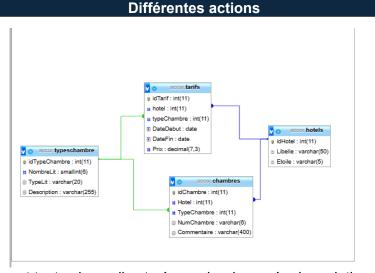
# BDD ET LANGAGE SQL TP3 : INTERROGER UNE BASE DE DONNÉES VIA LE LANGAGE SQL

**Nom: FALGUEYRETTES** 

Prénom : Aliyah Groupe : 1SIO-B

Mise en situation: Accor

# Tâches 1 – Requêtes d'extraction



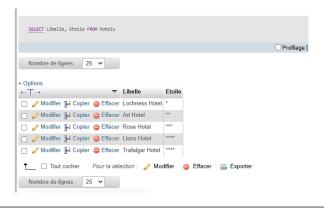
Avant toute chose, il est nécessaire de recréer les relations visiblement existantes entre les tables.

Pour toutes les requêtes, il conviendra d'utiliser les clauses SELECT, FROM et si besoin WHERE.

### 1e requête – SELECT Libelle, Etoile FROM hotels;

Il se trouve que sur la version utilisée de phpMyAdmin utilisée, l'onglet requête ne permettant pas de faire plusieurs requêtes d'affichage à la suite, le caractère '; 'provoquent une erreur. Après cet exemple, je n'utiliserai plus les points virgule afin de laisser la liberté de tester les commandes données.

Cette requête affiche ainsi les champs *Libelle* et *Etoile* (SELECT) de la table *hotels* (FROM).

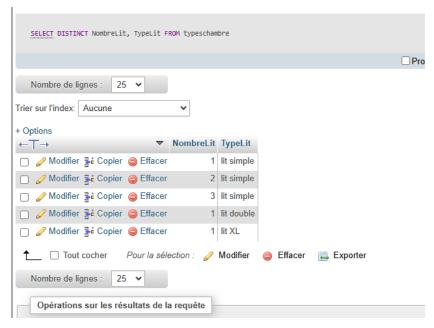


**2**<sup>e</sup> **requête** – Il est demandé d'afficher le nombre de lits et leur type, sans doublon.

Il faut donc afficher les types de chambres, sans prendre en compte leur description mais seulement sur le type et le nombre de lits qu'elles contiennent. Pour cela, l'on utilisera donc la commande suivante :

SELECT DISTINCT TypeLit, NombreLit FROM typeschambre

qui affiche le résultat suivant :



SELECT DISTINCT affiche la requête en éliminant les doublons.

**3° requête** – Il est demandé d'afficher la liste des hôtels par ordre alphabétique.

Une nouvelle clause fait son apparition : la clause ORDER BY, qui comme son nom l'indique, permet d'ordonner les résultats selon ce qui la suit. Cette clause peut-être utilisée sans être nécessairement liée à une requête d'affichage. L'ordre peut être croissant (ASC) ou décroissant (DESC).

L'on utilisera donc la requête suivante :

SELECT \* FROM hotels ORDER BY Libelle ASC



Cette requête demande les champs hotel, typeChambre et prix de la table tarifs, dont le champ DateDebut est compris entre le 01/12/2024 et le 31/12/2024, et de classer les résultats obtenus selon le champ Prix, dans l'ordre décroissant.

**5**<sup>e</sup> **requête** – ID des chambres de type 3 avec un commentaire « Vue sur le lac ».

La bonne requête est celle ci :

SELECT idChambre FROM chambres WHERE (TypeChambre = 3) AND (Commentaire = 'Vue sur le lac')

Mais comme aucune chambre ne correspond à la recherche, cette requête ne renvoie rien.

## 2 – Requêtes de modification

**6**° **requête** – Ajout d'enregistrements dans la table tarifs.

Il sera ici nécessaire d'utiliser la clause INSERT INTO. Fort heureusement, il est possible d'insérer plusieurs enregistrements à la fois.

INSERT INTO tarifs VALUES (57,1,1,'2025-04-15','2025-09-30','58.49'), (59,1,3,'2025-04-15','2025-09-30','81.49'), (60, 1, 4,'2025-04-15', '2025-09-30','69.99')

ce qui ne renvoie rien, mais en examinant désormais la table, l'on constate ceci :



Ce qui indique que la requête a fonctionné.

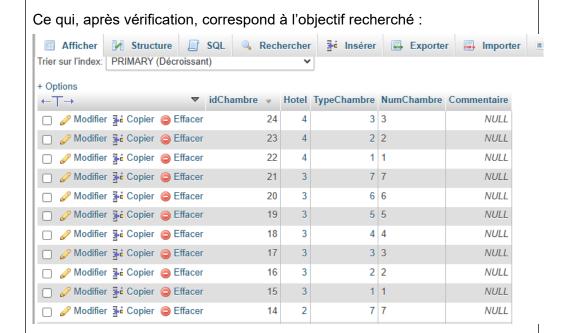
7º requête – Suppression des chambres ayant un identifiant supérieur à 25.

DELETE idChambre FROM chambres WHERE idChambre > 25

Je n'ai cependant pas réussi à réaliser cette commande, malgré que cela soit indiqué ainsi dans le cours. En me renseignant sur internet, j'ai observé que le fait de désigner le champ dans le DELETE pouvait mener à l'erreur à laquelle j'ai fait face.

J'ai donc effectué la commande qui suit :

DELETE FROM `chambres` WHERE `idChambre`>= 25



**8° requête** – Mise à jour des tarifs de +20 % pour l'hôtel « art hotel » et de 10 % pour l'hôtel « Lions Hotel ».

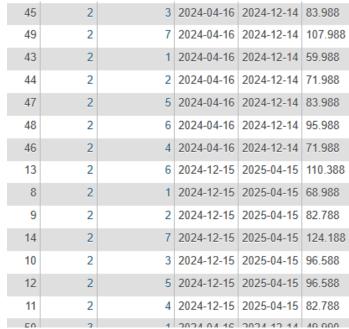
Pour cette requête, je n'étais pas sûre de s'il aurait fallu que je me serve d'un SELECT en plus de ma requête UPDATE, pour utiliser le nom de l'hôtel au lieu du chiffre qui lui correspond.

J'ai donc fait au plus simple, pour atteindre les objectifs recherchés, en séparant mes deux requêtes.

UPDATE tarifs SET Prix=Prix\*1.20 WHERE hotel=2 UPDATE tarifs SET Prix=Prix\*1.10 WHERE hotel=4

Il aurait fallu que je pense à faire une capture d'écran avant modification, mais voilà le résultat :





Nb: Noms des champs: idtarifs, hotel, typechambre, DateDebut, DateFin, Prix

**9° requête** – Mise à jour des commentaires : remplacement par 'VIP', pour la chambre 3 du « Rose Hotel ».

UPDATE chambres SET Commentaire='VIP' WHERE hotel=3 AND Num-Chambre=3

idChambre	¥	Hotel	TypeChambre	NumChambre	Commentaire
	24	4	3	3	NULL
	23	4	2	2	NULL
	22	4	1	1	NULL
	21	3	7	7	NULL
	20	3	6	6	NULL
	19	3	5	5	NULL
	18	3	4	4	NULL
	17	3	3	3	VIP
	16	3	2	2	NULL
	15	3	1	1	NULL

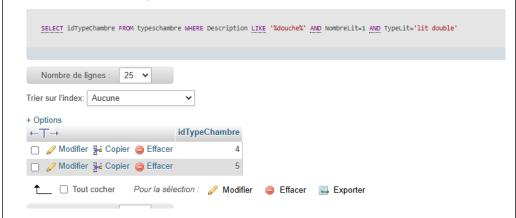
**10**° **requête** – INSERT INTO hotels VALUES (6, 'Massy Hotel', '\*\*\*\*\*')

Comme vu lors de la 6° requête, il n'y a aucun retour ; l'examen de la table permet de constater l'objectif atteint :



**11**° **requête** – Affichage de l'identifiant de toutes les chambres qui possèdent une douche et dont le type de lit est « lit double ».

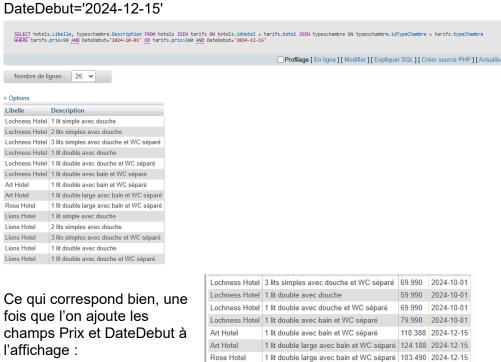
SELECT idTypeChambre FROM typeschambre WHERE Description LIKE '%douche %' AND TypeLit='lit double'



Remarque : contrairement à la clause LIKE, l'opérateur = est sensible à la casse. Pour plus de sécurité, il conviendra d'éviter, à moins de chercher une information identique à celle que l'on cherche, d'utiliser l'opérateur = dans la recherche de chaînes de caractère.

**12**° **requête** – Affichage des hotels et du type de chambre dont le prix est inférieur à 80€ et dont la date de début est au 1<sup>er</sup> octobre, ainsi que des hotels et types de chambre dont le prix est supérieur à 100€ et dont la date de début est fixée au 15 décembre.

SELECT hotels, typeChambre
WHERE prix<80 AND DateDebut='2024-10-01' OR prix>100 AND
DateDebut='2024-12-15'



Rose Hotel 1 III double rarge avec barrier vio separa 165.55

Lions Hotel 1 lit simple avec double 54.989 2024-10-01

Lions Hotel 3 lits simples avec douche et WC séparé 76.989 2024-10-01

Lions Hotel 1 lit double avec douche et WC séparé 76.989 2024-10-01

65.989 2024-10-01

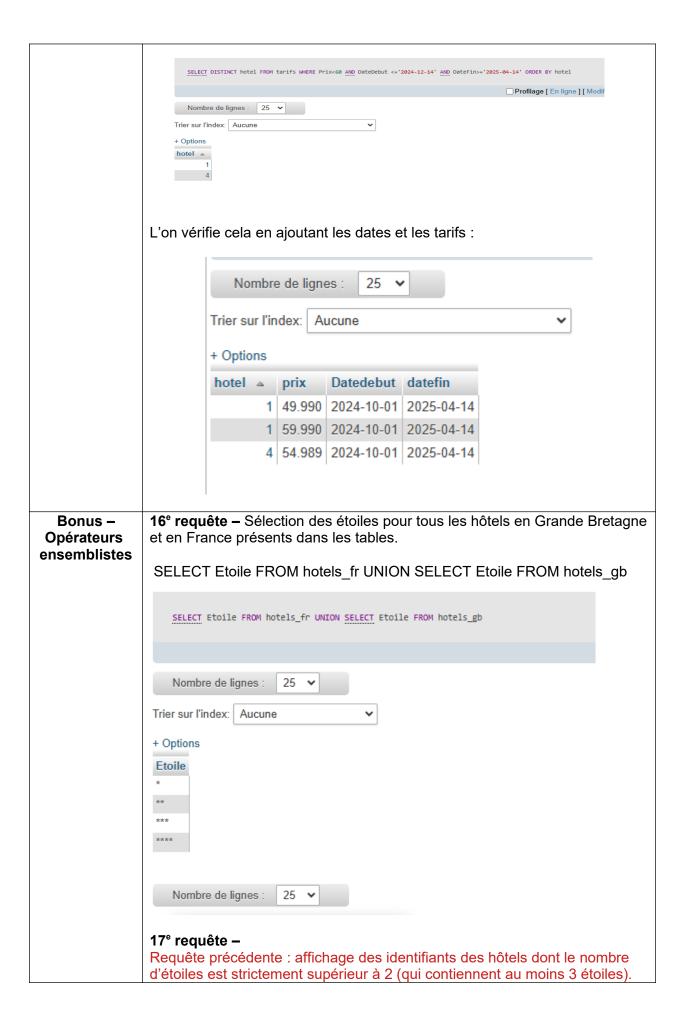
65.989 2024-10-01

Lions Hotel 2 lits simples avec douche

1 lit double avec douche

Lions Hotel

13° requête – Affichage des chambres qui possèdent un commentaire SELECT NumChambre FROM Chambres WHERE Commentaire is not null SELECT NumChambre FROM chambres WHERE Commentaire is not null Nombre de lignes : 25 🕶 Trier sur l'index: Aucune + Options NumChambre 5 14° requête – Affichage des identifiants des types de chambre qui possèdent un bain et un lit double, ou qui possèdent des WC et un ou plusieurs lits simples. SELECT idTypeChambre FROM typeschambre WHERE Description LIKE '%bain%' AND TypeLit='lit double' OR Description LIKE '%WC%' AND TypeLit='lit simple' SELECT idTypeChambre FROM typeschambre WHERE Description LIKE '%bain%' AND TypeLit='lit double' OR Description LIKE '%wC%' AND TypeLit='lit simple' Profilage [ En ligne ] [ Modifier ] [ Expliquer SQL ] [ Créer source Nombre de lignes : 25 🕶 Trier sur l'index: Aucune + Options ☐ // Modifier 3 € Copier ⊜ Effacer ☐ *⊘* Modifier **¾** Copier **⊝** Effacer 6 ↑ Tout cocher Pour la sélection : ⊘ Modifier ⊜ Effacer Exporter Nombre de lignes : 25 🕶 15° requête – Identifiant des hôtels dont le prix est inférieur à 60€, disponibles pour la période du 14 décembre 2024 au 14 avril 2025, donnés dans l'ordre croissant de leur identifiant. SELECT DISTINCT hotel FROM tarifs WHERE Prix<60 AND DateDebut <='2024-12-14' AND DateFin>='2025-04-14' ORDER BY hotel





Nouvelle requête : Affichage des identifiants de tous les hôtels qui possèdent au moins deux étoiles.

SELECT idhotel FROM hotels\_fr WHERE Etoile LIKE '\*\*%' UNION

SELECT idhotel FROM hotels gb WHERE Etoile LIKE '\*\*%'



#### Requêtes 18 et 19

Les clauses INTERSECT et EXCEPT ne fonctionnent pas ici, mais la théorie est appliquée selon la fiche ressource.

**18° requête –** SELECT Libelle FROM hotels\_fr INTERSECT SELECT Libelle FROM hotels\_gb

**19**° **requête** – SELECT Libelle FROM hotels\_fr WHERE Etoile LIKE '\*\*' EXCEPT SELECT Libelle FROM hotels\_gb WHERE Etoile LIKE '\*\*'

#### Questions 18 à 23

18 –

SELECT Nom, CASE

when Pays ID=1 then 'Europe'

when Pays ID=2 OR Pays ID=3 OR Pays ID=7 then 'Amérique'

when Pays ID=4 OR Pays ID=6 then 'Afrique'

when Pays ID=5 then 'Asie'

**END AS Continent** 

- 19 SELECT AVG(Prix) FROM Produit WHERE Libelle NOT LIKE '%ordinateur %'
- 20 SELECT COUNT(\*) FROM Commande WHERE Date BETWEEN '2024-07-01' AND '2024-08-31'
- 21 SELECT MIN(Prix) FROM Produit
- 22 Un alias est un moyen de « renommer une requête ». Cela n'a d'utilité que pour la lisibilité et n'affecte pas le nom de réels champs. La requête s'affichera sous le nom de l'alias grâce à l'instruction **AS**.

23 -

- a) SELECT AVG(Prix) AS 'Moyennne prix des produits qui ne sont pas des ordinateurs' FROM Produit WHERE Libelle NOT LIKE '%ordinateur %'
- b) SELECT COUNT(\*) AS 'Nombre de commandes été 2024' FROM Commande WHERE Date BETWEEN '2024-07-01' AND '2024-08-31'
- c) SELECT MIN(Prix) AS 'Prix le plus faible' FROM Produit