



**Université des sciences et de la Technologie Houari Boumediene
USTHB – Alger**

Département d'Informatique

MASTER SYSTÈMES INFORMATIQUES INTELLIGENTS

MASTER INFORMATIQUE VISUELLE

**ARCHITECTURE ET ADMINISTRATION DES
BASES DE DONNÉES**

LA MÉTA-BASE

(LES CATALOGUES)

**ORACLE, SQL SERVER, MYSQL
ET DB2**

PLAN

- Un peu de théorie
- Dictionnaire de données Oracle
- Les catalogues système de SQL SERVER
- Catalogues de MySQL
- Catalogues de DB2 d'IBM

RAPPEL

Administrateur

Requêtes DDL

Compilateur DDL

Utilisateur

Requêtes Interactives

Compilateur de Requêtes

Applications

Programmes

Pré-Compilateur

Commandes DML

Compilateur DML

Compilateur du langage de Prog.

Transactions Compilées

Gestionnaire des données

Processeur de BD

Exécute

BD

Module de contrôle des accès concurrents

Module de Restauration et de recouvrement

UN PEU DE THÉORIE

Une base de données contient des relations de base, des chemins d'accès et des vues.

Le SGBD, pour fonctionner et assurer ses objectifs doit pouvoir retrouver les différentes informations sur ces catégories d'information.

→ CATALOGUES

Les catalogues sont des relations (tables).

Ces relations contiennent des informations sur tous les objets de la base de données .

Sans ces informations le SGBD ne peut pas assurer ses objectifs.

PRINCIPAUX CATALOGUES

- La relation **RELATION** : Chacun de ses tuples décrit une relation de la base.
- La relation **ATTRIBUT**: chacun de ses tuples décrit un attribut d'une relation de la base
- La relation **INDEX**: chacun de ses tuples décrit une relation index créée sur une relation de base ou système.
- La relation **VUE**: chacun de ses tuples décrit une vue de la base.
- La relation **AUTORISATION**: chacun de ses tuples décrit un utilisateur avec les droits qui lui sont accordés sur chaque objet de la base(lecteur/écriture/modification...)

RELATION

- Chaque tuple est un descripteur d'une relation dans la base de donnée.
- Informations incluses :
 - Nom de la relation
 - Type de la relation
 - Taille
 - Cardinalité
 - Date de création,
 - Adresse du premier attribut

Nom relation (nom externe)	Type relation	Ident relation (nom interne)	Taille tuple (nbre octets)	Cardinalité (nombre tuples)	Degré (nombre attributs)	Date création	Version	Adresse 1 ^{er} attribut
Relation	système	1	100	50	09	25/03/02	1	@
Fournisseur	Base	10	110	1000	04	25/03/02	1	@

ATTRIBUT

- Chaque tuple décrit un attribut d'une relation de la base
- *Informations incluses :*
 - *Nom de l'attribut*
 - *Identificateur attribut*
 - *Identificateur relation*
 - *Type attribut*
 - *longueur*
 - *Position de l'attribut dans la table*

Nom attribut (nom externe)	Identificateur attribut	Identificateur relation	Type attribut	longueur	Position attribut dans la relation
Numf	11	02	Integer	06	01
Nomf	12	02	Char	10	02

INDEX

- Chaque tuple décrit UN index créé sur une relation de base ou système
- *Informations incluses :*
 - *Nom de l'index*
 - *Identificateur de la relation indexée*
 - *Type de l'index*
 - *Attributs indexés*

Nom index	Identification relation	Type (primaire , secondaire)	Liste arguments index
XFOUR	02	P	Numf
XSFOUR	02	S	villef

VUE

- Chaque tuple décrit une vue créée sur la base.
- *Informations incluses :*
 - *Identificateur de la vue*
 - *Identificateurs des relations de base utilisées par la vue*
 - *Requête définissant la vue*

Identificateur(s) des relations de base

Identificateur de la vue

Requête qui définit la vue

02

100

Select * from fournisseur where villef ='Tunis'

CATALOGUES - SGBD

- ❖ Les catalogues dépendent du SGBD
- ❖ Chaque SGBD possède une spécificité
- ❖ D'autres catalogues peuvent exister
- ❖ Les structures des catalogues peuvent être différentes selon les SGBDs.

ORACLE

LE DICTIONNAIRE DE DONNÉES ORACLE

- ❖ Organisé comme une base de données (ensemble de tables)
- ❖ La plupart de ses tables ne sont pas modifiables par les utilisateurs (seul l'administrateur de la base système, a accès à ces dernières) .
- ❖ Le contenu de ces tables est accessible par l'intermédiaire des vues.

CLASSEMENT DE VUES DE DICTIONNAIRE DE DONNÉES

USER_XXX

- décrit les objets appartenant à l'utilisateur connecté

ALL_XXX :

- décrit les objets accessibles à tous les utilisateurs

DBA_XXX :

- décrit tous les objets (vues autorisées aux administrateurs de la base seulement...)

V\$_XXX :

- pour le suivi des performances

QUELQUES VUES DE DICTIONNAIRE D'ORACLE

Nom de la vue	contenu
DICTIONARY (DICT)	Vues permettant de consulter le dictionnaire de données
USER_TABLES (TABS)	tables et vues créées par l'utilisateur
ALL_CATALOG (CAT)	Tables et vues sur lesquelles l'utilisateur a des droits (lecture, écriture, modification ...)
USER_TAB_COLUMNS (COLS)	Colonnes de chaque table ou vue créée par l'utilisateur courant
USER_VIEWS	Vues créées par l'utilisateur
USER_INDEXES (IND)	Indexes créées par l'utilisateur ou indexant des tables créées par l'utilisateur
USER_TAB_PRIVS	Objets sur lesquels l'utilisateur est propriétaire, donneur ou receveur d'autorisation
USER_CONSTRAINTS	Définition de contraintes pour les tables d'utilisateur

REQUÊTES UTILES

- ***DESCRIBE table/vue***
- pour avoir tous les attributs de cette table ou vue
- ***select comments from DICT where table_name='nom d'une vue de dictionnaire de données'***
- Utiliser l'attribut *comments* de la vue DICTIONARY pour connaître le contenu de la vue.

QUELQUES EXEMPLES

Structure d'une table

```
SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH, DATA_PRECISION      FROM USER_TAB_COLUMNS WHERE TABLE_NAME =  
'INSTALLER';
```

COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION
NPOSTE	VARCHAR2	7	
NLOG	VARCHAR2	5	
NUMINS	NUMBER	22	5
DATEINS	DATE	7	
DELAI	INTERVAL DAY(5) TO SECOND(2)	11	5

Tous les Objets d'un schéma

```
SELECT OBJECT_NAME, OBJECT_TYPE, CREATED FROM USER_OBJECTS;
```

OBJECT_NAME	OBJECT_TYPE	CREATED
ACCES_SECURISE	PACKAGE	03/09/03
ACCES_SECURISE	PACKAGE BODY	03/09/03
AFFICHEAVIONS	PROCEDURE	03/09/03
Compagnies	JAVA CLASS	17/08/03
EFFECTIFSHEURE	FUNCTION	16/09/03
ESPIONCONNEXION	TRIGGER	16/09/03
PILOTE	TABLE	18/09/03
PK_PILOTE	INDEX	18/09/03
VUEMULTICOMPPIL	VIEW	14/09/03

SQL SERVER

LES CATALOGUES SYSTÈME DE SQL SERVER

- ❖ Un ensemble de vues affichant les métadonnées décrivant les objets dans une instance de SQL Server
- ❖ Les métadonnées ne sont visibles que pour les éléments sécurisables qu'un utilisateur détient ou pour lesquels des autorisations lui ont été accordées.

L'ACCÈS AUX INFORMATIONS DES CATALOGUES SYSTÈME

- ❖ *Affichage catalogue.* (la plus recommandée)
- ❖ **Vues des schémas d'information.**
- ❖ **Ensembles de lignes de schémas OLE DB.**
- ❖ **Fonctions de catalogue ODBC.**
- ❖ **Fonctions et procédures stockées système.**

VUES D'AFFICHAGE CATALOGUE

OBJETS

Vue de catalogue	Contenu
Sys.objects	Contient une ligne pour chaque objet(table sys, table user, constraint check, primary key..)
Sys.tables(hérite de sys.objects)	Toutes les table de la base en cours (définies par l'utilisateur)
sys.columns, sys.computed_columns(calculée) sys.foreign_key_column sys.identity_columns	informations sur toutes les colonnes pour chaque table de la base, les autres vues sont pour des colonnes de type particulier (elles héritent de sys.columns)
sys.foreign_keys sys.foreign_key_columns	Informations sur les contraintes de type clé étrangères
sys.indexes	Information sur les indexes de tables de la base en cours(nom de l'index, la table sur laquelle l'index est créé)
sys.check_constraints, sys.default_constraints, sys.key_constraints	Informations sur les différentes contraintes .

EXEMPLE DE REQUÊTES

Toutes les tables définies par l'utilisateur dans une base de données spécifiée(simple):

Use <database_name>

go

*Select * from sys.tables*

go

MySQL

MYSQL

- ❖ MySQL dispose d'un catalogue système pour l'ensemble des bases de données présentes dans le SGBD.
- ❖ Le nom de cette base, accessible en lecture, est **INFORMATION_SCHEMA**.
- ❖ Une collection de vues systèmes et non pas de tables physiques.
- ❖ Pour se placer dans le catalogue, il suffit de taper la commande suivante : **USE INFORMATION_SCHEMA ;**

CATALOGUES ET DICTIONNAIRE DE DONNÉES

Les informations concernant les bases sont décrites dans la vue système SCHEMATA.

Pour obtenir toutes les informations sur une base de données on utilise l'attribut SCHEMA_NAME.

```
SELECT *  
  
FROM INFORMATION_SCHEMA.SCHEMATA  
  
WHERE SCHEMA_NAME = 'nom_base';
```

LES TABLES

Les métadonnées des tables sont présentes dans la vue système TABLES.

Les informations couramment utilisées dans cette vue système sont :

- TABLE_SCHEMA
- TABLE_NAME
- TABLE_TYPE
- ENGINE
- TABLE_ROWS
- DATA_LENGTH
- MAX_DATA_LENGTH
- INDEX_LENGTH
- AUTO_INCREMENT
- CREATE_TIME : date et heure de création de la table ;
- TABLE_COLLATION

LES COLONNES

**Les métadonnées sur les colonnes (ou champs) sont contenues dans la vue système
COLUMNS.**

Les informations les plus utiles dans cette vue système sont les suivantes :

- TABLE_SCHEMA
- TABLE_NAME
- COLUMN_NAME
- ORDINAL_POSITION
- COLUMN_DEFAULT : la valeur par défaut de la colonne ;
- IS_NULLABLE DATA_TYPE
- CHARACTER_MAXIMUM_LENGTH
- Etc.....

LES INDEX

La vue système STATISTICS contient les métadonnées sur les index

Une partie de cette information se retrouve dans la vue système décrivant les colonnes.

Les informations les plus utiles sont:

- TABLE_SCHEMA
- TABLE_NAME
- NON_UNIQUE
- INDEX_NAME
- SEQ_IN_INDEX
- COLUMN_NAME
- CARDINALITY
- INDEX_TYPE

LES CONTRAINTES

Pour les informations sur les contraintes d'une table, il faut utiliser la combinaison de deux nouvelles vues systèmes de métadonnées :

TABLE_CONSTRAINTS et KEY_COLUMN_USAGE.

La première des deux vues fournit principalement le type de la contrainte, alors que la seconde donne les tables et colonnes qui utilisent cette contrainte.

EXAMPLE

Afficher la liste des clés étrangères d'une table

```
SELECT *
  FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
 WHERE TABLE_SCHEMA = 'nom_base'
   AND TABLE_NAME = 'nom_table'
   AND CONSTRAINT_TYPE = 'FOREIGN KEY';
```

LES PROCÉDURES STOCKÉES, FONCTIONS ET DÉCLENCHEURS

Utilisation de la vue système ROUTINES

Exemple

**Les informations sur les procédures stockées et
fonctions (routines)**

```
SELECT *  
FROM INFORMATION_SCHEMA.ROUTINES
```

LES TRIGGERS :

Le catalogue fournit, depuis la version 5.0.10 de MySQL, des informations sur les déclencheurs (triggers).

Quelques données contenues dans le dictionnaire

- **TRIGGER_NAME**
- **EVENT_MANIPULATION**
- **EVENT_OBJECT_SCHEMA**
- **EVENT_OBJECT_TABLE**
- **ACTION_ORDER**
- **ACTION_STATEMENT**
- **ACTION_ORIENTATION**
- **ACTION_TIMING**
- **DEFINER**

LES PRIVILEGES

Quatre vues systèmes fournissent des informations sur les privilèges globaux sur un serveur MySQL

- *USER_PRIVILEGES* ;
- *SCHEMA_PRIVILEGES* ;
- *TABLE_PRIVILEGES* ;
- *COLUMN_PRIVILEGES*.

➤ La vue système *USER_PRIVILEGES* fournit des informations sur les utilisateurs déclarés. Ces informations sont issues de la table *MYSQL.USER*.

IBM DB2

LES TABLES

SYSCAT.TABLES

- Vue du catalogue contenant des informations sur les objets créés dans une base de données, tels que des tables, des vues, ...

Les informations (champs) couramment utilisées dans cette vue système

- TABSCHEMA
- TABNAME
- TYPE
- ColCount
- INDEXID

LES VUES ET LES COLONNES

SYSCAT.VIEWS

Permet de visualiser une ou plusieurs lignes par vue définie dans une instance de base de données. On peut utiliser cette vue catalogue pour en savoir plus sur les vues. Les informations enregistrées dans cette vue inclut nom de la vue, et la validité de la vue...

Les colonnes

SYSCAT.COLUMNS

- Utilisé pour stocker des informations sur les colonnes dans une table.

LES INDEX

SYSCAT.INDEXES

- Utilisé pour stocker des informations sur les index définis pour une table donnée.

Les informations remarquables dans cette vue système

- INDSHEMA
- INDNAME
- DEFINER
- TABSCHEMA
- TABNAME
- Colnames
- UNIQUERULE.

LES DÉCLENCHEURS

SYSCAT.TRIGGERS

- Utilisé pour stocker des informations sur les déclencheurs définis dans une base de données.

Informations contenues dans la vue

- TRIGSCHEMA
- TRIGNAME
- DEFINER
- TABSCHEMA
- TABNAME
- TRIGTIME
- TRIGEVENET
- GRANULARITE.

AUTRES CATALOGUES UTILES

- **SYSCAT.CHECKS** - contient des informations sur les contraintes de vérification dans un tableau
- **SYSCAT.COLAUTH** - contient des informations sur les droits d'accès des utilisateurs pour une colonne particulière
- **SYSCAT.DATATYPE** - contient des informations sur les types de données intégrés et définis par l'utilisateur disponible dans la base de données
- **SYSCAT.DBAUTH** - des informations d'autorisation des utilisateurs pour une instance de base de données particulière
- **SYSCAT.IND EXAUTH** - Avantages pour les utilisateurs sur un index particulier