Bases de Données

(G3SI2BD)

AIR1

Manel Zarrouk

zarrouk@lipn.univ-paris13.fr

Institut Galilée
UNIVERSITÉ PARIS 13

2019–2020

Bases de Données

Plan du cours

Part 1 - Introduction

Part 2 - L'algèbre relationnelle

Part 3 - SQL

Part 4 - Modélisation

Part 5 - Programmes et transactions

Bases de Données

Sources des transparents:

MOOC Bases de données relationnelles : apprendre pour utiliser

Philippe Rigaux et Serge Abiteboul

Lien:

https://www.fun-mooc.fr/courses/course-v1:CNAM+01041+session01/about

Je vous encourage fortement à regarder les vidéos de ce MOOC car ils sont complémentaires aux transparents

Bases de Données
Part 1 - Introduction

- Les bases de données
- Les modèles relationnels
- L'universalité
- L'abstraction
- L'indépendance
- Les autres modèles

Bases de Données
Part 1 - Introduction

- Les bases de données
- Les modèles relationnels
- L'universalité
- L'abstraction
- L'indépendance
- Les autres modèles

Les bases de données
BD et SGBD

Une base de données : une collection de données structurées stockées sur un support persistant

Le système de gestion de base de données : le logiciel qui gère les données de la base de données.

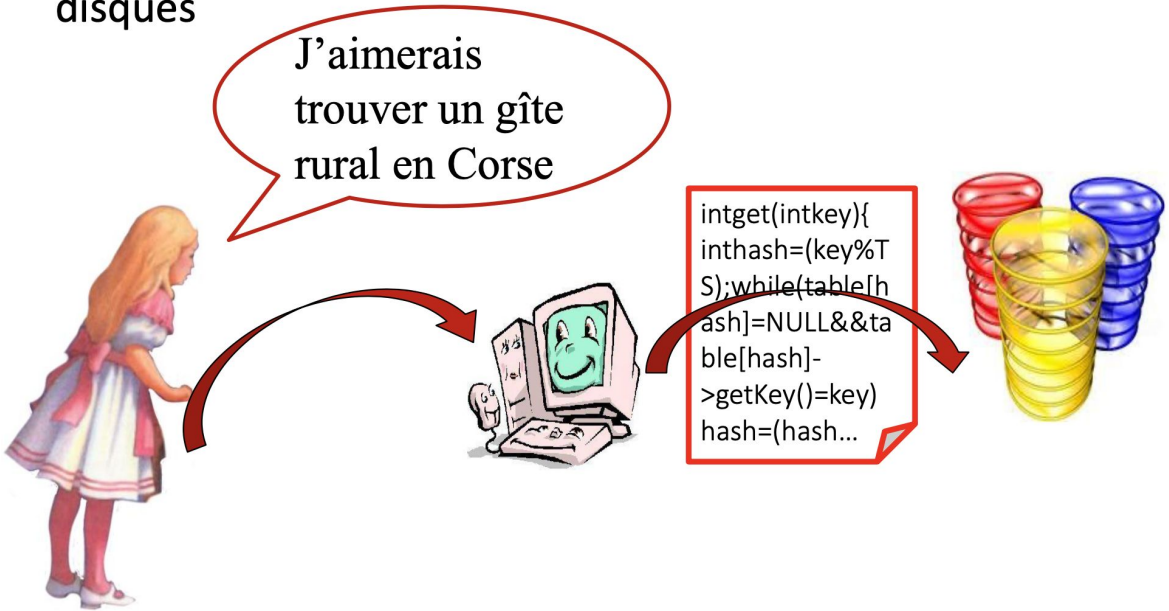
Caractéristiques des données

- Persistance au fil du temps (années) : stockage sur des disques, dans le cloud…
- Volume : gigaoctets, téraoctets...
- Répartition géographique (un bout à Paris, un à Boston…)
- Partage : entre plusieurs utilisateurs et programmes

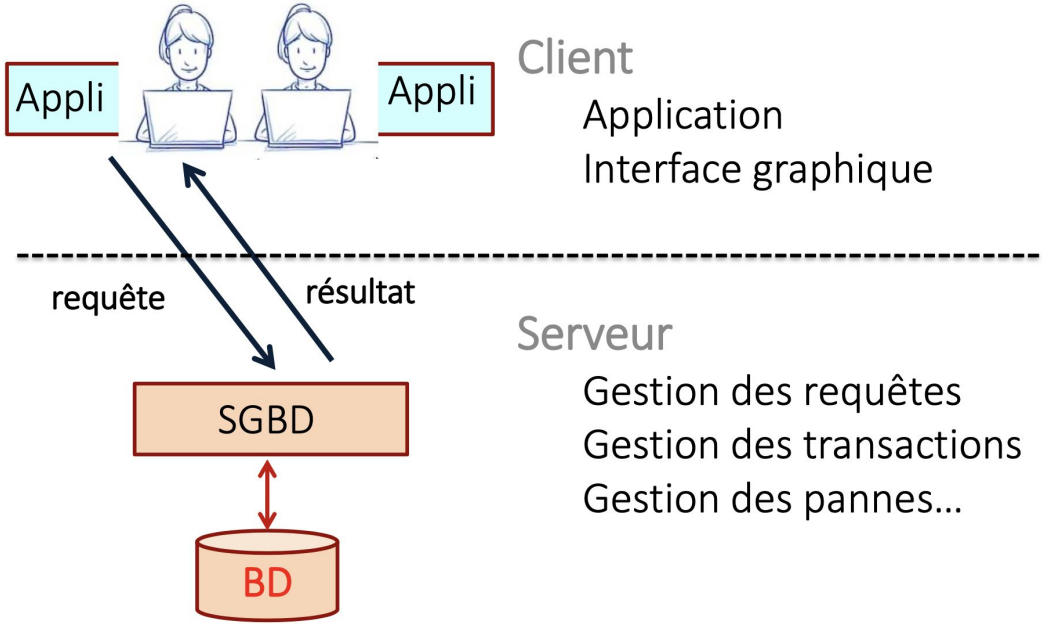
Ce sont des logiciels très complexes

Les bases de données
Le concept

Médiateurs entre les humains et les données sur des disques



Les bases de données
Architecture Client-Serveur



Client

Application

Interface graphique

Serveur

Gestion des requêtes

Gestion des transactions

Gestion des pannes…

Les bases de données
Les SGBD relationnels

- Modèle relationnel
  - Introduit par « Ted » Codd en 1970
- Un grand succès de l'informatique du 20e siècle
- Systèmes commerciaux comme Oracle, DB2, SQL Server…
- Logiciels libres comme MariaDB (suite de MySQL) et Postgres

Les bases de données
La base de données Voyageurs

Le schéma : définit la structure des données

Logement (id, nom, capacité, type, lieu)

Activité (idLogement, codeActivité, description)

Client (id, nom, prénom, ville, pays)

Séjour (id, idClient, idLogement, début, fin)

L'instance : définit leur valeur

Logement

id	nom	capacité	type	lieu
pi	U Pinzuttu	10	Gîte	Corse
ta	Tabriz	34	Hôtel	Alsace
be	Benbow	45	Auberge	Cévennes
re	Réforme	134	Hôtel	Alpes
sh	Tashilhunpo	28	Monastère	Bretagne

Client

id	nom	prénom	ville	pays
…				

Activité

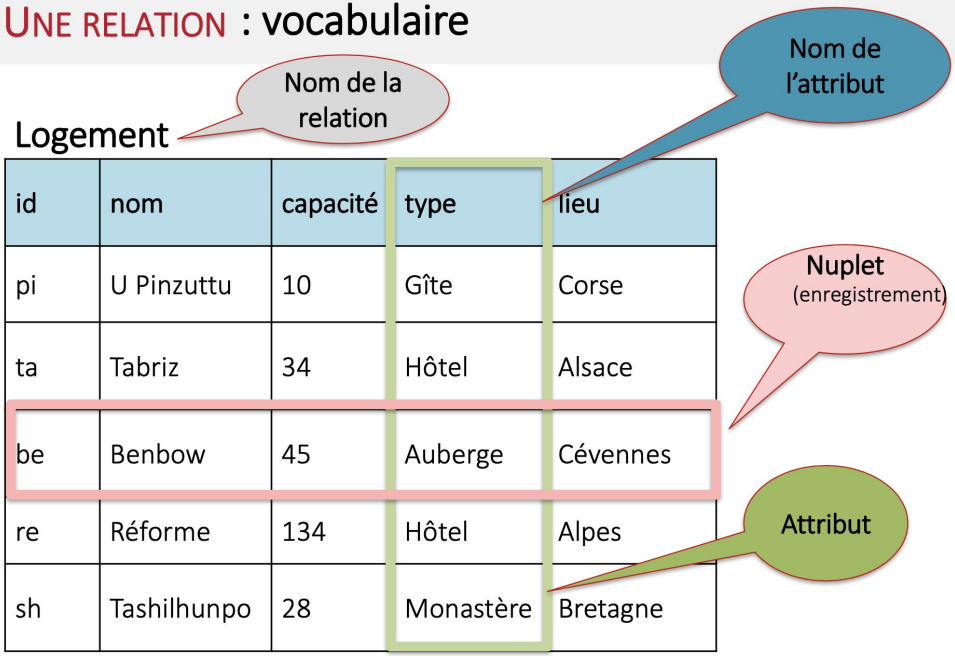
idlogement	codActivité	description
…		

Séjour

id	*idClient*	*idLogement*	début	fin
…				

Les bases de données
Une Relation

UNE RELATION : vocabulaire



id	nom	capacité	type	lieu
pi	U Pinzuttu	10	Gîte	Corse
ta	Tabriz	34	Hôtel	Alsace
be	Benbow	45	Auberge	Cévennes
re	Réforme	134	Hôtel	Alpes
sh	Tashilhunpo	28	Monastère	Bretagne

Les bases de données
À retenir

Les systèmes de gestion de bases de données sont des médiateurs entre les humains et les disques

Une base de données relationnelle est composée de tableaux à deux dimensions, des relations

Bases de Données
Part 1 - Introduction

- Les bases de données
- Les modèles relationnels
- L'universalité
- L'abstraction
- L'indépendance
- Les autres modèles

Les modèles relationnels
Formalisation d'une relation sur un ensemble U d'attributs

Un ensemble fini *U* d'attributs

- Pour chaque attribut *A* de *U*, type(*A*) est un ensemble de valeurs, par exemple, entier, string…

Un ***nuplet*** sur *U* est une fonction qui associe à chaque $A \in U$, une valeur de type(*A*)

Exemple de nuplet sur { id, nom, capacité, type, lieu }

- Id ⟼ be
- Nom ⟼ Benbow
- Capacité ⟼ 20:30
- Type ⟼ Auberge
- Lieu ⟼ Cévennes

Une ***relation*** sur *U* est un ensemble fini de nuplets sur *U*

Les modèles relationnels
Formalisation d'une base de données

Le ***schéma*** d'une base de données

1. Un ensemble fini de nom de relations
   - Exemple, { Logement, Activité, Client, Séjour }
2. Un type pour chaque relation
   - type(Logement) = { id, nom, capacité, type, lieu }
   - type(Activité) = { idlogement, codActivité, description }
   - type(Client) = { id, nom, prénom, ville, pays }
   - type(Séjour) = { id, idClient, idLogement, début, fin }

Une ***instance*** de la base de données associe à chaque nom *R* de relation, une relation sur type(*R*)

- Exemple, Voyage(Logement) est une relation sur type(Logement)

Les modèles relationnels
Pas une BD relationnelle

Les modèles relationnels
Les clés primaires

Logement

Clé primaire

Clé du nuplet

id	nom	capacité	type	lieu
pi	U Pinzuttu	10	Gîte	Corse
ta	Tabriz	34	Hôtel	Alsace
be	Benbow	45	Auberge	Cévennes
re	Réforme	134	Hôtel	Alpes
sh	Tashilhunpo	28	Monastère	Bretagne

La ***clé primaire*** d'une relation : un/plusieurs attributs qui identifient de manière unique un nuplet de la relation

Deux nuplets distincts d'une relation ne peuvent pas avoir la même clé

(et oui : nous verrons des clés non primaires)

Les modèles relationnels
Les clés primaires dans Voyageurs

En gras et rouge, les clés primaires

Logement (**id**, nom, capacité, type, lieu)

Activité (**idLogement**, **codeActivité**, description)

Client (**id**, nom, prénom, ville, pays)

Séjour (**id**, idClient, idLogement, début, fin)

Les modèles relationnels
À retenir

Bien sûr : les notions de nuplets, de relations, de schéma, et d'instance de bases de données

La clé primaire d'une relation : un/plusieurs attributs qui identifient de manière unique un nuplet de la relation

… et les 3 grands principes à venir…

Universalité - Indépendance - Abstraction

Bases de Données
Part 1 - Introduction

- Les bases de données
- Les modèles relationnels
- L'universalité
- L'abstraction
- L'indépendance
- Les autres modèles

L'Universalité
Principe

Les SGBD sont conçus pour capturer

- *toutes* les données d'une entreprise pour
- *tous* les utilisateurs et
- *toutes* les applications

Des fonctionnalités étendues pour arriver à gérer la variétés des besoins

L'Universalité
Différents type d'applications

Ajouter/trouver une information pertinente

- Agenda, courriel…
- Allocine, tripadvisor…

Online Transaction Processing – Transactionnel (OLTP)

- Transactions simples comme un retrait d'un DAB
- Banque, E-commerce…
- Peut-être gros volume de transactions par seconde

Online Analytical Processing – Analyse de données (OLAP)

- Calculs comme de comparer les ventes de produit en 2006 et 2007
- Requêtes très compliquées avec des fonctions « agrégat »
- Requêtes multidimensionnelles, date, pays, produit

L'Universalité
Fonctionnalités des SGBD

Transactions : gestion des accès concurrents aux données

Sureté : gestion des pannes aussi bien logicielles que matérielles

Sécurité : protection les données contre les attaques

Distribution : gestion de données distribuées géographiquement

Passage à l'échelle : laisser grossir l'application avec de bonnes performances

- En volume de données
- En volume de travail : nombre de requêtes ou de transactions

L'Universalité
Concurrence et transactions

Pouvoir partager des données de manière harmonieuse

Exemple : écriture de rapports

Propriétés ACID : atomicité, cohérence, isolation, durabilité

L'Universalité
Reprise sur pannes

Le SGBD doit résister aux pannes

- Journal des opérations
- Copies de sauvegarde

Disponibilité dans des délais raisonnables pour les applis

L'Universalité
Données distribuées

C'est le cas

- Quand on intègre des données de plusieurs sources (plusieurs entreprises, plusieurs branches d'une même compagnie, plusieurs ami.e.s…)
- Quand on distribue les données entre plusieurs systèmes pour améliorer les performances

Evaluation de requêtes quand les données sont distribuées

- Localiser qui peut contribuer à la réponse
- Envoyer des requêtes à ces systèmes
- Combiner les résultats

Transactions distribuées

- « Commit à deux phases »

L'Universalité
À retenir

Principe d'universalité

Les SGBD sont conçus pour

toutes les données d'une entreprise pour

tous les utilisateurs et pour

toutes les applications

Des fonctionnalités étendues pour arriver à ça : gestion de la concurrence, des pannes, de la distribution…

Bases de Données
Part 1 - Introduction

- Les bases de données
- Les modèles relationnels
- L'universalité
- **L'abstraction**
- L'indépendance
- Les autres modèles

L'Abstraction
Principe

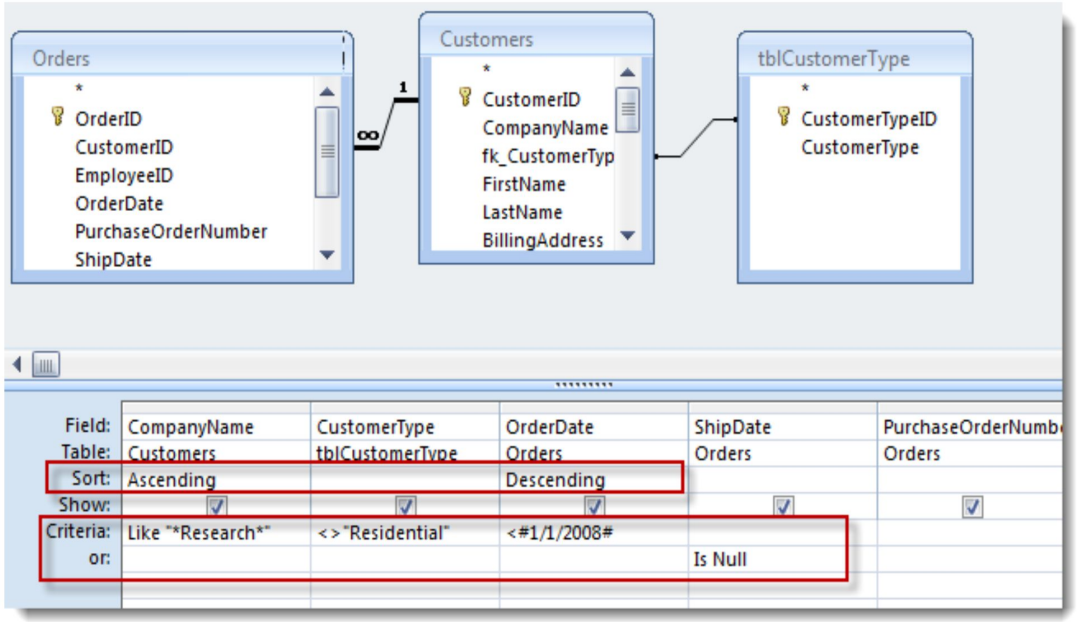
Le but

- Voir les données de façon abstraite, par exemple des tableaux à deux dimensions (en ignorant les détails de leur organisation réelle sur disque)
- Les manipuler avec des langages de haut niveau (et pas en écrivant du code informatique compliqué)

Des langages de haut niveau

- Un standard universel : SQL
- L'algèbre relationnelle : une boite à outils
- Le calcul relationnel : un langage déclaratif
- Et des langages graphiques

L'Abstraction
Reqêtes graphiques



Exemple : Microsoft Access

L'Abstraction
SQL: un langage pour exprimer des requêtes

(Donnez-moi la liste des noms et les lieux des hôtels)

```
select nom, capacité, lieu
from Logement
where type = 'Hôtel'
```

Un standard pour tous les SGBD relationnels

Compréhensible par un humain

Interprétable par une machine

Les systèmes relationnels utilisent aussi SQL entre eux

L'Abstraction
SQL: un langage pour exprimer aussi des mises-à-jour

```
insert into Logement (id,nom,capacité,type,lieu)
values (777, 'La Sauldre',25,'Hôtel', 'Sologne')

update Logement
set capacité = '15'
where nom = 'U Pinzuttu'
```

L'Abstraction
L'algèbre relationnelle : une boite à outils

$\Pi_{nom,\ lieu}(\sigma_{type='Hôtel'}(Logement))$

Sélection : Res1 = $\sigma_{type='Hôtel'}(Logement)$

id	nom	capacité	type	lieu
ta	Tabriz	34	Hôtel	Alsace
re	Réforme	134	Hôtel	Alpes

Projection : Res2 = $\Pi_{nom,\ lieu}(Res1)$

nom	lieu
Tabriz	Alsace
Réforme	Alpes

L'Abstraction
Le calcul relationnel : Lamgage déclaratif

(Donnez-moi la liste des noms et les lieux des hôtels)

$\{ y,t \mid \exists\, x,z \ \text{Logement}(x,y,z,\text{'Hôtel'},t) \}$

L'Abstraction
Raisons principales du succès des systèmes relationnels

- **L'abstraction** : Les données sont vues de façon abstraite et manipulées avec des langages compréhensibles par des humains
- Les requêtes sont compilées en des **requêtes algébriques** qui peuvent être évaluées automatiquement
- Des techniques d'optimisation et le parallélisme permettent de **passer à l'échelle de très grandes bases de données**

L'Abstraction
À retenir

Principe d'abstraction

On voit les données de façon abstraite : des tableaux à deux dimensions

On les interroge avec des langages de haut niveau.

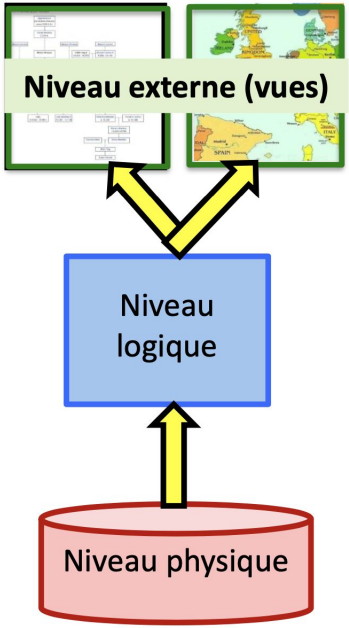
Nous verrons

- L'algèbre relationnelle
- SQL
- Le calcul relationnel

Bases de Données
Part 1 - Introduction

- Les bases de données
- Les modèles relationnels
- L'universalité
- L'abstraction
- L'indépendance
- Les autres modèles

L'indépendance
Principe



Architecture ANSI-SPARC (75)

Distinction de 3 niveaux

- Niveau physique : organisation sur disque
- Niveau logique : vue abstraite unifiée
- Niveau externe (vues) : vues particulières

Indépendance entre ces niveaux

- Physique et logique / externe : on peut changer l'organisation physique sans affecter les deux autres niveaux (sans changer les programmes d'application)
- Externe / logique et physique : on peut ajouter des vues sans affecter les deux autres niveaux

L'indépendance
Répartition des rôles

Niveau physique

- Gestion de mémoire secondaire : données (fichiers), schéma, index
- Gestion concurrence et pannes

Niveau logique

- Compilation
- Optimisation des requêtes
- Gestion de la confidentialité et intégrité

Niveau externe

- Environnement de programmation
- Interfaces graphiques
- Outils d'aides
- Débogueurs

L'indépendance
Exemple de Vue

La vue : « Qui va où ? »

QuiVaOù

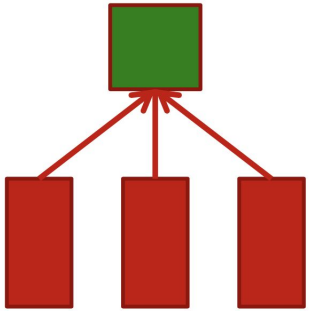
nomClient	prénomClient	nom	lieu
…			

Une requête permet de spécifier/évaluer cette vue

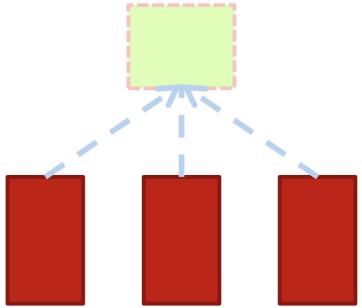
```
select nomClient, prénomClient, lieu, type
from Logement, Client, Séjour
where  Logement.id = Séjour.idLogement and
       Client.id = Séjour.idClient
```

L'indépendance
Vue matérialisée ou Intentionnelle

Matérialisée : entrepôt



Intentionnelle : médiateur



Dans la pratique parfois, cocktail des deux

L'indépendance
Requête Vs Maj : Le compromis des SGBD

Matérialisée	Intentionnelle
Requête : simple	Maj : rien à faire
Maj : propager	Requête : plus coûteux
• Base → vue (coûteux)	
• Vue → base (ambigu)	

L'indépendance
À retenir

Séparation en 3 niveaux

- Niveau physique : organisation sur disque
- Niveau logique : vue abstraite unifiée
- Niveau externe : vues particulières

Principe d'indépendance entre ces niveaux

Bases de Données
Part 1 - Introduction

- Les bases de données
- Les modèles relationnels
- L'universalité
- L'abstraction
- L'indépendance
- Les autres modèles

Les autres modèles
Histoire simplifiée

Les « vieux » modèles : réseau et hiérarchique	60-70
Le modèle relationnel	70-…
Le modèle objet	90-…
Les modèles « modernes » : XML, Json, RDF	00-…

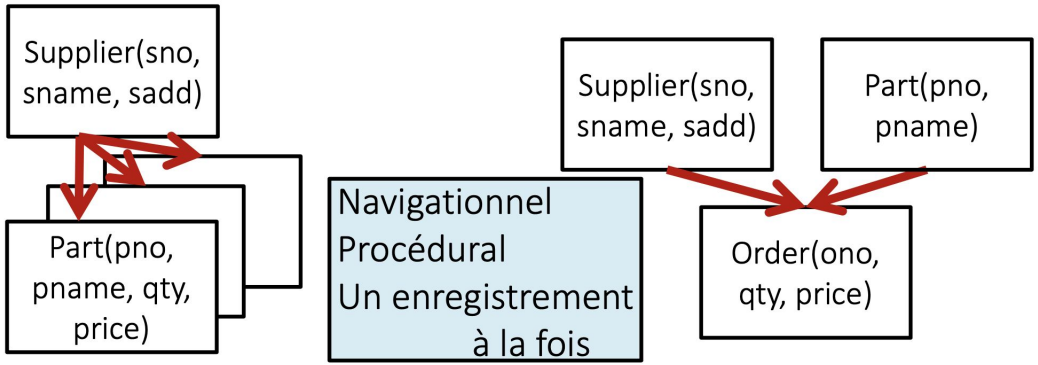
Les autres modèles
Les modèles historique 60-70

Modèle hiérarchique : arbres

- **IMS**, IBM
- Une hiérarchie d'enregistrements

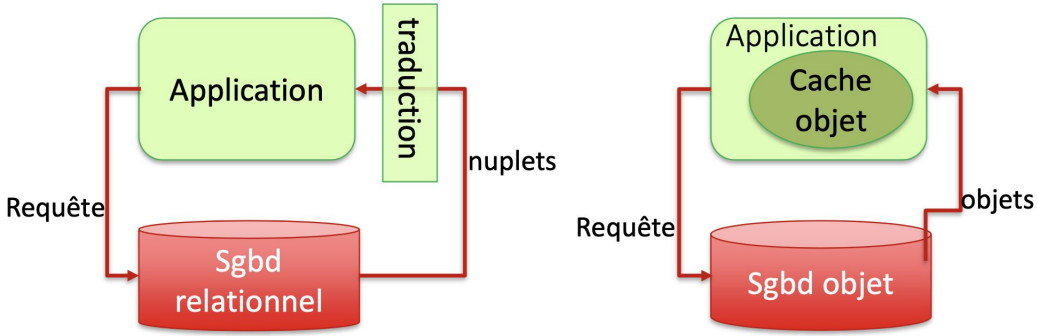
Modèle réseau : graphes

- **Codasyl**
- Un graphe d'enregistrements



Les autres modèles
Les bases de données Objet 90

Objet = données + comportement



Systèmes open-source comme db4o

Systèmes hybrides relationnel-objet

Les autres modèles
Le relationnel-Objet

Les systèmes relationnel-objet sont des extensions des SGBD relationnel avec une interface objet

Nous reviendrons là-dessus en plus de détail

Les autres modèles
Les SGBD modernes

Arbres

- **XML, Json**
- Données semistructurées
- Langage de requête : Xpath, Xquery

Graphes

- Web sémantique & **RDF**
- Représentation de connaissances
- Langage de requête : SPARQL

Les systèmes modernes

- **NoSQL**
- **Column store**
- **MapReduce**
- **Bases de données en mémoire**

Les autres modèles
À retenir

Les modèles « historiques » étaient très procéduraux

Le modèle relationnel est le plus utilisé

D'autres modèles sont utilisés souvent pour répondre à des problèmes rencontrés avec le modèle relationnel