I. Introduction ASE :

1. Installation ASE :

A- Bases de données installées obligatoirement :

Master : Contient les tables systèmes qui enregistrent les données relatives à la gestion de tout le système. Exemple : *sysdatabases, syslogins, sysdevices,…* ***(device min 6MB)***

Model : Une base template qui est utilisée pour créer les bases de données utilisateur ***(min 2MB/Master device)***.

Sybsystemprocs : Contient les tables qui enregistrent les procédures stockées système ***(device min 100MB)***.

Tempdb : Contient les tables temporaires ***(min 2MB/Master device)***. La taille recommandée est le max de : l'espace totale demandé lors de l'exécution du plan le plus large ; 10% de la taille des données ou 100 M ; 1 à 2 M par connexion utilisateur.

B- Bases de données additionnelles :

Sybsyntax : Contient une aide de syntaxe des commandes T-SQL avec la commande **sp_syntax "keyword"**

Dbccdb : contient les entrées sorties de la commande **dbcc checkstorage**

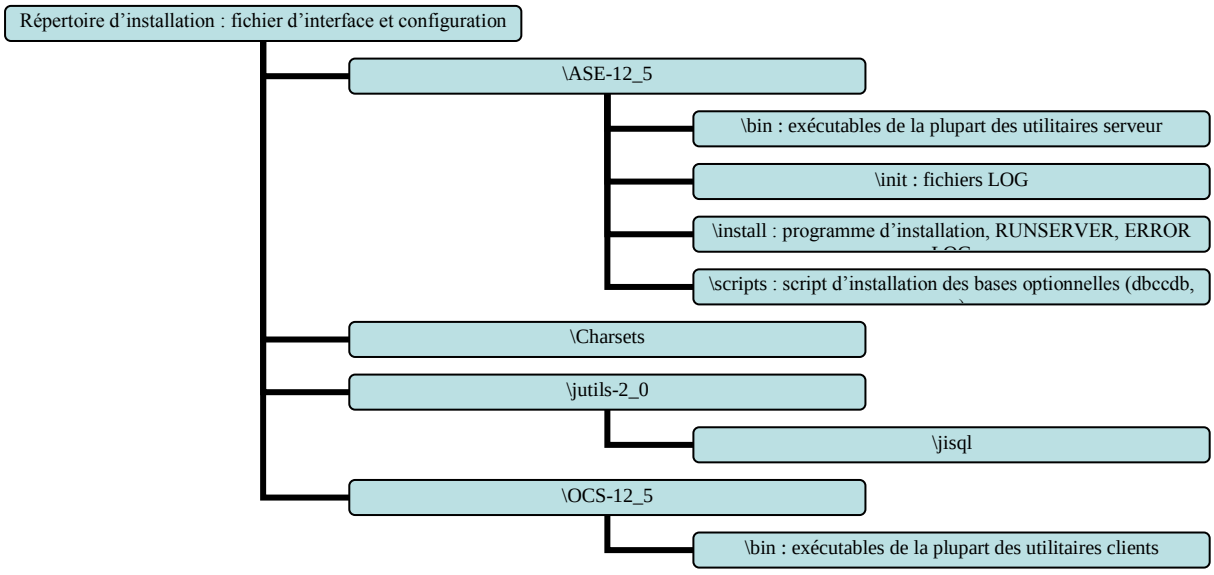
Bases de données exemple : pubs2, pubs3, …

2. Configuration ASE :

A- Système d'exploitation requis :

Solaris 2.8, HP/UX 11.1, IBM AIX 4.3.3, Intel NT 4 SP 6, Digital Unix TruUnix 5.0a 64 bits, SGI Irix 6.5, Intel Linux Kernel 2.2.14-50 glibc 2.1.3-15

B- Répertoires :



C- Fichier d'interface :

Sous UNIX : interfaces

Syntaxe :

server_name retry delay
service protocol network hostname port

Exemple :

SYBASE 3 10
query tcp ether post4 2000
master tcp ether post4 2000

Sous NT: sql.ini

Syntaxe :

[server_name]
Link_type=network_driver, connection_info

Exemple :

[SYBASE]
query=NLWNSCK,post4,2000
master= NLWNSCK,post4,2000

D- Fichier de configuration:

Contient les paramètres généraux de configuration de l'ASE.
Pour le lister : **sp_configure**
Pour lister un seul paramètre : **sp_configure "parameter_name"**
Pour modifier (notez qu'on peut l'éditer manuellement) : **sp _configure "parameter_name", parameter_value**
Pour revenir à la valeur par défaut : **sp _configure "parameter_name",0,"default"**

Pour écrire les valeurs configurées et actives pendant l'exécution d'ASE dans un fichier :
Sp_configure "configuration file", 0, write, "c:\...\new_file.cfg"

Pour écrire les valeurs configurées et pas nécessairement actives pendant l'exécution d'ASE dans un fichier :
Sp_configure "configuration file", 0, restore, "c:\...\new_file.cfg"

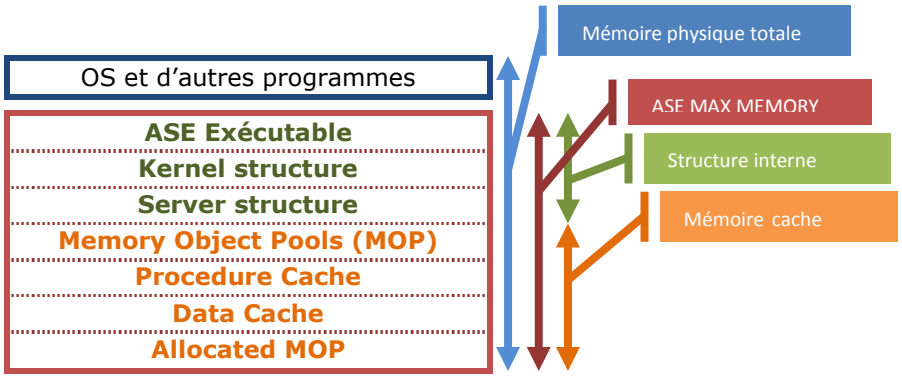
Pour vérifier un fichier :
Sp_configure "configuration file", 0, verify, "c:\...\new_file.cfg"

Pour vérifier les paramètres d'un fichier modifié manuellement et charger ceux qui passent :
Sp_configure "configuration file", 0, read, "c:\...\new_file.cfg"

Pour savoir quel fichier est utilisé :
Select value2 from master..syscurconfigs where config=14

Tables systèmes qui contiennent les informations sur le fichier de configuration sont :
sysconfigures : contient les valeurs des paramètres depuis le démarrage ou depuis la dernière modification
syscurconfigs : contient les valeurs utilisées des paramètres.

II. Configuration de la mémoire dans ASE :



MAX MEMORY :
La mémoire physique totale de l'ASE exprimée en page. Exemple en 2K pour 5000KO : **sp_configure "max memory", 2500**

ASE Exécutable :
Non configurable

Kernel structure :
Non configurable

Server structure :
Configurable en configurant les paramètres suivants :
number of user connections
number of open databases
number of open indexes
number of open objects
number of locks (recommendation : (user connection actuelle+worker process)x20)
number of devices.

Pour estimer la proportion de la mémoire souhaitée :
sp_helpconfig "parameter_name","size"

Exemple:
sp_helpconfig "number of user connections","100"

_____
Configuration parameter, 'number of user connections', will consume 8601K of memory if configured at 100.
...

Ou:
sp_helpconfig "number of user connections","50M"

_____
Configuration parameter, 'number of user connections', can be configured to 579 to fit in 50M of memory.

MOP :
Blocks de mémoire alloués dynamiquement en cas de besoin comme par exemple les buffers réseaux supplémentaires. Non configurable.

Procedure Cache :
Configurable = Max user connection x size of largest plan x 1.25 exprimé en taille de page. Pour obtenir la taille du plus grand plan on exécute la requête suivant :

select max(count(*)/8+1) from sysprocedures group by id
Notez que la requête nous retourne la taille en 2K.

Pour le configurer - exemple: **sp_configure "procedure cache size", 7000**

Data cache :
Charge les pages de données, d'indexe et de log. (on le verra après)

Autres paramètres :
dynamic allocation on demand *(default 1)* : si oui (1) on alloue la mémoire dynamiquement à la demande, sinon (0) la totalité de la mémoire configuré du paramètre est allouée au démarrage.

housekeeper free write percent *(default 1%)* : le pourcentage max des pages modifiées pour que la tâche housekeeper les flashe sur le disque.

III. Ressources physiques :

1. Devices :

Un ***DEVICE*** est une ressource physique qui peut être un ou plusieurs disques ou une portion ou plusieurs de disque. La taille max d'un device est 32GB. La taille min dépend de la taille de page : 2K > 1MB ; 4K > 1M ; 8K > 2MB ; 16K > 4M. Les informations sur les devices sont enregistrées dans la table système sysdevices.

Commandes relatives aux devices :

disk init : initialise les devices et les créé (la clause dsync détermine si oui ou non l'écriture est faite directement sur le disque sous UNIX seulement)
sp_dropdevice : supprime un device. Pour faire, on désalloue toutes les bases de données qui y sont liées. Cette commande ne supprime pas physiquement le fichier.
sp_helpdevice : affiche les informations sur tous les devices ou sur un de précis.
sp_diskdefault : change le device par défaut.
sp_deviceattr : change un paramètre de device – exemple : ***sp_deviceattr, dsync, false***
disk mirror : active le mirroring.
disk remirror : reactive le mirroring.
disk unmirror : désactive le mirroring.

2. Databases :

sp_estspace : pour estimer la taille d'une table ou un indexe. Exemple :

/* La taille que prendra titles si elle aura 10000 lignes*/
sp_estspace titles,10000

```
_____________
Total_Mbytes
------------
        2.51
```

create database : avec l'option « with override » si vous précisez « log on » sur le même device des données.
drop database
sp_helpdb
sp_changedbowner : pour changer le propriétaire de la base de données.
sp_dboption : configure les options de la base de données. Avec checkpoint pour que les changements prennent effet.
Alter database : pour augmenter la taille de la base de données.
sp_logdevice : pour convertir un nouveau device lié à la base de données en un device log. La recommandation pour la taille du log est : 10% à 25% de la taille de la base de données.
sp_configure "default database size", 4 change la taille par défaut (en MB) des bases de données.
db_name(), db_id() : renvoi respectivement le nom depuis l'id de la base, l'id depuis le nom de la base.
grant create database to : pour donner le droit de création des bases à un login.

Options des bases de données (sp_dboption) :

allow nulls by default : si à 1, permet d'attribuer la propriété null à une colonne dont la propriété n'a pas été mentionnée lors de la création d'une table.
auto identity : si à 1, ajoute une colonne d'indentité de 10 chiffres à une table qui n'a pas de clé primaire à la création. Notez que cette colonne ne figure pas dans un select *. Pour la faire apparaitre : select SYB_IDENTITY_COL. Pour changer sa taille (10 par défaut) changer le paramètre de configuration ASE : ***size of auto identity***.
dbo use only : la base ne peut être utilisée que par son propriétaire.
ddl in tran : si à 1, les commandes create, drop et alter peuvent être lancées depuis une transaction définie par utilisateur (user_defined trasactions). Les commandes suivantes ne peuvent

jamais être lancées depuis un user_defined trasactions : ***alter database, alter table … (un)partition, create database, disk init, dump database, dump transaction, drop database, load transaction, load database, select into, truncate table, update statistics***.

identity in nonunique index : si à 1, quand l'option auto identity est à true, et que la table a une colonne d'auto-identité, celle-ci va être ajoutée automatiquement à un indexe qui n'est pas unique.
read only : si à 1, la base est en lecture seule.
single user : seulement un seule utilisateur peut accéder à la base (le premier).
unique auto_identity index : si à 1, ajoute une colonne unique, si elle n'existe pas, avec un index unique non clusterisé à la création d'une table.
select into/bulkcopy/pllsort : si à 1, autorise les opérations non journalisées complètement.
no free space acctg : si à 1, seulement les thresholds sur le logsegment sont avtives, à 0, toutes les thresholds sont actives.
abort tran on log full : si à 1, le serveur abandonne toutes les transactions ouvertes si le log est rempli. Si à 0, il les suspend.
no chkpt on recovery : si à 1, l'enregistrement que présente un checkpoint après le recovery d'une base n'est pas ajouté au journal.
trunc log on chkpt : si à 1, le journal des transactions est tronqué (transactions validées sont supprimés) lors d'un checkpoint si seulement plus de 50 lignes sont écrites dans le journal.

Tables systèmes en jeu : sysdatabases, sysusages (enregistre les fragments d'une base)

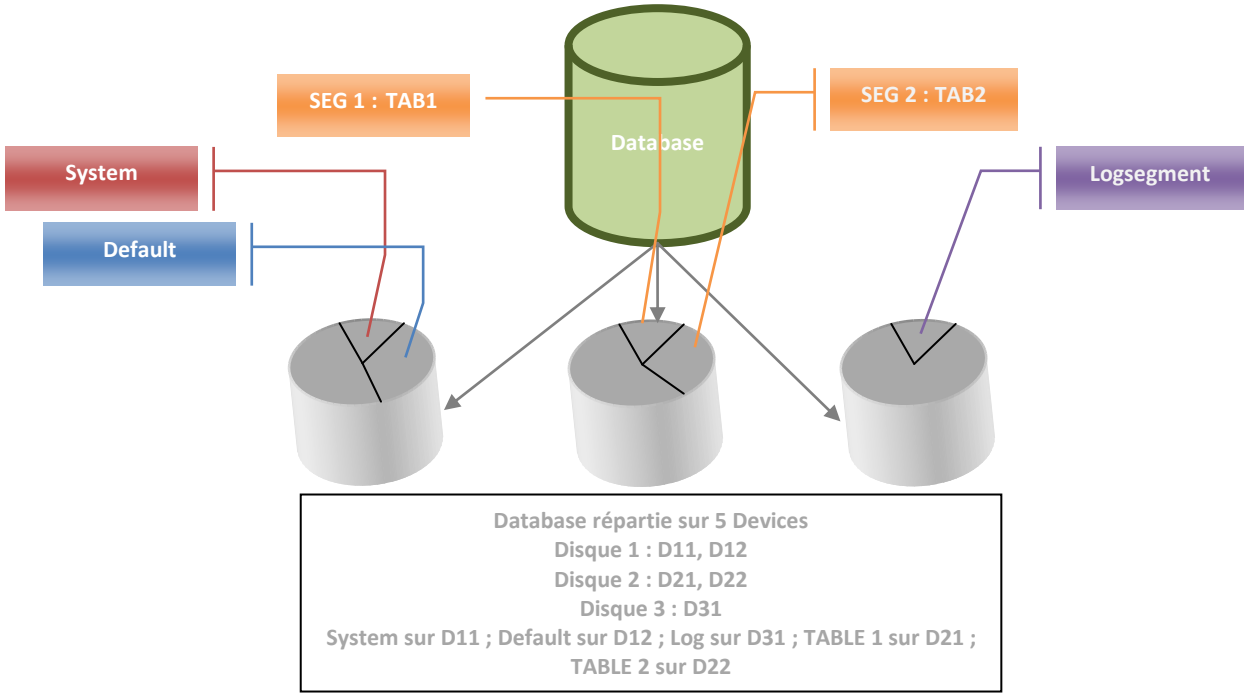
3. Segment :

Les segments sont des étiquettes, des pointeurs qui orientent l'écriture des objets d'une base de données à laquelle ils appartiennent sur un ou plusieurs fragments.
Par défaut lors de l'association base-device, 3 segments sont créés est ne peuvent jamais être supprimés :

System : pour l'écriture des données des tables systèmes.
Logsegment : pour l'écriture des données log.
Default : pour écrire les données utilisateur.

sp_addsegment : ajoute un segment.
sp_dropsegment : supprime un segment utilisateur ou réduit le scope d'un segment.
sp_extendsegment : élargie la scope d'un segment.
sp_helpsegment, sp_help table_name, sp_helpindex
create table … on segment_name ; create indexe … on segment_name : créé respectivement une table et indexe sur le segment souhaité. Notez que les données suivent toujours leur indexe clusterisé.
sp_placeobject : déplace un objet (table ou indexe) sur un segment. Seules les nouvelles entrées seront écrites sur le nouveau segment.

Tables systèmes en jeu : sur master > sysusages, sysdevices – sur la base de données > syssegments, sysindexes.



4. THRESHOLD :

Procédure stockée :

```
Create proc first_alert
        (@dbname varchar(30),
         @segmentname varchar(30),
         @space_left int,
         @status int)
As
        Print "Il ne reste que %1! de pages libres dans le segment %2! sur la base de donnée % !3",
        @space_left, @segmentname, @dbname
Return
```

sp_addthreshold pubs2, seg1, 200, first_alert

sp_modifythreshold pubs2, seg1, 200, first_alert, 250 (dans cette exepmle on a change le nombre de page déclenchant l'alerte.

sp_dropthreshold

Un segment peut avoir jusqu'à 128 thresholds.

IV. Gestion des utilisateurs :

ASE a une approche multilayer d'allocation des droits a l'utilisateur final qui accède aux données :

- ✓ L'utilisateur final doit avoir une permission pour accéder au serveur (server wide login)
- ✓ Il doit ensuite avoir le droit d'accéder à la base de données (database user)
- ✓ Enfin il doit avoir des permissions pour utiliser l'objet et/ou les objets qui enregistrent les données voulues.

1. System Role :

Les rôles systèmes sont un groupement de privilèges assignés à un login afin qu'il puisse réaliser certains tâches d'administration, de sécurité et de sauvegarde. Les rôles systèmes sont :

SA_ROLE, il : modifie tous les paramètres de configuration sauf ceux de sécurité ; gère les allocations de ressources physiques ; créé les utilisateurs des bases de données ; accorde et révoque SA rôle ; accorde les permissions aux utilisateurs du serveur ; sauvegarde et charge toutes les bases de données et log de transaction ; stop le serveur ou tue les processus ; diagnostique le serveur (exp : dbcc) ; il est traité comme étant propriétaire de toutes les bases de données ; …

SSO_ROLE, il : créé les logins serveur et initialise leur mot de passe ; modifie les logins ; change les mots de passe ; créé et gère les « user-defined role » ; accorde proxy authorization ; accorde et révoque SSO_ROLE et OPER_ROLE ; audit la sécurité du système ; verrouille déverrouille les logins ; supprime les logins ; …

OPER_ROLE, il : sauvegarde et charge toutes les bases de données et les logs de transaction. (les propriétaires des bases de données ont l'OPER_ROLE sur leur base)

SYBASE_TS_ROLE : est obligatoire pour lancer les commandes dbcc.

grant role role_name to login_name : grand role sa_role to Mohammed
revoke role role_name from login_name : revoke role sso_role from Ali

(commande alternative : **sp_role "grant",sa_role, Mohammed**)

set role role_name on|off : set role sa_role off .. active ou désactive un rôle. Le rôle désactivé est activé automatiquement après la fin de la cession et le démarrage d'une nouvelle)
sp_displaylogin login_name : sp_displaylogin mohammed .. affiche les informations relatives aux roles d'un login.
sp_addlogin … : ajoute un login. L'ordre des paramètres dans cette commande est obligatoire. Sinon, spécifiez le paramètre en ajoutant sans identifiant, exp : @language=french.

sp_addlogin loginame, passwd [, defdb]
[, deflanguage] [, fullname] [, passwdexp]
[, minpwdlen] [, maxfailedlogins]

Paramètres de configuration pour les valeurs par défaut de la commande sont : ***default language id ; systemwide password expiration ; minimum password length ; maximum failer login ; check password for digit***

sp_password oldpasswd, newpasswd : change le mot de passe. (en tant que SOO, précisez votre login au début pour changer un autre utilisateur : **sp_password ssoligin, newpasswd, logintochange**)

sp_droplogin : supprime un login
sp_modifylogin loginname, parametre, newvalue : change les paramètres d'un login.
sp_locklogin loginname, lock|unlock : verrouille ou déverrouille un login. Si la commande n'a pas de paramètres, elle retourne la liste des logins verrouillés.
suser_name(login_id) - suser_id(loginname) : retourne le loginname depuis sont id - l'id depuis le loginname.

grant set session authorization to loginname ou **grant set proxy to loginname** : accorde l'utilisation d'une cession proxy à un login
set proxy loginname ou **set session authorization loginname** : impersonalise un login au niveau serveur. **sp_who loginname** : pour savoir le login d'origine.

Tables systèmes en jeu : syslogins, sysloginroles, syssrvroles. (les valeurs de la colonne status dans syslogins sont : 0 rien de spécial, 1 mot de passe < 6, 2 login verrouillé, 4 mot de passe expiré)

2. Utilisateurs et groupes des bases de données :

Chaque base de données à un propriétaire qui : change ses options ; créé et supprime ses utilisateurs ; accorde et révoque les permissions de création d'objet ; exécute le checkpoint ; exécute les commandes dbcc ; sauvegarde et restauration de la base ; il a tous les privilèges sur ses objets. (**sp_changedbowner** pour changer le propriétaire d'une base ; **sp_dboption** pour changer les options d'une base)

sp_adduser loginame [, name_in_db [, grpname]] : ajoute un utilisateur de la base de données.
sp_dropuser… : supprime un utilisateur de la base. Notez que vous ne pouvez pas supprimer un utilisateur qui est propriétaire d'objets dans la base. Dans ce cas verrouillez-le !
sp_helpuser name_in_db : affiche les informations sur les utilisateurs de la base.
sp_adduser guest : ajoute un compte invité à la base de données (un compte guest est automatiquement ajouté à la base tempdb et master à la création qui ne peut pas être supprimé, juste verrouillé).
setuser ["user_name"] : moyen donné au dbo pour impersonnaliser un utilisateur de sa baase.
user_name(user_id) ; **user_id(user_name)** : l'id depuis le nom ; le nom depuis l'id…
sp_addalias loginname, name_in_db : ajoute un alias (autorisation d'accès) à un login qui n'est pas utilisateur de la base pour qu'il puisse y accéder.
sp_dropalias loginname : supprime l'alias.
sp_addgroup groupname : ajoute un groupe d'utilisateur.
sp_dropgroup : supprime un groupe.
sp_changegroup groupname, username : ajoute un utilisateur à un groupe.
sp_helpgroup : affiche les informations sur un groupe.

Notez que vous ne pouvez pas supprimer le groupe « public » ni détaché un utilisateur de ce groupe. Et qu'un utilisateur ne peut être affecté qu'à un groupe de plus que public. Chaque groupe est distinct est ne peut être contenu dans un autre. Pour y remédier, on implémente un « user defined role ».

create role rolename [with passwd password] : créé un user defined role.
grant permissionname on objectname to username : accorde un permission à un user defined role.
grant role rolename [,rolename] to rolename [,rolename] : assigne un user defined role à un utilisateur.
set role rolename [with passwd password] on|off : active ou désactive un user defined role. Le user defined role doit être activé avant d'être utilisé.
sp_modifylogin loginname, "add default role", rolename : permet d'activer un user defined role automatiquement à la connexion d'un utilisateur.

grant inner_role to outer_role : permet de contenir un user defined role dans un autre (donner les permission d'un à un autre).
alter role rolename add passwd password|drop passwd : permet d'ajouter ou supprimer un mot de passe à un user defined role.
revoke role rolename [,rolename] from rolename [,rolename] : révoque un user defined role à un utilisateur.
drop role rolename [with override] : supprime un user defined role. Avec l'option override, supprime toutes les permissions données dans toutes les bases de données.
sp_displayroles [loginname|rolename] [expand_up|expand_down] : affiche les informations sur un user defined role.
sp_helpprotect [objectname [,NULL [,NULL [,rolename] : pour afficher la liste des permissions d'un user defined role en relation avec un objet.
sp_activerole [expand_down] : affiche les user defined role actifs et seulement avec l'option down.
grant {all [privileges] | command_list} to {public | name_list | role_name} : accord des privilèges : create table, view, default, rule, procedure (**revoke … from**). **sp_helpprotect username** : pour afficher les privilèges d'un utilisateur. Les privilèges suivants ne peuvent être accordés : dbcc commande, dump, load, setuser.

Tables systèmes en jeu : sur master > syslogins, sysusers ; sur la base > sysusers, sysalternates

3. Utilisateurs et objets de la base de données :

```
grant {all [privileges]| permission_list}
    on { table_name [(column_list)]
       | view_name[(column_list)]
       | stored_procedure_name}
    to {public | name_list | role_name}
    [with grant option]
```

Permission list :

sur les tables : select, insert, update, delete, references
sur les views : select, insert, update, delete
sur les colonnes : select, update, references
sur les procédures : execute

with grant option : permet à un utilisateur auquel elle a été donnée d'accorder ce qu'il lui a été donné à un autre utilisateur.

```
revoke {all [privileges]| permission_list}
    on { table_name [(column_list)]
       | view_name[(column_list)]
       | stored_procedure_name}
    from {public | name_list | role_name}
    [cascade]
```

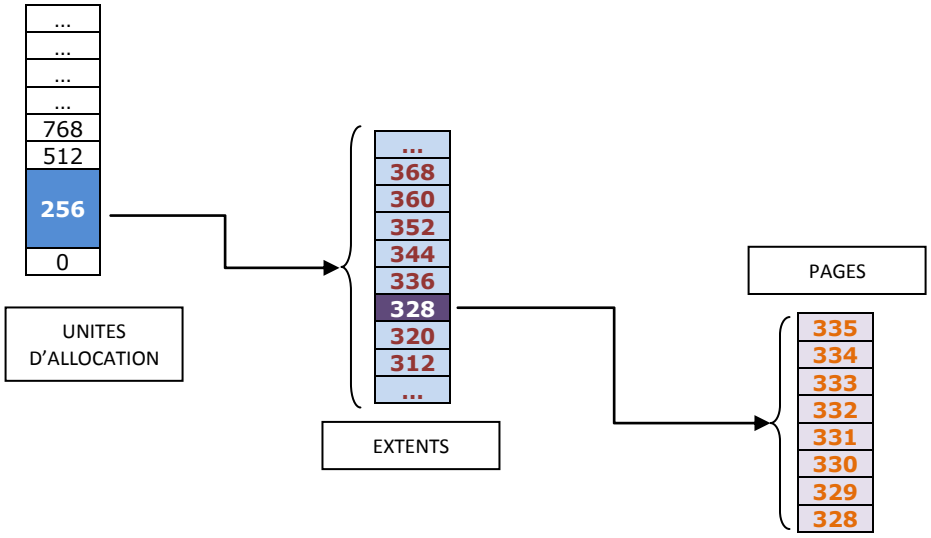
cascade : permet de supprimer toutes les permissions données par des utilisateurs avec le pouvoir **with grant option** aux autres utilisateurs.

revoke grant option for permission_list on objects from username [cascade]: revoque le pouvoir **with grant option** à un utilisateur. Avec l'option cascade, il revoque le pouvoir et revoque toutes les permissions données par cette utilisateur.
sp_helpprotect objectname : affiche la liste des autorisations sur un objet.
create schema authorization authorization_name create_object_statement [create_object_statement …][permission_statement …] : créé un schema d'autorisation pour un utilisateur.

V. Allocation d'objet

1. Structure physique: THE BIG PICTURE!

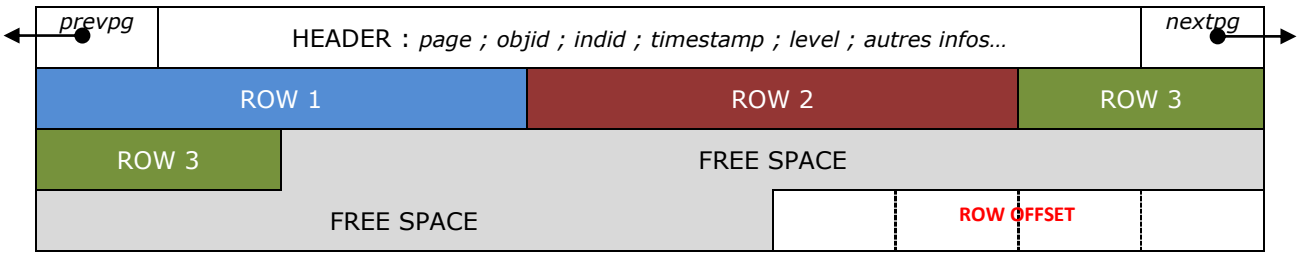
Le stockage physique des données se fait selon le schéma suivant : chaque device de la base de données se compose ***D'UNITE D'ALLOCATION***. Chaque unité d'allocation se compose de 32 ***EXTENTS***. Chaque extent se compose de 8 ***PAGES***.



Structure des pages : Toutes les pages ont la même structure (ou presque). On notera 6 Types de pages : ***APL – DOL*** pour les pages de données et ***PA, OAM, GAM, CA*** qui sont des pages système.

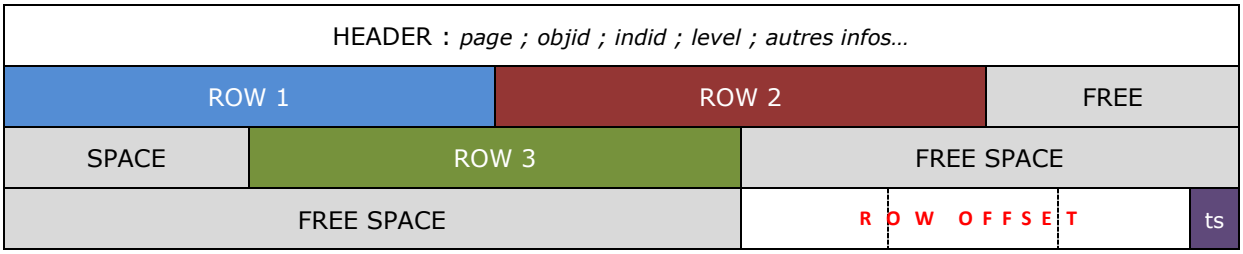
A- Schéma de la page APL :

Malgré qu'il existe 3 types de verrouillage : Allpages, Datapages et Datarows, il n'existe que deux type de pages : APL pour le Allpages et DOL pour le Datapages et Datarows. (avant la version 11.9.2 il n'existait que le verrouillage Allpages)



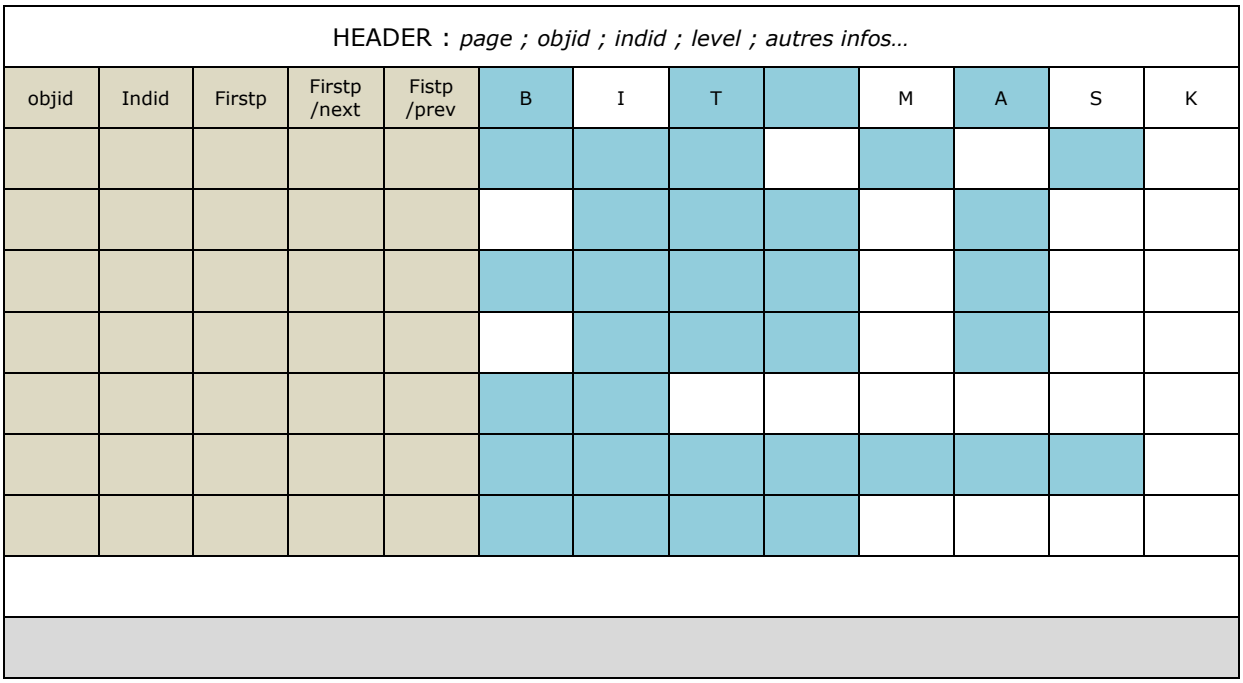
page : le numéro logique de la page.
objid : l'id de l'objet contenu dans cette page.
indid : l'id de l'indexe dont les données sont sur cette page.
prevpg, nextpg : pointeur sur la page d'avant et d'après.
timestamp : la dernière fois où des données ont été écrites sur cette page.
level : utilisé pour déterminer le niveau de l'indexe.
Autres infos : autres informations sur le statut de la page.
ROW 1, ROW 2,… : les lignes de données.
FREE SPACE : espace libre contigus.
ROW Offset : une table ou liste à la fin de la page, composé de 2 octet pour chaque ligne (ROW) qui pointe le début de chaque ligne et sauvegarde son numéro.

B- Schéma de la page DOL :



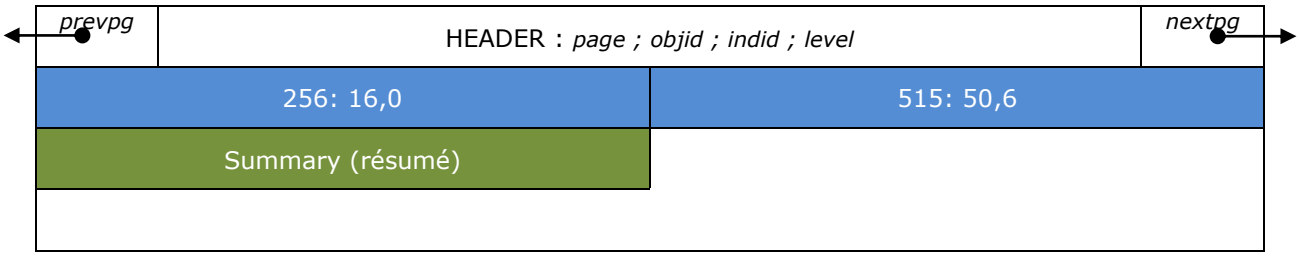
page : le numéro logique de la page.
objid : l'id de l'objet contenu dans cette page.
indid : l'id de l'indexe dont les données sont sur cette page.
timestamp : la dernière fois où des données ont été écrites sur cette page placé à la fin de la page (ts).
level : utilisé pour déterminer le niveau de l'indexe (voir ce qui suit).
Autres infos : autres informations sur le statut de la page.
ROW 1, ROW 2,… : les lignes de données.
FREE SPACE : espace libre non contigus.
ROW Offset : une table ou liste à la fin de la page pour chaque ligne (ROW) qui sauvegarde seulement son numéro.

C- Schéma de la page PA (OU AP → ALLOCATION PAGE) :



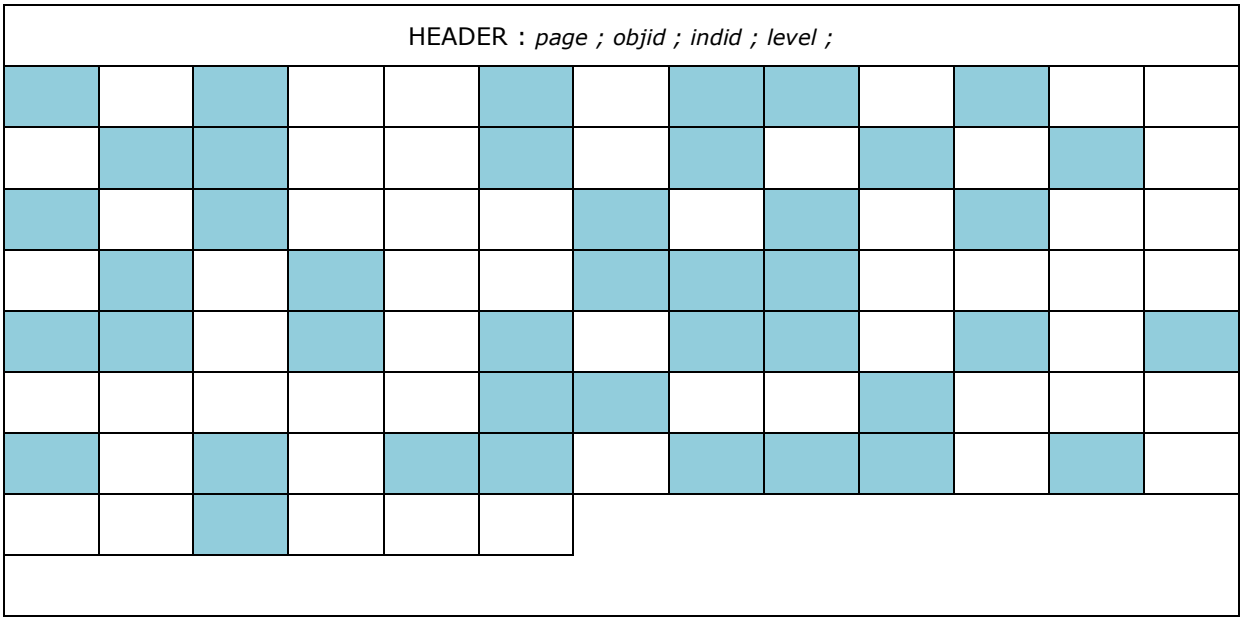
page : le numéro logique de la page : toujours un multiple de 256 : 0, 256, 512,… / objid : toujours 99 / indid : Toujours 0 / level : toujours 0.
Le contenu de la page PA : une ligne pour chaque extent dans l'unité d'allocation qu'elle représente. Chacune de ces lignes est une structure de 16 octets est contient les informations suivantes : objid : l'id de l'objet qu'utilise cet extent / indid : si l'objet est un indexe sont indexe id / firstp : un pointeur vers la première page de d'l'extent / firstp/next : un pointeur vers la première page de d'l'extent suivant utilisé par l'objet / firstpprev : un pointeur vers la première page de d'l'extent d'avant utilisé par l'objet / bitmask : trace l'utilisation des pages de l'extent par l'objet, bit à 1 → la page est utilisée, bit à 0 → la page est libre.

D- Schéma de la page OAM → OBJECT ALLOCATION MAP :



page : le numéro logique de la page OAM.
objid : l'id de l'objet que l'OAM trace.
indid : 0 heap tabel, 1 indexe clusterisé APL, 2-250 clusterisé DOL ou non clusterisé APL ET DOL.
prevpg, nextpg : pointeur sur la page OAM d'avant et d'après. (avec le temps si l'objet devient très grand il aura besoin de plus d'OAM qui seront chainées entre elles)
level : 10 pour la première OAM et 0 pour les suivantes.
Le contenu de la page OAM : une ligne pour chaque unité d'allocation où l'objet est présent. Chacune de ces lignes contient les informations suivantes : la première page de l'unité d'allocation, le nombre de pages que l'objet utilise sur cette unité d'allocation, le nombre de pages non utilisées mais réservées (généralement, la somme des deux est un multiple de 8 car un extent ne doit être utilisé que par un seul objet)
+ une ligne de résumé : le nombre total des pages utilisées pas l'objet, le nombre total estimé des lignes, le nombre estimé de lignes par page.

E- Schéma de la page GAM → GLOBAL ALLOCATION MAP :



page : le numéro logique de la page qui débute toujours à la 128ième page de la 1ère UA / objid : toujours 14
Le contenu de la page GAM : un bitmask qui trace toutes les unités d'allocation de la base de données. Un Bit représente une unité d'allocation, si à 1 → aucun extent n'est libre, si à 0 → au moins un extent n'est toujours pas affecté à un objet.

2. Vue sur la table système : SYSINDEXES

La table sysindexes contient une ligne pour chaque table, indexe et un LOB.

name	id	indid	doampg	ioampg	first	root
salesdetailind	160000570	3	0	944	952	954
taind	192000684	1	832	848	840	856
auidind	192000684	2	0	656	872	872
titleidind	192000684	3	0	880	888	888
titleidind	224000798	1	769	808	800	816
titleind	224000798	2	0	672	824	824
stores	256000912	0	680	0	681	681
discounts	288001026	0	696	0	697	697
au_pix	320001140	0	712	0	713	713
tau_pix	320001140	255	0	704	705	705

- Name : le nom de l'indexe ou de la table.
- Id : id de la table ou de l'id de la table à laquelle l'indexe appartient.
- Indid : 0 si une table, 1 si un indexe clusterisé APL, 2-250 si un indexe clusterisé DOL ou non clusterisé APL ou DOL, 255 si text, image, … LOB.
- Doampg : le numéro de la 1ère page OAM de la table.
- Ioampg : le numéro de la 1ère page OAM de l'indexe.
- First et root :

Indid	First	Root
0	La 1ère page de table non clusterisée (APLet DOL)	La dernière page de la table non clusterisée (APL seulement)
1	La 1ère page du dernier niveau de l'indexe clusterisé APL (qui n'est d'autre que les pages de données elles même)	La page racine de l'indexe clusterisé APL.
2 A 250	La première page du dernier niveau de l'indexe clusterisé DOL, indexe non clusterisé DOL ou indexe non clusterisé APL.	La page racine de l'indexe clusterisé DOL, indexe non clusterisé DOL ou indexe non clusterisé APL.
255	Pointe sur une page vide toujours utilisée comme target page pour une nouvelle entrée.	Non utilisé.

Pour examiner l'allocation physique des pages on utilise les différentes commandes DBCC.

3. Fragmentation :

A- Faible densité de page :

On parle d'une fragmentation des lignes ou FAIBLE DENSITE quand les pages sauvegardent moins de lignes que ce qu'elles peuvent théoriquement.
Pour la mesurer on calcul d'abord la taille théorique d'une ligne en additionnant les tailles des colonnes ou en se référant à la donnée **Data row size** d'optdiag.

Densité théorique (Nombre de linges théorique / page) = taille de la page / taille de la ligne

Après on calcul la densité réelle : nombre de ligne total / nombre de page total

Pour avoir ces informations :

sp_help tablename (pour calculer la taille de la ligne)

sp_spaceused tablename : rowtotal / (data/taille logique de la page)

name	rowtotal	reserved		data
titles	18	94 KB	4 KB	4 KB

dbcc checktable (tablename) : data rows / number of data pages

```
Checking titles: Logical pagesize is 2048 bytes
The total number of data pages in this table is 2.
Table has 18 data rows.
```

dbcc listoam (dbid,objid,inid) : rows / (used pgs –OAM pg cnt)

```
Objid:  224000798    indid:  0
OAM pg cnt:     1    Entry cnt:      1
Rows:          18    Rows Per pg:    9
Used pgs:       3    Unused pgs:    12
```

Optdiag statistics dbname..tablename –Usa –P… : data row count / data page count

On parlera alors d'une faible densité ou de fragmentation de page si

densité réelle < densité théorique

B- Fragmentation d'extent :

On parlera d'une fragmentation d'extent si un objet utilise des pages dispersées sur plusieurs extents d'une manière à ce que des pages sont restées libres. Par exemple un objet utilisant 5 pages réparties sur 3 extents laissant ainsi 19 pages libres → 2 extents utilisés en plus. Dans un cas optimal, cet objet devrait utiliser un seul extent.
On utilisera les commandes et informations suivantes pour déterminer si oui ou non on a une fragmentation d'extent :

sp_spaceused tablename : (unused / taille logique de la page (2K,4K,8K,16K))-1

name	rowtotal	reserved	data	index_size	unused
titles	18	94 KB	4 KB	4 KB	86 KB

dans l'exemple : (86/2)-1= 42 pages (pour data et indexe) → 5 extents

dbcc tablealloc ("tablename") : (total # of extents x 8) – used pages

TOTAL # of extents = 6

Alloc page 768 (# of extent=2 used pages=2 ref pages=2)
Alloc page 768 (# of extent=2 used pages=4 ref pages=4)
Alloc page 512 (# of extent=1 used pages=1 ref pages=1)
Alloc page 768 (# of extent=1 used pages=1 ref pages=1)
Total (# of extent=6 used pages=8 ref pages=8) in this database

Dans l'exemple : (6x8)-8= 40 → 5 extents

dbcc listoam (dbid,objid,inid) : unused

OAM pg # 769 has the following 1 entry (allocpg:used/unused):
768: 3/ 12

Dans l'exemple 12 pages unused → 1 extent (ça concerne seulement les pages de données car dans listoam on précise inid, dans l'exemple on a inid=0)

On parlera généralement d'une fragmentation d'extent si :

Nombre de pages inutilisées > 8

C- Fragmentation de chaînage ou extents traversés

On parlera d'une fragmentation de chainage si le chainage entre les pages n'est pas ordonné. Cela concerne seulement les tables APL, les pages OAM, les niveaux d'indexe chainés.
Pour le déterminer on utilise les commandes suivantes :

Dbcc pglinkage (…) : si les numéros de pages sont désordonnés on parlera de fragmentation de chainage.

Object ID for pages in this chain = 8.
Page : 770
Page : 773
Page : 771
Page : 772
Page : 775
Page : 774
Page : 776
Page : 777
Page : 778

Optdiag statistics dbname..tablename –Usa –P : data page cluster ratio ou index page cluster ratio doit être > 0.8

D- Fragmentation typique à DOL :

ROW FORWARDING : se produit lorsque la taille d'une ligne augmente et qu'il n'y a pas assez de place pour le faire, on supprime la ligne et on place un pointeur de 10 octets et on la déplace vers une nouvelle page.
DELETED ROW : se produit lors des suppressions de lignes dont l'espace est non récupéré automatiquement.
EMPTY PAGE : se produit lorsque tous les lignes d'une page deviennent des DELETED ROW.
Pour les détecter on se réfère au résultat d'**optdiag** : *Empty data page count, Forwarded row count, Deleted row count.*

E- Défragmentation :

TYPE de fragmentation	Diagnostique	Réparation
Faible densité	sp_help, dbcc checktable, sp_spaceused, dbcc listoam, optdiag	Suppression/Création Indexe clusterisé, BCP IN/OUT, Select into
Fragmentation d'extent	sp_spaceused, dbcc tablealloc, dbcc listoam	Suppression/Création Indexe clusterisé, BCP IN/OUT, Select into
Fragmentation de chaînage	dbcc pglinkage, optdiag	Suppression/Création Indexe clusterisé, BCP IN/OUT, Select into
DOL	Optdiag	Suppression/Création Indexe clusterisé, BCP IN/OUT, Select into, REORG

F- Prévention :

Fillfactor : ce paramètre utilisé lors de la création des indexes permet de spécifier un pourcentage libre sur les pages d'indexe pour une éventuelle augmentation de la taille des lignes, pour prévoir les ajouts de lignes et minimiser la division des pages,…
Create index …. With fikkfactor=75 → 25% de la taille de la page d'indexe restera vide seulement à la création de l'indexe, ceci dit, si un indexe est créé sur une table vide, cette commande sera inefficace !! par défaut, si vous ne précisez pas ce paramètre, ASE le prend pour 100.

exp_row_size : ce paramètre utilisé lors de la création d'une table DOL, réserve la taille exprimée pour chaque ligne. Exemple : create table … with exp_row_size=250 → chaque ligne ajouté occupera 250 octets même si réellement sa taille est inférieure à 250. Cela permet de se protéger des ROW FORWORDING en laissant de l'espace pour une éventuelle augmentation de la taille des lignes. Au niveau global, on précise le paramètre de configuration pour toutes les tables DOL : default exp_row_size percent (pourcentage calculé depuis la taille de la page).

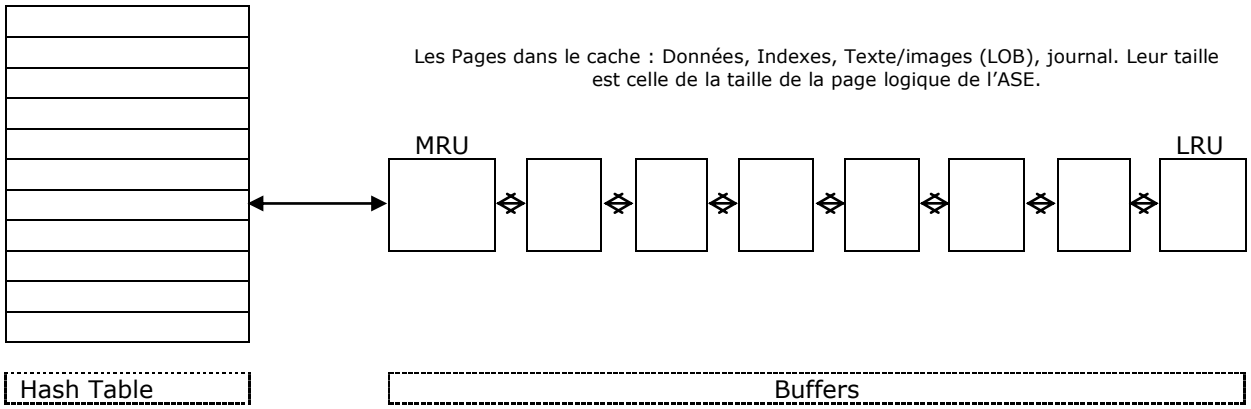
reservepagegap : ce paramètre utilisé lors de la création d'une table, réserve pour chaque nombre de page exprimé une page libre. Exepmle create table … with reservepagegap=8 → chaque 8 page utilisées on laisse la dernière libre. Cela permet de se protéger de la fragmentation d'extent en laissant une page libre lors du transfert de données par exemple en utilisant BCP ou select into.

VI. DATA CACHE :

1. Définition

Espace mémoire alloué pour la structure de données de l'ASE. Il consiste en un cache de données par défaut et éventuellement des caches de données nommés (utilisateurs).

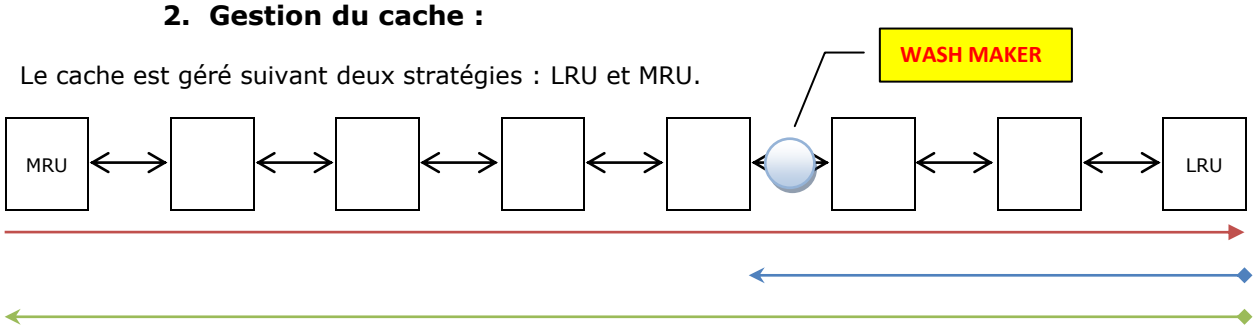
Le cache de données est composé d'une liste double chainée de bufferpools. Ces derniers sont utilisés pour contenir les pages de données dans le cache et sont gérés par la HASH-TABLE.
Un Bufferpool peut être constitué de : 1 page, 2 pages, 4 pages, ou 8 pages. Un buffer ne peut pas être une combinaison de bufferpool de différentes tailles.



Quand une page est demandée, ASE la cherche en premier lieu dans le cache, ou plus exactement dans la HASH-TABLE ; s'il ne l'a trouve pas, il doit obligatoirement la charger dans le cache avant de la traiter.

2. Gestion du cache :

Le cache est géré suivant deux stratégies : LRU et MRU.



Les pages dans le cache traversent la chaine MRU / LRU. Le marquer de vidage est un point logique qui détermine la plage de vidage qu'en la traversant, les pages sont flashées sur le disque.

A- <u>Stratégie LRU :</u>

Est la stratégie par défaut, elle assure que les pages utilisées sont retournées à l'extrémité MRU pour leur permettre une autre durée de vie dans le cache et par conséquence une éventuelle réutilisation. (Flèche verte)

B- <u>Stratégie MRU :</u>

Cette stratégie assure que les pages utilisées sont retournées juste avant le marquer de vidage pour s'en débarrasser le plus vite possible jugeant ainsi la non réutilisation de ces pages. Généralement utilisée pour les objets temporaires. (Flèche bleu)

Il existe une 3^{ième} stratégie gérant un cache dit relaxé. Elle consiste à maintenir un chaine fermée où la page victime est déterminée lisant le statut de la page avant le marquer de vidage et en

tournant suivant les aiguilles de la montre si la page ne présente pas un profil victime. Ce type de cache est utilisé généralement pour les pages du journal ou pour les applications avec un très faible ratio de remplacement des pages dans le cache.

3. Cache nommé :

On peut créer des caches nommés pour mieux gérer notre mémoire : placer les pages d'une base de données sur un cache à part, les pages d'objets volumineux, journal, indexe,…

Notez que le cache par défaut est toujours utilisé dans les cas suivant :

- Quand un objet n'est pas lié à un cache nommé,
- Lors de l'opération Load database et load transaction,
- Rcovery.
-

Le cache par défaut ne peut pas être supprimé.

La taille minimale d'un cache est la taille de 256 pages. Chaque cache doit avoir un buffer d'une page.

Complément, Commandes : Voir TP sur les caches de données.

Complément de documentation : Voir documents rattachés