MERISE 2

PAR

RALAIVAO Jean Christian

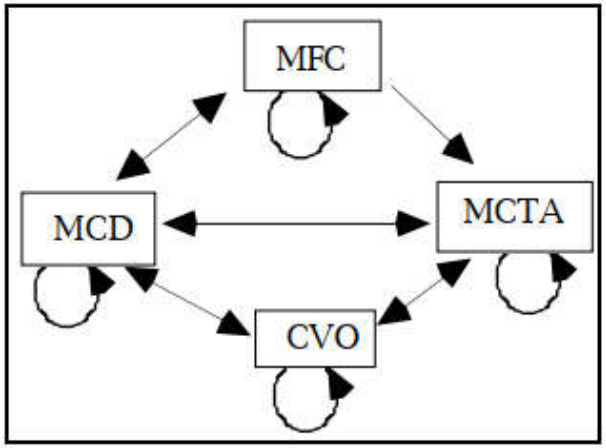
MODELES DE MERISE 2

	INTERFACE	APPLICATION		
		STATIQUE	DYNAMIQUE	ARCHITECTURE
quoi	MC	MCD	MCTA CVO	MFC
d'où qui quand	MOT	MOD	MOTA CVO	MFO
où comment	Maquettes IHM	MLD MLDr	MLT MLTr	SALMI, SAL SALR
		est	se comporte	fait

MC : modèle de contexte
MCD : modèle conceptuel des données
CVO : cycle de vie des objets
MCTA : modèle conceptuel des traitements analytique
MFC : modèle de flux conceptuel
MOT : modèle organisationnel des traitements
MOD : modèle organisationnel des données
MOTA : modèle organisationnel des traitements analytique
MFO : modèle de flux organisationnel
MLD : modèle logique des données
MLDR : modèle logique des données réparties
MLT : modèle logique des traitements
MLTR : modèle logique des traitements répartis
SALMI : schéma d'architecture logique des moyens informatiques
SAL : schéma d'architecture logique
SALR : schéma d'architecture logique répartie

NIVEAU CONCEPTUEL DE MERISE 2

- Utilisation des MFC
- MCT - -> MCTA
- Extension du MCD
- Introduction des CVO
- Démarche itérative

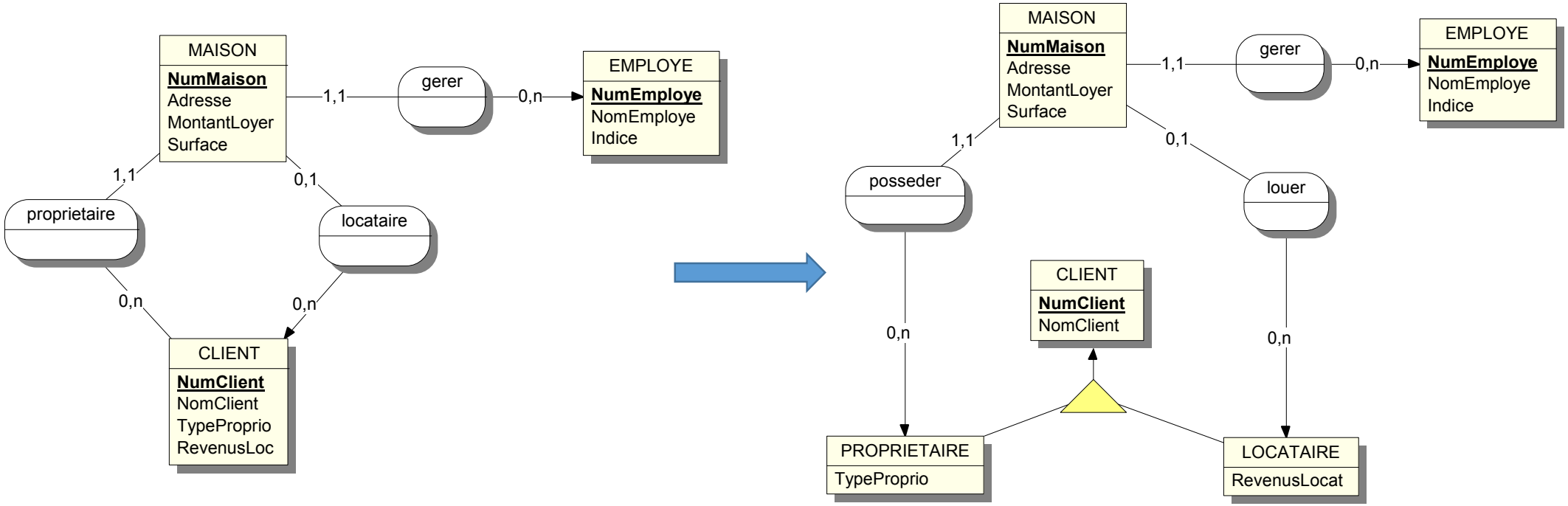


EXTENSION DU MEA

- Généralisation/Spécialisation des entités
- Contraintes d’intégrité de base entre sous-types
- Contraintes d’intégrité d’extension entre sous-types
- Passage au MLD
- Contraintes entre associations

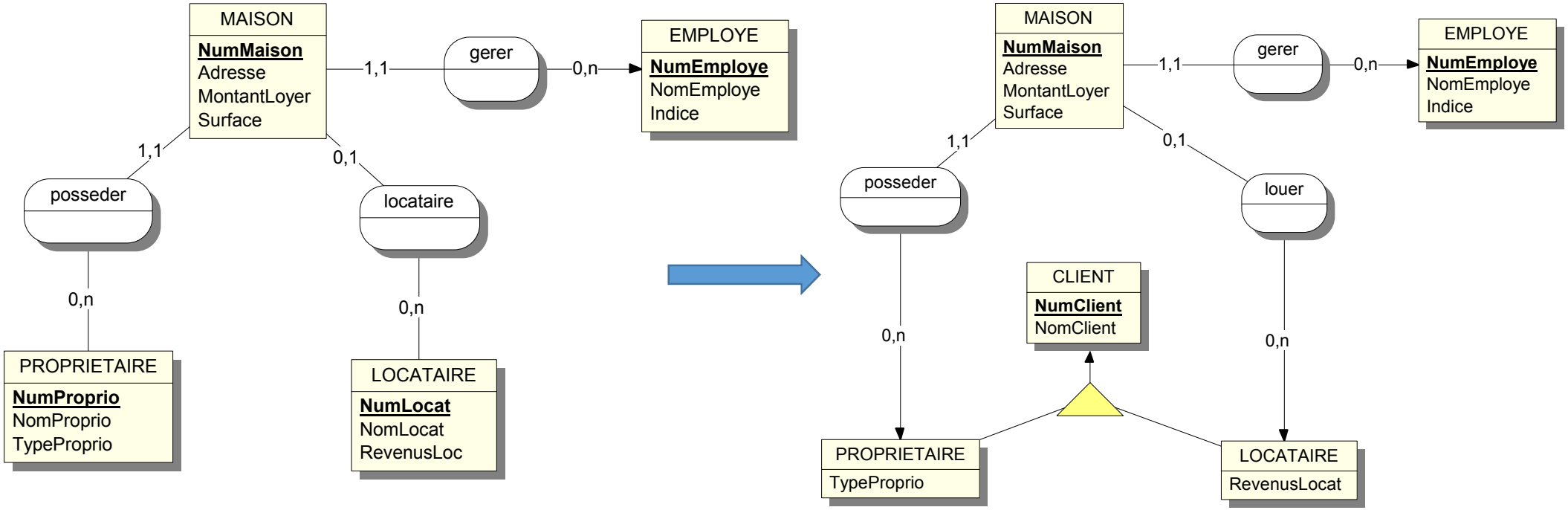
HERITAGE DES ENTITES

SPECIALISATION



HERITAGE DES ENTITES

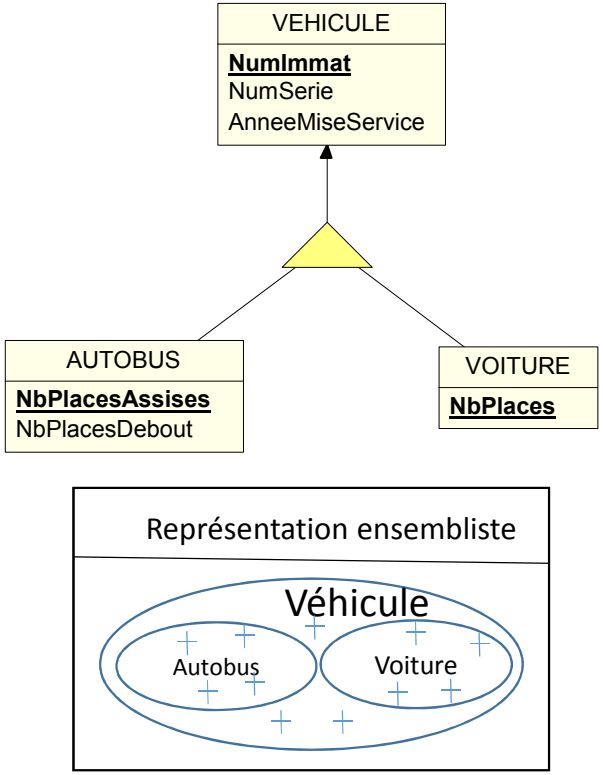
GENERALISATION



CONTRAINTES D'INTEGRITE DE BASE

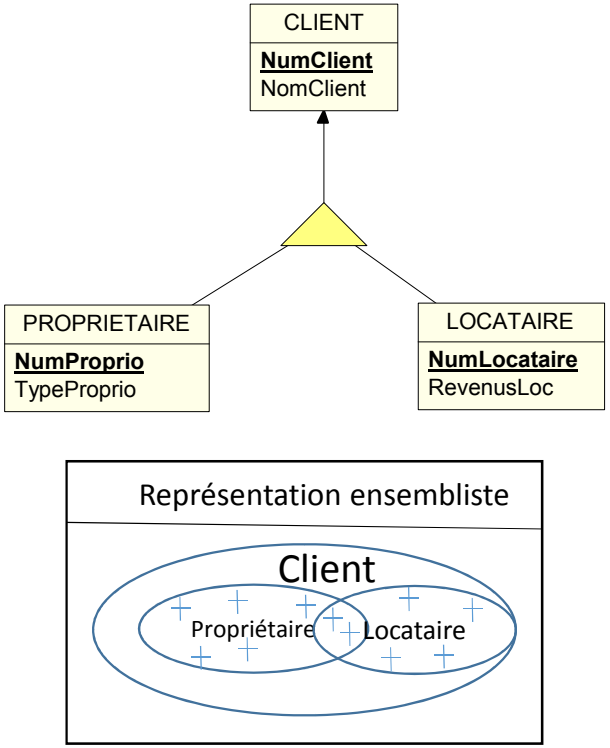
- Contrainte de disjonction

Toute occurrence de l'entité générique doit appartenir à un seul sous-type



- Contrainte de couverture

Toute occurrence de l'entité générique doit appartenir à au moins l'un des sous-types

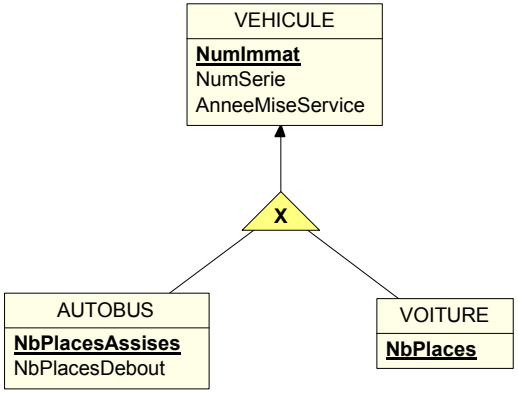


CONTRAINTES D'INTEGRITE D'EXTENSION

Exclusion

X

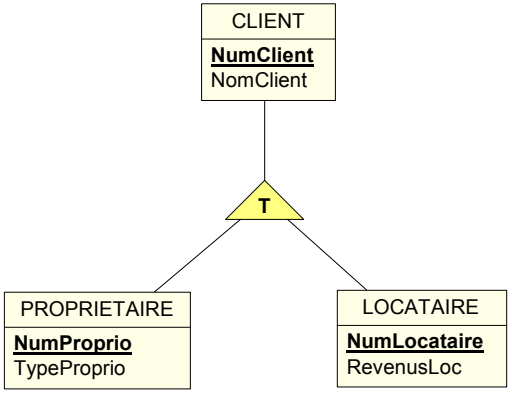
Disjonction + non Couverture



Totalité

T

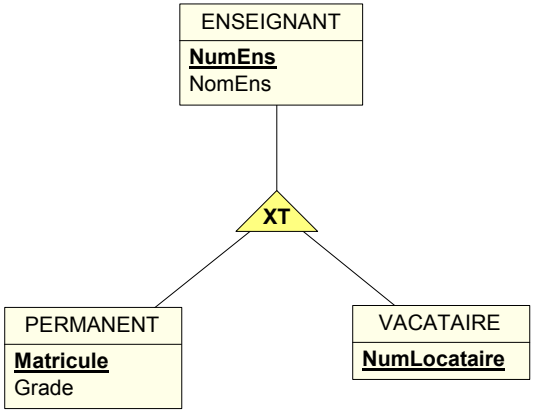
Couverture + non Disjonction



Partition

XT ou +

Couverture + Disjonction



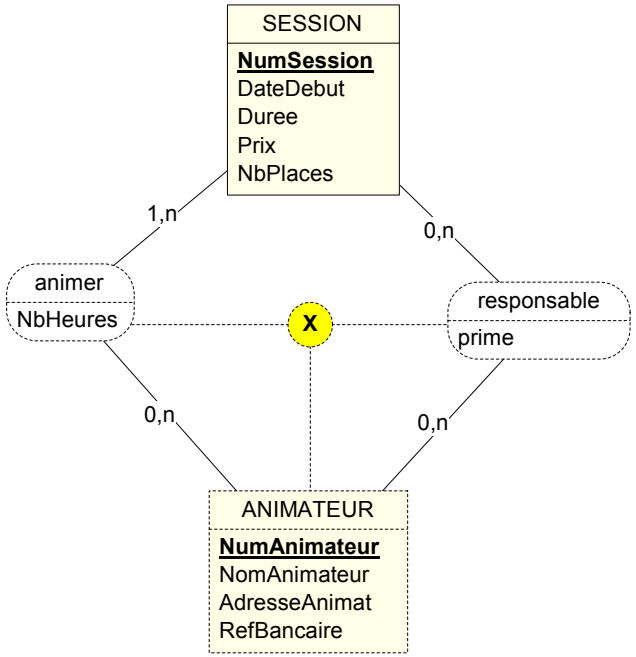
PASSAGE AU MLD

- Trois stratégies de base
  - Créer uniquement l'entité générique
  - Créer uniquement les entités spécifiques
  - Créer l'entité générique et les entités spécifiques avec héritage
    - Soit de toutes les propriétés
    - Soit de l'identifiant uniquement

CONTRAINTES ENTRE ASSOCIATION

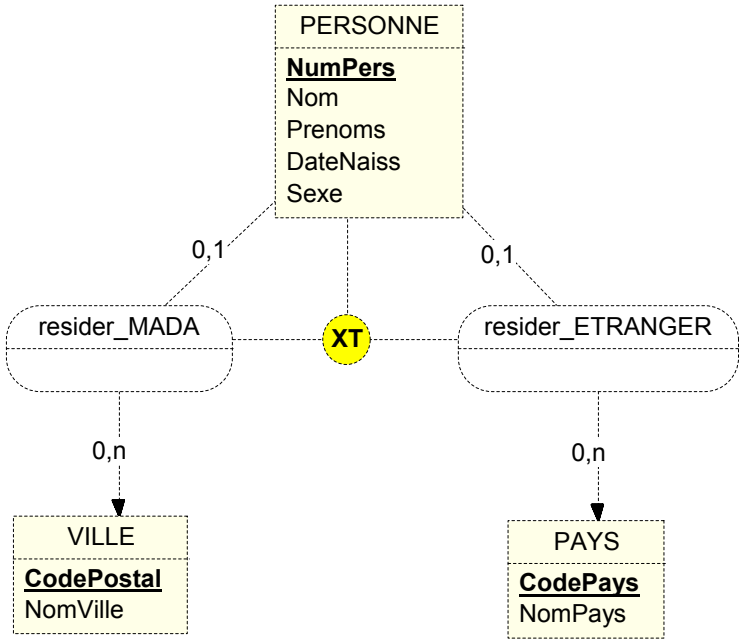
Pivot = ensemble des entités communes aux associations concernées par la contrainte

Contrainte d'exclusion : X



Un animateur peut animer ou être responsable d'une session mais ne peut pas à la fois animer et être responsable.

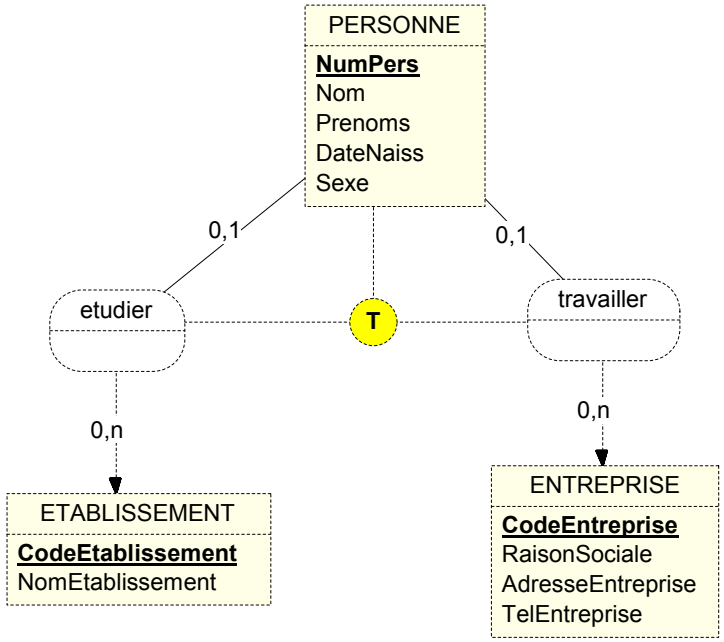
Contrainte de partition : XT ou +



Une personne est soit résidente à Mada, soit résidente à l'étranger mais ne peut pas être les deux.

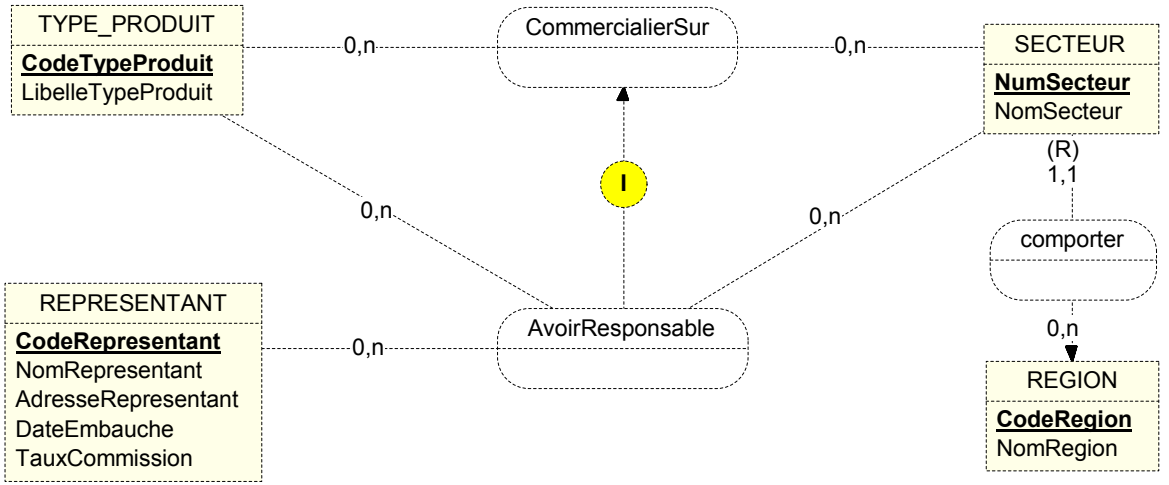
CONTRAINTES ENTRE ASSOCIATION

Contrainte de totalité : T



Une personne est étudiant dans un établissement ou salarié d'une entreprise ou les deux à la fois.

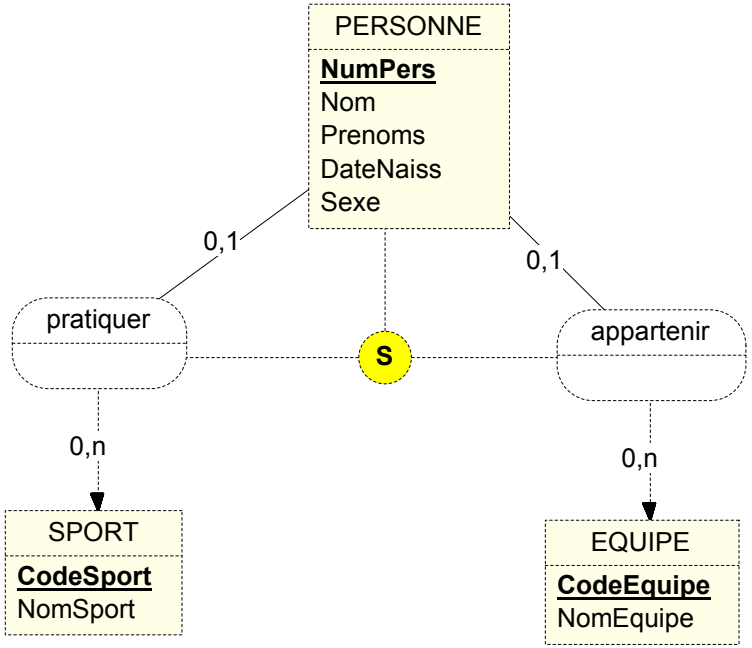
Contrainte d'inclusion : I



L'ensemble des produits du secteur qui ont un responsable est constitué uniquement de produits qui sont commercialisés sur le secteur.

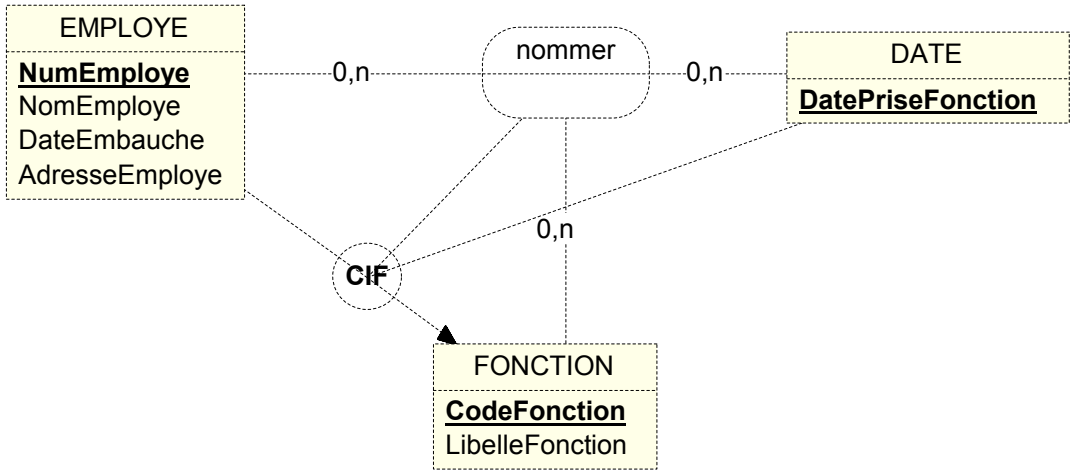
CONTRAINTES ENTRE ASSOCIATION

Contrainte d'égalité ou de simultanéité : = ou S



Toute personne qui pratique un sport fait partie d'une équipe et vice-versa.

Contrainte d'unicité : CIF ou 1



Un employé à une date donnée ne peut pas être nommé qu'à une seule fonction.

CVO

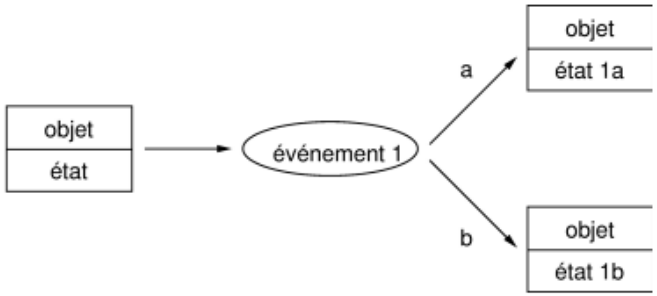
- Modèle proche du Diagramme d'Etats-Transition d'UML

Graphe orienté où les sommets correspondent aux états et aux événements (les sommets sont de deux types différents) et où les arcs correspondent aux transitions d'un état à un événement ou l'inverse

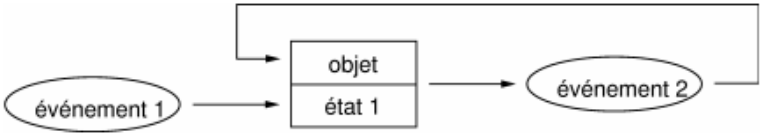
Création

Séquence

Alternative



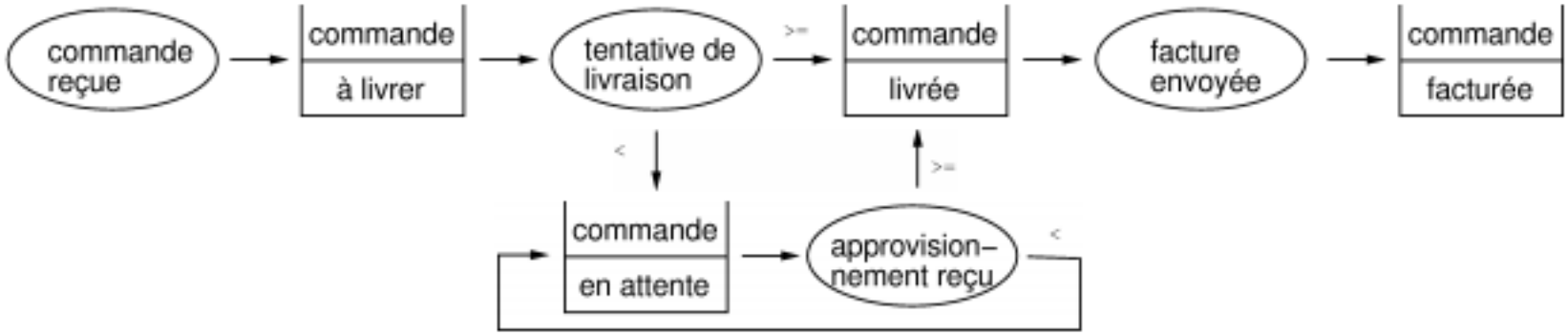
Itération



Suppression

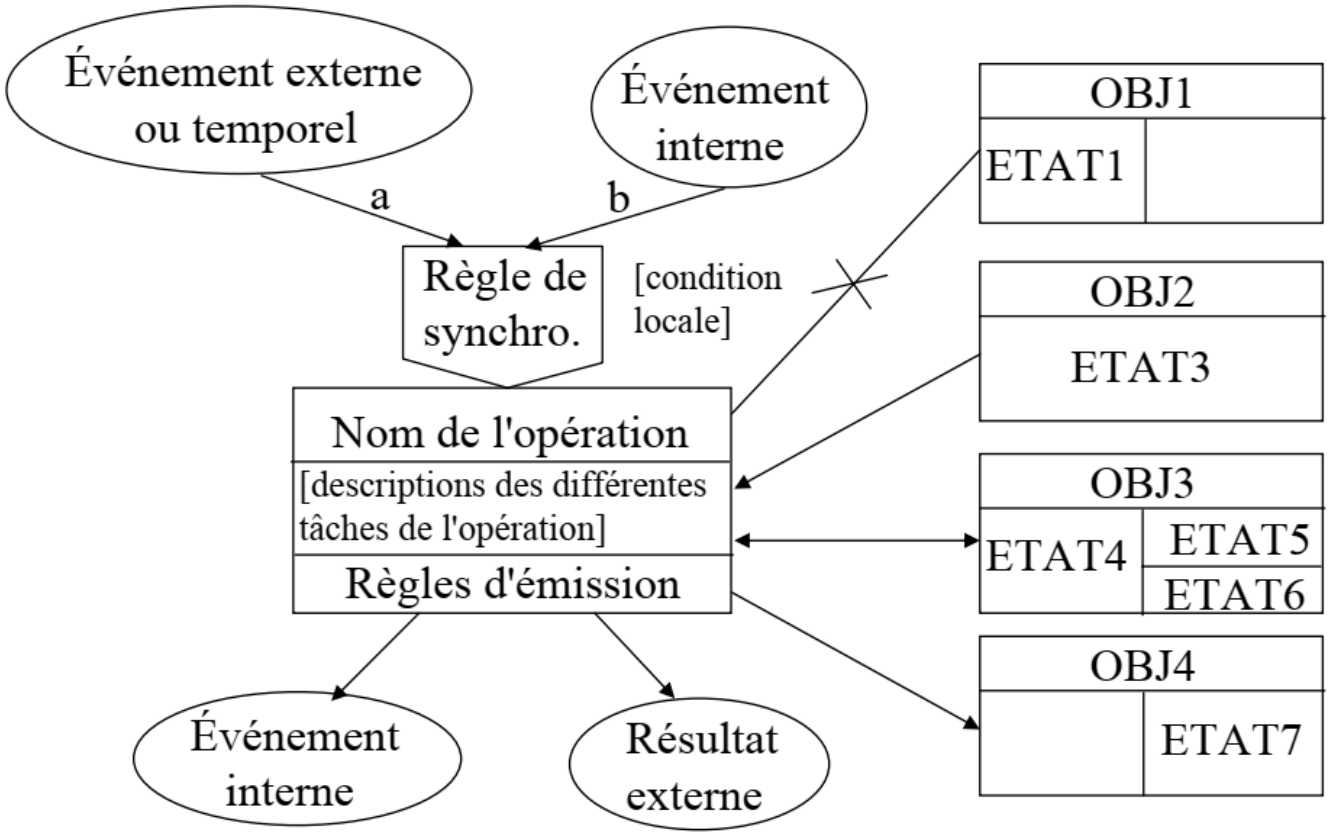
CVO

Exemple : une commande

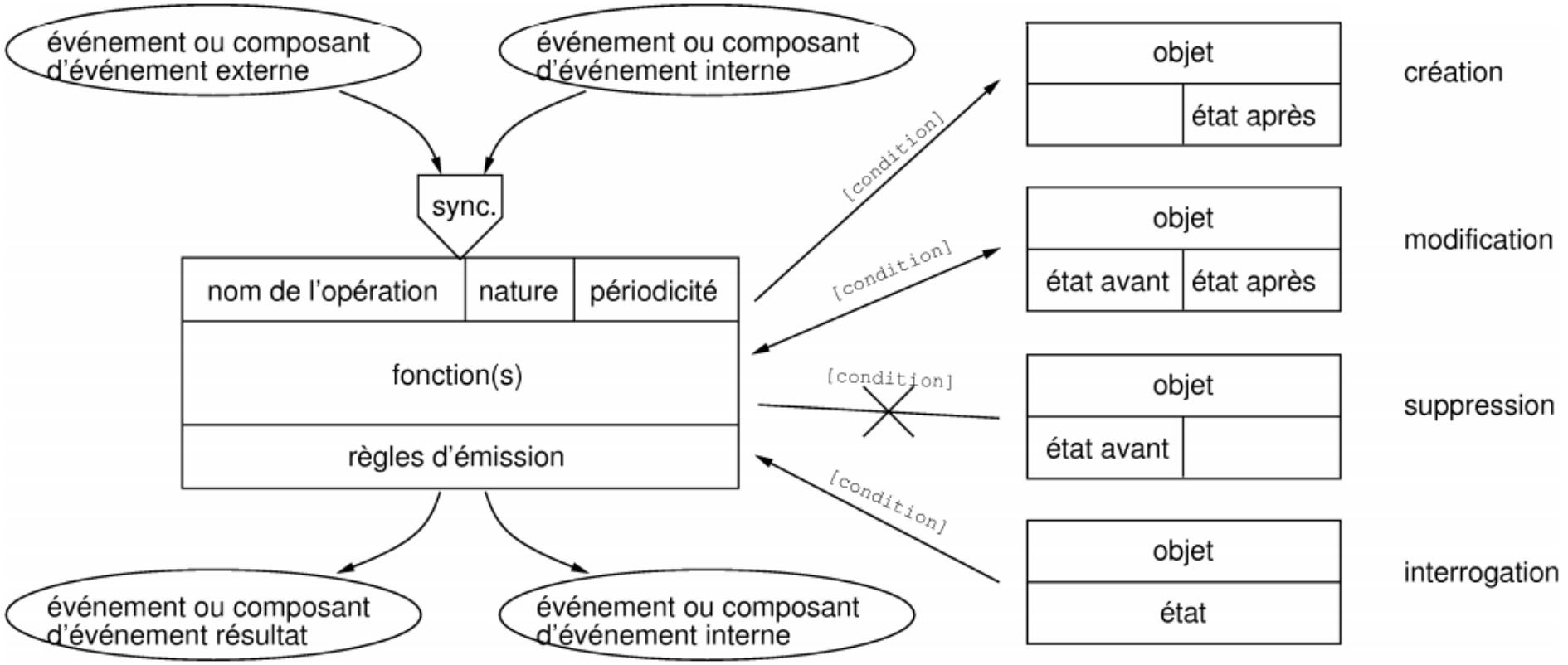


< Quantité en stock < Quantité commandée
>= Quantité en stock >= Quantité commandée

MCTA = MCD + Modèles Externes



MOTA


événement ou composant d'événement externe
événement ou composant d'événement interne
sync.
nom de l'opération
nature
périodicité
fonction(s)
règles d'émission
événement ou composant d'événement résultat
événement ou composant d'événement interne
[condition]
[condition]
[condition]
[condition]
objet
état après
création
objet
état avant
état après
modification
objet
état avant
suppression
objet
état
interrogation

OPTIMISATION DU MLD

- Concept d'optimisation
  - Approche pour optimiser
  - Etapes d'optimisation et facteurs
- Conception optimisée de la base
  - Redondance calculée
    - Dénormalisation intelligente
    - Diminution du nombre de tables
    - Mise en place des informations rédondantes
    - Partitionnement des tables
  - Indexation
    - Colonnes à indexer
    - Trop d'index tue l'index
  - Clusterisation

CONCEPT D'OPTIMISATION

- Approches pour optimiser
  - Processus optimale en termes de temps de réponse et de ressources
  - Deux éléments : puissance du matériel et réglage (tuning)
- Etapes d'optimisation et facteurs pris en compte
  - Un type d'optimisation par phase du cycle de vie en se basant sur des contraintes particulières :
    - au niveau du MLD
    - au niveau du MPD
    - au niveau de la

CONCEPTION OPTIMISEE DE LA BASE

- Objectif : Adapter la représentation (structure des données) aux besoins et spécificités des traitements
- Comment ?
  - Redondance calculée
  - Partitionnement des tables
  - Choix des index et des clusters
  - Choix des paramètres de stockage

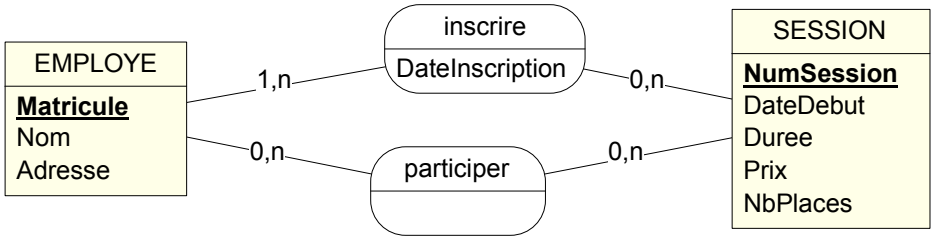
REDONDANCE CALCULEE

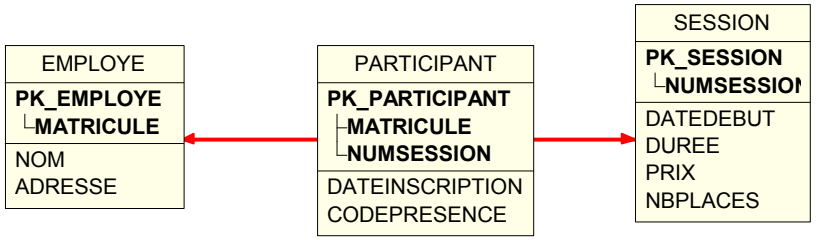
- Créer volontairement des redondances afin de favoriser certaines transactions
- Dénormalisation « intelligente »
  - Constat de baisse de performance pour un modèle « hautement » normalisé
  - Trop de tables nécessitant des jointures et des parcours d'index

  → Nécessité de dénormaliser physiquement la base de données
  - Conditions
    - S'effectue au niveau physique sur un modèle logique normalisé
    - Ne fait perdre aucun des avantages de la normalisation
    - Structure plus simple et plus optimisée que la structure normalisée

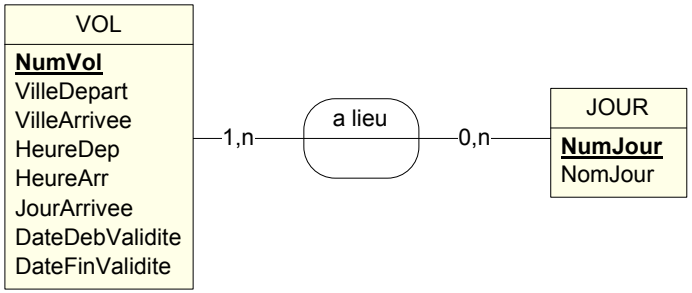
Dénormalisation « intelligente »

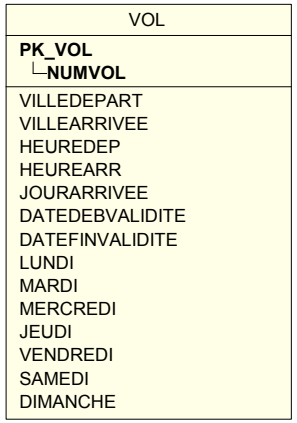
- Diminution du nombre de tables
  - Supprimer les tables qui ne comportent qu'une colonne clé
  - Fusionner les tables ayant une clé primaire identique





  - Fusionner les tables pour éviter les jointures





Les colonnes LUNDI au DIMANCHE sont de type Logique (vrai lorsque le vol a lieu.

  - Fusionner les tables de codification

Dénormalisation « intelligente »

- Mettre en place des informations redondantes
  - Redondance sélective
    - Minimiser le temps d'accès
    - Minimiser la complexité de recherche
    - Règles à vérifier
      - La redondance est valable si le champ est stable (peu fréquemment mis à jour)
      - Lors de la mise à jour, il faut la propager sur les données redondantes
  - Stockage des valeurs calculables

Partitionnement des tables (BD répartie)

- Rapprocher les données des sites qui les utilisent le plus
- Partitionner la base de données en « fragments »
  - Partitionnement horizontal (suivant les lignes)
  - Partitionnement vertical (suivant les colonnes)
  - Partitionnement mixte
- Créer une vue englobant tous les fragments pour les traitements travaillant sur la table entière

INDEXATION

- Objectif : Eviter de parcourir toute la table
- Deux types : sans doublon (unique) ou avec doublon
- Colonnes à indexer
  - Clés primaires
  - Clés étrangères (pour la jointure)
  - Colonnes servant de critère de recherche
  - Colonnes servant de critère de tri et de regroupement
- Attention ! TROP d'index tue l'index
- Inconvénients :
  - mise à jour (suppression, modification, insertion)
  - Pénalisation de certaines requêtes

CLUSTERISATION DES DONNEES

- Objectif : Ranger dans un même bloc des lignes (qui peuvent provenir d'une ou de plusieurs tables) ayant une valeur colonne commune (clé de cluster).
- Attention à la taille d'un bloc : l'existence de blocs chainés est néfaste pour la performance
- Ne pas clustériser des tables dont la clé de cluster change fréquemment