

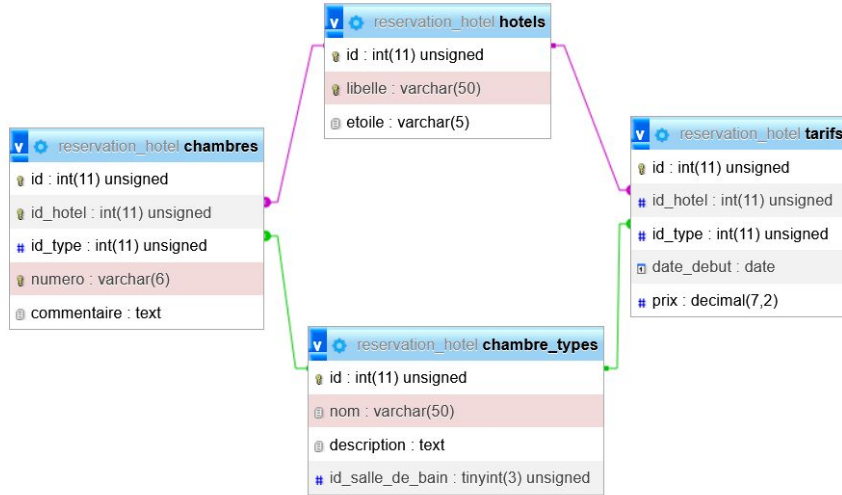
Projet Hotel

Composants SQL

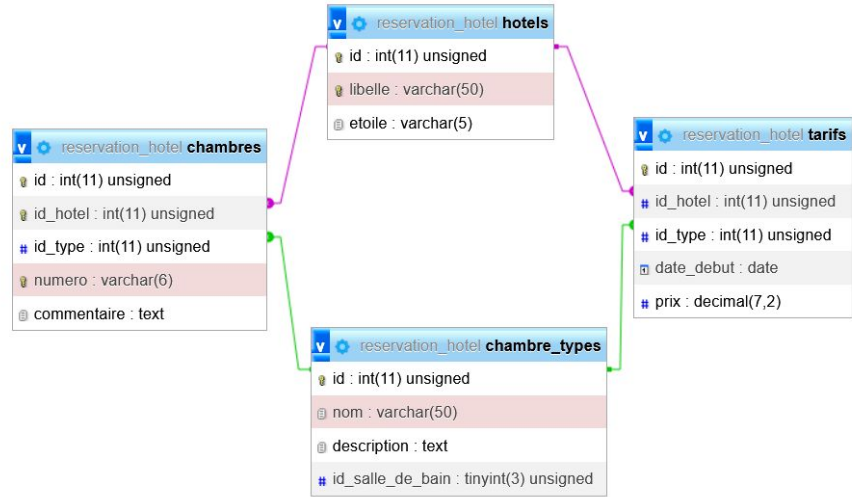


Les Hôtels

Un hôtel a plusieurs chambres de type différent.
Les prix des chambres peuvent varier.



Contraintes de clé étrangère



- **id_hotel** fait référence à **id** dans la table **hotels**
- **id_type** fait référence à **id** dans la table **chambre_types**

Jointure

Rappel

Structure d'une jointure :

SELECT * FROM _____

JOIN _____

ON _____ = _____ ;

nom table 1

nom table 2

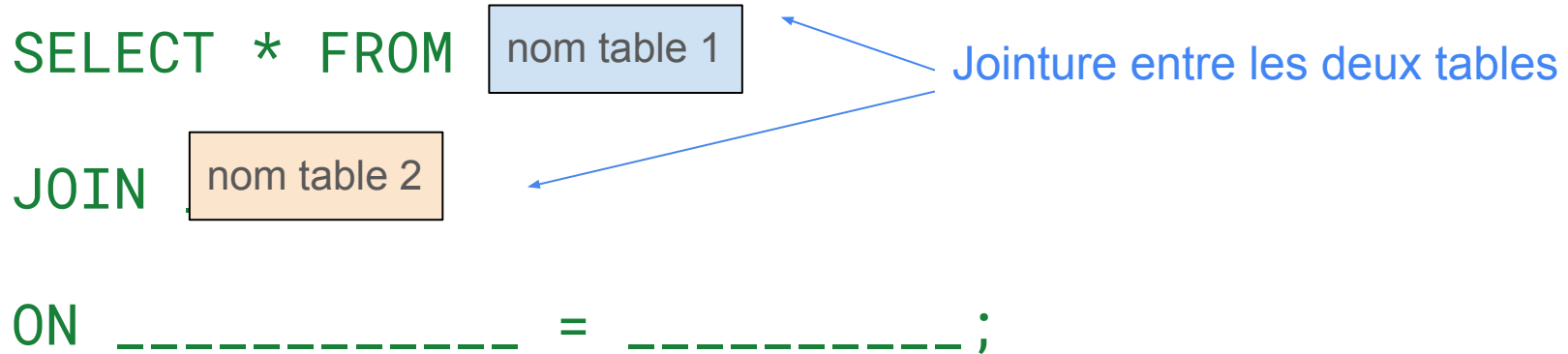
Colonne 1

Colonne 2

Condition de jointure

Rappel : Structure d'une jointure :

`SELECT * FROM` nom table 1 > Jointure entre les deux tables
`JOIN` nom table 2 >
`ON` _____ `=` _____ `;`



Condition de jointure

Rappel : Structure d'une jointure

SELECT * FROM nom table 1

JOIN nom table 2

ON Colonne 1 = Colonne 2

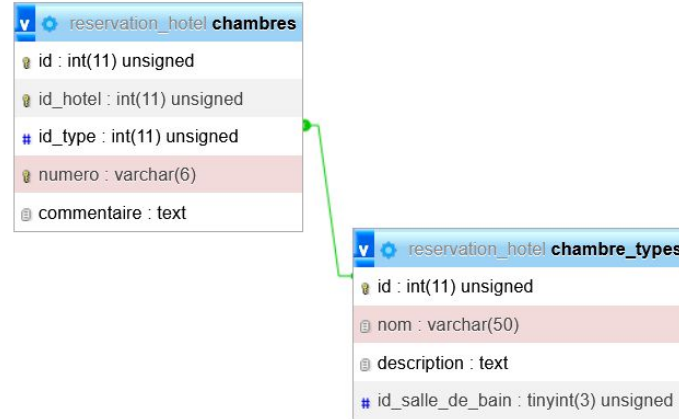
Condition de jointure
= lien entre les clés

Exemple 1 de jointure



```
SELECT * FROM chambres  
JOIN hotels ON chambres.id_hotel = hotels.id;
```

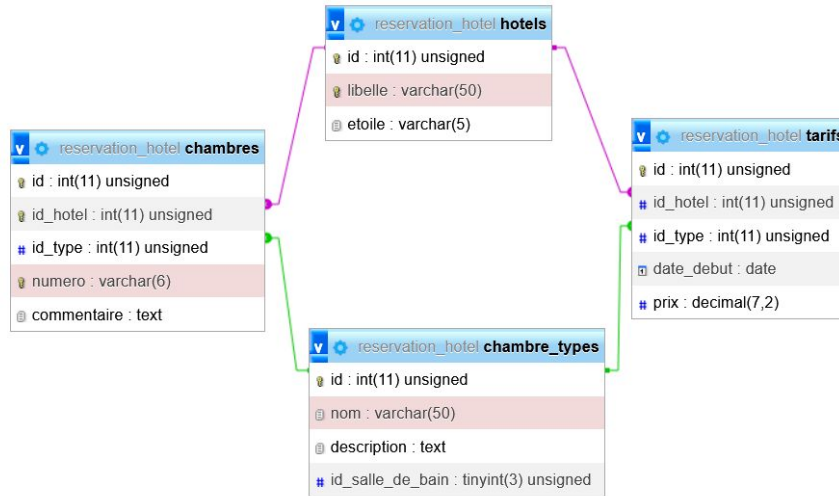
Exemple 2 de jointure



```
SELECT * FROM chambres c
JOIN chambre_types ct ON c.id_type = ct.id ;
```

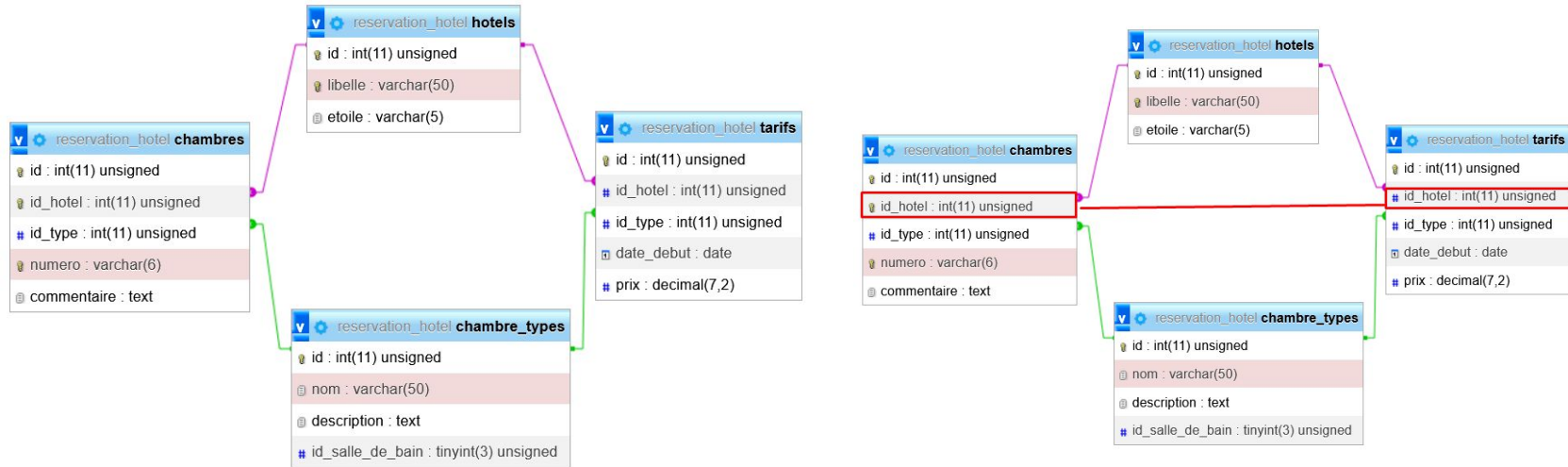

Question :

Comment afficher le tarif de chaque chambre ?



TP Hotel

Question : Comment afficher le prix de chaque chambre ?



- `chambres.id_hotel = hotels.id = tarifs.id_hotel`
- `chambres.id_type = chambre_types.id = tarifs.id_type`

TP Hotel

Question : Comment afficher le prix de chaque chambre ?

reservation_hotel	chambres
id : int(11) unsigned	
id_hotel : int(11) unsigned	
# id_type : int(11) unsigned	
numero : varchar(6)	
commentaire : text	

reservation_hotel	tarifs
id : int(11) unsigned	
# id_hotel : int(11) unsigned	
# id_type : int(11) unsigned	
date_debut : date	
# prix : decimal(7,2)	

```
SELECT chambres
JOIN tarifs
ON chambres.id_hotel = tarifs.id_hotel
AND chambres.id_type = tarifs.id_type
```

Agrégation

La clause **GROUP BY** regroupe les enregistrements qui ont la même valeur dans une colonne spécifiée. Elle est souvent utilisée pour les **fonctions d'agrégation** comme **COUNT()**, **SUM()**, **AVG()**, **MAX()**, **MIN()** ... pour effectuer des opérations sur chaque groupe de lignes.

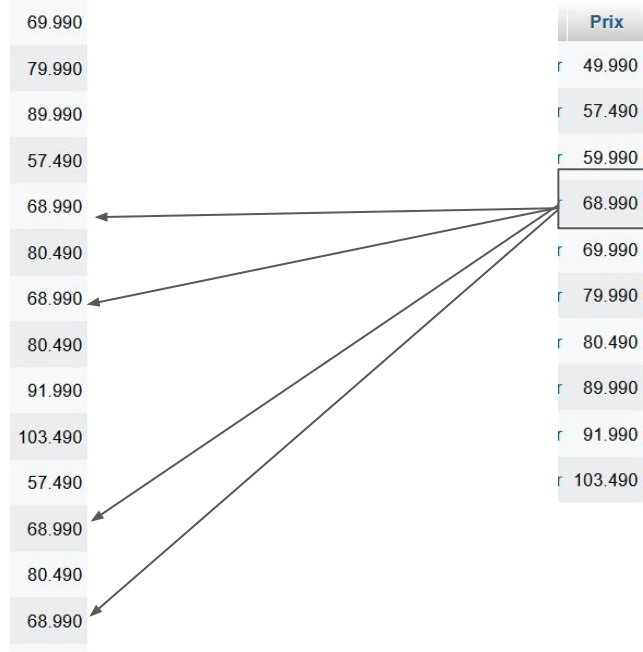
Group By

```
SELECT Prix FROM Tarifs;
```

Prix
49.990
59.990
69.990
59.990
69.990
79.990
89.990
57.490
68.990
80.490
68.990
80.490
91.990
103.490
57.490
68.990
80.490
68.990

```
SELECT Prix FROM Tarifs GROUP BY Prix;
```

Prix
49.990
57.490
59.990
68.990
69.990
79.990
80.490
89.990
91.990
103.490



Fonction d'agrégat

Fonctions	Description
AVG (attribut)	Calcule la moyenne des valeurs dans l'attribut.
SUM (attribut)	Calcule la somme des valeurs dans l'attribut.
MIN (attribut)	Détermine la plus petite valeur dans l'attribut.
MAX (attribut)	Détermine la plus grande valeur dans l'attribut.
COUNT (attribut)	Compte le nombre d'occurrences dans l'attribut.
VARIANCE (attribut)	Variance de l'attribut
STDDEV (attribut)	Ecart type de l'attribut

Prix moyen

❏ Calculer le prix moyen par chambre

Hotel	N° Chambre	Prix moyen
Ski Hotel	1	53.74
Ski Hotel	2	64.49
Ski Hotel	3	75.24
Ski Hotel	4	64.49
Ski Hotel	5	75.24
Ski Hotel	6	85.99
Art Hotel	1	53.74
Art Hotel	2	64.49
Art Hotel	3	75.24
Art Hotel	4	64.49
Art Hotel	5	75.24
Art Hotel	6	85.99
Art Hotel	7	96.74
Rose Hotel	1	53.74
Rose Hotel	2	64.49

Hotel	N° Chambre	Prix moyen
Rose Hotel	3	75.24
Rose Hotel	4	64.49
Rose Hotel	5	75.24
Rose Hotel	6	85.99
Rose Hotel	7	96.74
Lions Hotel	1	53.74
Lions Hotel	2	64.49
Lions Hotel	3	75.24
Lions Hotel	4	64.49
Lions Hotel	5	75.24
Lions Hotel	6	85.99
Lions Hotel	7	96.74

Résultats à 27 lignes

Exercices

- ❑ Exo 1 : Calculer la somme des prix des chambres par type ?
- ❑ Exo 2 : Trouver le tarif le plus cher ?
- ❑ Exo 3 : Trouver le tarif le plus cher et le moins cher par type de chambre ?
- ❑ Exo 4 : Trouver le prix moyen des chambres par type de chambre ?

Les limites que l'on peut rencontrer ?

Exemple

❏ Calculer le nombre de jour d'application de chaque tarif

idTarif	hotel	typeChambre	prix	nb_jours
1	1	1	49.99	196
2	1	2	59.99	196
3	1	3	69.99	196
4	1	4	59.99	196
5	1	5	69.99	196
6	1	6	79.99	196
7	1	7	89.99	196
8	2	1	57.49	122
9	2	2	68.99	122
10	2	3	80.49	122
11	2	4	68.99	122
12	2	5	80.49	122
13	2	6	91.99	122
14	2	7	103.49	122
15	3	1	57.49	122

idTarif	hotel	typeChambre	Prix	nb_jours
16	3	2	68.990	122
17	3	3	80.490	122
18	3	4	68.990	122
19	3	5	80.490	122
20	3	6	91.990	122
21	3	7	103.490	122
22	4	1	49.990	196
23	4	2	59.990	196
24	4	3	69.990	196
25	4	4	59.990	196
26	4	5	69.990	196
27	4	6	79.990	196
28	4	7	89.990	196
29	1	1	57.490	183
30	1	2	68.990	183

idTarif	hotel	typeChambre	Prix	nb_jours
31	1	3	80.490	183
32	1	4	68.990	183
33	1	5	80.490	183
34	1	6	91.990	183
35	1	7	103.490	183
36	4	1	57.490	183
37	4	2	68.990	183
38	4	3	80.490	183
39	4	4	68.990	183
40	4	5	80.490	183
41	4	6	91.990	183
42	4	7	103.490	183
43	2	1	49.990	243
44	2	2	59.990	243
45	2	3	69.990	243

idTarif	hotel	typeChambre	Prix	nb_jours
46	2	4	59.990	243
47	2	5	69.990	243
48	2	6	79.990	243
49	2	7	89.990	243
50	3	1	49.990	243
51	3	2	59.990	243
52	3	3	69.990	243
53	3	4	59.990	243
54	3	5	69.990	243
55	3	6	79.990	243
56	3	7	89.990	243

Résultats à 56 lignes

Retrouver l'information

Comment trouver
le nombre de jours

❏ Calculer le nombre de jour d'application de chaque tarif

idTarif	hotel	typeChambre	prix	nb_jours
1	1	1	49.99	196
2	1	2	59.99	196
3	1	3	69.99	196
4	1	4	50.00	196
5	1	5	60.00	196
6	1	6	70.00	196
7	1	7	80.00	196
8	2	1	50.00	196
9	2	2	60.00	196
10	2	3	70.00	196
11	2	4	80.00	196
12	2	5	90.00	196
13	2	6	100.00	196
14	2	7	110.00	196
15	3	1	57.49	122

reservation_hotel tarifs

- id : int(11) unsigned
- # id_hotel : int(11) unsigned
- # id_type : int(11) unsigned
- date_debut : date
- # prix : decimal(7,2)

idTarif	hotel	typeChambre	Prix	nb_jours
16	3	2	68.990	122
17	3	3	80.490	122
18	3	4	68.990	122
19	3	5	80.490	122
20	3	6	91.990	122
21	3	7	103.490	122
22	4	1	49.990	196
23	4	2	59.990	196
24	4	3	69.990	196
25	4	4	59.990	196
26	4	5	69.990	196
27	4	6	79.990	196
28	4	7	89.990	196
29	1	1	57.490	183
30	1	2	68.990	183

idTarif	hotel	typeChambre	Prix	nb_jours
31	1	3	80.490	183
32	1	4	68.990	183
33	1	5	80.490	183
34	1	6	91.990	183
35	1	7	103.490	183
36	4	1	57.490	183
37	4	2	68.990	183
38	4	3	80.490	183
39	4	4	68.990	183
40	4	5	80.490	183
41	4	6	91.990	183
42	4	7	103.490	183
43	2	1	49.990	243
44	2	2	59.990	243
45	2	3	69.990	243

idTarif	hotel	typeChambre	Prix	nb_jours
46	2	4	59.990	243
47	2	5	69.990	243
48	2	6	79.990	243
49	2	7	89.990	243
50	3	1	49.990	243
51	3	2	59.990	243
52	3	3	69.990	243
53	3	4	59.990	243
54	3	5	69.990	243
55	3	6	79.990	243
56	3	7	89.990	243

Résultats à 56 lignes

Affichage

Comment utiliser un système de pagination ?

❏ Calculer le nombre de jour d'application de chaque tarif

idTarif	hotel	typeChambre	prix	nb_jours
1	1	1	49.99	196
2	1	2	59.99	196
3	1	3	69.99	196
4	1	4	59.99	196
5	1	5	69.99	196
6	1	6	79.99	196
7	1	7	89.99	196
8	2	1	57.49	122
9	2	2	68.99	122
10	2	3	80.49	122
11	2	4	68.99	122
12	2	5	80.49	122
13	2	6	91.99	122
14	2	7	103.49	122
15	3	1	57.49	122

idTarif	hotel	typeChambre	Prix	nb_jours
16	3	2	68.990	122
17	3	3	80.490	122
18	3	4	68.990	122
19	3	5	80.490	122
20	3	6	91.990	122
21	3	7	103.490	122
22	4	1	49.990	196
23	4	2	59.990	196
24	4	3	69.990	196
25	4	4	59.990	196
26	4	5	69.990	196
27	4	6	79.990	196
28	4	7	89.990	196
29	1	1	57.490	183
30	1	2	68.990	183

idTarif	hotel	typeChambre	Prix	nb_jours
31	1	3	80.490	183
32	1	4	68.990	183
33	1	5	80.490	183
34	1	6	91.990	183
35	1	7	103.490	183
36	4	1	57.490	183
37	4	2	68.990	183
38	4	3	80.490	183
39	4	4	68.990	183
40	4	5	80.490	183
41	4	6	91.990	183
42	4	7	103.490	183
43	2	1	49.990	243
44	2	2	59.990	243
45	2	3	69.990	243

idTarif	hotel	typeChambre	Prix	nb_jours
46	2	4	59.990	243
47	2	5	69.990	243
48	2	6	79.990	243
49	2	7	89.990	243
50	3	1	49.990	243
51	3	2	59.990	243
52	3	3	69.990	243
53	3	4	59.990	243
54	3	5	69.990	243
55	3	6	79.990	243
56	3	7	89.990	243

Résultats à 56 lignes Page 4

Page 1

Page 2

Page 3

PP Hôtel

Calcul

Comment faire des calculs un peu complexe ?

- ❑ Calculer les tarifs et le nombre de jour d'application de ce tarif pour la chambre 1

idChambre	prix	nb_jours
1	49.99	196
1	57.49	183

Résultats à 2 lignes

- ❑ En déduire le prix moyen pour la chambre 1

Hotel	N° Chambre	Moyenne pondérée
Ski Hotel	1	53.61

$$\begin{array}{c} \textbf{Moyenne pondérée} \\ 49.99 * 196 + 57.49 * 183 \\ \hline (196 + 183) \end{array}$$

Fonction



Comparer des résultats

❏ Comparons l'impacte des nombres de jours sur les tarifs

Hotel	N° Chambre	Prix moyen
Ski Hotel	1	53.74
Ski Hotel	2	64.49
Ski Hotel	3	75.24
Ski Hotel	4	64.49
Ski Hotel	5	75.24
Ski Hotel	6	85.99
Art Hotel	1	53.74
Art Hotel	2	64.49
Art Hotel	3	75.24
Art Hotel	4	64.49
Art Hotel	5	75.24
Art Hotel	6	85.99
Art Hotel	7	96.74
Rose Hotel	1	53.74
Rose Hotel	2	64.49

Hotel	N° Chambre	Prix moyen au prorata
Ski Hotel	1	53.61
Ski Hotel	2	64.34
Ski Hotel	3	75.06
Ski Hotel	4	64.34
Ski Hotel	5	75.06
Ski Hotel	6	85.78
Art Hotel	1	52.50
Art Hotel	2	63.00
Art Hotel	3	73.50
Art Hotel	4	63.00
Art Hotel	5	73.50
Art Hotel	6	84.00
Art Hotel	7	94.50
Rose Hotel	1	52.50
Rose Hotel	2	63.00

Comparer des résultats

❏ Comparons l'impacte des nombres de jours sur les tarifs

Hotel	N° Chambre	Prix moyen
Ski Hotel	1	53.74
Ski Hotel	2	64.49
Ski Hotel	3	75.24
Ski Hotel	4	64.49
Ski Hotel	5	75.24
Ski Hotel	6	85.99
Art Hotel	1	53.74
Art Hotel	2	64.49
Art Hotel	3	75.24
Art Hotel	4	64.49
Art Hotel	5	75.24
Art Hotel	6	85.99
Art Hotel	7	96.74
Rose Hotel	1	53.74
Rose Hotel	2	64.49



Hotel	N° Chambre	Prix moyen au prorata
Ski Hotel	1	53.61
Ski Hotel	2	64.34
Ski Hotel	3	75.06
Ski Hotel	4	64.34
Ski Hotel	5	75.06
Ski Hotel	6	85.78
Art Hotel	1	52.50
Art Hotel	2	63.00
Art Hotel	3	73.50
Art Hotel	4	63.00
Art Hotel	5	73.50
Art Hotel	6	84.00
Art Hotel	7	94.50
Rose Hotel	1	52.50
Rose Hotel	2	63.00

Comment ré-utiliser des requêtes qu'on a déjà faite, et qu'on doit consulter souvent ?

*Nos seuls **outils** pour imposer des règles ...*

Contrainte d'unicité

Contrainte d'intégrité

Contrainte nullité

Contrainte de type

Check

Des composants SQL

Vue (VIEW)

Réutilisation de requête basé sur un affichage virtuel.



TRIGGER

Un bloc de code qui s'exécute automatiquement à un événement.



```
trigger.sql

CREATE TRIGGER nom_trigger
[BEFORE | AFTER] [INSERT | UPDATE | DELETE]
ON nom_table
FOR EACH ROW
BEGIN
    -- instructions SQL
END;
```



Procédure stockée

Ensemble d'instruction, **une routine**, **ne renvoyant pas de valeur**, qui appelée pour exécuter des actions répétitives.



```
JS procedure.js

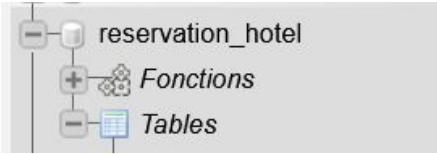
CREATE PROCEDURE nom_procedure(param1 TYPE, param2 TYPE)
BEGIN
    -- instructions SQL
END;
```

CodeImage



Fonction stockée

Permet d'encapsuler du code, de calculer et retourner une valeur



Procédures stockées

☐ Tout cocher

 Exporter

 Supprimer

	Nom	Type	Retourne	
<input type="checkbox"/>	get_tarif_jour	FUNCTION	decimal(7,2)	<div> Éditer</div> <div> Exécuter</div> <div> Exporter</div> <div> Supprimer</div>



```
function.sql

CREATE FUNCTION nom_fonction(param TYPE)
RETURNS TYPE
DETERMINISTIC
BEGIN
    -- instructions SQL
    RETURN valeur;
END;
```

De nouveaux outils ...

Composant	Rôle principal	Exemple clé
Vue	Réutilisation, + lisible, personnaliser les colonnes	<code>CREATE VIEW</code>
Procédure	Une fonction sans retour (hors select)	<code>CREATE PROCEDURE</code>
Fonction	Une fonction avec retour (dans un select)	<code>CREATE FUNCTION</code>
Trigger	Se déclenche à un événement : automatisation	<code>CREATE TRIGGER</code>
Check	Valider une condition dans une table	<code>CHECK (valeur > 0)</code>
Index	Accélérer les recherches	<code>CREATE INDEX</code>