

---

## OBJECTIF

---

Prendre en compte différentes évolutions de la base de données, en effectuant diverses manipulations sur la structure de cette base. Il s'agit, également, d'utiliser les opérations de mises à jour des données.

---

## BASE DE DONNEES

---

Vous travaillerez sur la base de données *voyage*. Le script se trouve sur la plateforme, ouvrez-le et générez la BDD sous Postgres, la BDD est créée dans le schéma *TP2\_bd\_voyage*.

Le modèle de données est fourni en page 3

---

## TRAVAIL A FAIRE

---

### **1 PREMIERE ETAPE : MODIFICATIONS DE LA STRUCTURE DE LA BASE**

---

#### **1.1 AJOUT DE COLONNE**

On souhaite intégrer dans la relation *planning* un deuxième tarif de voyage pour les enfants. Effectuer l'ajout de cet attribut *tarif enf* dans la relation en utilisant le même type de données que pour la colonne *tarif*. De manière similaire, le nombre d'enfants doit être ajouté dans la relation *reservation*. Donnez à cet attribut *nbenf* le même type que celui de *nbpers*.

#### **1.2 MODIFICATION DE TYPE D'UN COLONNE**

On souhaite étendre la définition du type de l'attribut *libelle* de la table *optionv*, en augmentant la taille du nombre de caractères admissibles de 10 caractères. Consultez la définition de cet attribut puis effectuez la modification correspondante et vérifiez-la.

#### **1.3 CONTRAINTE DE VERIFICATION DE DOMAINE**

1. On souhaite intégrer dans la base une contrainte de domaine pour l'attribut *categorie* de la table *client* qui prendra ses valeurs suivantes : {PRIVILEGIE, BON, MAUVAIS}.  
On va pour cela créer un domaine de valeur (que vous nommerez *DOM\_cat\_cli*), puis modifier le type de l'attribut *categorie* pour qu'il prenne le type du domaine précédemment créé.  
Effectuez cet ajout de contrainte.
2. Modifier la valeur par défaut de la colonne *categorie* de la table *client* pour qu'elle soit MAUVAIS. Vous ferez en sorte que les valeurs NULL soient remplacées par la valeur MAUVAIS
3. Spécifiez une contrainte de valeur pour l'attribut *nbetoiles* dans la table *voyage* afin que ses valeurs appartiennent à l'intervalle [1 .. 5]. (Sans création de domaine cette fois ci !!!), utiliser le alter Table T1 ADD constraint ck\_nom check .....

# Création – modification de la structure d'une BDD et manipulation de données sous Postgres

**TP2.1**

R2.06 exploitation d'une base de données

## 1.4 AJOUT DE TABLE (RELATION)

On souhaite intégrer dans la base une nouvelle relation permettant de connaître par hôtel et type de chambre, la capacité offerte (nombre de chambres). Des modifications sont apportées pour intégrer cette demande, le diagramme de classe en annexe 1 illustre ces modifications :

Vous devez réaliser le script qui permet ces modifications, les colonnes de la table voyage qui doivent disparaître le seront uniquement après transfère des données correspondantes dans la table HOTEL.

Le principe est simple :

1. Ajout des tables *capacite* et *hotel* (on peut créer un type énuméré pour le type de chambre)
2. Création des colonnes clés étrangères (attention de ne pas oublier la table voyage qui doit avoir un lien vers l'hôtel)
3. Transfère des données de la table *VOYAGE* vers la table *HOTEL* (attention aux doublons)
4. Valorisation de la colonne *id\_hotel* de la table *VOYAGE*, Il faut utiliser un update, avec un select corrélé dans le SET
5. Vérification des valeurs avant de supprimer les colonnes de la table *VOYAGE*
6. Supprimer les colonnes de la table *VOYAGE*
7. Insertion des données dans la table *CAPACITE* (on peut utiliser une table temporaire)

Insérer les tuples dans la relation *CAPACITE* qui sont donnés dans le fichier *capacite\_hotel.csv*

**Attention** : c'est à vous de trouver l'id de l'hôtel avec le nom de l'hôtel, les noms sont dans la table hôtel ainsi que leur identifiant.

**Si vous utiliser une table temporaire, vous pouvez importer le fichiers csv dans la table temporaire**



## 2 DEUXIEME ETAPPE : MISES A JOUR DES DONNEES

*Sur la base modifiée, on vous demande de réaliser les opérations de mise à jour des données suivantes :*

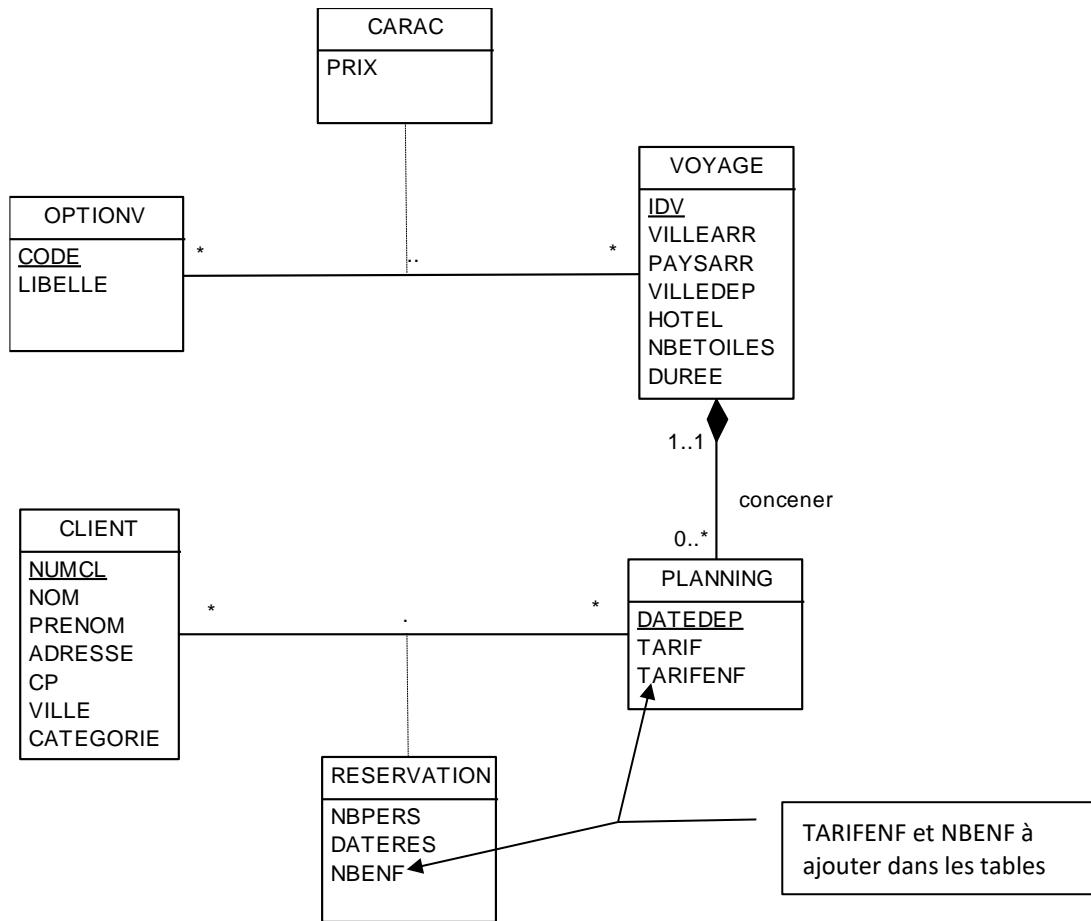
1. Mettez à jour l'*attribut nbenf* en lui affectant la valeur 2 pour toutes les réservations du client 2103, et en lui donnant la valeur 1 pour les réservations du client *Thomas Jarolim*.
2. Pour tous les tuples de la relation *planning*, effectuez la modification de *tarif* en lui affectant la valeur de *tarif* réduite de moitié.
3. Insérez deux nouveaux tuples dans la relation *client* en vérifiant la mise en œuvre de la contrainte de domaine sur *categorie*. Créez une réservation pour un de ces clients et un voyage existant.
4. Mettez à jour la relation *hotel* en tentant de donner une valeur invalide à l'*attribut nbetoile*.

# Création – modification de la structure d'une BDD et manipulation de données sous Postgres

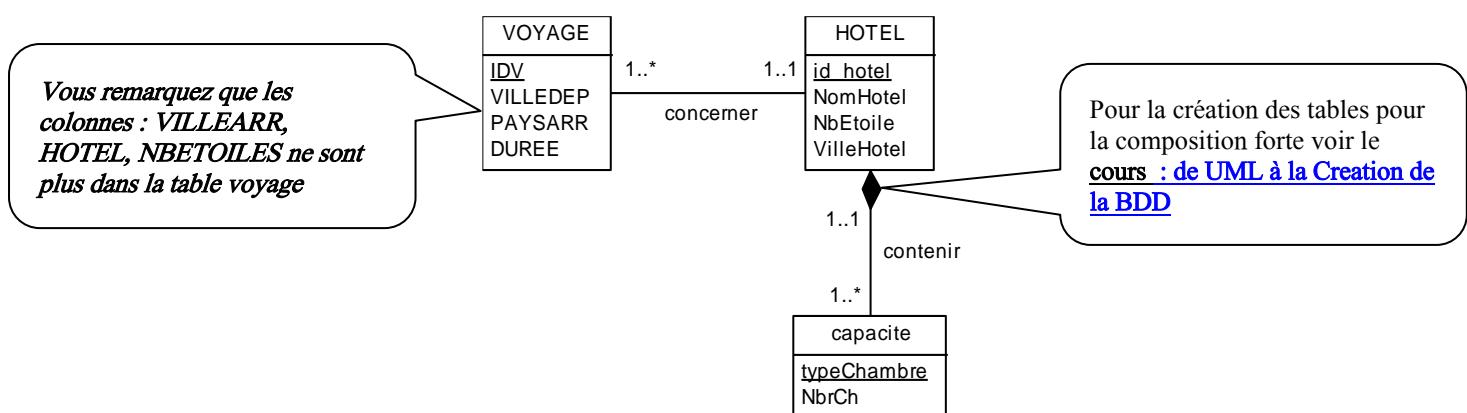
**TP2.1**

R2.06 exploitation d'une base de données

## DIAGRAMME DE CLASSES DE LA BASE VOYAGE



## ANNEXE 1 : VUE PARTIELLE DU DIAGRAMME DE CLASSES MODIFIÉ



colonnes	type	Contrainte
Id hotel	Entier	auto incrémenté
nomHotel	Même type que la colonne HOTEL de voyage	Defaut NULL
NbEtoile	Même type que la colonne NBETOILES de voyage ( <b>ne pas oublier la contrainte de domaine réalisée au 1.3.3)</b>	Defaut NULL
VilleHotel	Même type que la colonne VILLEARR de voyage	Defaut NULL
TypeChambre	Enumération { <i>SINGLE, DOUBLE, DOUBLE LUXE, SUITE, SUITE JUNIOR, SUITE PRESTIGE</i> }	
NbrCh	Entier	Defaut 0