

Rappels sur les bases de données relationnelles

Bernard ESPINASSE

Professeur à l'Université d'Aix-Marseille

Septembre 2015

- Niveaux de description d'une base de données
- Rappel sur le modèle Entité-Relation (E-R)
- Dérivation d'un MLD-R à partir d'un MCD en Entité-Relation
- Dimensionnement d'une BD Relationnelle
- Sous schéma de données et valorisation de l'activité des traitements sur la BD

Plan

- 1. Niveaux de description d'une base de données**
 - Niveau « Réel perçu » et « Niveau Interne ou Logique »
 - Niveau « Conceptuel » et « Externe »
- 2. Rappel sur le modèle Entité-Relation (E-R)**
 - Historique, concepts, graphisme
 - Cardinalités (ou multiplicités)
 - Dépendances fonctionnelles
- 3. Dérivation d'un MLD-R à partir d'un MCD en Entité-Relation**
 - Problématique du MLD
 - Formalisme graphique de Merise
 - Dérivation d'un MLD-R à partir d'un MCD en Entité-Relation
 - Création de tables en langage SQL (clé primaires et étrangères)
- 4. Dimensionnement d'une BD Relationnelle**
 - Multiplicités moyennes des liens relationnels et propagation des multiplicités moyenne
 - Calculs cumulés des volumes des tables
- 5. Sous-schéma de données, valorisation de l'activité des traitements sur la BD**
 - Transformation de sous-schémas conceptuels en sous-schémas logiques
 - Valorisation des primitives logiques

1. Niveaux de description d'une base de données

- Niveau « Réel perçu » et « Niveau Interne ou Logique »
- Niveau « Conceptuel » et « Externe »

Niveaux « Réel perçu » et « Niveau Interne ou logique » (1)

Niveau « Réel perçu » :

- Représentation du réel que l'on se construit selon les finalités, le phénomène observé
- Exprimé dans le langage naturel + vocabulaire du domaine étudié.

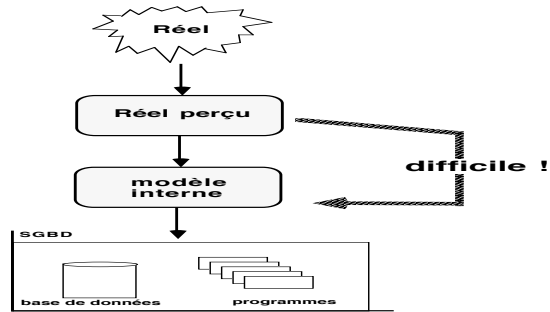
Niveau « interne ou logique » :

- Définit dans le système informatique la réalisation de la structure de données selon le SGBD choisi (notamment relationnel) et d'objectifs d'optimisation
- Exprimé dans le formalisme informatique lié à l'outil informatique :

Modèle Interne (ou logique) de Données (MLD)

Modèle Logique de Données Relationnel (MLDR)

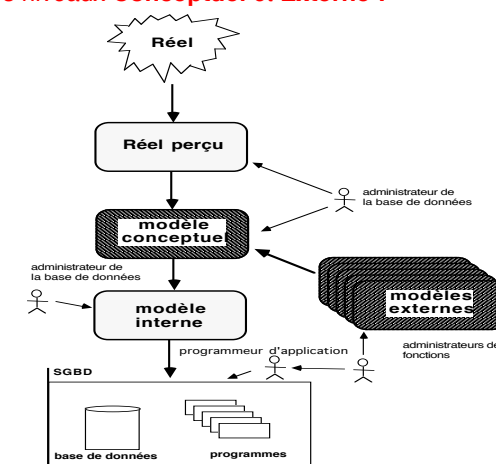
Niveaux « Réel perçu » et « Niveau Interne ou logique » (2)



Grande **difficulté** à décrire le « **réel perçu** » dans un « **modèle interne** » informatique :
 -> émergence d'un niveau **intermédiaire** (rapport ANSI/Sparc 1975) : **le niveau conceptuel**

Niveaux « Conceptuel » et « Externe » (1)

• Emergence des niveaux **Conceptuel** et **Externe** :



Remarque : définition des rôles dans l'organisation

Niveaux « Conceptuel » et « Externe » (2)

- Description en termes **d'objets, propriétés** et **relations** du réel perçu, permettant :
 - **l'unicité** et **stabilité** la structure de mémorisation des informations
 - la **rencontre** gens du domaine/informaticiens
- Exprimé dans un **formalisme plutôt naturel** (pour gens du domaine) tout en étant rigoureux (pour les informaticiens):

Modèle Conceptuel de Données (MCD): exprimé en formalisme **Entité-Relation (E-R)**

Niveau externe :

- Description en terme d'objets, propriétés et relations d'une utilisation particulière d'information du domaine :

Modèle Externe (MED) : exprimé dans le même formalisme qu'au niveau conceptuel

On parlera souvent de « Sous-Schémas conceptuels externes »

2. Rappel sur le modèle Entité-Relation (E-R)

- **Historique, concepts, graphisme**
- **Cardinalités (ou multiplicités)**
- **Dépendances fonctionnelles**

Historique

- pour l'élaboration des **modèles de données** (MCD):
- nécessité d'un **formalisme graphique**
 - approche ENTITE-RELATION (P.Chen 75)
 - formalisme INDIVIDUEL (D.Nanci, D.Pascot, H.Tardieu 75)
 - norme ISO: ENTITY-RELATIONSHIP

• 3 concepts de base :

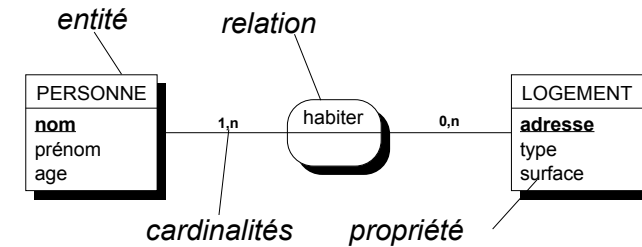
- **PROPRIETE**
- **ENTITE**
- **RELATION (association)**

Graphisme

3 concepts de base :

- **PROPRIETE** : *description*
- **ENTITE** (individu): *structure*
- **RELATION** (association): *structure*

Une représentation graphique :



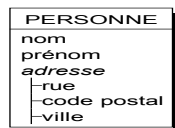
Propriété

Propriété = particule élémentaire d'information

| nom | valeur |
|---------------|-----------------------|
| TYPE | OCCURENCE |
| couleur | rouge bleu vert |
| âge | 22 23 24 56 |

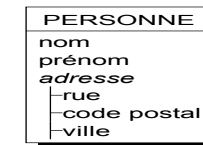
attention !!!! : dans un modèle de données, on ne représente pas les valeurs mais les TYPES des valeurs

Propriété composée : adresse : rue, code postal, ville



Entité

Entité = représentation d'une famille d'objets **distinguables** perçus comme **stables** et **homogènes**

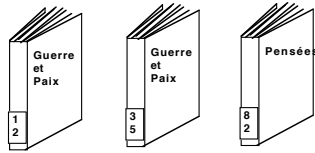


Règles :

- 1) pour toute occurrence d'entité, il y a au plus une valeur pour chacune de ses propriétés (**stabilité**)
- 2) au moins une propriété, identifiante, permet de distinguer les occurrences d'entité entre elles (**distinguabilité**)
- 3) toute propriété doit avoir un sens pour toute occurrence d'entité même si elle est inconnue à un certain moment (**homogénéité**)

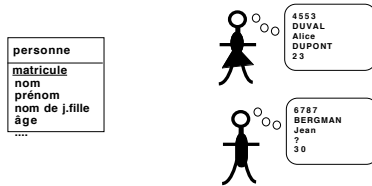
Entité

Types/occurrences :



Combien y a-t-il de livres?

Homogénéité :



Attention, distinguer:

propriétés non significatives, valeur inconnue, valeur nulle,...

Relation

Relation = classe d'information définies par rapport à un ensemble d'entités



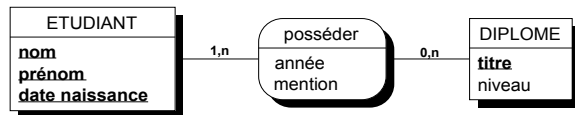
Conseil : les nommer par un verbe à l'**infinitif** (ACHETER)

Règles :

- les relations n'ont **pas d'existence propre** :
la relation "ACHETER" **n'existe que** si des **occurrences** d'entités "PERSONNE" et "VOITURE" existent
- elles peuvent avoir des **propriétés propres**: *date achat*
- elles sont **identifiées par les identifiants de leur collection** :
 - collection**: {entités participant à la relation}
 - dimension**: nb d'entité(s) de la collection

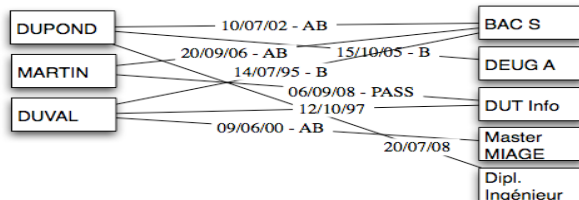
Relation

Exemple :



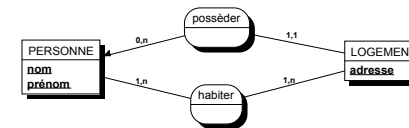
- collection: {ETUDIANT, DIPLOME}
- dimension: 2
- identifiants: (nom) x (titre)

Un graphe des occurrences possible :

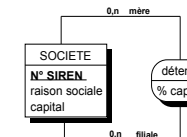


Variété des relations

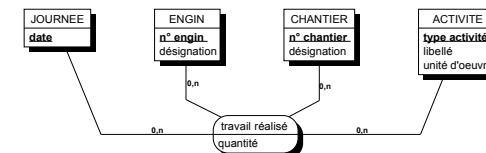
Partage d'une même collection :



Relation sur une même entité :



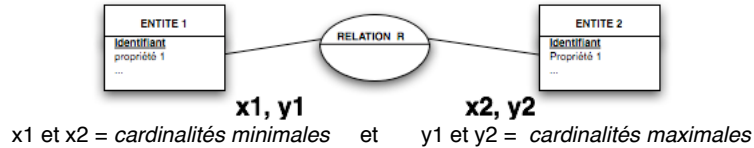
Relation de dimension quelconque :



Cardinalité (ou multiplicité) d'une entité dans une relation (1)

Cardinalité ou multiplicité : caractérise le rôle d'une entité dans une relation

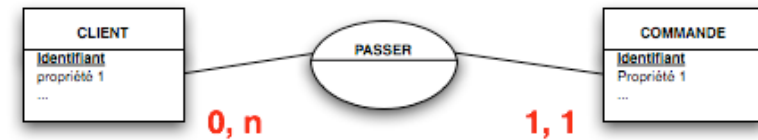
- permet d'enrichir le modèle (niveau des types) en connaissances du niveau des occurrences



Cardinalités fréquemment utilisées :

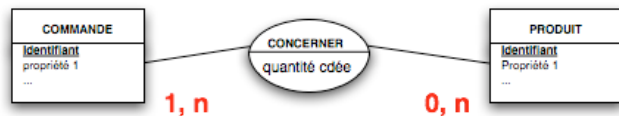
| Participation | Optionnelle | Obligatoire |
|---------------|-------------|-------------|
| Unique | 0,1 | 1,1 |
| Multiple | 0,n | 1,n |

Cardinalité d'une entité dans une relation (1)



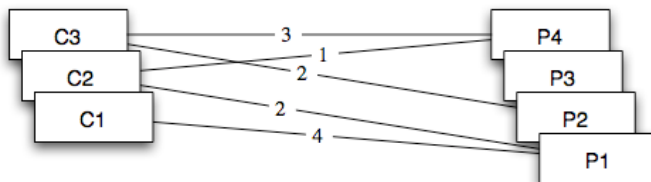
- **0, n** : une occurrence de CLIENT est en relation par la relation PASSER avec une ou plusieurs occurrences de COMMANDE
- **1, 1** : une occurrence de COMMANDE est en relation par la relation PASSER avec une et une seule occurrence de CLIENT

Cardinalité d'une entité dans une relation (2)



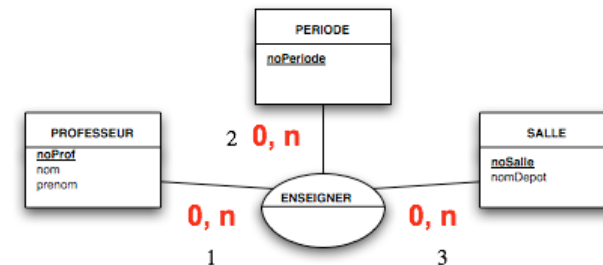
- **1, n** : une occurrence de COMMANDE est en relation par la relation CONCERNER avec une ou plusieurs occurrences de PRODUIT
- **0, n** : une occurrence de PRODUIT est en relation par la relation CONCERNER avec aucune ou plusieurs occurrences de COMMANDE

un graphe des occurrences possible :



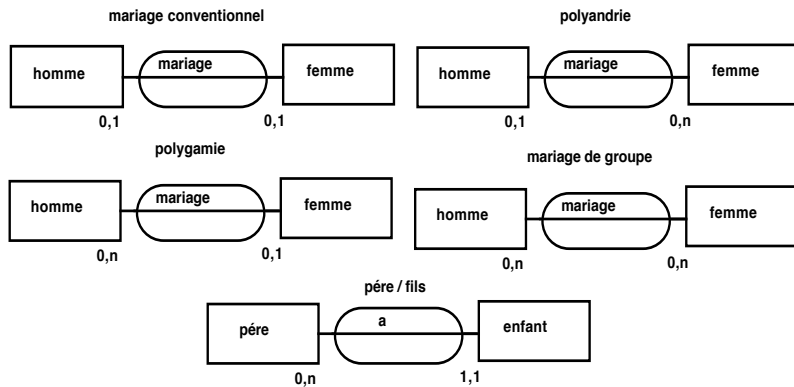
Cardinalité d'une entité dans une relation (3)

Cardinalités sur relation n-aire



- 1) **(0, n)** : une occurrence de PROFESSEUR est en relation par la relation ENSEIGNER avec aucun ou plusieurs couples d'occurrences (SALLE, PERIODE)
- 2) **(0, n)** : une occurrence de PERIODE est en relation par la relation ENSEIGNER avec aucun ou plusieurs couples d'occurrences (PERIODE, SALLE)
- 3) **(0, n)** : une occurrence de SALLE est en relation par la relation ENSEIGNER avec aucun ou plusieurs couples d'occurrences (PROFESSEUR, PERIODE)

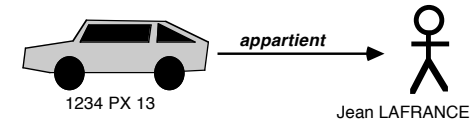
Modèles génériques de cardinalités individuelles



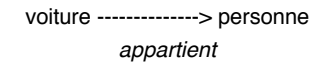
Dépendances fonctionnelles

Contraintes d'intégrité fonctionnelle

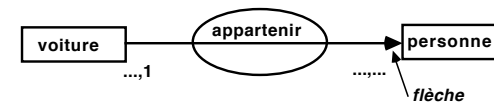
Relation binaire fonctionnelle :



soit:



Représentation :



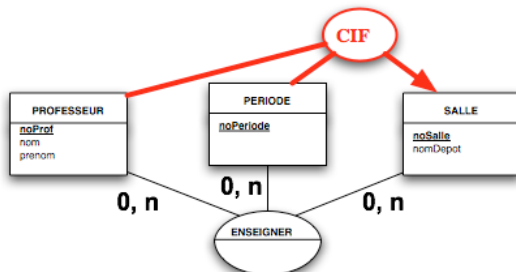
Dépendances fonctionnelles sur relations n-aire

Soit la contrainte à exprimer :

« pour toute **période** d'emploi du temps, (le mercredi de 9 h à 12 h), un **professeur** n'a **cours** que dans une seule **salle** »

soit: PERIODE x PROFESSEUR -----ENSEIGNER-----> SALLE

Limitation du formalisme Entité-Relation de base => besoin d'extension



Types de dépendances fonctionnelles :

- simples (à un émetteur: a --> b)
- composées (à n émetteurs: a x b --> c) englobant pas la totalité de la collection
- composées (à n émetteurs: a x b --> c) englobant la totalité de la collection

3. Dérivation d'un MLD-R à partir d'un MCD en Entité-Relation

- Problématique du MLD
- Formalisme graphique de Merise
- Dérivation d'un MLD-R à partir d'un MCD en Entité-Relation
- Création de tables en langage SQL (clé primaires et étrangères)

Problématique du MLD

Modèle Conceptuel de Données (MCD) :

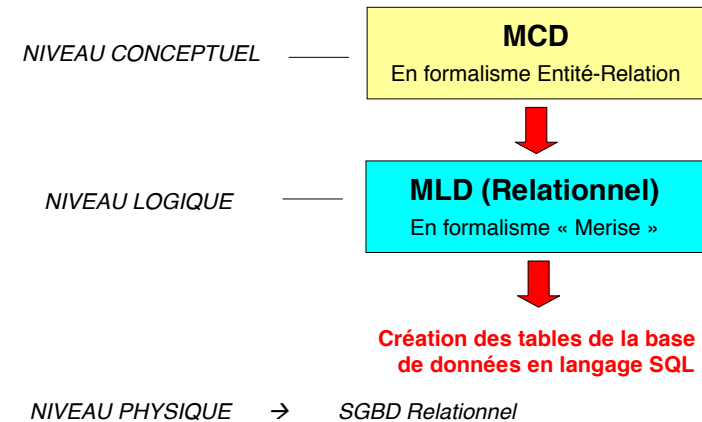
- permet de modéliser la **sémantique des informations** d'une façon **compréhensible par l'utilisateur** de la future base de données
- utilise le formalisme (graphique) **Entité-Relation**
- **ne permet pas d'implémentation informatique de la base de données** dans un SGBD donné

Modèle Logique de Données (MLD) :

- permet de modéliser la **structure selon laquelle les données seront stockées** dans la future base de données
- est **adapté à une famille de SGBD** : ici les **SGBD relationnels** (MLD Relationnels ou **MLD-R**)
- utilise le formalisme graphique **Merise**
- permet **d'implémenter** la base de données dans un **SGBD donné**

Démarche d'élaboration d'un MLD Relationnel

- **MCD** : *Modèle Conceptuel de Données*
- **MLD-R** : *Modèle Logique de Données Relationnel*



Formalisme graphique "Merise" pour le MLD-R (1)

Table

Table EMPLOYE (représentation graphique) :

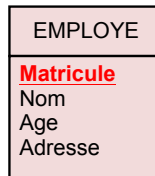


Schéma de la table EMPLOYE :

EMPLOYE (Matricule, nom, age, adresse)

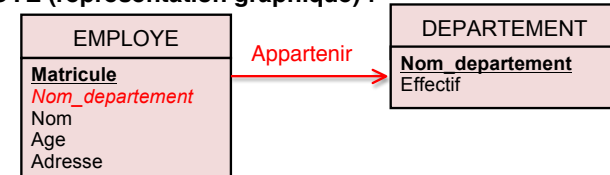
Attributs de la table EMPLOYE :

- Matricule : clé primaire
- Nom, Age, Adresse : autres attributs

Formalisme graphique "Merise" pour le MLDR (2)

Lien entre tables : contrainte d'intégrité référentielle

Table EMPLOYE (représentation graphique) :

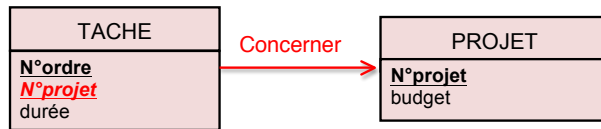


Schémas des tables :

- Table DEPARTEMENT (Nom_departement, Effectif) :
 - Nom_departement : clé primaire
- Table EMPLOYE (Matricule, *Nom_departement*, Nom, Age, Adresse)
 - Matricule : clé primaire
 - *Nom_departement* : **clé étrangère** vers table DEPARTEMENT

Formalisme graphique "Merise" du MLD-R (3)

Lien entre tables : clé primaire composée référentielle

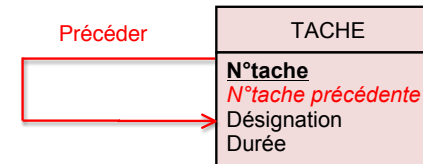


Schémas des tables :

- Table PROJET (N°projet, budget)
 - N°projet : clé primaire
- Table TACHE (N°ordre, N°projet, durée)
 - N°ordre, N°projet : **clé primaire composée**
 - N°projet : **clé étrangère** vers table PROJET

Formalisme graphique "Merise" du MLD-R (4)

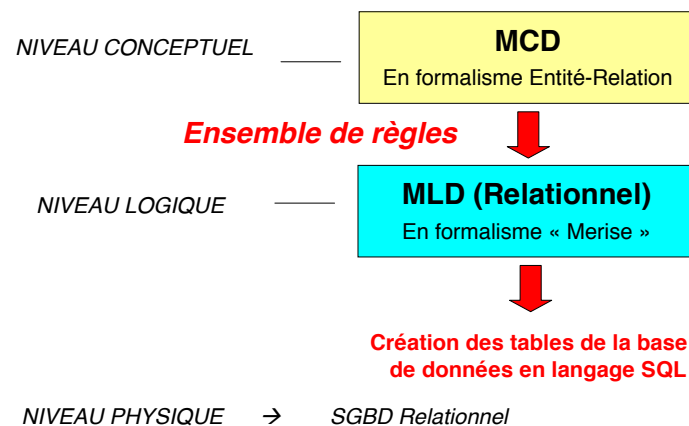
Lien entre tables : contraintes d'intégrité référentielle réflexive



Schémas des tables :

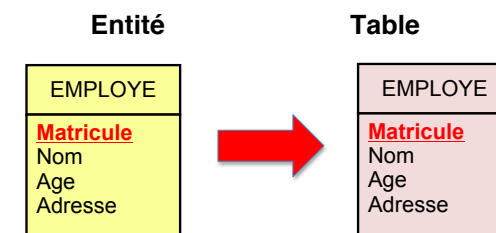
- Table TACHE (n°tache, n°tache_précédente, désignation, durée)
 - N°tache : **clé primaire**
 - N°tache_précédente : **clé étrangère** vers table TACHE

Dérivation d'un MLD-R à partir d'un MCD en Entité-Relation



MCD -> MLD : dérivation des entités

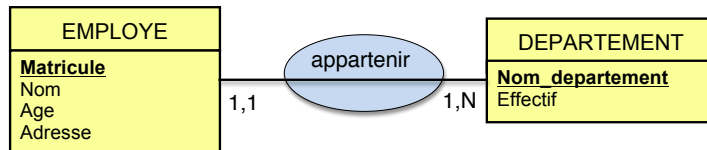
Règle : toute entité du MCD se dérive en une table du MLD



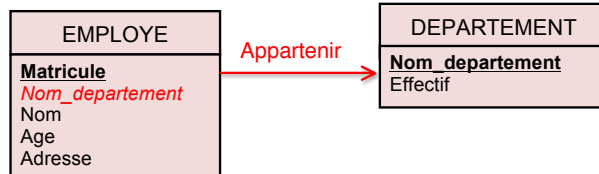
la **propriété identifiante** de l'entité devient la **clé primaire** de la table

MCD -> MLD : relations (*,N)-(1,1)

MCD :



MLD :

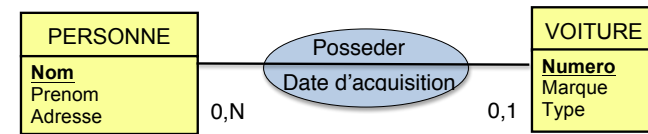


Schémas relationnels :

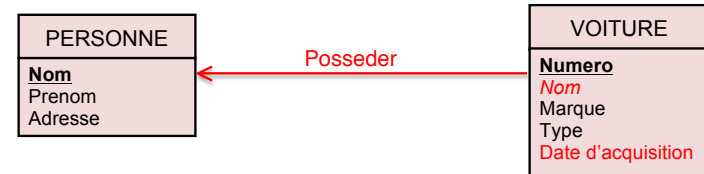
- Table DEPARTEMENT (Nom_departement, Effectif) ;
 - Table EMPLOYE (Matricule, *Nom_departement*, Nom, Age, Adresse)
- Nom_departement* : **clé étrangère** vers table DEPARTEMENT

MCD -> MLD : relations (*,N)-(0,1)

MCD :



MLD :

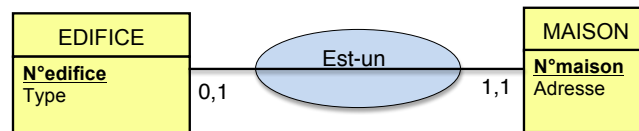


Schémas relationnels :

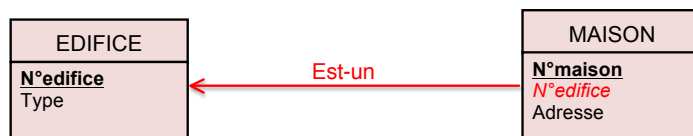
- PERSONNE (Nom, Prenom, Adresse) ;
- VOITURE (Numéro, *Nom*, Marque, Type, *Date_acquisition*) ;

MCD -> MLD : relations (0,1)-(1,1)

MCD :



MLD :

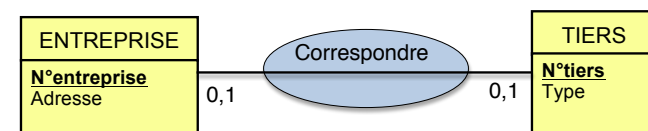


Schémas relationnels :

- EDIFICE (N°edifice, Type) ;
- MAISON (N°maison, *N°édifice*, Adresse).

MCD -> MLD : relations (0,1)-(0,1)

MCD :



MLD (solution 1) :



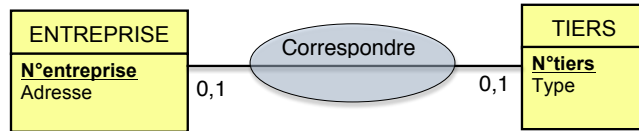
Schémas relationnels :

- ENTREPRISE (N°entreprise, Adresse) ;
- TIERS (N°tiers, *N°entreprise*, Adresse)

La cardinalité (0,1) pose le problème d'accepter des valeurs nulles sur l'attribut migrant pouvant fixer le sens de migration (par exemple la taille des clés).

MCD -> MLD : relations (0,1)-(0,1)

MCD :



MLD (solution 2) :



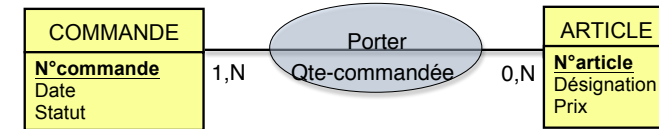
Schémas relationnels :

- ENTREPRISE (**N°entreprise**, *N°tiers*, Adresse) ;
- TIERS (**N°tiers**, Adresse)

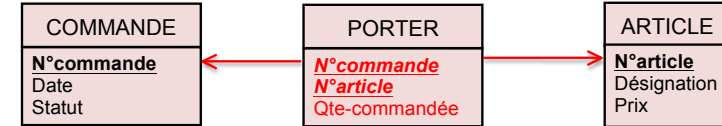
Idem : la cardinalité (0,1) pose le problème d'accepter des valeurs nulles sur l'attribut migrant pouvant fixer le sens de migration (par exemple, la taille des clés).

MCD -> MLD : relations (0/1,N)-(0/1,N)

MCD :



MLD :

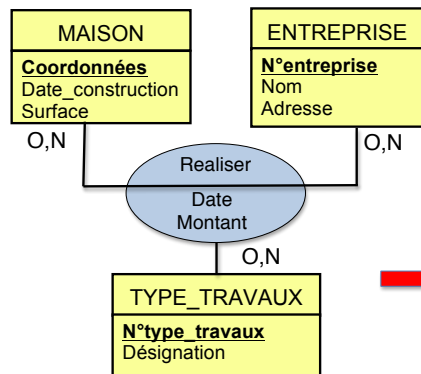


Schémas relationnels :

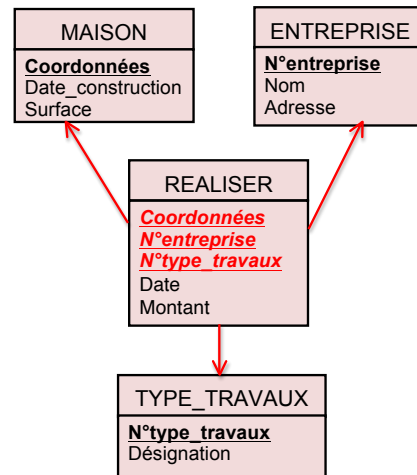
- COMMANDE (**N°commande**, Date, Statut) ;
- PORTER (*N°article*, *N°commande*, Qte_commandée) ;
- ARTICLE (**N°article**, Désignation, Prix).

MCD -> MLD : relations ternaires ou plus (1)

MCD



MLD



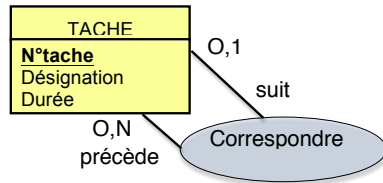
MCD -> MLD : relations ternaires ou plus (2)

Schémas relationnels associés :

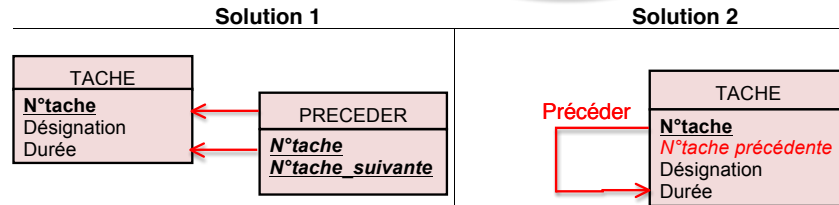
- MAISON (**Coordonnées**, Date_construction, Surface) ;
- TYPE_TRAVAUX (**N°type_travaux**, Désignation) ;
- RÉALISER (*N°entreprise*, *Coordonnées*, *N°type_travaux*, Date, Montant) ;
- ENTREPRISE (**N°entreprise**, Nom, Adresse).

MCD -> MLD : Relations réflexives (0,N)-(0,1)

MCD :



MLD :



Schémas relationnels :

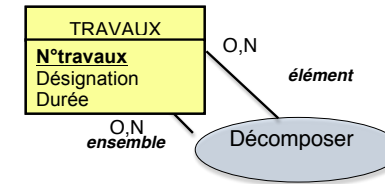
Solution 1 :

- TACHE (N°tache, Désignation, Durée) ;
- PRÉCÉDER (N°tache, N°tache_suivante)

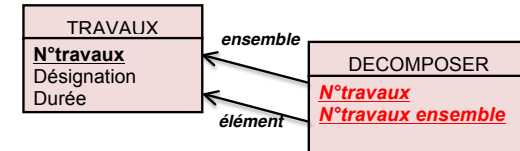
Solution 2 : TACHE (N°tache, N°tache_précédente, Désignation, Durée)

MCD -> MLD : Relations réflexives (*,N)-(*,N)

Entité-Relation :



Relationnel dérivé :

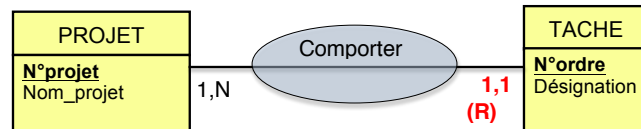


Schémas relationnels :

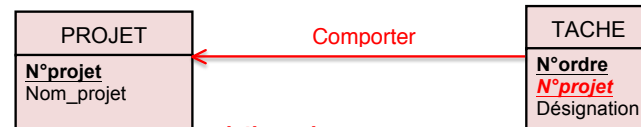
- TRAVAUX (n°travaux, désignation, durée) ;
- DÉCOMPOSER (n°travaux, n°travaux_ensemble).

Règle de dérivation : Identifiant relatif

MCD :



MLD :

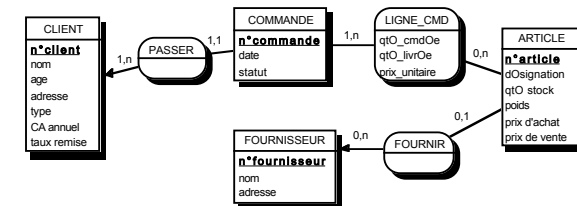


Schémas relationnels :

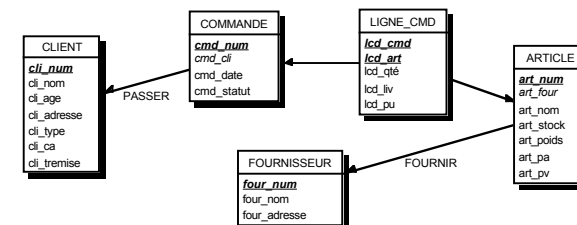
- PROJET (N°projet, Nom_projet)
- TRANCHE (N°ordre, N°projet, Désignation)

Exemple de passage E-R au Relationnel

Entité-Relation :

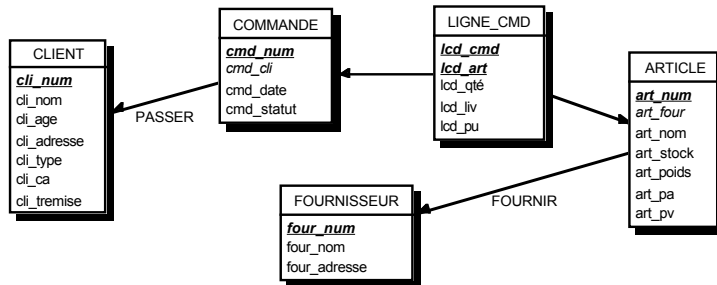


Relationnel dérivé :



Exemple de passage E-R au relationnel

• Modèle relationnel dérivé :



• Schémas relationnels :

- CLIENT (cli_num, cli_nom, cli_age, cli_adresse, cli_type, cli_ca, cli_tremise)
- ARTICLE (art_num, art_nom, art_four, art_stock, art_poids, art_pa, art_pv)
- COMMANDE (cmd_num, cmd_cli, cmd_date, cmd_statut)
- LIGNE_CMD (lcd_cmd, lcd_art, lcd_qté, lcd_liv, lcd_pu)
- FOURNISSEUR (four_num, four_nom, four_adresse)

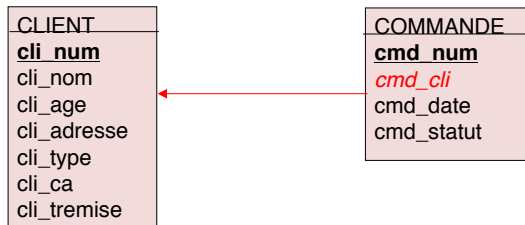
Introduction au langage SQL

- **Origine** : SQL (Structured Query Language) est un langage de requêtes standard pour les SGBD relationnels
- **3 niveaux de normes** :
 - SQL86 (standard ANSI en 86 puis ISO en 87) : la base puis SQL89 ou SQL1 : l'intégrité:
 - SQL91 ou SQL2
 - SQL3 (98) : SQL devient un langage de programmation et évolue vers l'objet

Dans ce chapitre nous ne considérerons que la création BASIQUE de tables par la commande CREATE TABLE de SQL2,

Création d'une table en langage SQL (1)

Soit le MLD suivant :



Schémas relationnels :

Table CLIENT (cli_num, cli_nom, cli_age, cli_adresse, cli_type, cli_ca, cli_tremise)

- cli_num = clé primaire

Table COMMANDE (cmd_num, cmd_cli, cmd_date, cmd_statut)

- cmd_num = clé primaire
- cmd_cli, = clé étrangère

Création d'une table (2) : clé primaire

Table CLIENT (cli_num, cli_nom, cli_age, cli_adresse, cli_type, cli_ca, cli_tremise)

```
CREATE TABLE Client
(cli_num CHAR(8) NOT NULL,
 cli_nom CHAR(25) NOT NULL,
 cli_age INTEGER NOT NULL,
 cli_adresse VARCHAR(80),
 cli_type VARCHAR(16),
 cli_ca INTEGER,
 cli_tremise INTEGER NOT NULL,
 PRIMARY KEY (cli_num)) ;
```

NOT NULL : on n'accepte pas que l'attribut puisse avoir une valeur nulle (valeur inconnue)

PRIMARY KEY (cli_num): l'attribut cli_num est **clé primaire** de la table Client.

Remarque : un attribut déclaré clé primaire doit être défini avec l'option NOT NULL

Création d'une table (3) : clé étrangère

Table **COMMANDE** (**cmd_num**, **cmd_cli**, cmd_date, cmd_statut)

Création de la table **COMMANDE** :

```
CREATE TABLE Commande
(cmd_num CHAR(8) NOT NULL,
cmd_cli CHAR(8) NOT NULL,
cmd_date DATE NOT NULL,
cmd_statut VARCHAR(16),
PRIMARY KEY (cmd_num),
FOREIGN KEY (cmd_cli) REFERENCES client);
```

PRIMARY KEY (cmd_num): l'attribut cmd_num est clé primaire de la table Commande.

FOREIGN KEY (cmd_cli) **REFERENCES** client: l'attribut cmd_cli est une clé étrangère qui réfère à la table Client

Remarques :

- 1- la table client doit déjà exister
- 2- l'attribut cmd_cli de la table commande est du même type que celui de la clé primaire de la table client.

Création d'une table (4) : clé étrangère

Table **ARTICLE** (**art_num**, art_nom, art_four, art_stock, art_poids, art_pa, art_pv) ;

• Création de la table **ARTICLE** :

```
CREATE TABLE Article
(art_num CHAR(8) NOT NULL,
art_nom VARCHAR(25) NOT NULL,
art_four CHAR(8) NOT NULL,
art_stock INTEGER NOT NULL,
art_poids NUMERIC (8,1),
art_pa INTEGER NOT NULL,
art_pv INTEGER NOT NULL,
PRIMARY KEY (art_num),
FOREIGN KEY (art_four) REFERENCES Fournisseur);
```

PRIMARY KEY (art_num): l'attribut art_num est clé primaire de la table article.

FOREIGN KEY (art_four) **REFERENCES** Fournisseur: l'attribut art_cli est une clé étrangère qui réfère à la table Fournisseur.

Création d'une table (5) : clé primaire composée

Table **LIGNE_CMD** (**lcd_cmd**, **lcd_art**, lcd_qte, lcd_liv, lcd_pu)

```
CREATE TABLE Ligne_cmd
(lcd_art CHAR(8) NOT NULL,
lcd_cmd INTEGER NOT NULL,
lcd_qte INTEGER NOT NULL,
lcd_liv INTEGER,
lcd_pu INTEGER NOT NULL,
FOREIGN KEY (lcd_cmd) REFERENCES Commande,
FOREIGN KEY (lcd_art) REFERENCES Article,
PRIMARY KEY (lcd_cmd, lcd_art)) ; <- Clé primaire composée
```

FOREIGN KEY (lcd_cmd) **REFERENCES** commande: l'attribut lcd_cmd est une clé étrangère qui réfère à la table Commande.

FOREIGN KEY (lcd_art) **REFERENCES** commande: l'attribut lcd_art est une clé étrangère qui réfère à la table Article.

PRIMARY KEY (lcd_cmd, lcd_art): la clé primaire de la table Ligne_cmd est composée des attributs lcd_cmd et lcd_art qui sont ici clés étrangères.

Insertion d'enregistrement dans une table en SQL

Table **CLIENT** (**cli_num**, cli_nom, cli_age, cli_adresse, cli_type, cli_ca, cli_tremise)

```
CREATE TABLE Client
(cli_num CHAR(8) NOT NULL,
cli_nom CHAR(25) NOT NULL,
cli_age INTEGER NOT NULL,
cli_adresse VARCHAR(80),
cli_type VARCHAR(16),
cli_ca INTEGER,
cli_tremise INTEGER NOT NULL,
PRIMARY KEY (cli_num)) ;
```

Insertion d'un nouveau enregistrement dans la table Client :

```
INSERT INTO client VALUES ('C2345', 'Tranvouez', 29,
'Marseille', 'particulier', 3680, 25) ;
```

Suppression d'enregistrement dans une table

Table CLIENT (cli_num, cli_nom, cli_age, cli_adresse, cli_type, cli_ca, cli_tremise) ;

```
CREATE TABLE Client
(cli_num CHAR(8) NOT NULL,
 cli_nom CHAR(25) NOT NULL,
 cli_age INTEGER NOT NULL,
 cli_adresse VARCHAR(80),
 cli_type VARCHAR(16),
 cli_ca INTEGER,
 cli_tremise INTEGER NOT NULL,
 PRIMARY KEY (cli_num)) ;
```

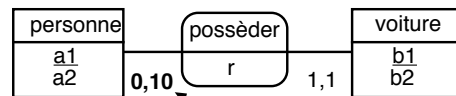
Suppression d'un enregistrement dans la table client :

```
DELETE FROM client WHERE (cli_nom = 'Tranvouez');
```

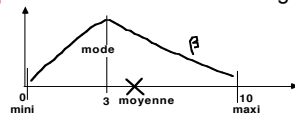
4. Dimensionnement d'une BD Relationnelle

- Niveau « Réel perçu » et « Niveau Interne ou Logique »
- Niveau « Conceptuel » et « Externe »

Multiplicités moyennes des liens relationnels



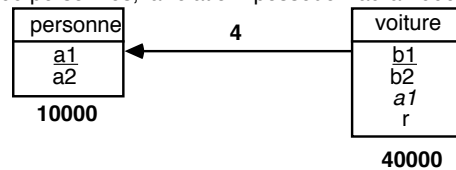
supposons une distribution triangulaire:



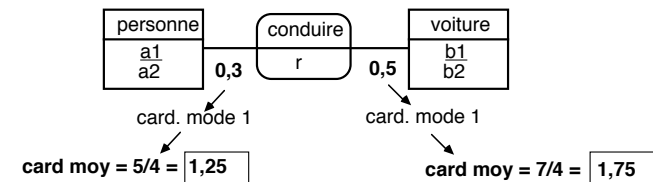
selon la distribution, ici une loi triangulaire d'où moy = (mini+2(mod)+max)/4

d'où **cardinalité moyenne** = (0+2(3)+10)/4 = 4

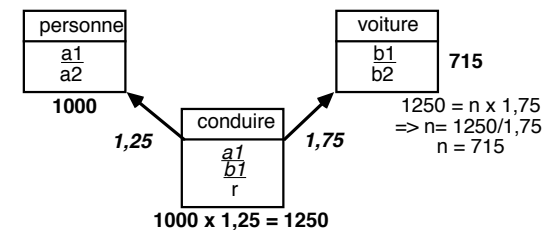
si 10000 personnes, la relation "posséder" aura 40000 tuples



Propagation des multiplicités moyenne



d'où:



nb
person
ne x
1,25 =
nb
voiture
x 1,75
soit:

Calculs cumulés des volumes des tables

- Il s'agit de faire sur un tableau, par exemple sous Excel, le cumul du volume estimé de toutes les tables composant la base de données
- Ce tableau peut être paramétré
- Il permet d'estimer l'évolution du volume de la base de données selon différents scénarii de croissance
- ...

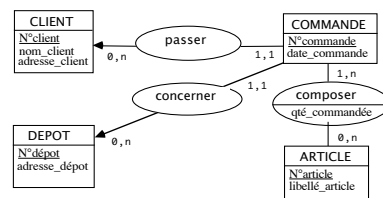
5. Sous schéma de données et valorisation de l'activité des traitements sur la BD

- Transformation de sous-schémas conceptuels (SSC) en sous-schémas logiques (SSL)
- Valorisation des primitives logiques

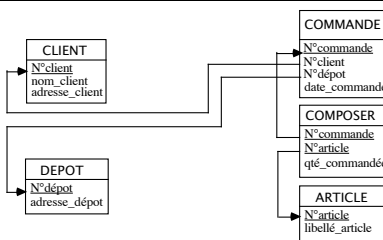
Valorisation de l'activité des traitements sur la BD (1)

1. Transformation de sous-schémas conceptuels (SSC) en sous-schémas logiques (SSL)

Sous-Schéma Conceptuel (SSC)



Sous-Schéma Logique (SSL) associé



Valorisation de l'activité des traitements sur la BD (2)

2. Fréquence d'une tâche associée à un modèle externe

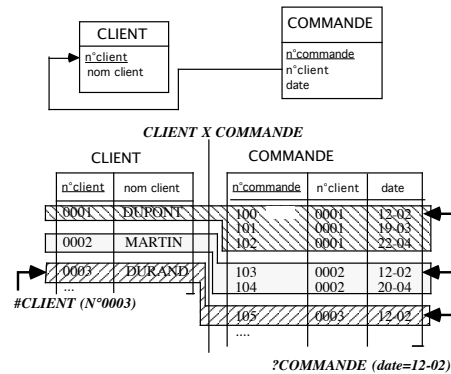
- fréquence du message associé au modèle externe exprimée par rapport à la période de référence (généralement le mois moyen)
- basée sur la fréquence de survenance des événements initiateurs des procédures où figure cette tâche

3. Transformation des actions en primitives

- **#T** : Accès à un tuple d'une table relationnelle par accès sur clé primaire, connaissant une valeur de cette clé
- **?T** : Accès à un tuple d'une table relationnelle par accès sur un autre attribut que la clé primaire, ou sélection/restriction de tuples d'une table selon une qualification
- **T+** : Ajout d'un tuple à une table relationnelle
- **T-** : Suppression d'un tuple à une table relationnelle. Cette primitive est notée T- nom de la table.
- **Tm** : Modification d'un tuple d'une table relationnelle, par la modification de valeurs d'un ou plusieurs attributs de ce tuple
- **TXT'** : Jointure entre deux tables relationnelles T et T', par la modification de valeurs d'un ou plusieurs attributs de ce tuple. Cette primitive est notée TXT' noms des tables concernées .

Valorisation de l'activité des traitements sur la BD (3)

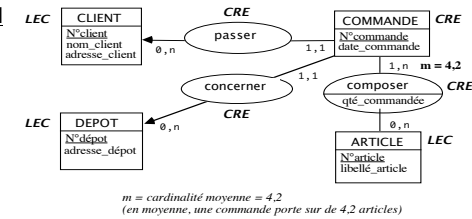
Illustration des primitives relationnelles :



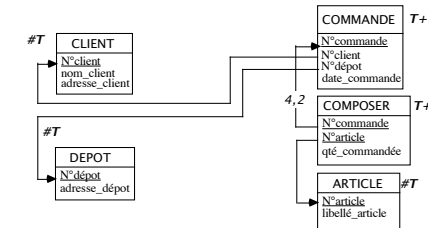
Valorisation de l'activité des traitements sur la BD (4)

• Actions associées à la prise en compte d'une nouvelle occurrence de COMMANDE:

Sous-Schéma Conceptuel (SSC)



Sous-Schéma Logique (SSL) associé



Valorisation de l'activité des traitements sur la BD (4)

4. Récapitulatif de l'activité valorisée par primitive

- pour chaque modèle externe : traduction des actions, en primitives
- effectuer ce calcul d'abord pour l'activité unitaire de la tâche, puis de la multiplier par la fréquence de la tâche sur la période de référence.
- cumul par primitive de l'activité supportée par le modèle (total par ligne). Du fait de l'hétérogénéité des primitives en termes d'équivalent-accès, seul le cumul par primitive a une signification :

| primitives | activité unitaire | activité mensuelle |
|-------------|-------------------|--------------------|
| #T CLIENT | 1 | 10 000 |
| #T DEPOT | 1 | 10 000 |
| T+ COMMANDE | 1 | 10 000 |
| T# ARTICLE | 1x4,2 | 42 000 |
| T+ COMPOSER | 1x4,2 | 42 000 |

→ repérer les primitives à forte activité, et retrouver les modèles externes responsables

• en pratique:

en l'absence d'outil logiciel prenant en charge cette valorisation fortement algorithmique, pratiquer ces calculs que sur un échantillon d'une vingtaine de modèles externes remarquables par leur fréquence, d'aspect critique (comme dans d'autres domaines la loi des 80-20 se constate également ici).

Valorisation de l'activité des traitements sur la BD (5)

5. Poids relatif des primitives en accès logiques

Définir des équivalents-accès-logiques = poids relatifs entre les différentes primitives.

Conventions pour la valorisation des primitives logiques

- 1- unité d'accès logique = #T, toutes les autres seront exprimées en fonction de #T.
- 2- le nombre de tuples occupées par une table T sera noté **NbT(T)**
- 3- le nombre de pages occupées par une table T sera noté **NbP(T)**

avec :

- $NbP(T) = NbT(T) / \text{nombre moyen de tuples dans la page}$
- le nombre moyen de tuples dans la page dépendra :
 - de la taille de la page, noté P
 - de la taille du tuple, noté T
 - du taux de remplissage de la page
- ces valeurs seront liées au SGBD et à l'organisation de la table adoptés.
- on pourra faire des calculs en première approximation avec un taux de remplissage de 80% et des pages de 4K.

Exemple :

$NbT(T) = 10000$; $P = 4K$ octets; taux de remplissage de 80%; $T = 200$ octets;
 Nombre moyen de tuples par page = $4000 \times 0,8 / 200 = 16$
 $NbP(T) = 10000 / 16 = 625$ pages.

Valorisation des primitives logiques (1)

• valeurs **moyennes** pour SGBDR actuellement commercialisés, une **réflexion nécessaire** sur la pertinence de ces valeurs dans le contexte de la BD

#T = 1 : sélection selon clé primaire = unité d'accès logique

?T = NbP(T) : sélection selon un attribut non clé primaire sans index secondaire : balayage séquentiel

?T = Sx(NbP(T)+NP(I)): si le critère de sélection comprend plusieurs termes dont un faisant intervenir un index secondaire I (*avec S = facteur de sélectivité du critère, cad la probabilité qu'un tuple satisfasse le critère; NbP(T) = nb pages de la table T; NbP(I) = nb pages de l'index I*)

T+ = T- = 4: le tuple est atteint par 1 accès à l'index + 1 accès à la page contenant le tuple cherché + 1 mise à jour d'une page + 1 mise à jour de l'index.

Tm = 3: le tuple est atteint par 1 accès à l'index + 1 accès à la page contenant le tuple cherché + 1 écriture sur page.

TXT' = S(NbP(T) + (N(T)xNbP(T'))) : jointure, *avec NbP(T) = nombre page de la table T, S = facteur de sélectivité du critère sur T, N(T) = nombre de tuples de T satisfaisant le critère, NbAP(T') = nombre page de la table T'*

Valorisation des primitives logiques (2)

Globalement on pourrait retenir :

| primitive | équivalent accès |
|---------------------|----------------------------|
| #T | 1 |
| Tm | 3 |
| T+=T- | 4 |
| ?T | NbP(T) |
| ?T si critère/index | Sx(NbP(T)+NP(I)) |
| TXT' | S(NbP(T) + (N(T)xNbP(T'))) |