



# Administration des SGBD

Conception d'un SGBD

M1 CNS

Damien Ploix

# *Conception d'un SGBD*

- Application
  - Base/Schéma de données
  - Code
- Composants technique
  - stockage
  - traitement

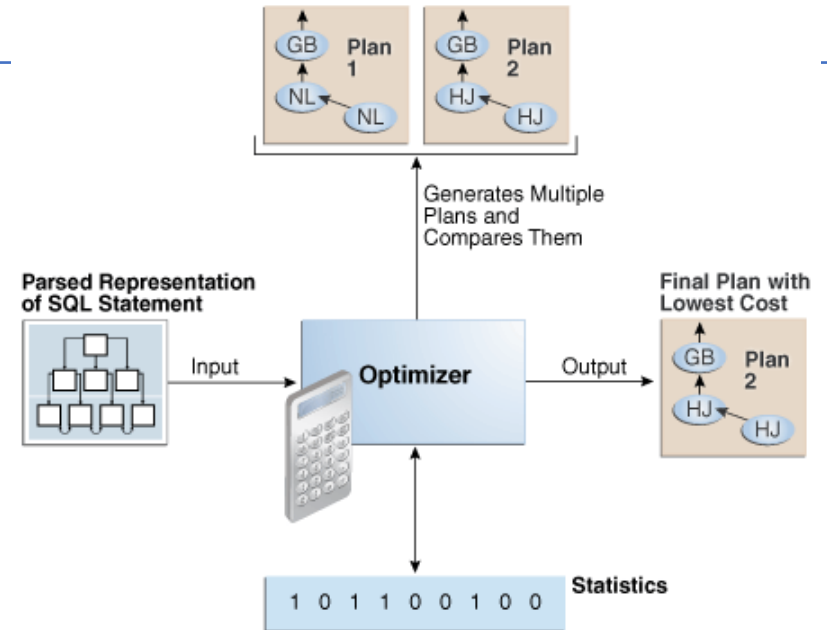
# *Conception du modèle de base de données*

- Le modèle de base de données définit :
  - Des « tables/documents/... » contenant des données conçues suite à une analyse « fonctionnelle/conceptuelle »
  - Des traitement (requêtes) des informations (données)
    - En lecture ou écriture (CRUD)
    - Intégrant des calculs
    - Nécessitant un contrôle d'accès (confidentialité)
  - Des « optimisations » liées aux requêtes sur les données :
    - Indexation,
    - Dénormalisation,
    - ...

# Construction d'une base de données

## Fonctionnement des SGBD

- Les moteurs analysent :
  - Les requêtes SQL,
  - Les plans d'exécution :
    - Méthodes de jointure (le cas échéant)
    - Indexes et distribution des données
  - Les statistiques sur les données
- Afin de déterminer la méthode qui sera mise en œuvre.



- La conception des modèles de données se basera sur la logique en œuvre dans le moteur de SGBD cible.
  - Moteur transactionnel : respect des formes normales,
  - Moteur analytique : faits / dimensions
  - Moteur BigData : clé/valeur, table, ...

## Construction d'une base de données rappel sur les formes normales

- Formes normales : un enregistrement est « en 1<sup>ère</sup> ssi tous les domaines sous-jacents ne contiennent que des données atomiques »
- *Objectif : ne pas avoir à rechercher du texte dans du texte ou à parcourir des tables incluses dans des tables !*
  - Exemple : mettre la table suivant en 1FN

Client	Jour 1	Jour 2	Montants
Ho David	Lundi		19,20 19,20
Ho David	Mercredi	Jeudi	-80
Ho Marian	Jeudi		100,00 150,00 -40,00

## Construction d'une base de données rappel sur les formes normales

Groupes multiples par séparation de lignes

Client	Jour 1	Jour 2	Montant
Ho David	Lundi		19,2
Ho David	Lundi		19,2
Ho David	Mercredi	Jeudi	-80
Ho Marina	Jeudi		100
Ho Marina	Jeudi		150
Ho Marina	Jeudi		-40

Transactions identiques via ID unique

ID achat	Client	Jour 1	Jour 2	Montant
1	Ho David	Lundi		19,2
2	Ho David	Lundi		19,2
3	Ho David	Mercredi	Jeudi	-80
4	Ho Marina	Jeudi		100
5	Ho Marina	Jeudi		150
6	Ho Marina	Jeudi		-40

Plusieurs éléments significatifs  
dans 1 seule colonne

ID achat	Nom Client	Prénom Client	Jour 1	Jour 2	Montant
1	Ho	David	Lundi		19,2
2	Ho	David	Lundi		19,2
3	Ho	David	Mercredi	Jeudi	-80
4	Ho	Marina	Jeudi		100
5	Ho	Marina	Jeudi		150
6	Ho	Marina	Jeudi		-40

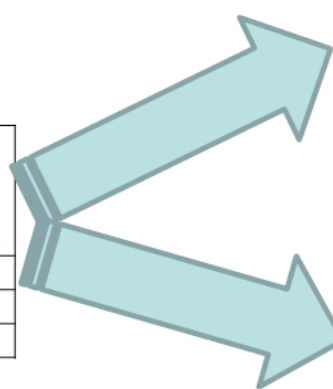
Plusieurs colonne représentant le même fait

ID achat	Nom Client	Prénom Client	Jour	Montant
1	Ho	David	Lundi	19,2
2	Ho	David	Lundi	19,2
3	Ho	David	Mercredi	-80
4	Ho	Marina	Jeudi	100
5	Ho	Marina	Jeudi	150
6	Ho	Marina	Jeudi	-40
7	Ho	David	Jeudi	-80

## Construction d'une base de données rappel sur les formes normales

- Formes normales : un enregistrement est « en 2<sup>ème</sup> ssi 1<sup>ère</sup> + toutes les données n'étant pas des clés dépendent entièrement de clés »
- *Objectif : éviter les redondances des informations dans plusieurs enregistrement !*

ID Produit Clé primaire	ID Fournisseur clé primaire	Nom du fournisseur	Prix	Adresse
65	2	Lautre	59.99	Paris
73	2	Lautre	20.00	Paris
65	1	Leclerk	59.99	Lyon



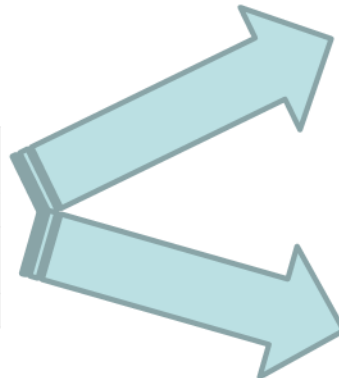
ID Fournisseur clé primaire	Nom du fournisseur	Adresse
1	Lautre	Paris
2	Leclerk	Lyon

ID Produit Clé primaire	ID Fournisseur clé primaire	Prix
65	2	59.99
73	2	20.00
65	1	59.99

## Construction d'une base de données rappel sur les formes normales

- Formes normales : un enregistrement est « en 3<sup>ème</sup> ssi 2<sup>ème</sup> + tous les éléments hors clés primaire sont indépendants transitivement de celle si »
- *Objectif : définir clairement le contenu d'une table et éviter les redondances (tables cachées dans d'autres tables) !*

ID Produit Clé primaire	Nom du fournisseur	Adresse
1000	Toyota	Park Avenue
1001	Mitsubishi	Lincoln Street
1002	Toyota	Park Avenue



Nom du fournisseur clé primaire	Adresse
Toyota	Park Avenue
Mitsubishi	Lincoln Street

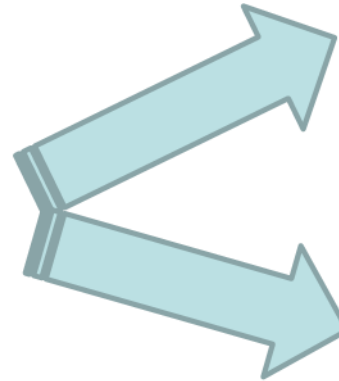
ID Produit clé primaire	Nom du Fournisseur
1000	Toyota
1001	Mitsubishi
1002	Toyota



## Construction d'une base de données rappel sur les formes normales

- Formes normales : un enregistrement est « en Boyce-Codd ssi 3<sup>ème</sup> + les clés sont les uniques sources de dépendance fonctionnelle »
- *Objectif : définir clairement le contenu d'une table et éviter les redondances (tables cachées dans d'autres tables) !*

Clé primaire			
étudiant	matière	enseignant	note
Jules	réseau	Petit	10
Jules	ASGBD	Ploix	12



clé primaire		
étudiant	matière	note
Jules	réseau	10
Jules	ASGBD	12

matière	enseignant
réseau	Petit
ASGBD	Ploix

# *Modèle Logique / Physique*

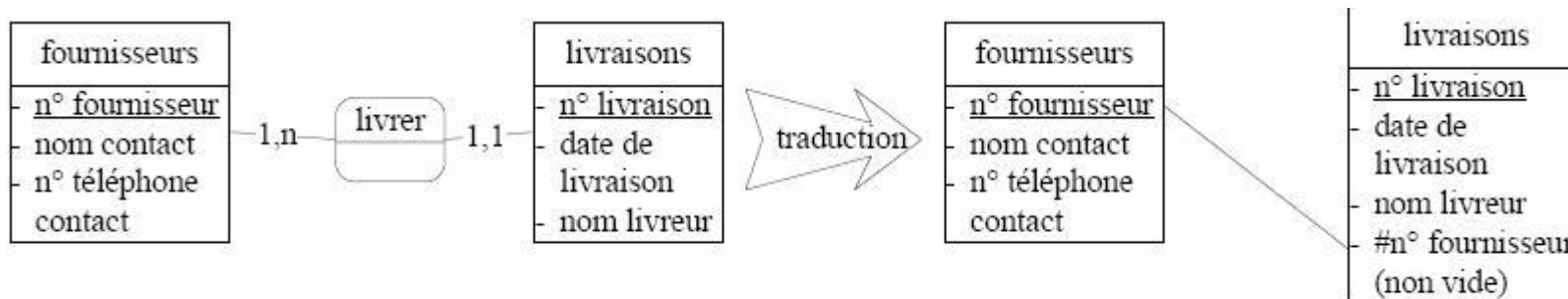
- Introduction :
  - Le modèle conceptuel pour des situations transactionnelles conçu est de type « entité » (nom), « relation » (verbe) et il respecte (à minima) les 3 premières formels normales,
  - Le modèle logique/physique définit les tables et les liens entre les tables via des références (clés étrangères),
  - Il permet d'estimer la taille (physique) nécessaire au stockage des données,
  - Il respecte les caractéristiques de définition des tables propres aux SGBD cibles.

## *Construction du modèle logique / physique à partir du modèle conceptuel*

- Toute entité devient une table dans laquelle les attributs deviennent les colonnes. L'identifiant de l'entité constitue alors la clé primaire de la table.
- Gestion des PK / FK
  - Si pas de clé primaire dans une entité, on construit une clé primaire « technique »
  - Si une clé primaire complexe est référencée, on construit une clé primaire « technique » (optimisation de l'espace)

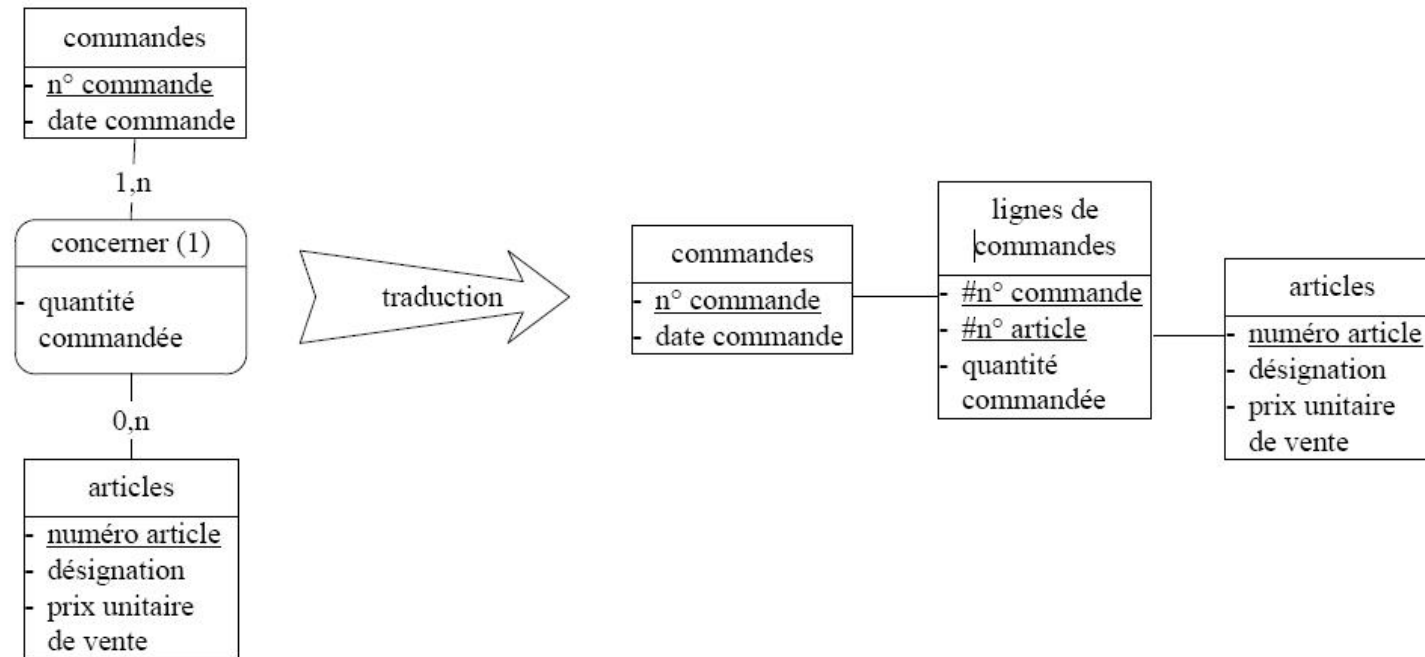
## Construction du modèle logique / physique à partir du modèle conceptuel

- Une association binaire de type 1 : n disparaît, au profit d'une clé étrangère dans la table coté 0,1 ou 1,1 qui référence la clé primaire de l'autre table. Cette clé étrangère ne peut pas recevoir la valeur vide si la cardinalité est 1,1



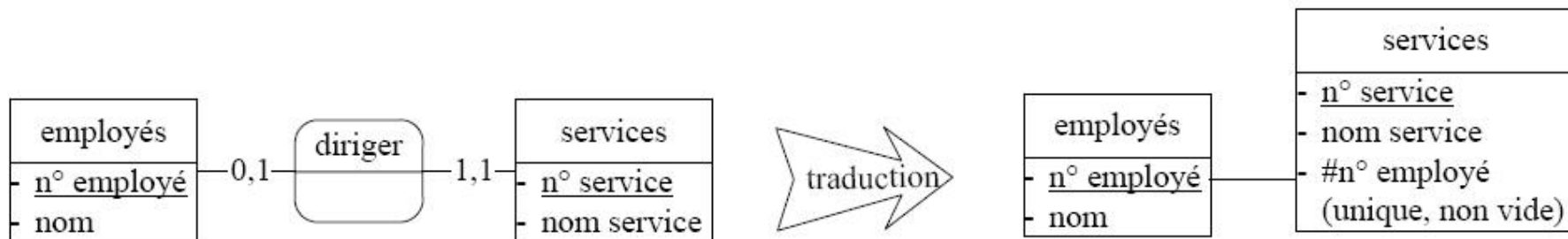
## Construction du modèle logique / physique à partir du modèle conceptuel

- Une association binaire de type n :m devient une table supplémentaire (table de jonction) dont la clé primaire est composée des deux clés étrangères.



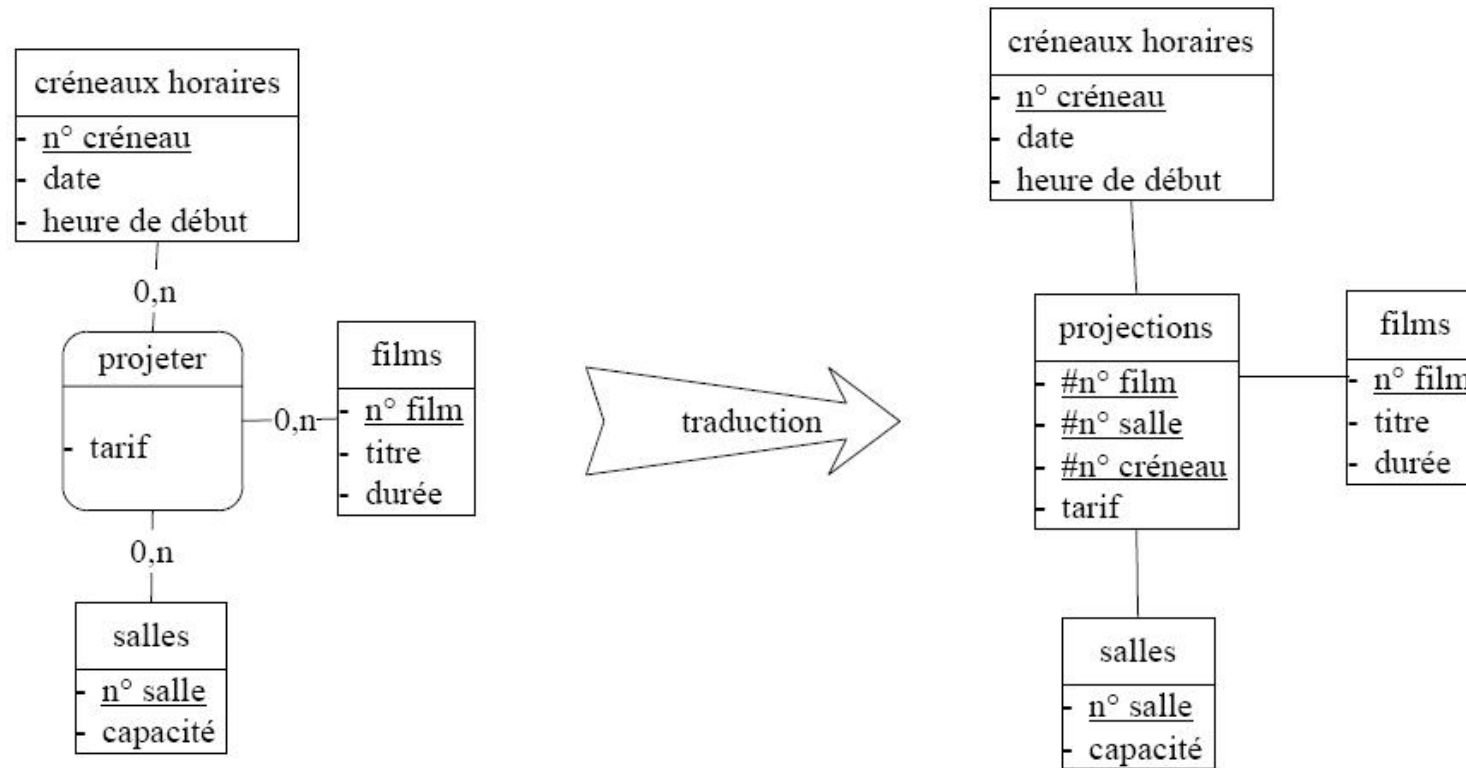
## Construction du modèle logique / physique à partir du modèle conceptuel

- Une association binaire de type 1 :1 est traduite comme une association binaire de type 1 :n sauf que la clé étrangère se voit imposer une contrainte d'unicité en plus d'une éventuelle contrainte de non vacuité (cette contrainte d'unicité impose à la colonne correspondante de ne prendre que des valeurs distinctes).



## Construction du modèle logique / physique à partir du modèle conceptuel

- Une association non binaire est traduite par une table supplémentaire dont la clé primaire est composée d'autant de clés étrangères que d'entité en association. Les attributs de l'association deviennent les colonnes de cette nouvelle table.



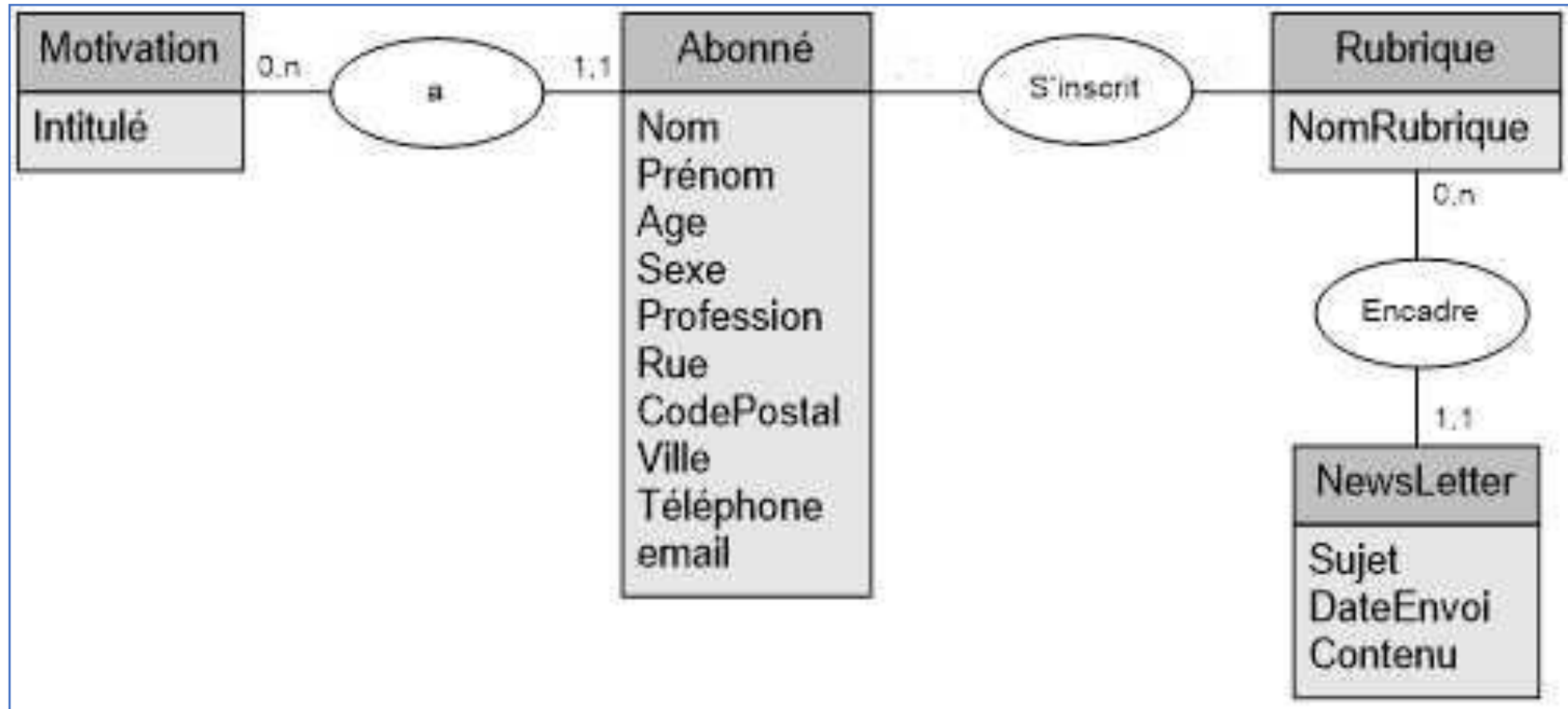
## Construction du modèle logique / types de données

Type OCL	Type SQL2	Oracle	MySQL
Boolean	BIT(1)	Non supporté (utiliser CHAR(1) + CHECK)	BIT, BOOL, BOOLEAN
Integer	INTEGER ou SMALLINT	NUMBER(n)	TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT
String	CHARACTER (CHAR) (n), CHARACTER VARYING (VARCHAR) (n)	VARCHAR2(n), LONG ou LONG VARCHAR (chaîne jusqu'à 2G), CLOB (chaîne jusqu'à 4G), NCLOB (chaîne pour caractères encodés sur plusieurs octets)	CHAR(n), VARCHAR(n), BLOB, TEXT
Real	NUMERIC(p,s) (précision exacte), DECIMAL(p,s), REAL, DOUBLE PRECISION, FLOAT(p)	NUMBER(p,s)	SQL2
Enum {v1,...vn}	CHARACTER (CHAR) ou VARCHAR + CHECK ... IN (v1,...,vn) (possibilité de création de domaine)	Domaine non supporté	ENUM
	DATE	DATE inclut TIME	SQL2
	TIME		SQL2
	TIMESTAMP		SQL2
	BIT(n), BIT VARYING(n)	RAW(n : max = 255), LONG RAW (binaire jusqu'à 2G), BLOB (binaire jusqu'à 4G)	SQL2
		BFILE (pointeur à un fichier externe)	

Référence plus complète : <https://www.databasestar.com/sql-data-types/>



# exemple

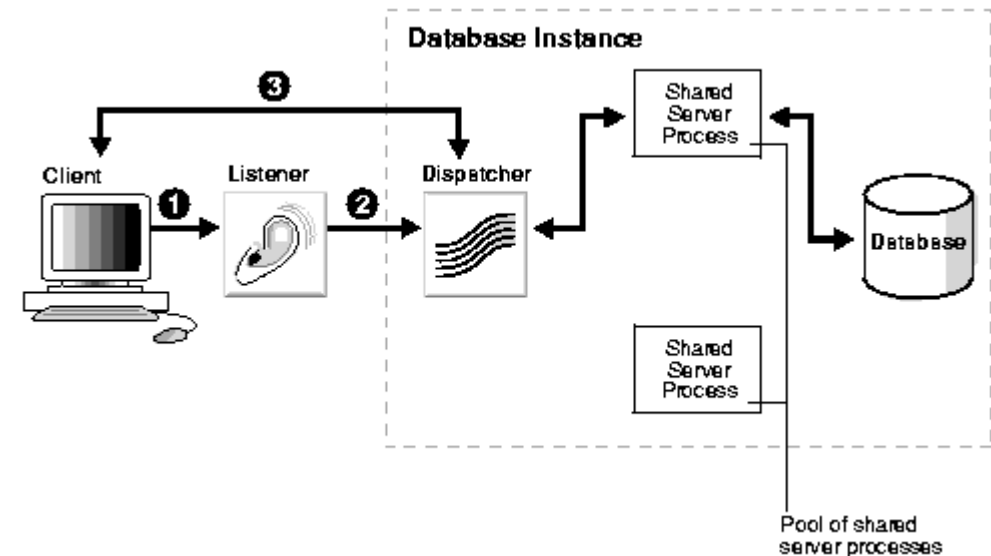


# *Construction d'un SGBD*

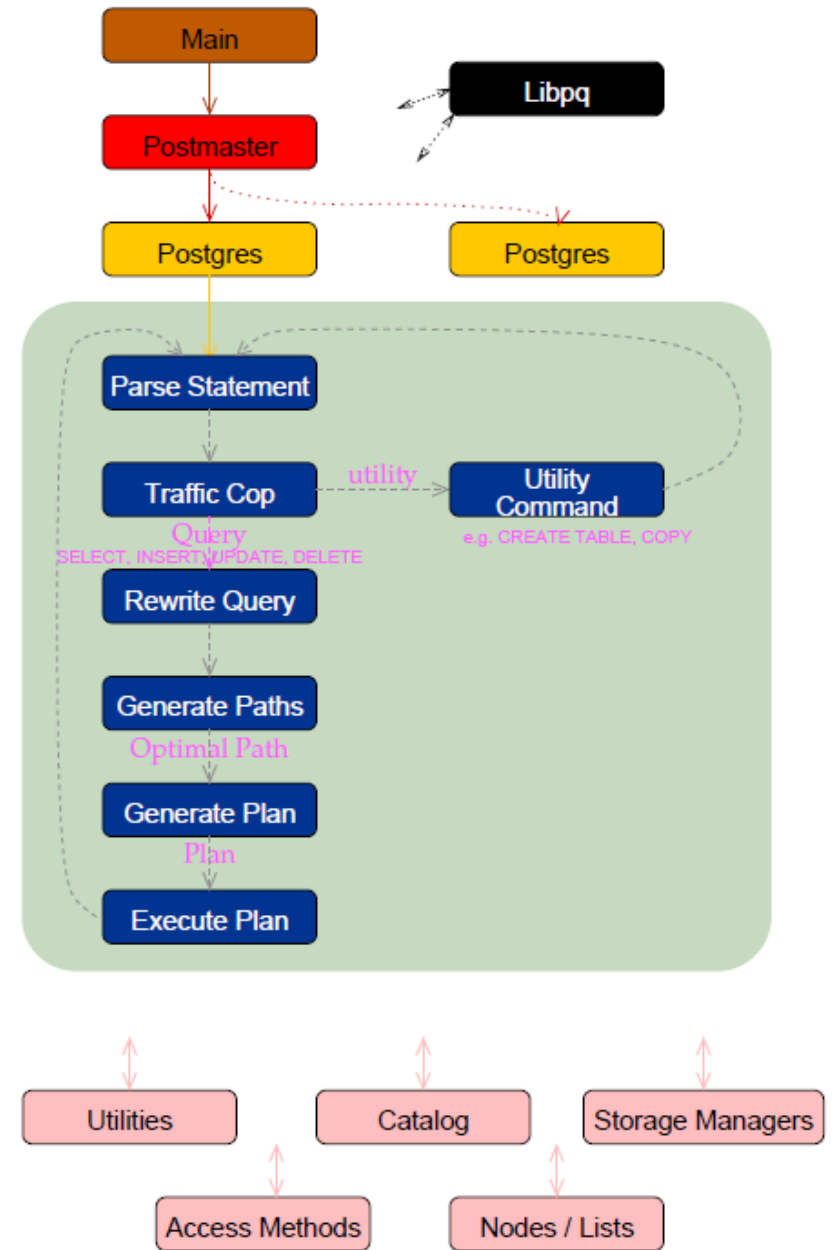
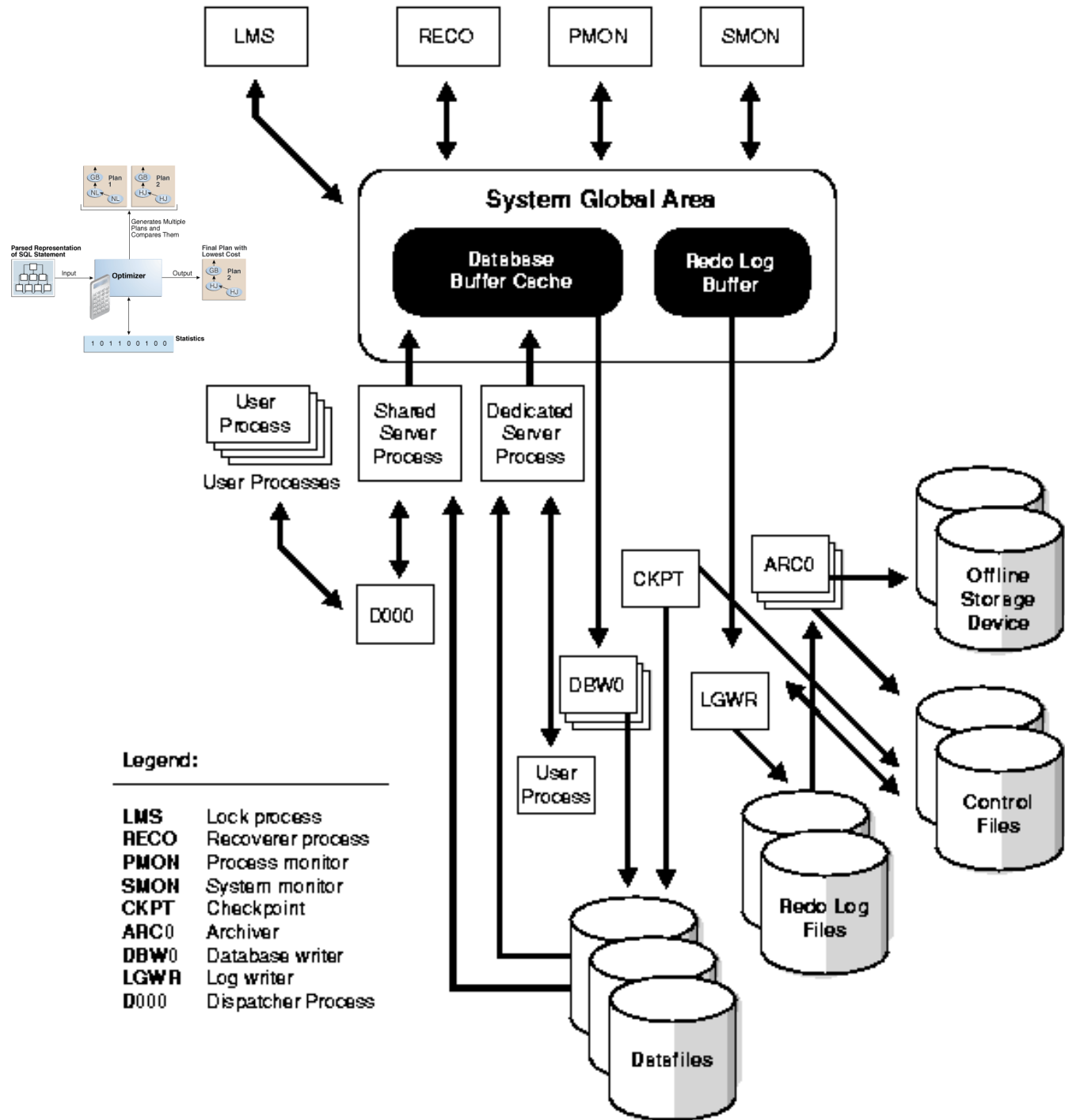
- Application
  - Base/Schéma de données
  - Code
- Composants technique
  - stockage
  - traitement

# Concepts de base / Services de connexion

- Listener – dispatcher – base de données
  - À l'écoute du réseau pour donner accès à une base
  - N instances possibles sur une machine
    - Versions multiples d'oracle
    - Différentes configurations
  - Bascule automatique
    - Client load balancing
    - Connection load balancing
    - Connect time failover
    - Transparent application failover

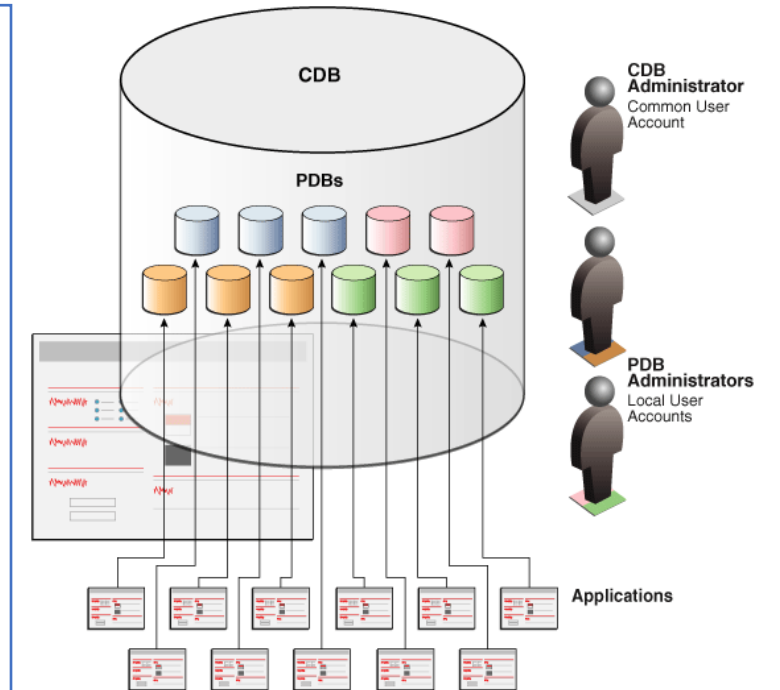


- Porte le rôle de boîtier de répartition de charge (de type BIG-IP par exemple)



# Concepts de base : vocabulaire

- Base Container / Pluggable
  - La base container “contient” des bases de données “enfichables” (pluggable)
- Logiques (Oracle/PostgreSQL):
  - Instance/database = base de données
  - Utilisateurs = qui est connecté
    - A des droits sur des objets présents dans des schémas
  - Schéma = ce qui est stocké
    - ensemble d’objets (tables, vues, indexes, cluster...) qui appartiennent à un utilisateur et qui en porte le nom
- Physiques (Oracle) :
  - Data Blocks, Extents, et Segments
    - organisation des éléments physiques
  - Mémoire SGA et Processus serveurs
    - Contrôle et optimisation du fonctionnement d’Oracle
  - Listener
    - accessibilité depuis le réseau
  - Profils de ressources
- Faisant le lien (Oracle/PostgreSQL)
  - Tablespace = espace de stockage du pdv de ce qui est stocké
    - groupement de structures logiques

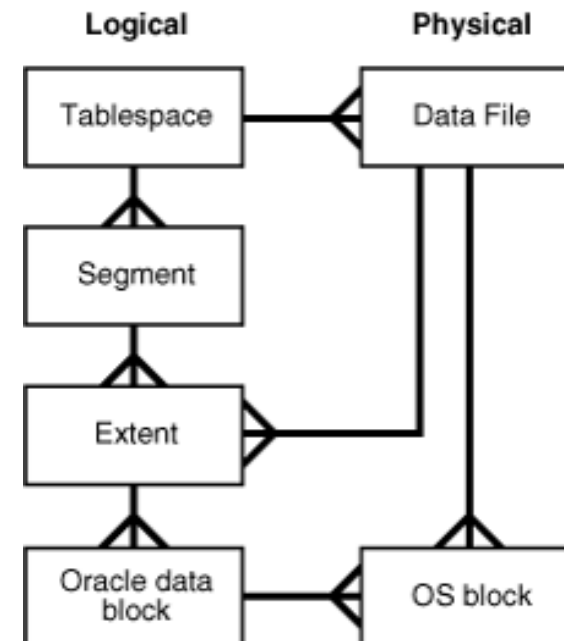
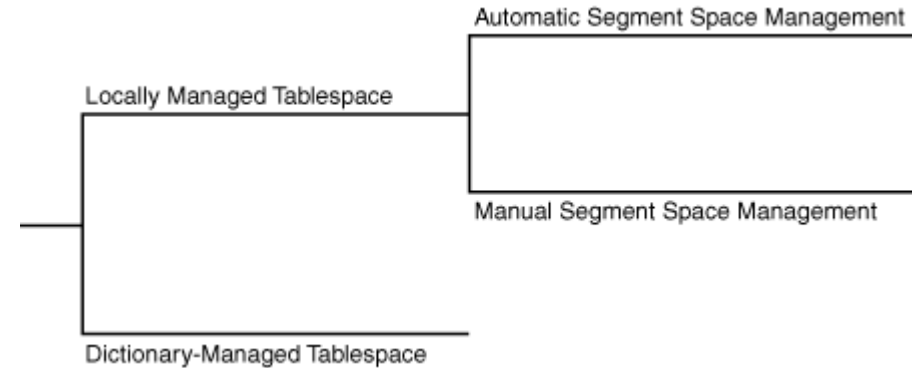


# *Concepts de base / Structure Physique*

- **Datafiles :**
  - Conteneur des données (structurés en blocks, ...)
- **Redo Log files :**
  - Trace dynamique des modifications effectuées sur les données
- **Archive log files :**
  - Archive des traces des modifications effectuées
- **Control files :**
  - Description de la structure physique de la base

# Concepts de base / Stockage

- Stockage logique :
  - Un « espace de stockage » (tablespace) contenant des éléments logiques (applicatifs) et lié à des éléments physiques
- Stockage physique :
  - Fichiers gérés par un FS



## Concepts de base / Utilisateurs & sécurité

- Utilisateurs liés à l'OS :
  - Sous Unix : groupe dba (utilisateur *oracle/postgres*)
  - Sous Windows : groupe ORA\_DBA
  - Peuvent se connecter « CONNECT / AS SYSDBA » sans saisir de mot de passe.
- Utilisateurs liés à un rôle SGBD :
  - Utilisateurs avec le rôle DBA ou autres rôles d'administration
- Utilisateurs liés à des bases de données
  - Utilisateurs propriétaire des éléments de la base de données : tables, procédures stockées, triggers, indexes, ...
    - ➔ définit un « schéma » ayant pour nom le nom de l'utilisateur
  - Utilisateurs accédant aux données d'une base de données (CRUD)
    - ➔ ayant des droits (GRANT) sur les données/objets qu'il doit manipuler



# Sécurité : création des utilisateurs

- Rappel : un « utilisateur » dans Oracle définit un ensemble de droits ainsi qu'un moyen d'accéder à des objets (tables, données, procédures, ...).

Création d'un utilisateur avec des droits minimaux :

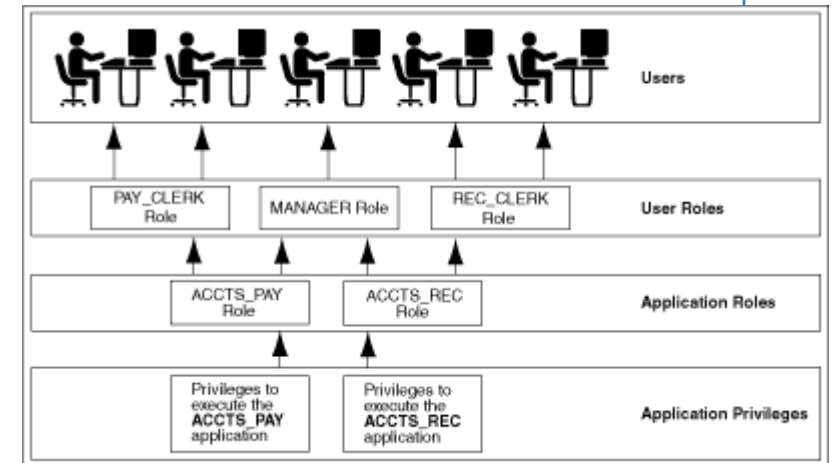
<pre>CREATE USER jward IDENTIFIED BY password DEFAULT TABLESPACE example QUOTA 10M ON example TEMPORARY TABLESPACE temp QUOTA 5M ON system PASSWORD EXPIRE;  GRANT CREATE SESSION TO jward;</pre>	<p>Création de l'utilisateur « jward » Mot de passe = « password » Espace de stockage par défaut « example » Peut ajouter 10M de données dans « example » Espace de stockage nécessaire aux calculs : temp Peut créer 5M de données dans le TBS « system » Il faut changer le mot de passe !</p> <p>L'utilisateur jward à le droit de se connecter.</p>
---	---

Exemple de droits minimaux pour l'utilisateur de migration de données :

<pre>GRANT CREATE SESSION TO DATAREP_USER; GRANT RESOURCE TO DATAREP_USER; GRANT SELECT ANY TABLE TO DATAREP_USER; GRANT SELECT ANY DICTIONARY TO DATAREP_USER; GRANT ALTER SESSION TO DATAREP_USER; GRANT EXECUTE ON DBMS_FLASHBACK TO DATAREP_USER;</pre>	<p>Peut se connecter Insertion de données sans limites dans tous les TSB Peut voir toutes les données de toutes les tables Peut voir toutes les définition des tables Modifie les caractéristiques de la session en cours Gestion des points de sauvegarde</p>
---	--

# Sécurité : principe du moindre privilège

- **Un privilège est conféré à un utilisateur si et seulement si il en a besoin**
- Définition des familles de privilèges (rôles, droits, ...) :
  - Liés aux activités d'administration de la base de données
    - Droit de sauvegarde / restauration,
    - Droit de création de base,
    - Droit de création d'utilisateurs ayant accès à la base,
    - Droit de modification des caractéristiques de la base,
    - ...
  - Liés aux activités des bases de données :
    - Création de tables, d'indexes, ...,
    - CRUD pour chaque attribut de chaque table (cf RGDP),
    - Optimisation des requêtes,
    - ...



➔ Les privilèges sont liées à des **activités**.

➔ Le rôle du responsable d'une activité est de s'assurer que chaque privilège confié à un acteur correspond bien à une mission de cet acteur.

➔ Oracle permet d'attribuer des rôles à des packages (procédures PL/SQL).

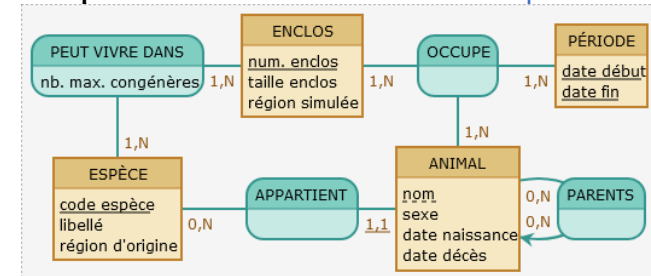
# Sécurité : exemple

- Contexte (sujet 2021) : « Afin de construire une application de gestion de l'ensemble des animaux d'un groupe de zoos, le modèle conceptuel de données a été réalisé. Le groupe gère environ 5 millions d'animaux de 1000 espèces différentes répartis dans 50 000 zoos ayant chacun une centaine d'enclos. Les traitements les plus fréquents prévus pour l'application sont la recherche d'animaux par espèce et la recherche de la catégorie des pièces. Après avoir vérifié que le modèle respecte bien les 3 premières formes normales, décrivez l'ensemble des éléments permettant la création d'une base de données Oracle. »

- On définit les caractéristiques liées à l'application :

- Utilisateur(s) « applicatifs »,
- Tables, types de tables et espaces de stockage.

- Une base de données **container** « dbc\_applications\_zoos » est créée pour accueillir la base de données « pluggable » de l'application.
- Une base de données **pluggable** « LesZoos » liée à l'application est créée dans la base dbc\_applications\_zoos.
- Les espaces de stockage « tbs\_leszoos » et « tbs\_index\_leszoos » sont créés.
- L'utilisateur « ApplicationLesZoos » propriétaire du schéma (tables, indexes, ...) de l'application « LesZoos » est créé.
- Les droits de connexion et de CRUD sans limite sont donnés sur les tbs de l'application.



# Tables Oracle

- Spécifications du modèle de stockage des données (caractéristiques des tables) :
  - Indication du taux de remplissage des blocks de données : influence l'espace disque utilisé
    - PCTFREE 10 : % préservé pour des données future
    - PCTUSED 40 : % d'occupation pour d'être remis dans le pool des blocks libres
    - PCTFREE + PCTUSED ≤ 100
  - Taille de bloc, taille initiale (initial) et de progression (next)
  - Logging ou nologging ?
- Contraintes de validité des colonnes :
  - Valeur contrainte (VALUES) par une liste ou un maximum (LESS THAN)
  - Champ type CONSTRAINT <nom> (NOT NULL | check(<expression>))
  - Constraint <nom> [primary|foreign] key (champ(s)) [deferrable];
  - ...

# Calcul de la taille nécessaire au stockage

- Deux méthodes :
  - Estimations large via des règles de calcul
  - Estimation via des outils :
    - Dans les outils de conception (Toad, ...)
    - Dans Oracle (DBMS\_SPACE)
- Caractéristique pct\_free des tables et indexes :
  - Correspond au % de chaque data block laissé libre en cas de modification des enregistrements.

## Best Practice :

Pour les indexes dont les valeurs indexées changent rarement : 2

Pour les tables dont les valeurs changent rarement : 2

Pour les tables dont les valeurs changent fréquemment : 10 à 30

Il faut toujours laisser un peu (2 min) car Oracle à besoin de cet espace dans certain cas.

## Calcul de la taille nécessaire au stockage règles de calcul d'une estimation globale

Permet l'estimation globale en amont de la création de la base de données

La somme des tailles des attributs donnent la taille maximale d'un enregistrement

Le nombre d'enregistrement à considérer est :

- Le nombre initial (à la création)
- Le taux de croissance doit permettre, si capacity planning il y a, d'optimiser l'occupation d'espace

Table ETUDIANT

Enregistrement	Type	Taille Max	Taille Moyenne
NOM	VARCHAR2	100	10
PRENOM	VARCHARS	100	10
ID_ETUDIANT	NUMBER	10	10

Taille Enregistrement 30

Nombre Initial 1 000 000

Taille Table 30 000 000 (30 Mo)

Croissance 10 % par an, taille à 5 ans : 48 315 300 (50 Mo)

Prise en compte du pct\_free de 10 % : 55 Mo

## Calcul de la taille nécessaire au stockage règles de calcul d'une estimation globale

```
SET SERVEROUTPUT ON;
```

```
DECLARE
```

```
v_used_bytes NUMBER(10);
```

```
v_Allocated_Bytes NUMBER(10);
```

```
v_type sys.create_table_cost_columns;
```

```
BEGIN
```

```
v_Type := sys.create_table_cost_columns(  
sys.create_table_cost_colinfo('NUMBER',9),  
sys.create_table_cost_colinfo('VARCHAR2',50),  
sys.create_table_cost_colinfo('VARCHAR2',15),  
sys.create_table_cost_colinfo('DATE',NULL),  
sys.create_table_cost_colinfo('DATE',NULL)  
);
```

```
DBMS_SPACE.CREATE_TABLE_COST('USERS',v_Type,10000,7,v_used_Bytes,v_Allocated_Bytes);
```

```
DBMS_OUTPUT.PUT_LINE('Used Bytes: ' || TO_CHAR(v_used_Bytes));
```

```
DBMS_OUTPUT.PUT_LINE('Allocated Bytes: ' || TO_CHAR(v_Allocated_Bytes));
```

```
END;
```

Used Bytes: 696 320

Allocated Bytes: 720 896

→ taille utilisée par 10 000 enregistrements de la table (théorique de 900 000)

→ taille occupée (% free à 7), surcoût de 3,5 %.

## Calcul de la taille nécessaire au stockage règles de calcul d'une estimation globale

```
declare
  calc_used_bytes NUMBER;
  calc_alloc_bytes NUMBER;
begin
  DBMS_SPACE.CREATE_INDEX_COST (
    ddl => 'create index BOOK_CAT on BOOKSHELF (CategoryName) tablespace
BOOKS_INDEX',
    used_bytes => calc_used_bytes,
    alloc_bytes => calc_alloc_bytes
  );
  DBMS_OUTPUT.PUT_LINE('Used bytes = ' || calc_used_bytes);
  DBMS_OUTPUT.PUT_LINE('Allocated bytes = ' || calc_alloc_bytes);
end;
```

Imprimera la taille nécessaire au stockage de l'indexe sur la colonne CategoryName de la table BOOKSHELF dans le tablespace BOOKS\_INDEX

Utile, mais uniquement si la base existe déjà et qu'il y a déjà des données dans la table... donc en cas d'ajout d'un indexe sur une table existante



## *Mais aussi ...*

- Après la conception logique (fonctionnelle) et le travail sur les tables, le DBA va pouvoir :
  - Identifier les index à ajouter et leur type
  - Identifier les tables à partitionner
  - Identifier les contentions à gérer
  - Identifier la segmentation des objets et percevoir les TBS
  - Identifier les paramètres à spécialiser pour l'instance Oracle (Vx)
  - Identifier les paramètres à spécialiser pour les sessions Oracle (Vx)
  - Identifier les tables à « suivre » de près
  - Identifier les futures entretiens et leur rythme
  - Cibler les exports, en plus des sauvegardes

# Concepts de base / objets oracle

## Liens entre bases de données :

- Database links

## Éléments de stockage des données :

- Tables et index-organized tables
- Clusters de tables
- Indexes et index types
- Dimensions
- Views
- Materialized views et materialized view logs
- Object tables, object types, et object views

## Opérateurs internes :

- Sequences
- Synonyms
- Operators

## Calculs intégrés dans la base via :

- Du PLSQL :
  - Stored functions, procedures, et packages
  - Et triggers
- Du Java :
  - Java classes, Java resources, et Java sources
- Des bibliothèques externes :
  - External procedure libraries

# *Concepts de base Oracle vs MySQL et SQL Server*

- Une instance Oracle ⇔ un serveur MySQL
- Les objets d'une applications sont
  - Dans les schémas des utilisateurs Oracle de l'application et répartis dans les différents tablespaces eux-mêmes constitués de fichiers de données.
  - Dans une « base de données » (MySQL – un répertoire+fichiers / SQL Serveur 1 à N fichiers DBF)
  - Les utilisateurs ont des droits sur les objets de la/des bases gérées par le serveur.
- SQL Serveur est intrinsèquement lié au système (Windows) – alors que les deux autres sont OS indépendants.
- Le client contact
  - Un Listener Oracle qui va lui indiquer la localisation du serveur
  - Un moteur installé sur une machine (MySQL / SQLServer)