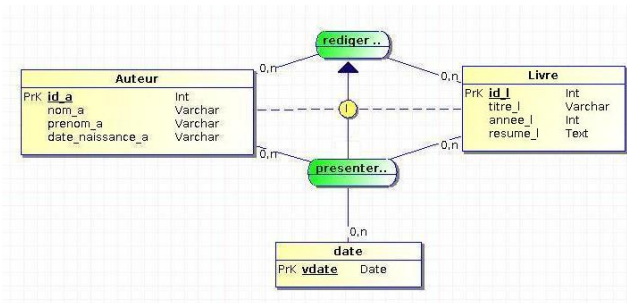
- le TA "est_parent_de"
- le TE Parents comprend l'attribut "nb_enfants".

⇒ Contrainte d'intégrité :

« le "nb_enfants" d'un parent est égal au nombre d'occurrences dans le TA "est_parent_de" qui lie ce parent. »

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

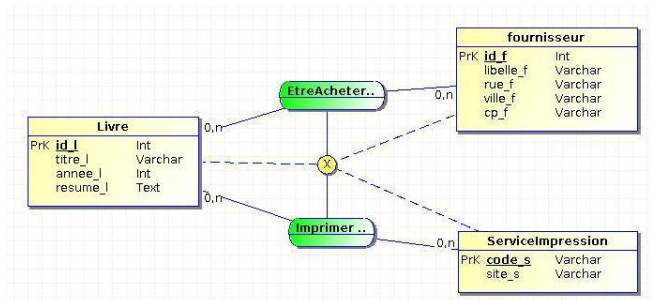
NoSQL

Normalisation
d'une relation

"Inclusion" : si un auteur présente son livre, c'est qu'il l'a écrit !

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

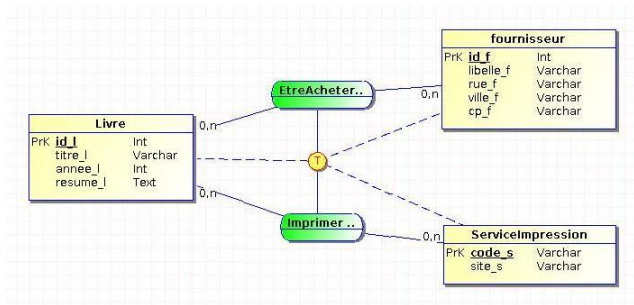
NoSQL

Normalisation
d'une relation

"ou exclusif" : si un livre est présent, soit il a été acheté à un fournisseur, soit il a été imprimé dans un des services d'impression du groupe.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

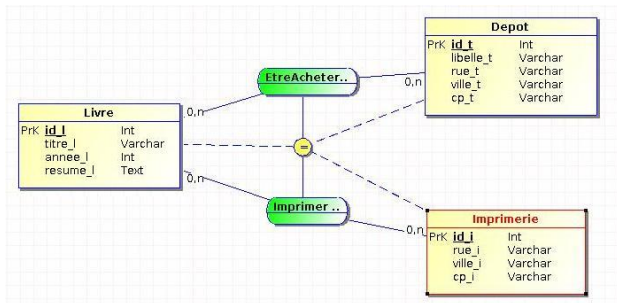
NoSQL

Normalisation
d'une relation

"Totalité" : il n'y a pas d'autre choix...

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

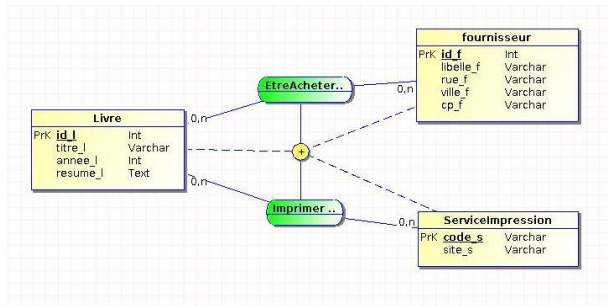
NoSQL

Normalisation
d'une relation

"Egalité" : Un livre (vu par une librairie) acheté dans un dépôt est également imprimé dans une imprimerie et vice et versa.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

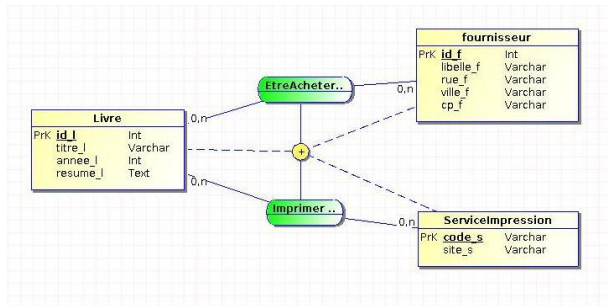
NoSQL

Normalisation
d'une relation

"Totalité + ou exclusif" noté "TX" ou "+" : le livre sera soit imprimé dans un service interne, soit acheté.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
AssociationsConcepts de base et
Diagramme EAgénéralisation /
spécialisationSpécification des
diagrammes EAContraintes
d'intégrité

NoSQL

Normalisation
d'une relation

"Totalité + ou exclusif" noté "TX" ou "+" : le livre sera soit imprimé dans un service interne, soit acheté.

Pour conclure sur le modèle Entités-Associations : un schéma conceptuel Entités-Associations est un ensemble de descriptions de types d'entités et de types d'associations (attributs, liens de généralisations...) et de contraintes d'intégrités.

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

- 1 Les concepts généraux
- 2 Notions essentielles pour le modèle relationnel
- 3 Algèbre Relationnelle et SQL
- 4 Le modèle conceptuel Entités-Associations
- 5 NoSQL
 - NoSQL
 - La famille
 - Théorème CAP
 - NoSQL vs. Jointures
- 6 Normalisation d'une relation

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

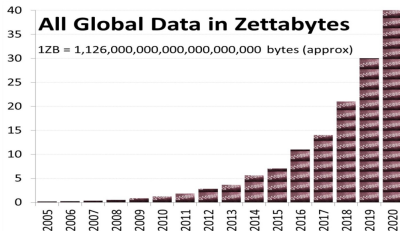
NoSQL vs. Jointures

Normalisation
d'une relation

Application et plateforme web

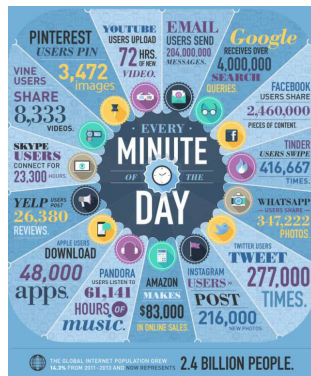
Croissance exponentielle :

×2 tous les 2 ans



Un volume à gérer sans précédent :

- Besoin de distribuer les données sur plusieurs sites :
 - Données
 - Calcul
- Clusters : grand nombre de serveurs
 Ex : Google, Amazon, Facebook
 Data center Google : 70000 serveurs/data center, 1M de serv.
 Facebook : 1 pétaoctet de données (10^{15} octets)



Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentiels

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

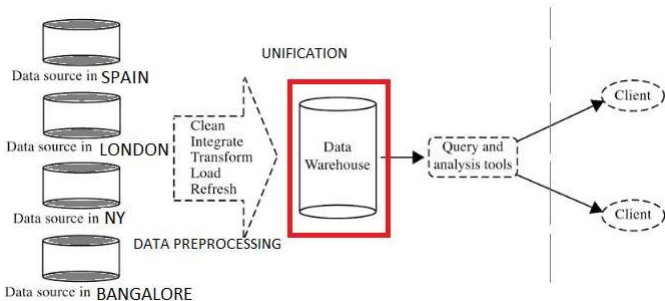
NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation



- **Le volume :**
 - Conçu pour stocker des Go/To
 - Mais a besoin de Po (péta-octets, 10^{15}), voire de Eo (exa-octets, 10^{18})
- **Variété**
Données hétérogènes
Texte, semi-structuré
- **Vitesse :**
Production rapide de données $\Rightarrow$ Big Data

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

- Données hétérogènes, complexes et souvent liées
 - produites par des applications différentes
 - par des utilisateurs différents
 - avec des liens explicites (p.e. citations, url...) ou implicites (à extraire/apprendre)
- Nombreux serveurs/clusters
 - un serveur unique ne peut stocker cette quantité d'information, garantir des temps d'accès pour un grand nombre d'utilisateur, faire les calculs rapides...
- Besoin de distribuer les calculs et les données
 - besoins d'algos permettant le calcul et la distrib. de données à large échelle

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

- Fonctionnalités
 - Jointures entre les tables
 - Langage d'interrogation riche
 - contraintes d'intégrité solide
- Limite dans le contexte distribué
 - Comment distribuer / partitionner les données
 - liens entre entités → même serveur
 - mais plus on a de liens, plus le placement des données est complexes

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
générauxNotions
essentielsAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

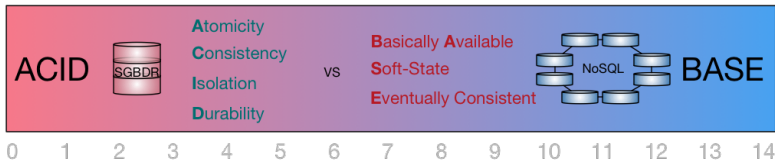
NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

- **Atomicité** : Une transaction s'effectue entièrement ou pas du tout
- **Cohérence** : Le contenu d'une base doit être cohérent au début et à la fin d'une transaction
- **Isolation** : Les modifications d'une transaction ne sont visibles/modifiables que quand celle-ci a été validée
- **Durabilité** : Une fois la transaction validée, l'état de la base est permanent (non affecté par les pannes ou autre)
- **Basically Available** : le système garantit un taux de disponibilité de la donnée
- **Soft-state** : La base peut changer lors des mises à jour ou lors d'ajout/suppression de serveurs. La base NoSQL n'a pas à être cohérente à tout instant
- **Eventually consistent** : À terme, la base atteindra un état cohérent

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

NoSQL : Not Only SQL

- Nouvelle approche de stockage/gestion des données
- Passage à l'échelle du système (via la distribution)
- Gestion des métadonnées complexes (sans schéma)
⇒ pas de langage riche pour l'interrogation



Ne remplacez pas le SGBD !!

Cible NoSQL :

- Très gros volume de données (PetaBytes)
- Temps de réponse très court
- La cohérence n'est pas obligatoire

Attention au coûts : serveurs, bandes bassantes, disques...

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentielsAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

- Pas de relation
 - pas de schéma physiques ou dynamiques
 - notions de collections
- Données éventuellement complexes
 - imbrication, tableaux (dénormalisations, on oublie la 1ère FN par exemple)
- Distribution de données (milliers de serveurs)
 - parallélisation des traitements (Map/Reduce)
- Réplication des données (pannes)
 - disponibilité vs. cohérence (pas de transactions)
 - peu d'écriture, beaucoup de lectures

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

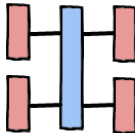
Base de données SQL

WayToLearnX

Relationnelle

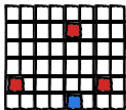


OLAP

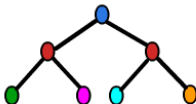


Base de données NoSQL

Orienté colonne



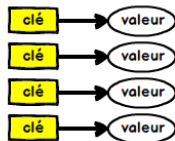
Document



Graphe



Clé-Valeur



Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

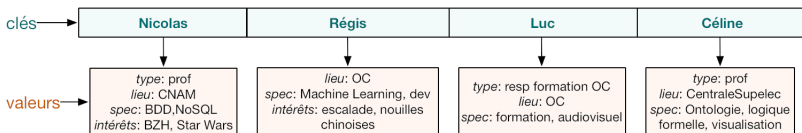
Normalisation
d'une relation

Différents systèmes NoSQL

- Clé-valeur (Key-Value Store)
 - Données identifiées par clé unique (utilisées pour l'interrogation)
 - DynamoDB, Voldemort, Redis, Riak, MemcacheDB
- Colonnes (Column data)
 - Relation 1-n “one-to-many” (messages, posts)
 - HBase, Cassandra, Hypertable
- Documents
 - Données complexes, attributs/valeurs
 - MongoDB, CouchDB, Terrastore, Cassandra
- Graphes
 - Réseaux sociaux...
 - Neo4j, OrientDB, FlockDB

1- NoSQL & Clé-Valeurs

- Similaires à un “HashMap” distribué
- Couple Clé+Valeur
 - Pas de schéma pour la valeur (chaîne, objet, entier, binaires...) qui peut donc être différente pour chaque
- Conséquences :
 - Pas de structure ni de types
 - Pas d'expressivité d'interrogation (pré/post traitement pour manipuler concrètement les données)
- DynamoDB (Amazon), Redis (VMWare), Voldemort (LinkedIn)



Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

1- NoSQL & Clé-Valeurs

- Opérations CRUD (HTTP)
 - Create(key,value)
 - Read(key)
 - Update(key,value)
 - Delete(key)
- Passage à l'échelle horizontal
partitionnement/distribution des tuples
Modèle très simple
Modification facile du contenu associé à une clé
- Interrogation difficile car accès par clé
⇒ la complexité reportée au pré/post traitement

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

- Stockage des données par colonnes
 - SGBD : tuples (lignes)
- Facile d'ajouter une colonne (pas une ligne !)
 - Schéma peut être dynamique (d'un tuple à l'autre)
- BigTable/Hbase (Google), Cassandra (Facebook&Apache), SimpleDB (Amazon)

Stockage orienté lignes				
id	type	lieu	spec	intérêts
Nicolas	prof	CNAM	BDD, NoSQL	BZH, Star Wars
Régis		OC	Machine Learning, Dev	escalade, nouilles chinoises
Luc	resp formation OC	OC	formation, audiovisuel	
Céline	prof	CentraleSupélec	Ontologie, logique formelle, visualisation	

Stockage orienté colonnes							
id	type	id	lieu	id	spec	id	intérêts
Nicolas	prof	Céline	CentraleSupélec	Nicolas	BDD	Nicolas	BZH
Céline	prof	Nicolas	CNAM	Nicolas	NoSQL	Nicolas	Star Wars
Luc	resp formation OC	Régis	OC	Régis	Machine Learning	Régis	escalade
		Luc	OC	Régis	Dev	Régis	nouilles chinoises
				Luc	formation		
				Luc	audiovisuel		
				Céline	Ontologie		
				Céline	logique formelle		
				Céline	visualisation		

Avantages :

- Supporte le semi-structuré
- Indexation de chaque colonne
- Passage à l'échelle horiz.
- Compression possible

Inconvénients :

- Lecture de données complexes difficile
- Difficulté de relier les données
- Requêtes pré-définies (pas à la volée)

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

NoSQL

NoSQL

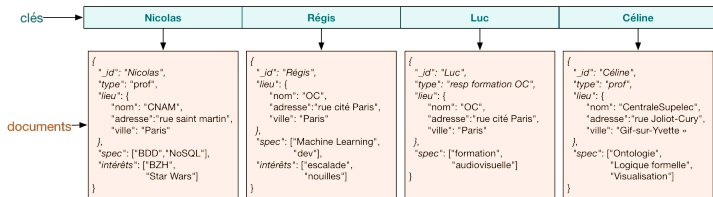
La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation d'une relation

- Basé sur le modèle clé-valeur
 - Ajout de données semi-structurées (JSON ou XML)
 - chaque valeur est un document
- Requêtes : Interface HTTP
 - Plus complexe que CRUD
- MongoDB, CouchDB (Apache), RavenDB, Terrastore



Un document est composé de champs et de valeurs associées

- Types simples (Int, String, Date)
- Schéma non nécessaire (peut différer d'un document à l'autre)
- Imbrication de données (schéma arborescent)
- Chaque donnée du document peut être interrogée

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

NoSQL

NoSQL

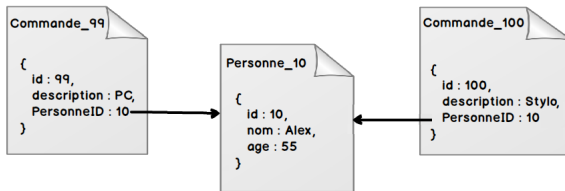
La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation d'une relation

- Basé sur le modèle clé-valeur
 - Ajout de données semi-structurées (JSON ou XML)
 - chaque valeur est un document
- Requêtes : Interface HTTP
 - Plus complexe que CRUD
- MongoDB, CouchDB (Apache), RavenDB, Terrastore



Un document est composé de champs et de valeurs associées

- Types simples (Int, String, Date)
- Schéma non nécessaire (peut différer d'un document à l'autre)
- Imbrication de données (schéma arborescent)
- Chaque donnée du document peut être interrogée

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

- Basé sur le modèle clé-valeur
 - Ajout de données semi-structurées (JSON ou XML)
 - chaque valeur est un document
- Requêtes : Interface HTTP
 - Plus complexe que CRUD
- MongoDB, CouchDB (Apache), RavenDB, Terrastore

Avantages :

- Modèle simple mais bonne puissance d'expression (structure imbriquée)
- Interrogation de tout attribut (+indexation)
- Passe facilement à l'échelle (surtout si sharding supporté)

Inconvénients :

- Inter-connexion de données complexe
- Interrogation sur la clé (+index)

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

NoSQL

NoSQL

La famille

Théorème CAP

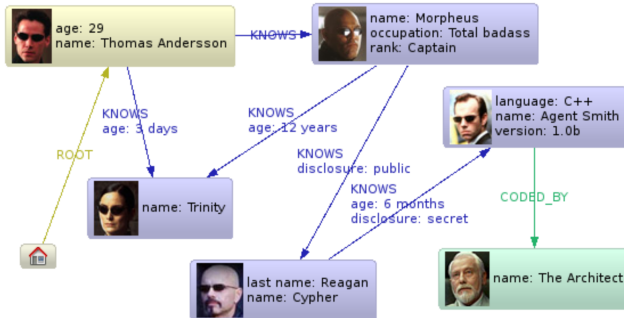
NoSQL vs. Jointures

Normalisation d'une relation

Stockage des noeuds, relations et propriétés

- Théorie des graphes
- Interrogation par traversées de graphe
- Appel des données sur demande (parcours performants)
- Modélisation non triviale

Neo4j, OrientDB (Apache), FlockDB (Twitter)



Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation d'une relation

Stockage des noeuds, relations et propriétés

- Théorie des graphes
- Interrogation par traversées de graphe
- Appel des données sur demande (parcours performants)
- Modélisation non triviale

Neo4j, OrientDB (Apache), FlockDB (Twitter)

Avantages

- Modèle adapté pour le stockage de grands graphes
- Offre des fonctionnalités de calculs sur grands graphes
- les objets (cf. documents)
- les arcs (avec propriétés)

Inconvénients :

- Sharding/partitionnement difficile

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités-Associations

NoSQL

NoSQL

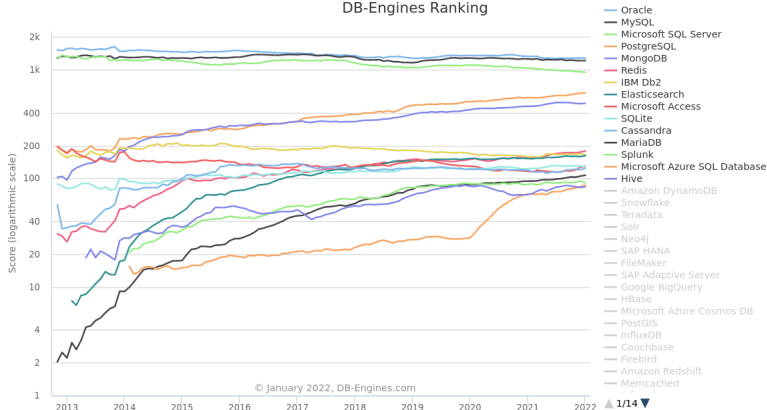
La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation d'une relation

DB-Engines Ranking



https://db-engines.com/en/ranking_trend

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation

Théorème de Brewer (2000) Dit : “Théorème de CAP” :

3 propriétés fondamentales pour les systèmes distribués :

- 1 **Consistency** : Tous les serveurs voient la même données (valeur) en même temps (ou Cohérence)
- 2 **Availability** : Si un serveur tombe en panne, les données restent disponibles
- 3 **Partition Tolerance** : Le système même partitionné doit répondre correctement à toute requête (sauf en cas de panne réseau)

Théorème :

Dans un système distribué, il est impossible que ces 3 propriétés co-existent, vous devez choisir 2 d'entre elles.

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

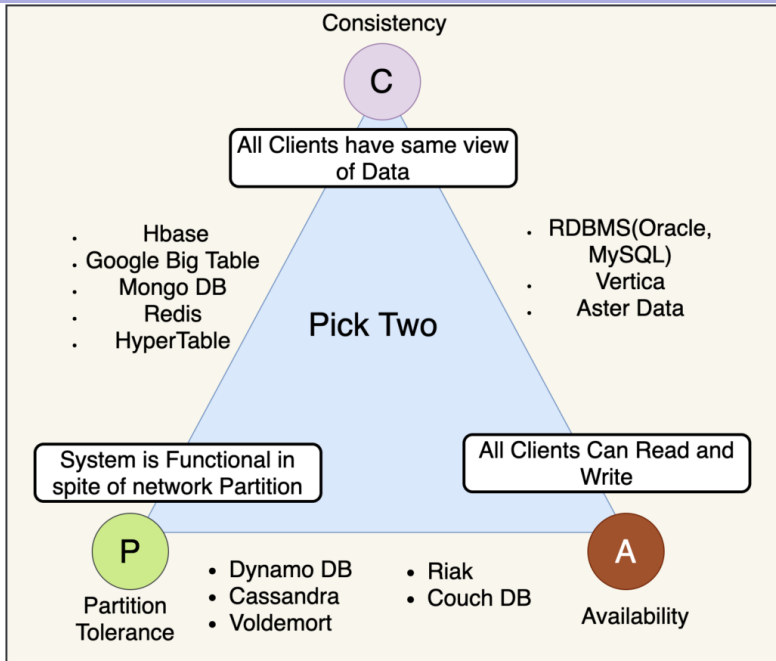
NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation
d'une relation



Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

le modèle Entités- Associations

NoSQL

NoSQL

La famille

Théorème CAP

NoSQL vs. Jointures

Normalisation d'une relation

Contexte distribué incompatible avec les liens entre les collections

- Besoin d'un modèle de coût
 - Coût basé sur les communications réseaux - MapReduce
 - Jointure multi-serveurs

Si l'on considère deux collections de données

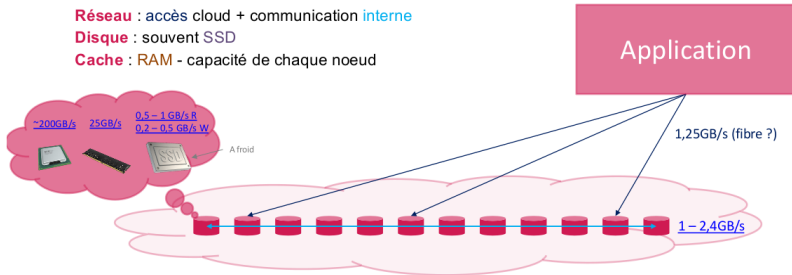
Pour une jointure entre les deux collections

- Sur quel serveur se fait la jointure ?
- Distribution des données à joindre ?
- Problème de coût réseau (plus cher qu'en local)

Réseau : accès cloud + communication interne

Disque : souvent SSD

Cache : RAM - capacité de chaque noeud



Contexte distribué incompatible avec les liens entre les collections

- Besoin d'un modèle de coût
 - Coût basé sur les communications réseaux - MapReduce
 - Jointure multi-serveurs

Si l'on considère deux collections de données

Pour une jointure entre les deux collections

- Sur quel serveur se fait la jointure ?
- Distribution des données à joindre ?
- Problème de coût réseau (plus cher qu'en local)

- Accès aux serveurs (S) => **Map**
 - Récupérer les éléments de **collection 1** filtré : **C1**
 - Pour chaque élément de **C1**, interroger **collection 2**, filtré : **C2**
 - Peut être très coûteux
 - Equivalent relationnel d'une boucle imbriquée avec index non optimisée et coup réseau supplémentaire
- ⇒ $(S+C1) + C1 * (S+C2)$

