



## R2.06 - BD2 - TD-TP-01-02

### Objectifs

- Comprendre et analyser la base de données : Magasin Réparation Vélos
- Construction du modèle entité-association
- Étude des dépendance de référence
- Étude des contraintes complexes de validation des données
- Écriture des fichiers de création de la base de données : ddl et dr
- Utilitaires de la base de données
- Utilisation de la méta-base pour extraire des informations concernant la base de données
- Génération de rapport HTML : documentation de la base de données
- Séquences et application à la base de données
- Construction des utilitaires de la base de données

### 1 Comprendre et analyser la base de données

L'enseignement R2.06-BD2 s'appuie sur un exemple d'application portant sur la réparation de vélos. Nous avons fourni une spécification détaillée de cette base de données.

#### Travail à faire

- Lire et comprendre la spécification de la base de données
- Analyser les dictionnaires des tables et des colonnes de la base de données

## 2 Modélisation de la base de données

### Travail à faire

- Construire un modèle entité-association
- Établir le graphe des dépendances de référence

## 3 Création de la base de données

Dans cette partie, nous allons étudier l'ordre SQL de création de la table `reparation` en prenant compte l'ensemble des contraintes statiques définies sur cette table :

1. prise en charge de la clé primaire `id_reparation`
2. prise en charge de la clé unique `date_debut_prevue, velo`
3. prise en charge de la dépendance de référence :  
`reparation (velo) ref velo (id_velo)`
4. attributs facultatifs : `date_debut_reelle, date_fin_reelle`
5. prise en charge du domaine des attributs :
  - (a) valeur par défaut de `cout_total` est 0
  - (b) valeur par défaut de `etat` est 'ouverte'
  - (c) domaine de valeurs de `etat` = 'ouverte', 'encours', 'annulee', 'terminee', 'reglee'
  - (d) `date_fin_prevue >= date_debut_prevue`
  - (e) si `date_debut_reelle` est connue alors  
`date_debut_reelle >= date_debut_prevue`
  - (f) si `date_fin_reelle` est connue alors :
    - `date_debut_reelle` est connue
    - `date_fin_reelle >= date_debut_reelle`

### Travail à faire

Écrire le DDL de la table `reparation` en respectant son dictionnaire de données et ses contraintes d'intégrité.

## 4 Utilisation de la méta-base pour extraire des informations concernant la base de données

### 4.1 Information sur les utilisateurs : `_USERS`

Ci-dessous un extrait de la description de la table `ALL_USERS` de la méta-base. Les tables `DBA_USERS` et `USER_USERS` sont en relation avec la table `ALL_USERS`.

Nom	NULL ?	Type
-----	-----	-----
USERNAME	NOT NULL	VARCHAR2(128)
USER_ID	NOT NULL	NUMBER
CREATED	NOT NULL	DATE

1. Donner l'ordre SQL qui compte le nombre des utilisateurs de la BD INFO1.
2. Donner l'ordre SQL qui vérifie que vous faites parti des utilisateurs de la BD INFO1 et qui affiche la date de création de votre utilisateur.

## 4.2 Information sur les tables : \_TABLES

Ci-dessous un extrait de la description de la table ALL\_TABLES de la méta-base. Les tables DBA\_TABLES et USER\_TABLES sont en relation avec la table ALL\_TABLES.

Nom	NULL ?	Type
-----	-----	-----
OWNER	NOT NULL	VARCHAR2(128)
TABLENAME	NOT NULL	VARCHAR2(128)

1. Donner l'ordre SQL qui affiche les noms des tables de votre schéma en utilisant : ALL\_TABLES et USER\_TABLES.
2. Donner l'ordre SQL qui affiche les noms des tables de votre schéma en utilisant : TABS et TAB.

## 4.3 Information sur les contraintes définies sur les tables

1. Donner l'ordre SQL qui affiche les contraintes d'unicité définies dans votre schéma.  
Schéma de la relation résultat : RES(TABLE, CONTRAINTE, TYPE, COLONNE)
2. Donner l'ordre SQL qui affiche les contraintes de domaine (CHECK) définies dans votre schéma.  
Schéma de la relation résultat : RES(TABLE, CONTRAINTE, TYPE, COLONNE, COND)
3. Donner l'ordre SQL qui affiche les contraintes de dépendance de référence définies dans votre schéma.  
Schéma de la relation résultat : RES(TABLE, RTABLE, CONTRAINTE, TYPE, COLONNE, RCOLONNE)

## 4.4 Description de la base de données

1. Donner l'ordre SQL permettant d'extraire le tableau « Dictionnaire des tables de la base de données ».
2. Donner l'ordre SQL permettant d'extraire le tableau « Dictionnaire des colonnes des tables de la base de données ».

## 5 Séquences et application à la base de données

### 5.1 Création et utilisation des séquences

Soit l'ordre SQL de création de la table `test1` :

```
create table test1 (  
    id                number,  
    description       varchar2(30),  
    constraint pk_test primary key (id)  
);
```

1. Donner les ordre SQL de création de la séquence nommée `seq_test1` tels que :
  - l'incrémentation égale à 1 ;
  - le départ du compteur égale à 1 ;
  - la valeur minimale du compteur égale à 0 ;
  - interdire la séquence à reprendre le compteur à partir de son départ ;
  - mettre 25 valeurs dans le cache.
2. Donner l'ordre SQL pour obtenir la valeur courante et la valeur suivante de la séquence `seq_test1`
3. Donner l'ordre SQL pour définir la séquence `seq_test1` comme valeurs par défaut pour l'attribut `test1.id`
4. Donner un programme de test de la relation `test1` mettant en œuvre la séquence `seq_test1` comme générateur par défaut des valeurs de l'attribut `test1.id`
5. Donner l'ordre SQL de création de la table `test2` (même schéma que la table `test1`) en définissant les valeurs de l'attribut `test2.id` générées automatiquement.
6. Donner un programme de test de la relation `test2` mettant en œuvre la génération automatique des valeurs de l'attribut `test2.id`

### 5.2 Séquences : application à la base de données

1. Quelles sont les séquences à définir sur le schéma de la base de données ?
2. Donner les ordres SQL de création de ces séquences.
3. Quels impacts auront la définition de ces séquences sur le DDL de la base de données ?

## 6 Travail à faire en TP

Dossier de travail : BD2/TP-01-02
-----------------------------------

### 6.1 Création de la base de données : Magasin Réparation Vélos

Les ressources du TP-01-02 sont disponibles sur la plateforme MOODLE, cours BUT-INFO-S2-R2.06.

1. Créer le dossier : `ddl`. Il doit contenir les fichiers suivants.
2. `ddl-Magasin-Reparation-Velos.sql` : le ddl de la base de données. Il contient les ordres sql de création des tables de la base de données en tenant compte des contraintes d'intégrité :
  - valeur obligatoire/facultative ;
  - clé primaire et clé unique ;
  - contrainte du domaine des valeurs des attributs (check).
3. `dr-Magasin-Reparation-Velos.sql` : le dr de la base de données. Il contient les ordres sql de création des contraintes de dépendance de référence sur les tables.
4. `seq-Magasin-Reparation-Velos.sql` : ce fichier contient les ordres SQL de création des séquences à définir sur la base de données.
5. `ddl-modif-Magasin-Reparation-Velos.sql` : ce fichier contient les ordres SQL pour mettre à jour les tables de la base de données afin de prendre en compte les définitions des séquences.
6. Exécution des programmes SQL, dans l'ordre :
  1. `seq-Magasin-Reparation-Velos.sql`
  2. `ddl-Magasin-Reparation-Velos.sql`
  3. `dr-Magasin-Reparation-Velos.sql`
  4. `ddl-modif-Magasin-Reparation-Velos.sql`

### 6.2 Génération de la documentation HTML

Les ressources de documentation de la base de données sont disponibles sur la plateforme MOODLE, cours BUT-INFO-S2-R2.06, dossier TP-01-02/dict.

1. Créer le dossier : `dict`. Il doit contenir les fichiers suivants.
2. `def-dict-tables-Magasin-Reparation-Velos.sql` : le programme de définition des descriptions de toutes les tables de la base de données.
3. `def-dict-cols-Magasin-Reparation-Velos.sql` : le programme de définition des descriptions de toutes les colonnes de chaque table de la base de données.

4. `extract-dict-tables-Magasin-Reparation-Velos.sql` : programme d'extraction des descriptions de toutes les tables de la base de données.
5. `extract-dict-cols-Magasin-Reparation-Velos.sql` : programme d'extraction des descriptions des colonnes de chaque table de la base de données.
6. Exécution des programmes SQL, dans l'ordre :
  1. `def-dict-tables-Magasin-Reparation-Velos.sql`
  2. `extract-dict-tables-Magasin-Reparation-Velos.sql`
  3. `def-dict-cols-Magasin-Reparation-Velos.sql`
  4. `extract-dict-cols-Magasin-Reparation-Velos.sql`
7. Afficher les fichiers HTML résultats.

### 6.3 Création des utilitaires de la base de données

Des fichiers modèles des utilitaires sont disponibles sur la plateforme MOODLE, cours BUT-INFO-S2-R2.06, dossier TP-01-02/utilitaires.

1. Créer le dossier : `utilitaires`. Il doit contenir les fichiers suivants.
2. `vide-contenu-base-Magasin-Reparation-Velos.sql` : suppression des nuplets des tables de la base de données.
3. `affiche-contenu-base-Magasin-Reparation-Velos.sql` : affiche le contenu des tables de la base de données.
4. `presentation-Magasin-Reparation-Velos.sql` : ordres SQLPLUS COLUMN pour la présentation du contenu des colonnes de la base de données de type VARCHAR.
5. `modele-insertion-Magasin-Reparation-Velos.sql` : modèle d'insertion pour la base de données.