

5. CONCEPTION PHYSIQUE RELATIONNELLE

Version 2 - Septembre 2012

Support du chapitre 20, *Conception physique d'une base de données relationnelle*
de l'ouvrage *Bases de données*, J-L Hainaut, Dunod 2018.

5. CONCEPTION PHYSIQUE RELATIONNELLE

Contenu

- 5.1 Introduction**
- 5.2 Détermination des index**
- 5.3 Technologie des index**
- 5.4 Sélection des espaces de stockage**
- 5.5 Optimisation avancée**
- 5.6 Exemple**
- 5.7 Aide au développement d'un schéma physique**

5.1 INTRODUCTION

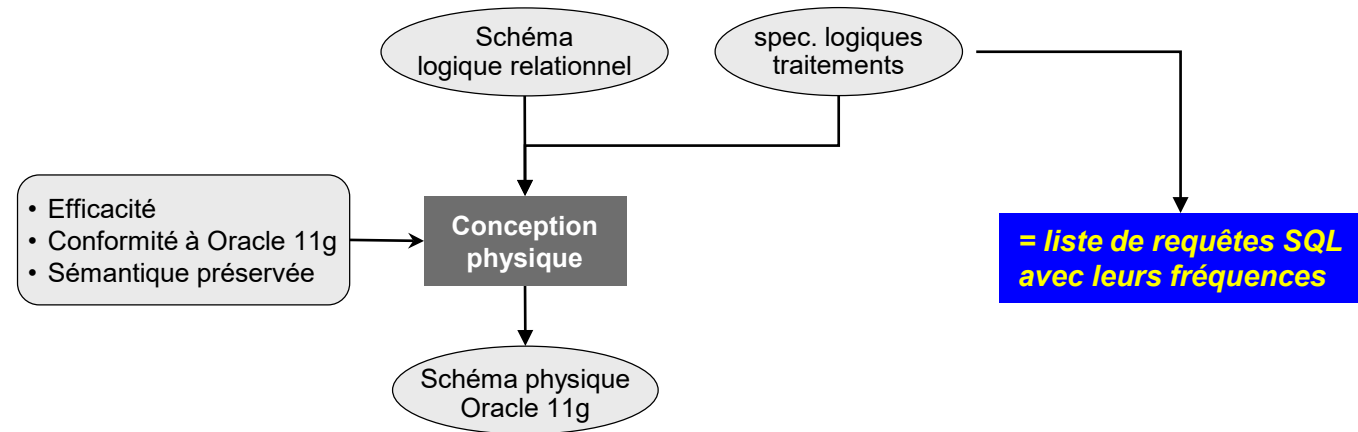
5.1 Introduction

En (très) bref :

La conception physique consiste à rendre un schéma logique conforme à un SGBD particulier et à lui ajouter les constructions physiques lui assurant de bonnes performances.

***Accessoirement* : évaluation a priori des volumes et des principaux temps d'accès**

5.1 Introduction



Mise en conformité physique : selon l'*Administration Guide*

- Constructions physiques :**
- choix des index
 - technologie des index
 - espaces de stockage
 - *selon SGBD* : clusters, clustering index, etc.

5.1 Introduction

Notion de performance d'une base de données

Volumétrie à minimiser

- Espace occupé en mémoire externe : espaces de stockage, tables, index, journaux, etc.
- Espace occupé en mémoire interne : tampons, verrous, etc.

Temps d'exécution à minimiser

- Temps d'accès à la mémoire externe : accès séquentiel, accès via les index, tri, modification, etc.
- Temps de traitement : SE, SGBD, programmes d'application, etc.

Compromis espace/temps

- Minimisation de l'espace \Rightarrow coûts en temps plus élevés
- Minimisation du temps \Rightarrow coûts en espace plus élevés

5.1 Introduction

Notion de performance d'une base de données

Autres facteurs techniques

fréquence et coût unitaire des réorganisations, coût des sauvegardes (volume et temps), mécanismes de protection contre les incidents (sauvegardes, journaux, redondance de données), coût de restauration en cas d'incident, niveau de parallélisme des transactions (gestion des verrous).

Autres facteurs non techniques

évolutivité, coût de la maintenance, intégration de données existantes, coûts d'exploitation, facilité de développement des programmes d'application (et donc coûts et délais), compétences nécessaires, sécurité et contrôle d'accès.

5.1 Introduction

Conception physique : 4 processus

- **Détermination des index**
- **Technologie des index**
- **Sélection des espaces de stockage**
- **Optimisation avancée**
- **(Estimation des volumes et des temps de traitement)**

5.2 Choix des index

6.2 Détermination des index

On identifie 4 types de règles

1. index induits par les structures logiques;
2. index induits par l'utilisation des données;
3. index à écarter
4. maintenance des index

5.2 Choix des index

1. Index induits par les structures logiques

Règle 1

On associe un index à chaque identifiant



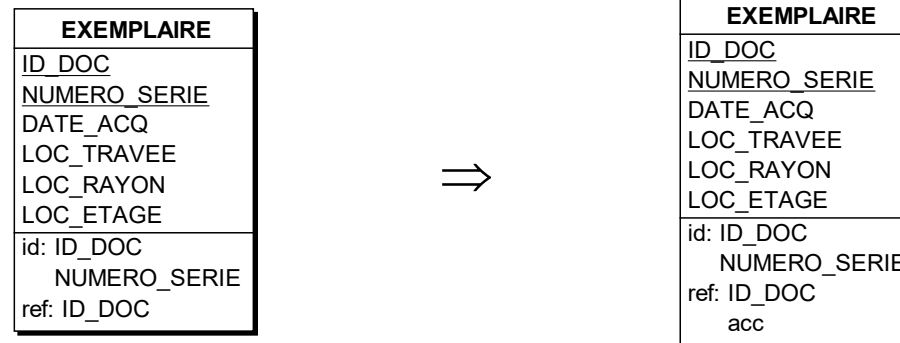
**Même si l'index est inutile pour les performances.
L'intégrité a priorité sur les performances !**

5.2 Choix des index

1. Index induits par les structures logiques

Règle 2

On associe un index à chaque clé étrangère



5.2 Choix des index

2. Index induits par l'utilisation des données

Règle 3

On associe un index à chaque colonne (ou groupe de -) qui fait l'objet de comparaisons fréquentes dans des conditions d'égalité ou d'inclusion (in)

```
select NCLI ,NOM ,ADRESSE  
from CLIENT  
where LOCALITE = :LOC ;
```



CLIENT
<u>NCLI</u>
NOM
ADRESSE
LOCALITE
id: NCLI
acc: LOCALITE

5.2 Choix des index

2. Index induits par l'utilisation des données

Règle 4

On associe un index à chaque colonne (ou groupe de -) servant fréquemment de critère d'intervalle

```
select NCLI,NOM  
from CLIENT  
where CAT like :C||'%' ;
```



CLIENT
<u>NCLI</u>
NOM
ADRESSE
LOCALITE
CAT[0-1]
id: NCLI
acc: CAT

5.2 Choix des index

2. Index induits par l'utilisation des données

Règle 5

On associe un index à chaque colonne servant fréquemment de critère de groupement ou de tri

```
select DATECOM, count(*)  
from   COMMANDE  
group by DATECOM;
```

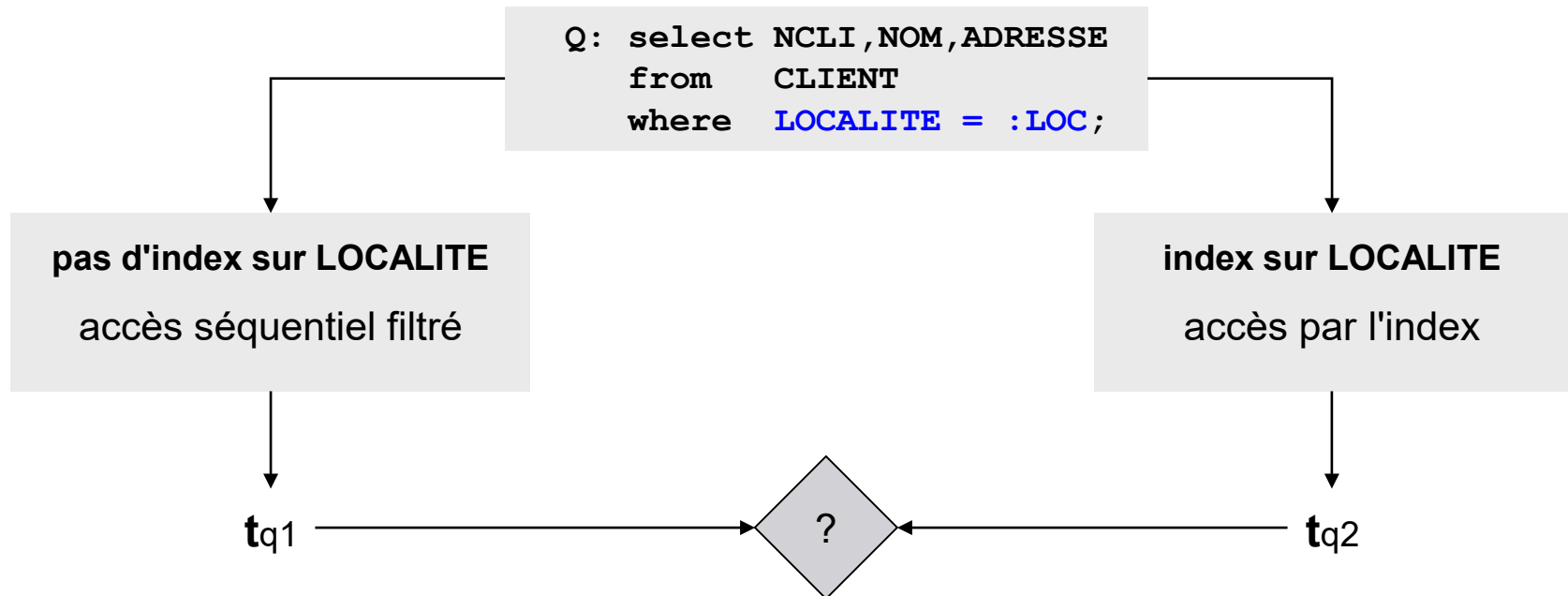


COMMANDE
NCOM
NCLI
DATECOM
acc: DATECOM

5.2 Choix des index

2. Index induits par l'utilisation des données

Utilité d'un index : comment décider ? (rappel Implémentation, section 4.10)



5.2 Choix des index

2. Index induits par l'utilisation des données

Utilité d'un index : comment décider ?

pas d'index : accès séquentiel filtré

$$t_{q1} = N_p \times t_{ls1}$$

5.2 Choix des index

2. Index induits par l'utilisation des données

Utilité d'un index : comment décider ?

index : accès par l'index

- l'index possède N_c valeurs et donc N_c entrées;
- chaque entrée référence $N_{rpc} = \lceil N_r / N_c \rceil$ lignes

1. **index secondaire sur fichier en vrac** (lignes distribuées aléatoirement)

- ces lignes sont stockées dans N_{rpc} pages différentes (si $N_{rpc} \ll N_p$)

$$t_{q2} = N_{rpc} \times t_{la1}$$

2. **index secondaire sur fichier ordonné** (lignes sélectionnées dans pages consécutives)

- ces lignes occupent $n_{pic} = \lfloor N_{rpc} / N_{rpp} + 0,5 \rfloor + 1$ pages

$$t_{q2} = n_{pic} \times t_{ls1}$$

5.2 Choix des index

2. Index induits par l'utilisation des données

Utilité d'un index : comment décider ?

digression : et si N_{rpc} est très grand devant N_r ?

Alors la probabilité qu'une page contienne plus d'une ligne sélectionnée n'est plus négligeable.

- l'index possède N_c valeurs;
- chaque entrée référence $N_{rpc} = \lceil N_r / N_c \rceil$ lignes *sélectionnées*
- chaque page contient $m_c = N_{rpc} / N_p$ lignes *sélectionnées*
- probabilité qu'une page contienne 0 ligne sélectionnée : e^{-m_c} (loi de Poisson)
- probabilité qu'une page contienne au moins une ligne sélectionnée : $1 - e^{-m_c}$

1. **index secondaire sur fichier en vrac** (lignes sélectionnées distribuées aléatoirement)

- ces lignes occupent $q_c = N_p \times (1 - e^{-m_c})$ pages

$$t_{q2} = q_c \times t_{a1}$$

5.2 Choix des index

2. Index induits par l'utilisation des données

Utilité d'un index : comment décider ?

Exemple

```
select NCLI ,NOM,ADRESSE  
from CLIENT  
where LOCALITE = :LOC ;
```

?
⇒

CLIENT
<u>NCLI</u>
NOM
ADRESSE
LOCALITE
id: NCLI
acc: LOCALITE

$N_r = 500.000$
 $N_{loc} = 5.000$
 $N_p = 25.000$

$t_{ls1} = 0,184 \text{ msec}$
 $t_{la1} = 12,300 \text{ msec}$

5.2 Choix des index

2. Index induits par l'utilisation des données

Utilité d'un index : comment décider ?

Exemple : accès séquentiel filtré

$$t_{q1} = N_p \times t_{ls1} = 25.000 \times 0,184 = 4.600 \text{ msec} = \mathbf{4,6 \text{ sec}}$$

5.2 Choix des index

2. Index induits par l'utilisation des données

Utilité d'un index : comment décider ?

Exemple : accès par index sur fichier en vrac (calcul simplifié)

$$N_{rploc} = \lceil N_r / N_{loc} \rceil = 500.000 / 5.000 = 100 \text{ lignes par valeur de LOCALITE}$$

$$t_{q2} = N_{rploc} \times t_{la1} = 100 \times 12,3 = 1.230 \text{ msec} = \mathbf{1,23 \text{ sec}}$$

5.2 Choix des index

2. Index induits par l'utilisation des données

Utilité d'un index : comment décider ?

Exemple : accès par index sur fichier trié

$$N_{rploc} = \lceil N_r / N_{loc} \rceil = 500.000 / 5.000 = 100 \text{ lignes par valeur de LOCALITE}$$

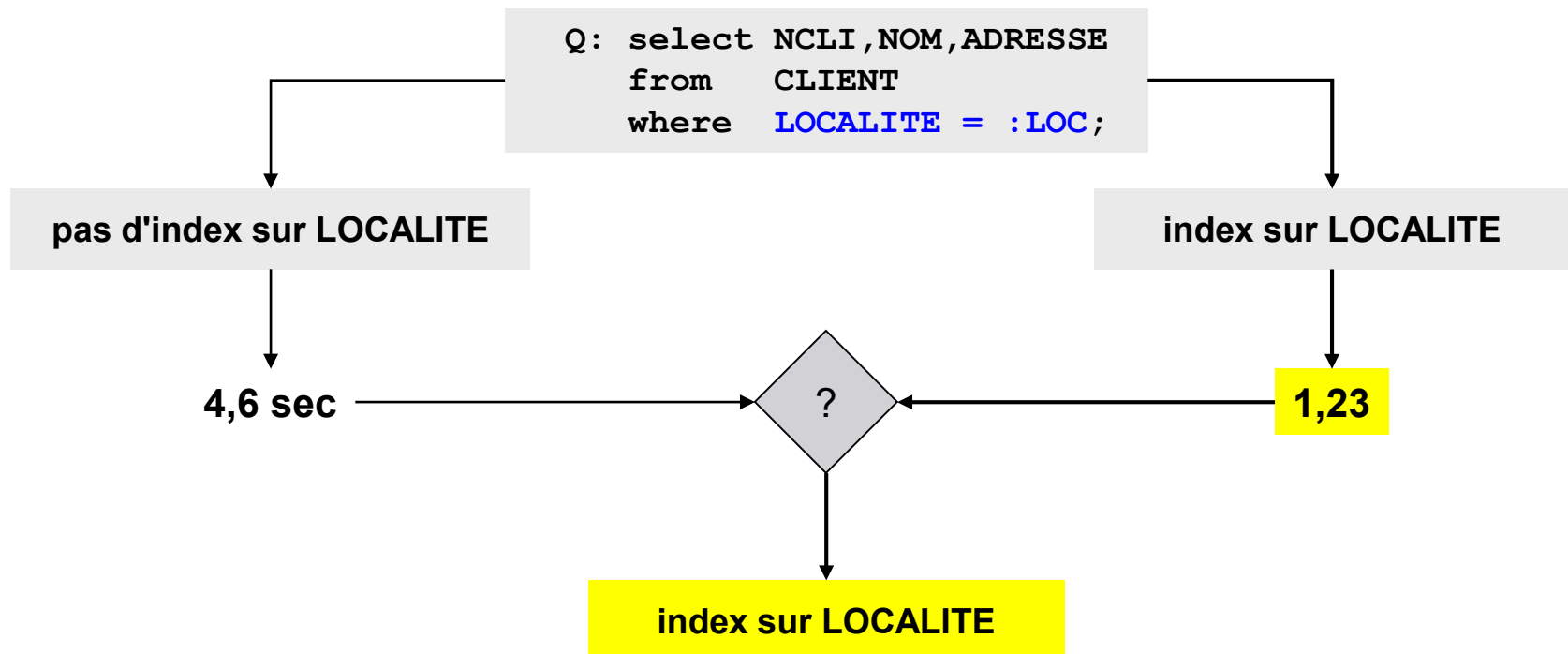
$$n_{piloc} = \lfloor N_{rploc} / N_{rpp} + 0,5 \rfloor + 1 = \lfloor 100 / 16 + 0,5 \rfloor + 1 = 7 \text{ pages contenant les 100 lignes}$$

$$t_{q2} = n_{piloc} \times t_{ls1} = 7 \times 0,184 = 1,288 \text{ msec} = \mathbf{0,0013 \text{ sec}}$$

5.2 Choix des index

2. Index induits par l'utilisation des données

Utilité d'un index : comment décider ?



5.2 Choix des index

3. Index à écarter

Règle 6

Un index est inutile pour les conditions d'**inégalité**, d'**exclusion** ou **masque non préfixe**

```
select NPRO,LIBELLE
from   PRODUIT
where  NPRO not in (select NPRO from DETAIL);
```

ne justifie pas (à lui seul)
un index sur DETAIL.NPRO

```
select NCLI,NOM
from   CLIENT
where  LOCALITE <> :L;
```

ne justifie pas un index
sur CLIENT.LOCALITE

```
select NCLI,NOM
from   CLIENT
where  ADRESSE like '%'||:C||'%' ;
```

ne justifie pas un index
sur CLIENT.ADRESSE

5.2 Choix des index

3. Index à écarter

Règle 7

Un index préfixe d'un autre est inutile lorsque le dictionnaire de valeurs du second est ordonné (= "index trié")

RESERVATION
<u>NUMPERS</u>
<u>ID_VOL</u>
DATERESERVATION
id: ID_VOL
NUMPERS
acc
ref: ID_VOL
acc
ref: NUMPERS
acc



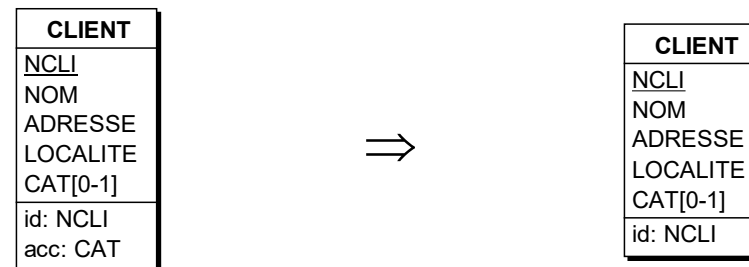
RESERVATION
<u>NUMPERS</u>
<u>ID_VOL</u>
DATERESERVATION
id: ID_VOL
NUMPERS
acc
ref: ID_VOL
ref: NUMPERS
acc

5.2 Choix des index

3. Index à écarter

Règle 8

Un index non identifiant peut être inutile si son domaine de valeurs est très petit et si les lignes de la table ne sont pas ordonnées selon cet index

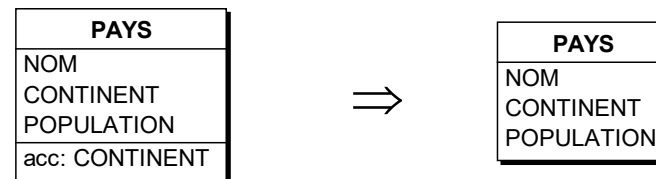


5.2 Choix des index

3. Index à écarter

Règle 9

Un index non identifiant peut être inutile s'il est associé à une table de petite taille



5.2 Choix des index

4. Maintenance d'un jeu d'index

Règle 10

Lors de l'exploitation, on surveillera le taux d'utilisation des index ainsi que les plans d'exécution élaborés par l'optimiseur du SGBD pour les requêtes les plus critiques

5.3 Technologie des index

5.3 Technologie des index

Pour chaque index, choix de la technique d'implémentation :

- index primaire en séquentiel indexé (B-tree)
- index primaire en calculé (hashing)
- index secondaire, dictionnaire en séquentiel indexé
- index secondaire, dictionnaire en calculé
- *autres*

Techniques de clustering

= "index trié"

5.3 Technologie des index

Règle 11

Les techniques basées sur un **dictionnaire de valeurs ordonné** (= "index trié") seront préférées lorsque l'accès par intervalle à de petites séquences d'enregistrements est fréquent.

Lorsqu'en outre l'accès par intervalle risque de produire des séquences relativement longues ou lorsqu'un accès séquentiel ordonné est fréquent, alors on recommandera un **index primaire en séquentiel indexé** ou un **index secondaire sur fichier ordonné** (clustering index)

5.3 Technologie des index

Règle 12

Les techniques basées sur une organisation **séquentielle indexée** (table et/ ou dictionnaire de valeurs) sont recommandées,

1. lorsque le volume de données croît de manière imprévisible,
2. lorsque le temps d'accès par index doit être uniforme pour tous les enregistrements
3. lorsque les réorganisations sont fréquentes.

Dans les autres cas, les **techniques calculées** pourront être sélectionnées.

5.3 Technologie des index

Règle 13

Les techniques basées sur une **organisation calculée** (table ou dictionnaire de valeurs) sont appropriées lorsque

1. le temps **moyen** d'accès par clé doit être très court
2. la place mémoire occupée sur disque et dans le tampon doit être minimale.

Règle 14

Si un index est susceptible d'être supprimé et recréé à tout moment, sans suspension du fonctionnement de la base de données, alors il doit être secondaire.

5.4 Espaces de stockage

5.4 Sélection des espaces de stockage

Règle 15

Deux tables appartenant au même cluster sont assignées au même espace

Règle 16

Dans une grande base de données, s'il est possible d'identifier un ensemble de tables utilisées simultanément mais de manière sporadique, alors on envisagera de les stocker dans un ou plusieurs espaces qui leur sont exclusivement réservés.

5.5 Optimisation avancée

5.5 Optimisation avancée

Selon disponibilité : index clustering, clusters, index augmentés, index bitmap, etc.

En principe, la conception physique ne modifie pas le schéma logique.

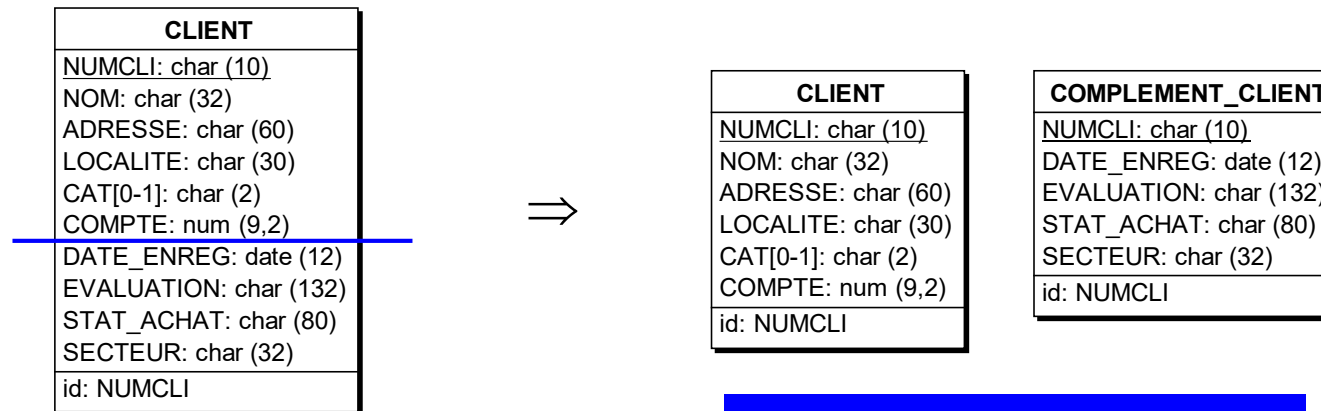
Mais résultats pas toujours satisfaisants : des transformations structurelles du schéma logiques peuvent alors s'avérer nécessaires, ***bien que déconseillées***

- restructurations sans redondance
- redondances structurelles
- dénormalisation

5.5 Optimisation avancée

5.5 Optimisation avancée - Restructurations sans redondance

Partitionnement vertical d'une table

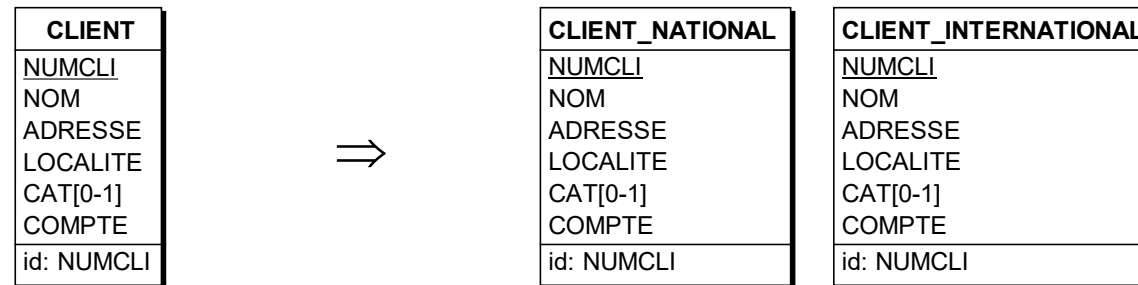


mêmes lignes, colonnes différentes

5.5 Optimisation avancée

5.5 Optimisation avancée - Restructurations sans redondance

Partitionnement horizontal d'une table

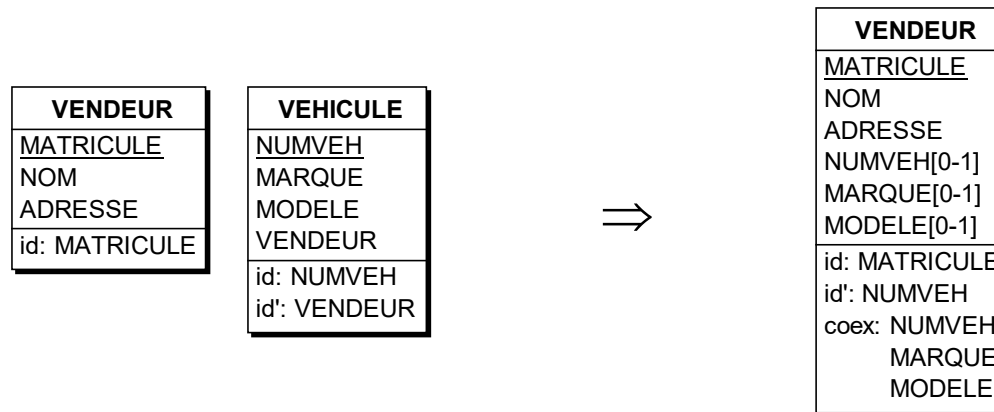


mêmes colonnes, lignes différentes

5.5 Optimisation avancée

5.5 Optimisation avancée - Restructurations sans redondance

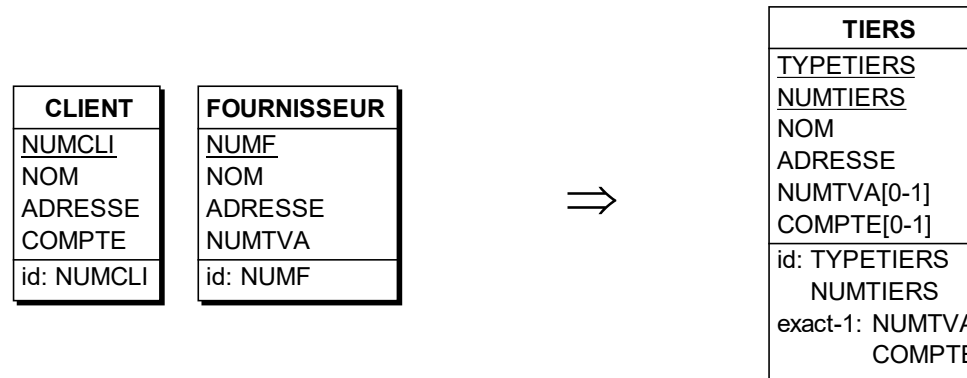
Fusion verticale de tables



5.5 Optimisation avancée

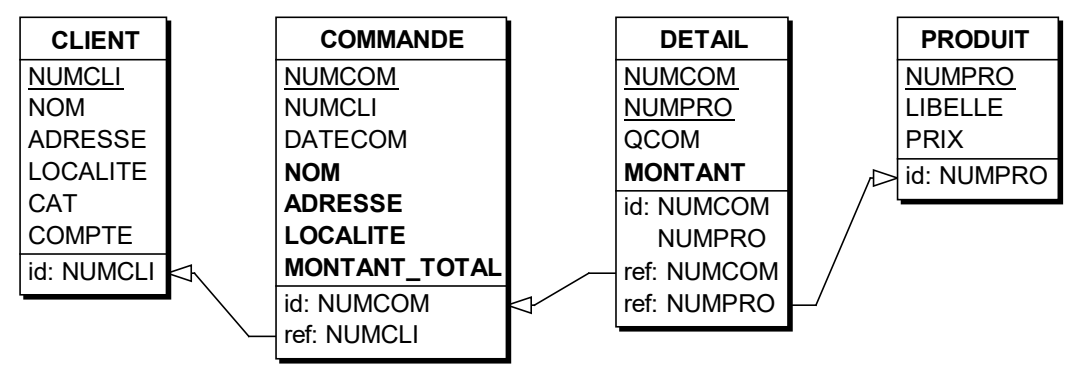
5.5 Optimisation avancée - Restructurations sans redondance

Fusion verticale de tables



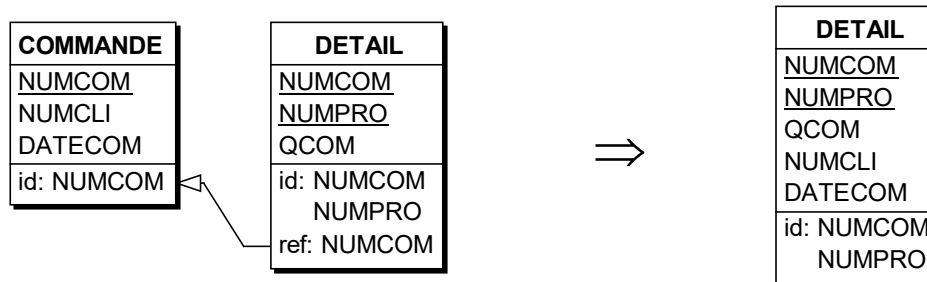
5.5 Optimisation avancée

5.5 Optimisation avancée - Redondance structurelle



5.5 Optimisation avancée

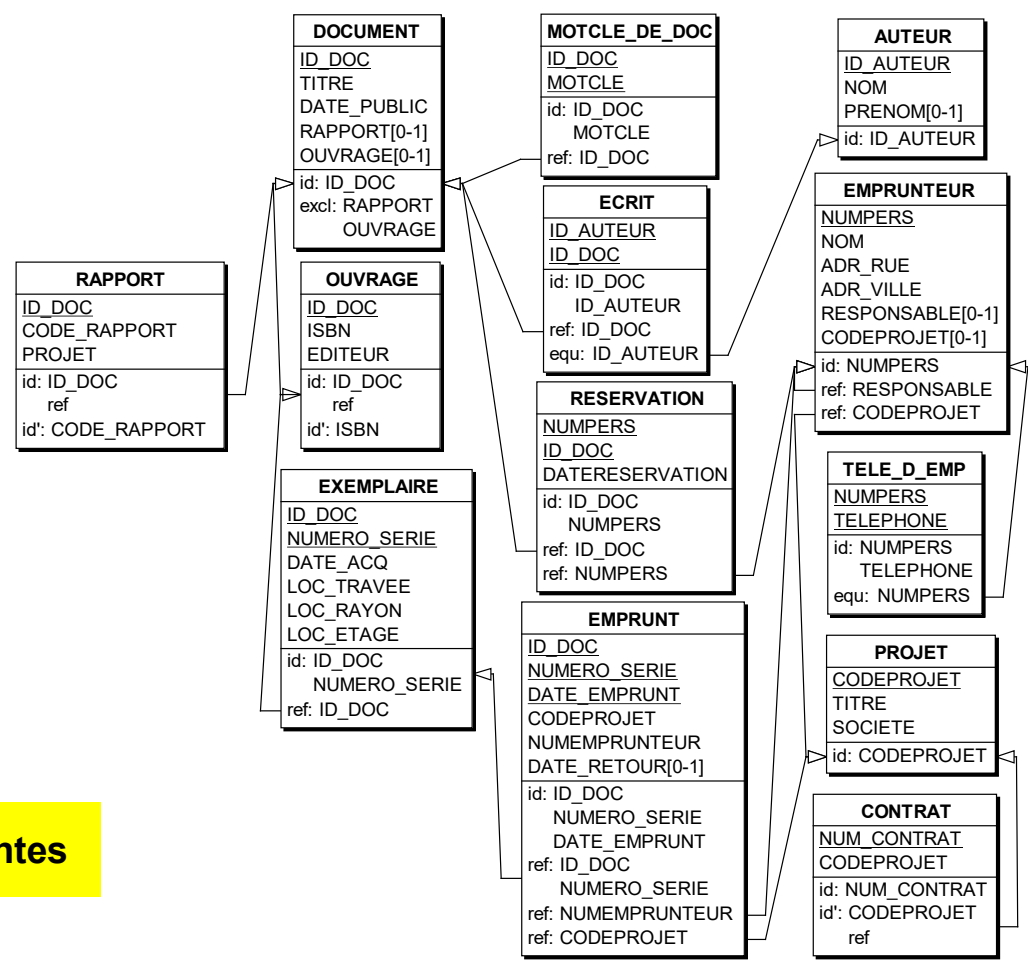
5.5 Optimisation avancée - Redondance interne



5.6 Exemple

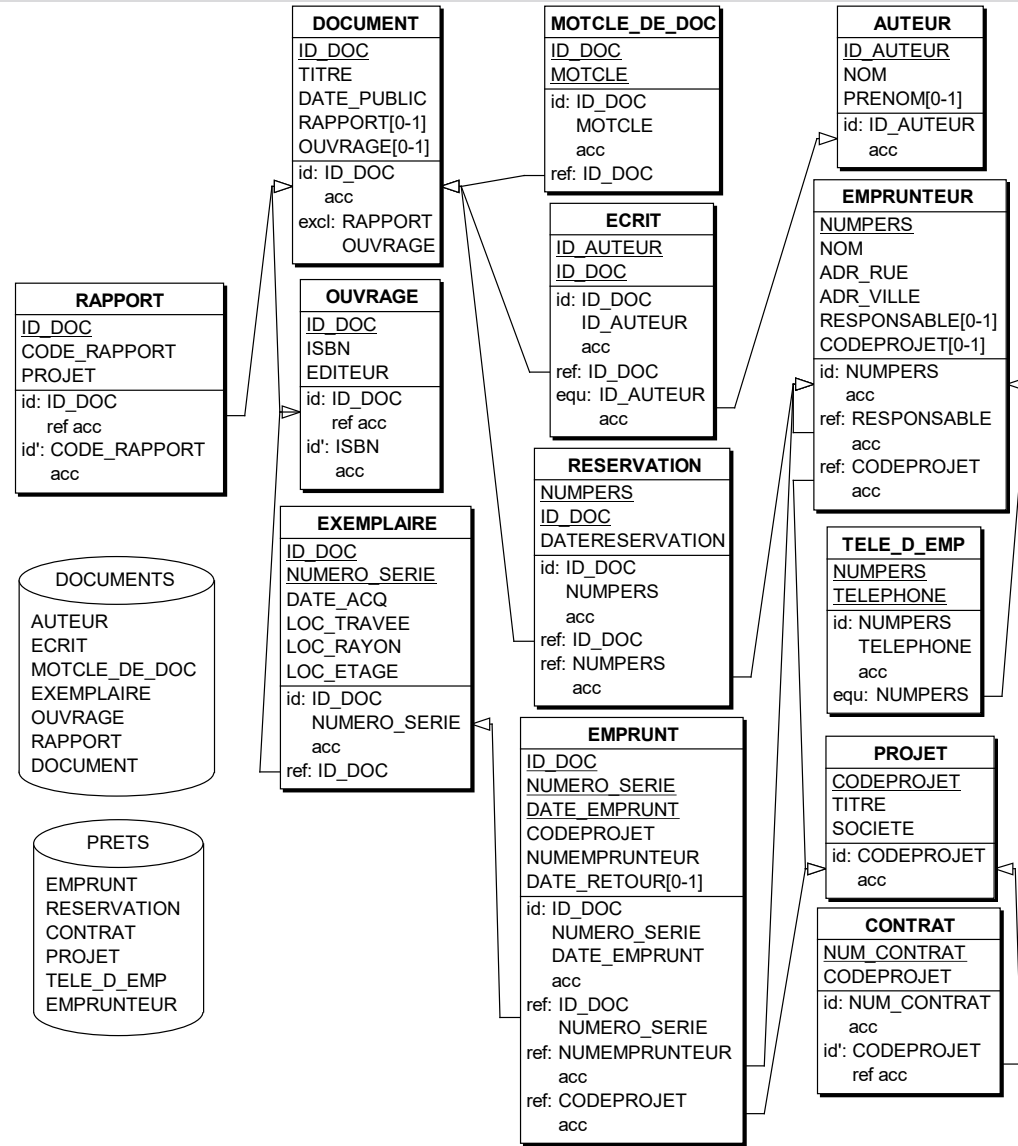
5.6 Exemple

5.6 Exemple - Schéma logique



+ contraintes

5.6 Exemple - Schéma physique

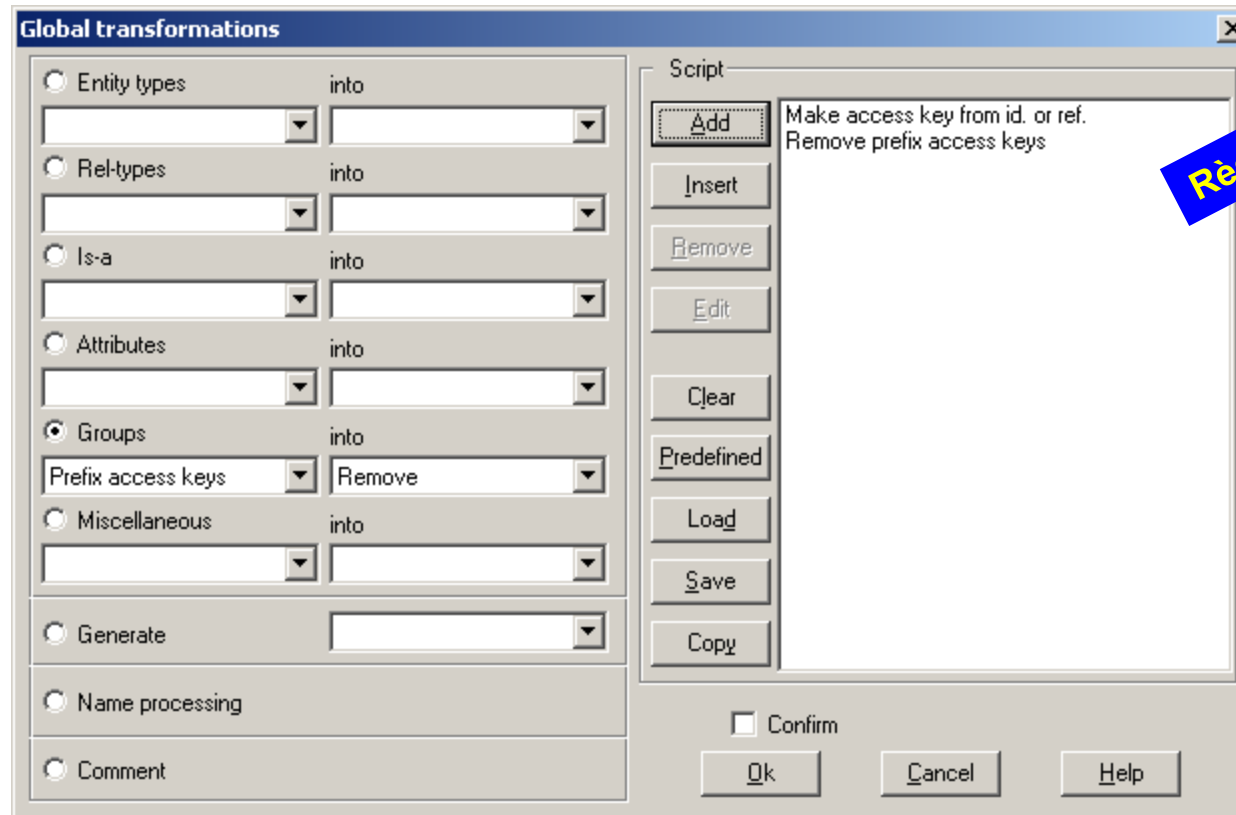


Règles 1, 2 et 7

+ contraintes

5.7 Aide au développement du schéma physique

5.7 Aide au développement du schéma physique



5.7 Aide au développement du schéma physique

Scan or index?		Instructions
Logical parameters		Table sequential scan ...
Nr	1.000.000 records	... through random page read Tlta : 960.94 sec
Lr	250 bytes	... through sequential page read Tlts : 12,76 sec
Nc	1.500 values	
Physical parameters		Index access
Lp	4.096 bytes	selected records Nsc : 667 rec/index entry
τ base	0,8	selected records per page mc : 0,0085 rec/page
Nrpp	12,8 rec/page	selected pages qc : 665 pages
Mrpp	16 rec/page	Reading selected pages (random) Tli : 8,18 sec
Np	78.125 pages	
pages/track	100 pages	Scan or index?
rand. page read	0,0123 sec	thresholds for index choice
seq. page read	0,0002 sec	- min threshold Thr1 : 0,85
rand. track read	0,0163 sec	- max threshold Thr2 : 0,95
		Conclusion: Index recommended

Règle 3

Fin du module 5

Module suivant :
6. Production du code

1. Méthodologie des BD
2. Le modèle Entité-association
3. Analyse conceptuelle
4. Conception logique relationnelle

- 5. Conception physique**
6. Production du code
7. Rétro-ingénierie