

# Bases de Données

## C1-Introduction et Rappels

Lina Soualmia

Université de Rouen  
LITIS - Équipe TIBS-CISMeF  
[lina.soualmia@chu-rouen.fr](mailto:lina.soualmia@chu-rouen.fr)

9 septembre 2015

# Sources

- J.Ullman <http://infolab.stanford.edu/~ullman/>
- J.Darmont <http://eric.univ-lyon2.fr/~jdarmont/>
- M.Manouvrier  
<http://www.lamsade.dauphine.fr/~manouvri/>
- T.Hamon, Univ Paris 13.
- S.Feuerstein, Oracle PL/SQL, guide de programmation.  
O'Reilly (traduit de l'anglais)

# Organisation des séances

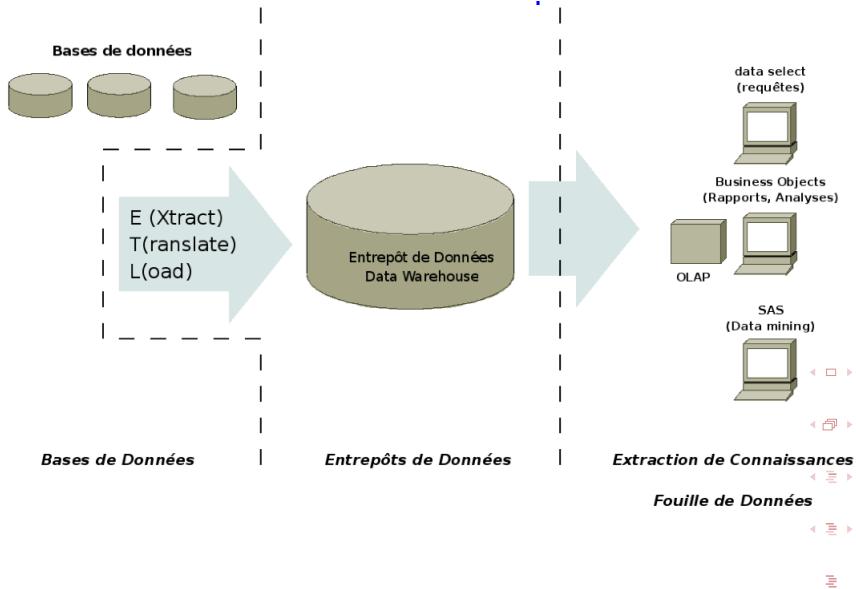
- Cours les mercredis : 9 séances de 2 heures de 8h15 à 10h15
  - ▶ des bases de données aux entrepôts de données
  - ▶ exploitation intelligente des données
- TPs : 12 séances de 2 heures
  - ▶ Mise en œuvre des concepts vus en cours
  - ▶ Oracle
- Évaluations :
  - ▶ Pratique (TPs ET mini-projet) 50%
  - ▶ Écrit (examen) 50%

# Plan

- Introduction et Rappels, Conception et Modélisation
- PL/SQL
- UML et SQL 2/3
- Administration, Structure logique et physique d'une base de données
- Optimisation
- Sécurité, Confidentialité et sauvegarde/restauration

## Programme

- Rappels de SQL
- Conception et modélisation de bases de données
  - ▶ Méta-modélisation, formalismes utilisés (ER, EER, UML ...)
  - ▶ Expression et cohérence des contraintes (SQL2/3, PL/SQL...)
- Implantation de bases de données
  - ▶ Relationnel étendu, orienté objet (de UML à SQL2/3, JDBC, PL/SQL ...)
  - ▶ Optimisation de requêtes, évaluation de requêtes
  - ▶ Architecture et introduction à l'administration de bases de données



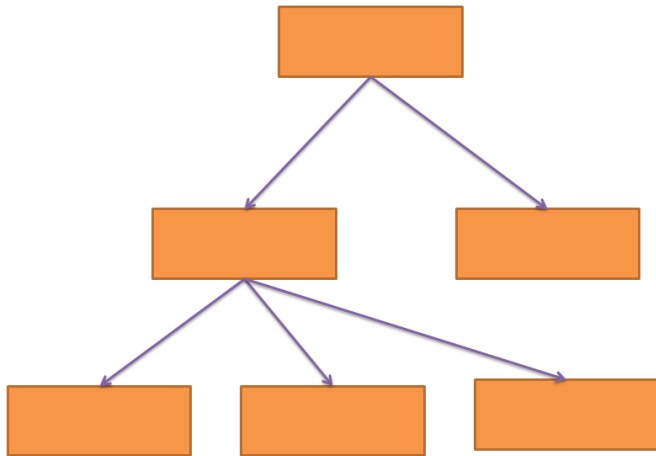
# Rappels : Historique

## Le modèle hiérarchique - Années 60

Premier modèle de SGBD, les données sont classées **hiérarchiquement** selon une arborescence descendante.

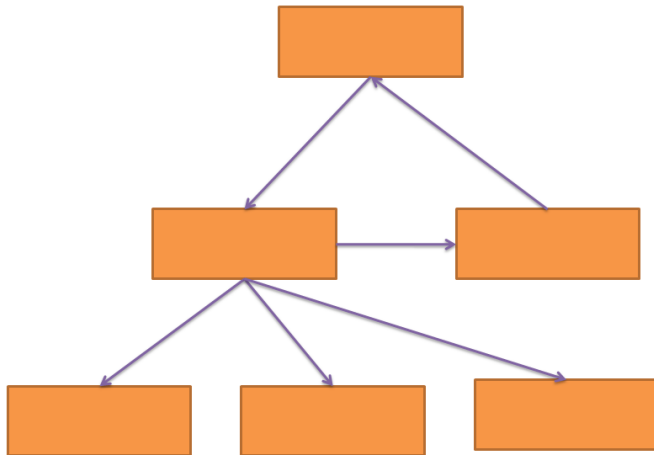
Ce modèle utilise des **pointeurs** entre les différents enregistrements, organisés dans une structure arborescente de façon à ce que chaque enregistrement n'ait qu'un seul possesseur.





## Le modèle réseau - Années 70

- Il lève de nombreuses limites du modèle hiérarchique grâce à la possibilité d'établir des liaisons de type **n-n**, les liens entre objets pouvant exister sans restriction.
- Pour retrouver une donnée dans ce modèle, il faut connaître le chemin d'accès (les liens), ce qui rend les programmes dépendants de la structure de données.
- Comme le modèle hiérarchique ce modèle utilise des **pointeurs** vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant.



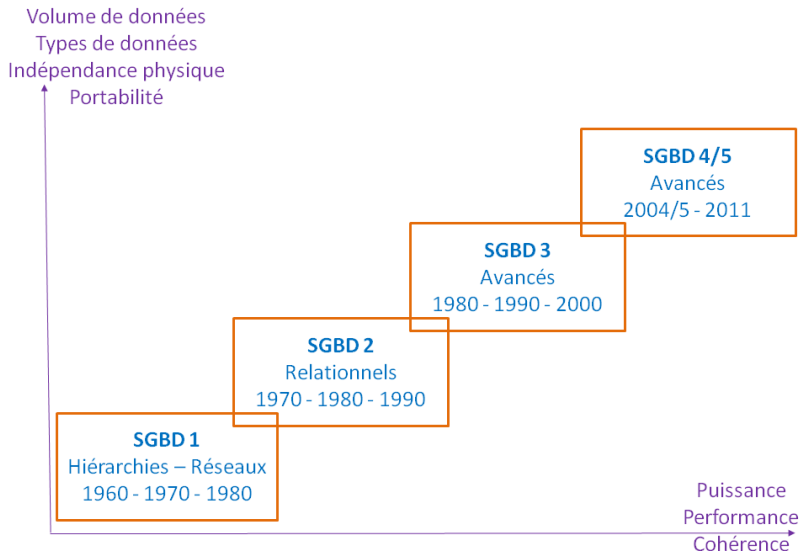
## Le modèle relationnel - Fin 60

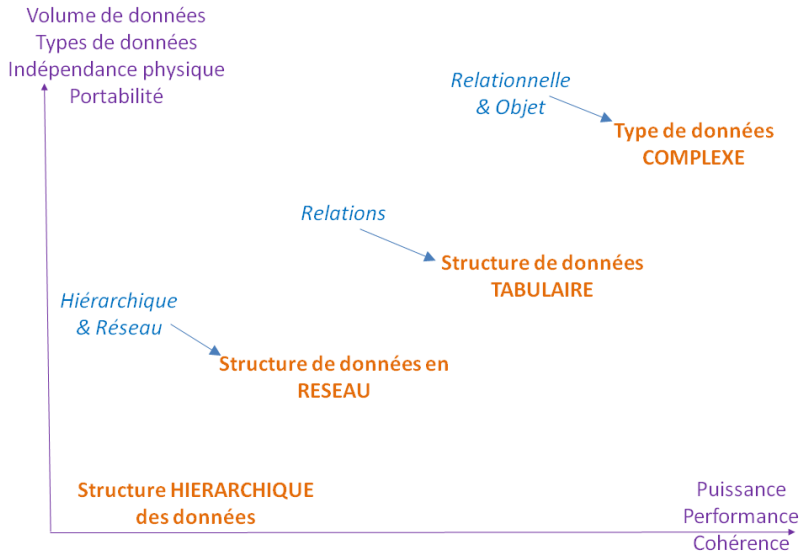
- Le modèle Relationnel (Codd, fin des années 60).
- Les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes) sous forme de **n-uplets**. La manipulation de ces données se fait selon la théorie mathématique des relations.
- Modèle le plus utilisé : celui que nous considérerons dans la suite. A donné lieu au langage SQL, extension de l'algèbre relationnelle, standardisation en 1987.
- Près de 80% des utilisations en entreprise.



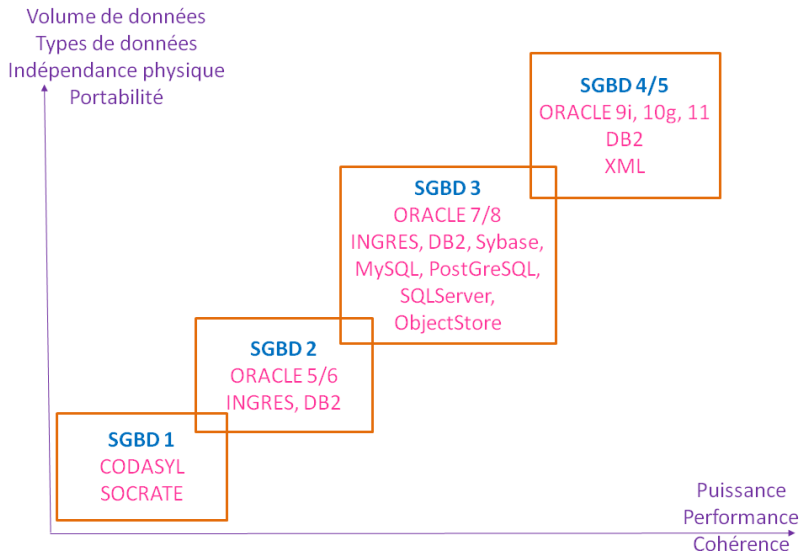
## Autres modèles

- Le modèle **n-uplets** (géré par des OODBMS), les données sont des objets.
- Le modèle **Multidimensionnel**. Les données sont représentées sous la forme d'un cube. Très utilisé en analyse de données et extraction de connaissances.
- Le modèle **Semi-structuré** (fichiers XML). Les données sont représentées sous la forme d'arbre (option M2 GIL).









# Rappels : SQL (pour Oracle)



## Commandes SQL

SQL : Structured Query Language ; langage de gestion de données relationnelles

Plusieurs sortes de commandes parmi lesquelles :

- **LDD** Langage de Définition de données
- **LMD** Langage de Manipulation des Données composé d'un
  - ▶ **LMJ** (Langage de Mise à Jour)
  - ▶ et d'un **LID** (Langage d'Interrogation des Données)
- **LCD** Langage de Contrôle des Données

## Repères historiques

- **1974** SEQUEL (Structured English Query Language)  
ancêtre de SQL
- **1979** premier SGBD basé sur SQL par Relational  
Software Inc. (rebaptisé Oracle)
- **1986** SQL1 1ière norme ISO
- **1989** ajout des contraintes d'intégrité de base (clé  
primaire et étrangère)
- **1992** SQL2 2ième norme extension de SQL1 (nouveaux  
types et nouveaux opérateurs)
- **1999** SQL3 extension de SQL2 (introduction des types  
orientés objet)

- **Oracle** est un SGBD qui utilise SQL
- **PL/SQL** est un langage procédural
- programmes :
  - ▶ **SQL\*PLUS** SQL interactif (TPs)
  - ▶ **SQL\*FORMS** saisie et visualisation des données avec des formulaires
  - ▶ **WebDB** bases interfacées via le Web
  - ▶ **SQL\*Developer** (M2)

## LDD - Langage de Définition des Données

- Ensemble de commandes qui définit une base de données et les objets qui la composent
- La définition d'un objet inclut
  - ▶ sa création : CREATE
  - ▶ sa modification : ALTER
  - ▶ sa suppression : DROP

## Identificateurs

- lettre suivie par : lettre ou chiffre ou `_` ou `#` ou `$`
- chaîne de caractères entre guillemets
- maximum 30 caractères
- différent d'un mot clé `ASSERT`, `ASSIGN`, `AUDIT`, `COMMENT`, `DATE`, `DECIMAL`, `DEFINITION`, `FILE`, `FORMAT`, `INDEX`, `LIST`, `MODE`, `OPTION`, `PARTITION`, `PRIVILEGE`, `PUBLIC`, `SELECT`, `SESSION`, `SET`, `TABLE`
- **pas de distinction** entre majuscules et minuscules

## Tables

- les relations d'un schéma relationnel stockées sous **tables**
- **table** : formée de lignes et de colonnes
- **SQL2** : nom d'une table précédé du nom d'un schéma
- **ORACLE** :
  - ▶ nom de schéma remplacé par le nom d'utilisateur qui a créé la table
  - ▶ par défaut le schéma est le nom de l'utilisateur connecté



## Tables

- toutes les données d'une colonne sont du même type
- identificateur unique pour les colonnes d'une même table
- 2 colonnes dans 2 tables différentes peuvent avoir le même nom
- nom complet d'une colonne comprend le nom complet de la table à laquelle elle appartient
- exemple : `DEPARTEMENTS.DEPARTEMENT_ID` ou `HR.DEPARTEMENTS.DEPARTEMENT_ID`

## Types de données de SQL2 :

- types pour les chaînes de caractères
- types numériques
- types temporels (date, heure, ... )
- SQL2 n'a pas de type pour les données volumineuses (images, sons)
- SQL2 ne permet pas à l'utilisateur de créer ses propres types

## SQL2 : Types pour les chaînes de caractères

- **CHAR**(taille)
  - ▶ chaînes de caractères de longueur fixe
  - ▶ codage en longueur fixe : remplissage de blancs
  - ▶ taille comprise entre 1 et 2000 octets
- **VARCHAR**(taille max)
  - ▶ chaînes de caractères de longueur variable
  - ▶ taille comprise entre 1 et 4000 octets
- **constantes**
  - ▶ chaînes de caractères entre guillemets

## ORACLE : Types pour les chaînes de caractères

- **CHAR**(taille) ou **NCHAR**(taille)
  - ▶ **NCHAR**(5) : chaînes de 5 caractères
- **VARCHAR**(taille max)

MAIS de préférence

- **VARCHAR2**(taille max) ou **NVARCHAR2**(taille max)
  - ▶ **VARCHAR2**(20) : chaînes de 20 caractères au plus
- **constantes**
  - ▶ chaînes de caractères entre côtes
  - ▶ 'Administration', 'Marketing'

## SQL2 : Types numériques

- types numériques pour les entiers :
  - ▶ **SMALLINT** pour 2 octets
  - ▶ **INTEGER** pour 4 octets
- types numériques pour les décimaux à virgule flottante :
  - ▶ **REAL**
  - ▶ **DOUBLE PRECISION** ou **FLOAT**
- types numériques pour les décimaux à virgule fixe :
  - ▶ **DECIMAL**(nb\_chiffres, nb\_décimales)
  - ▶ **NUMERIC**(nb\_chiffres, nb\_décimales)
- constantes
  - ▶ exemples : 43.8, -13, 5.3E-6

## ORACLE : Types numériques

- Oracle accepte tous les types numériques de SQL2 mais il les traduit dans ses propres types
- **NUMBER** : nombre en virgule flottante jusqu'à 38 chiffres significatifs
- **NUMBER**(nb\_chiffres, nb\_décimales) : nombre en virgule fixe

## SQL2 : Types temporels

- **DATE** : pour les dates
- **TIME** : pour les heures, minutes et secondes
- **TIMESTAMP** : pour un moment précis : date et heure, minutes et secondes (précision jusqu'à la microseconde)

## ORACLE : Types temporels

- le type **DATE** : remplace DATE et TIME de SQL2
- **DATE** correspond à une date avec une précision jusqu'à la seconde
- **constantes** : '12/09/2013' ou '12 SEPTEMBER 2013'
- le type booléen n'est pas supporté par ORACLE

## ORACLE : Types pour objets larges

**LOB** : large objet formé par :

- **valeur du LOB** : une grande donnée (jusqu'à 4 Go)
- **index du LOB** : structure d'accès
- le type **localisateur du LOB** pointeur vers l'endroit où il est stocké

### Types pour objets larges

- **CLOB** ou **NCLOB** : pour le stockage de grandes chaînes de caractères
- **BLOB** : pour le stockage de grandes chaînes d'octets
- **BFILE** : pour le stockage de données binaires dans un fichier extérieur à la base



## ORACLE : Autres types

### Les types chaînes d'octets

- **RAW**(taille) : 2000 octets max
- **LONG RAW** : 2 Go max
  - ▶ conversion automatique d'une chaîne d'octets en une chaîne de caractères représentant un nombre hexadécimal, et inversement

### Le type adresse de ligne

- **ROWID** : une valeur de type est un nombre en base 64

### Absence de valeur

- **NULL** : représente l'absence de valeur pour **tous les types** de données. Ce n'est pas une valeur.

## Création de table : contrainte de table

contrainte\_de\_table : :=

[**CONSTRAINT** nom]

type\_de\_contrainte\_de\_table

type\_de\_contrainte\_de\_table : :=

**PRIMARY KEY** (liste de *nom\_de\_colonne*) ou

**NOT NULL** (liste de *nom\_de\_colonne*) ou

**UNIQUE** (liste de *nom\_de\_colonne*) ou

**CHECK** (*condition\_sur\_ligne*) ou

**FOREIGN KEY** liste de *nom\_de\_colonne* **REFERENCES**

*nom\_de\_table* (liste de *nom\_de\_colonne*)

## Suppression de table

**DROP TABLE** nom ;

Quand une table est supprimée, ORACLE :

- efface tous les index qui y sont attachés quelque soit le propriétaire
- efface tous les privilèges qui y sont attachés MAIS les vues et les synonymes se référant à cette table **ne sont pas supprimés**

## Modification de table

**ALTER TABLE** nom\_de\_table modification\_de\_table ;

modification\_de\_table ::= **ADD COLUMN**

définition\_de\_colonne

**ADD CONSTRAINT** contrainte\_de\_table

**DROP COLUMN** nom\_de\_colonne

**DROP CONSTRAINT** nom\_de\_contrainte

## LMD - Langage de Manipulation des Données

- Ensemble de commandes qui permet la consultation et la mise à jour des objets créés par le langage de définition des données
- Consultation : **SELECT**
- La mise à jour inclut :
  - ▶ l'insertion de nouvelles données : **INSERT**
  - ▶ la modification de données existantes : **UPDATE**
  - ▶ la suppression de données existantes : **DELETE**

## Insertion de lignes dans une table

**INSERT INTO** nom\_de\_table [liste\_de\_colonnes] **VALUES**  
liste\_de\_valeurs ;

ou

**INSERT INTO** nom\_de\_table [liste\_de\_colonnes] requête ;

## Suppression de lignes d'une table

**DELETE** [**FROM**] nom\_de\_table [ **WHERE** condition] ;

## Modification de lignes dans une table

**UPDATE** nom\_de\_table **SET** liste expression colonne [**WHERE**  
condition] ;

## Interrogation

- **SELECT** <liste champ(s)> **FROM** <liste nom\_table(s)>  
[**WHERE** condition(s)] [options];

## Interrogation

requête : := **SELECT** [**DISTINCT**] projection  
    **FROM** liste de (*nom\_de\_table* [[**AS**] *nom*])|(requête **AS** *nom*)  
    **WHERE** condition  
    [**GROUP BY** liste de *nom\_de\_colonne*]  
    [**HAVING** condition]  
    [**ORDER BY** liste de ((*nom\_de\_colonne* | *rang\_de\_colonne*)  
(**ASC**|**DESC**))];

requête : := requête (**UNION**|**INTERSECT**|**EXCEPT**) requête



## Les vues

- une vue est une table virtuelle résultat d'une requête
- rôle d'une vue
  - ▶ réduire la complexité syntaxique des requêtes
  - ▶ définir les schémas externes.
  - ▶ définir des contraintes d'intégrité.
  - ▶ définir un niveau additionnel de sécurité en restreignant l'accès à un sous-ensemble de lignes et/ou de colonnes.

**Création d'une vue** `CREATE [OR REPLACE][FORCE|NO FORCE]  
VIEW table [(liste de colonne)] AS requête [WITH CHECK  
OPTION|WITH READ ONLY];`

**Interrogation d'une vue** : comme une table

**Suppression d'une vue** : `DROP VIEW nom_de_vue;`

**Mises à jour** : `INSERT, UPDATE, DELETE`

- Ces instructions ne s'appliquent pas aux vues qui contiennent :
  - ▶ une jointure un opérateur ensembliste : `UNION, INTERSECT, MINUS`
  - ▶ une clause `GROUP BY, CONNECT BY, ORDER BY` ou `START WITH`
  - ▶ la clause `DISTINCT`, une expression ou une pseudo-colonne dans la liste de sélection des colonnes.

## LCD - Langage de Contrôle des Données

- Ensemble de commandes de contrôle d'accès aux données
- Le contrôle d'accès inclut :
  - ▶ l'autorisation de réaliser une opération : **GRANT**
  - ▶ l'interdiction de réaliser une opération : **DENY**
  - ▶ l'annulation d'une commande de contrôle précédente : **REVOKE**
  - ▶ l'autorisation de modifier des enregistrements : **UPDATE**
  - ▶ l'interdiction de modifier des enregistrements : **READ**
  - ▶ l'autorisation de supprimer des enregistrements : **DELETE**

## Sécurité des données

- confidentialité :
  - ▶ gestion des rôles et des utilisateurs
  - ▶ attribution de privilèges aux rôles et aux utilisateurs
  - ▶ définition de filtres (protection de données confidentielles, contrôle d'intégrité)
- pérennité
  - ▶ gestion des transactions
- intégrité
  - ▶ gestion des transactions

- transaction : séquence d'opérations manipulant des données
- vérifient les propriétés suivantes :
  - ▶ atomicité
  - ▶ cohérence
  - ▶ indépendance
  - ▶ permanence
- contrôle des transactions :
  - ▶ **COMMIT** : valide la transaction en cours
  - ▶ **ROLLBACK** : annule la transaction en cours

## Gestion des utilisateurs et des privilèges : rôle

- création de rôle
  - ▶ `CREATE ROLE nom_de_rôle [IDENTIFIED BY mot_de_passe];`
- ajout, modification, suppression de mot de passe
  - ▶ `ALTER ROLE nom_de_rôle [IDENTIFIED BY mot_de_passe];`
- suppression de rôle
  - ▶ `DROP ROLE nom_de_rôle;`

## Gestion des utilisateurs et des privilèges : utilisateurs

- création d'utilisateur
  - ▶ **CREATE USER** nom\_utilisateur [**IDENTIFIED BY** mot\_de\_passe] ;
- ajout, modification, suppression de mot de passe
  - ▶ **ALTER USER** nom-utilisateur [**IDENTIFIED BY** mot\_de\_passe] ;
- suppression de rôle
  - ▶ **DROP USER** nom\_utilisateur ;

## Gestion des utilisateurs et des privilèges : privilèges

- attribution de privilèges
  - ▶ `GRANT` `systeme_privileges`|`ALL` [`privileges`]  
`TO` `liste_roles_utilisateurs`|`PUBLIC`  
`[WITH ADMIN OPTION]` ;
- `systeme_privileges` :
  - ▶ `CREATE ROLE`
  - ▶ `CREATE SEQUENCE`
  - ▶ `CREATE SESSION`
  - ▶ `CREATE SYNONYM`
  - ▶ `CREATE TABLE`
  - ▶ `CREATE USER`
  - ▶ `CREATE VIEW`



## Gestion des utilisateurs et des privilèges : privilèges

- attribution de privilèges sur des objets oracle
  - ▶ **GRANT** liste-droits  
**ON** nom-composant  
**TO** liste-roles-utilisateurs  
**[WITH GRANT OPTION]** ;
- liste\_droits : **SELECT, INSERT, UPDATE, DELETE, ALTER, ALL [PRIVILEGES]**

- suppression de privilèges
  - ▶ **REVOKE** liste\_système\_privileges  
**FROM** liste\_roles\_utilisateurs ;
- suppression de privilèges sur des objets oracle
  - ▶ **REVOKE** liste\_privileges  
**ON** nom\_composant  
**FROM** liste\_roles\_utilisateurs ;

- attribution de rôles
  - ▶ **GRANT** liste-roles  
    **TO** liste-roles-utilisateurs  
    **[WITH ADMIN OPTION]** ;
- suppression de rôles
  - ▶ **REVOKE** liste-roles  
    **FROM** liste-roles-utilisateurs ;

À suivre :

- C2 : Conception et Modélisation