

# Projet Pédagogique Hôtel

## Application

**Objectif :** concevoir, déboguer, comprendre les composants SQL et les automatismes que l'on peut ajouter sur une base de données.

**Remarque :** Si vous arrivez au résultat attendu, vous avez réussi.


**Comment rendre le travail :** Créer un repos git accessible, qui contiendra les différents scripts SQL, des fichiers Markdown qui contiendront les différentes réponses. **Un fichier .md par tâche.**

## Tâche 1 : Debugging

### Partie 1 : importation

Essayer d'importer la base de données. Des erreurs se sont glissées dans le script SQL.

Trouvez-les et corrigez-les. Adopter une approche logique de débogage.

 tâche 1. 1 Expliquer votre démarche

Vérifier que tout est bien importé en regardant le **concepteur**.

### Partie 2 : peuplement


Peupler la base de données avec le script fourni.

Vous pouvez comparer avec les données originelles fournies par le client :

chambre	1	Ski Hotel	*	1 lit simple avec douche (1 personne )	
chambre	2	Ski Hotel	*	1 lit simple avec douche (1 personne )	NB : En travaux
chambre	4	Ski Hotel	*	1 lit simple avec douche (1 personne )	
chambre	3	Ski Hotel	*	2 lits simples avec douche et wc séparé (2 personnes )	
chambre	5	Ski Hotel	*	1 lit double avec douche et WC séparé (2 personnes)	NB : Réservée aux VIP
chambre	6	Ski Hotel	*	1 lit double, 2 lits superposés avec douche et WC séparé (4 personnes)	
chambre	1	Art Hotel	**	1 lit simple avec douche (1 personne )	
chambre	2	Art Hotel	**	2 lits simples avec douche ( 2 personnes )	

chambre	3	Art Hotel	**	2 lits simples avec douche et wc séparé (2 personnes)	
chambre	4	Art Hotel	**	1 lit double avec douche (2 personnes)	
chambre	5	Art Hotel	**	1 lit double avec douche et WC séparé (2 personnes)	NB : Vue sur le lac
chambre	6	Art Hotel	**	1 lit double, 2 lits superposés avec douche et WC séparé (4 personnes)	
chambre	7	Art Hotel	**	1 lit double large avec bain et WC séparé (2 personnes)	
chambre	1	Rose Hotel	***	1 lit simple avec douche (1 personne)	
chambre	2	Rose Hotel	***	2 lits simples avec douche ( 2 personnes )	
chambre	3	Rose Hotel	***	2 lits simples avec douche et wc séparé (2 personnes)	
chambre	4	Rose Hotel	***	1 lit double avec douche (2 personnes)	
chambre	5	Rose Hotel	***	1 lit double avec douche et WC séparé (2 personnes)	
chambre	6	Rose Hotel	***	1 lit double, 2 lits superposés avec douche et WC séparé (4 personnes)	
chambre	7	Rose Hotel	***	1 lit double large avec bain et WC séparé (2 personnes)	
chambre	1	Lions Hotel	****	1 lit simple avec douche (1 personne)	
chambre	2	Lions Hotel	****	2 lits simples avec douche ( 2 personnes )	
chambre	3	Lions Hotel	****	2 lits simples avec douche et wc séparé (2 personnes)	
chambre	4	Lions Hotel	****	1 lit double avec douche (2 personnes)	
chambre	5	Lions Hotel	****	1 lit double avec douche et WC séparé (2 personnes)	
chambre	6	Lions Hotel	****	1 lit double, 2 lits superposés avec douche et WC séparé (4 personnes)	
chambre	7	Lions Hotel	****	1 lit double large avec bain et WC séparé (2 personnes)	

Rencontrez-vous des difficultés ? Sur quel point faut-il être vigilant pour l'importation de données ?


 tâche 1. 2 Expliquer sur quel point il faut être vigilant

## Cas particulier :

La chambre 2 de l'hôtel "Ski Hôtel" est en travaux. Cette chambre ne sera pas disponible à la réservation.

chambre 2	Ski Hotel	*	1 lit simple avec douche (1 personne)	NB : En travaux
-----------	-----------	---	---------------------------------------	-----------------

Comment traiter le fait qu'elle n'est pas disponible actuellement ?

 tâche 1. 3 Expliquer comment vous feriez

## Tâches 2 : Delete on cascade

===Avant de faire cette partie, faites une sauvegarde de la base de données.===

Si vous avez exploré des applications web de gestion pour le SGBD, comme phpMyAdmin, vous avez pu remarquer ceci :

The screenshot shows the phpMyAdmin interface for defining a foreign key constraint. The 'ON DELETE' and 'ON UPDATE' dropdown menus are both set to 'CASCADE'. The 'id\_hotel' table is selected as the referenced table, and the 'id' column is selected as the referenced column. The 'hotels' table is the target table. The 'FK\_Chambres\_Hotels' constraint name is visible on the left.

Lors de la création d'une contrainte de clé étrangère, vous pouvez choisir en option définir ce qu'il se passe dans le cas d'une suppression ou d'une mise à jour.

La clause `'ON DELETE CASCADE'` signifie que, si une clé primaire (PK) est supprimée de sa table d'origine, toutes les lignes d'enregistrement contenant la même valeur d'ID, c'est-à-dire contenant les clés étrangères, sont **supprimées**.

**Important :** C'est la suppression de la clé primaire (PK) dans la table référencée qui déclenche la suppression en cascade.

**Recherche :** Voir aussi les autres options : RESTRICT, NULL, UPDATE ...


Application :

Par défaut, l'option ON DELETE est positionnée sur RESTRICT.

Vérifier et essayer de supprimer un enregistrement dans la table hotels. Par exemple l'Hôtel avec l'identifiant 1.


```
DELETE FROM hotels WHERE `hotels`.`id` = 1
```

Que se passe-t-il ?

 tâche 2.1 Expliquer ce que vous constatez


Modifier les contraintes de clés étrangères dans la vue relationnelle pour passer à `ON DELETE ON CASCADE`. Refaire le teste "suppression de l'hôtel 1" et regardez ce qu'il se passe...

Que se passe-t-il ?

 tâche 2.2 Expliquer ce que vous constatez

Testez les options `ON UPDATE ON CASCADE` et `ON DELETE ON NULL`.

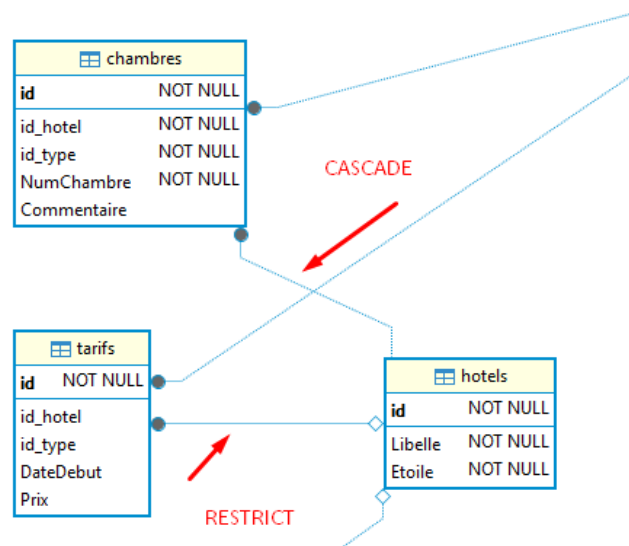
Que se passe-t-il ?

 tâche 2.3 Expliquer ce que vous constatez


Remettre la base de données en ordre à la fin des tests, pour la suite des exercices.

Question :

Si la contrainte entre chambres et hôtels est en cascade et celle entre hôtels et tarifs est en restriction. RESTRICT.



Votre avis ?

 tâche 2.4 A votre avis, est-ce que la suppression est permise ou non ?

=== Après cette partie, recharger votre sauvegarde .===

## Tâche 3 : Type de chambre

Le type d'une chambre permet de déterminer son prix. Il sera souvent interrogé.

### Requête 1 :

Préciser la salle de bain pour chaque type de chambre :

Résultat attendu :

id	type	sanitaire	
1	Chambre Standard	Douche	
2	Chambre Twin	Douche	
3	Chambre Twin +	Douche et WC séparé	
4	Chambre Double	Douche	
5	Chambre Double +	Douche et WC séparé	
6	Chambre Familiale	Douche et WC séparé	
7	Chambre Familiale +	Baignoire et WC séparé	
8	Chambre King Size	Baignoire et WC séparé	

### Requête 2 :

Faire une requête qui permet d'afficher aussi le couchage de chaque type de chambre.

Résultat attendu :

id	type	sanitaire	qte	nom	nb places
1	Chambre Standard	Douche	1	Lit simple	1
2	Chambre Twin	Douche	2	Lit simple	1
3	Chambre Twin +	Douche et WC séparé	2	Lit simple	1
4	Chambre Double	Douche	1	Lit double	2
5	Chambre Double +	Douche et WC séparé	1	Lit double	2
6	Chambre Familiale	Douche et WC séparé	1	Lit double	2
6	Chambre Familiale	Douche et WC séparé	2	Lit superposé	1
7	Chambre Familiale +	Baignoire et WC séparé			
8	Chambre King Size	Baignoire et WC séparé	1	Lit XL	2

## Requête 3 :

Afficher les résultats sur **une seule ligne**, en calculant le nombre de personnes :

id	type	sanitaire	details	nb personnes
1	Chambre Standard	Douche	1xLit simple	1
2	Chambre Twin	Douche	2xLit simple	2
3	Chambre Twin +	Douche et WC séparé	2xLit simple	2
4	Chambre Double	Douche	1xLit double	2
5	Chambre Double +	Douche et WC séparé	1xLit double	2
6	Chambre Familiale	Douche et WC séparé	1xLit double, 2xLit superposé	4
7	Chambre Familiale +	Baignoire et WC séparé		1
8	Chambre King Size	Baignoire et WC séparé	1xLit XL	2

## Création d'une vue :

La requête 3 étant compliqué à faire et pouvant pourtant être utile et donc souvent interroger.  
Créer une vue de cette sélection. (nom de la vue : `vue_chambre_types`)

 tâche 3.1 Donner le script de la vue `vue_chambre_types`

## Bien choisir une vue.

Les vues sont tout de même gourmandes en ressource. On va bien les choisir et pas en abuser.  
Proposer une autre vue possible et utile dans la base de données de réservation.

 tâche 3.2 Proposer une vue, en précisant les colonnes, et en expliquant votre choix.

## Tâches 4 : Historisation

En vous inspirant du support de cours, **mettre en place un trigger** qui enregistre une date de mise à jour pour un hôtel. (nom du trigger : `trigger_hotel_after_update` )

 tâche 4.1 Script du trigger `trigger_hotel_after_update`


**Créer un autre trigger** qui enregistre l'ancienne version de l'enregistrement dans une table historique, à chaque modification

- nom de la table : `histo_hotel`
- nom du trigger : `trigger_hotel_before_update`


 tâche 4.2 Script du trigger `trigger_hotel_before_update`

### Questions :

**Peut-on ajouter deux triggers à une même table ?**

 tâche 4.3 Répondez

**Peut-on ajouter deux triggers à une même table et sur un même événement ?**

 tâche 4.4 Répondez

## Tâche 5 : Tarif

### La date de fin

Pour le tarif, on n'enregistre pas la date de fin en base de données, car elle est **calculatoire**. En effet, l'ancien tarif s'arrête là où le nouveau tarif commence.

Mais concrètement, comment calculer cette date de fin ?


**Recherche** : Trouver la date de fin d'application de chaque tarif

Exemple pour l'hôtel 2 et le type de chambre 1, vous devez avoir ce résultat.


- du 16 avril 2021 au 14 décembre 2021 : 49.99

- du 15 décembre jusqu'à aujourd'hui : 57.49

id	id_hotel	id_type_chambre	date_debut	prix	date_fin
43	2	1	2021-04-16	49.99	2021-12-14
8	2	1	2021-12-15	57.49	

 tâche 5.1 Donnez la requête SQL qui affiche la date de fin de chaque tarif


Créer une vue pour utiliser cette date de fin plus tard. Nom de la vue : `vue_tarif_date_fin`

 tâche 5.2 Donnez l'exportation de la vue `vue_tarif_date_fin`

Lorsqu'on travaille sur des périodes, on peut se demander comment connaître le prix de la chambre pour un jour en particulier. Pour cela :

Créer une fonction `get_tarif_jour()`

Cette fonction calcule le prix d'une chambre pour un jour donné (DATE), dans un hôtel donné et pour un type de chambre donné, en allant chercher le tarif valide dans la table `tarifs`.


 tâche 5.3 Donnez l'exportation de la vue `vue_tarif_date_fin`



## Validation

Imaginons une règle métier sur le prix : les prix ne peuvent qu'augmenter, jamais diminuer.


**Créer un trigger BEFORE UPDATE** sur la table `tarif` qui provoque une erreur si le prix est inférieur ou égale aux autres tarifs et oblige une augmentation constante.

 tâche 5.4 Donnez le script de création du trigger (l'exportation) et expliquez votre choix pour le nom du trigger.

## Tâche 6 : Conception


Il manque quelque chose à la base de données. On ne peut pas enregistrer de réservation. **Comment modéliser la réservation ?** Quelles transformations cela implique dans la base de données ?

Votre tâche consiste à réaliser les modifications en utilisant le LDD.


 tâche 6.1 Donnez le script correspondant aux modifications

**Conceptualiser des composants SQL** (trigger, procédure stockée ou fonction stockée) autour du sujet de la réservation. C'est-à-dire, imaginez les composants que vous pouvez ajouter dans ce domaine.


Trigger :

 tâche 6.2 Donnez un trigger documenté (Expliquer ce qu'il fait)

Procédure stockée :

 tâche 6.3 Donnez une procédure stockée documentée

Fonction stockée :

 tâche 6.4 Donnez une fonction stockée documentée