

⑧ Vues et Confidentialité de la base de données

CAMPINGS (idCamping, nomCamping, villeCamping, nbEtoilesCamping)

BUNGALOWS (idBungalow, nomBungalow, superficieBungalow, idCamping#)

SERVICES (idService, nomService, categorieService)

PROPOSER (idBungalow#, idService#)

CLIENTS (idClient, nomClient, prenomClient, villeClient)

LOCATIONS (idLocation, idClient#, idBungalow#, dateDebut, dateFin, montantLocation)

EMPLOYES (idEmploye, nomEmploye, prenomEmploye, salaireEmploye, idCamping#, idEmployeChef#)

Première partie – Créer et utiliser des vues :

1) Vue *BungalowsLFB*

- Créer une vue *BungalowsLFB* qui contient toutes les informations qui concernent les bungalows qui se trouvent dans le camping 'Les Flots Bleus'. Cette vue doit avoir la structure suivante :

BUNGALOWSLFB (idBungalow, nomBungalow, superficieBungalow)

```
SELECT *
FROM BungalowsLFB ;
```

IDBUNGALOW	NOMBUNGALOW	SUPERFICIEBUNGALOW
B2	Le Palace	60
B3	Le Royal	65
B4	Le Titanic	35
B8	L'Impérial	40
B11	Le Polymorphe	45

- Ecrire une requête SQL qui, en utilisant la vue *BungalowsLFB*, retourne le nombre de bungalows qui se trouvent dans le Camping 'Les Flots Bleus'.

```
NB BUNGALOWS
-----
5
```

2) Vue *LocationsLFB*

- En utilisant la vue *BungalowsLFB* créée précédemment, écrire une vue *LocationsLFB* qui contient les locations qui ont été réalisées dans le camping 'Les Flots Bleus'. Cette vue doit avoir la structure suivante :

LOCATIONSLFB (idLocation, idClient, nomClient, prenomClient, idBungalow, nomBungalow)

```
SELECT *
FROM LocationsLFB ;
```

IDLOCATION	IDCLIENT	NOMCLIENT	PRENOMCLIENT	IDBUNGALOW	NOMBUNGALOW
L18	C2	Zetofrais	Mélanie	B11	Le Polymorphe
L10	C2	Zetofrais	Mélanie	B4	Le Titanic
L8	C4	Terrieur	Alex	B3	Le Royal
L6	C5	Tare	Guy	B2	Le Palace
L7	C6	Bricot	Judas	B2	Le Palace
L9	C7	Stico	Judas	B3	Le Royal
L17	C10	Nemard	Jean	B8	L'Impérial

- Ecrire une requête SQL qui, en utilisant la vue *LocationsLFB*, retourne pour chacun des bungalows du camping 'Les Flots Bleus' qui ont été loués, le nombre de locations.

IDBUNGALOW	NOMBUNGALOW	NBLOCATIONS
B11	Le Polymorphe	1
B3	Le Royal	2
B8	L'impérial	1
B4	Le Titanic	1
B2	Le Palace	2

3) Vue *LocationsBungalows*

- Créer une vue *LocationsBungalows* qui contient, pour tous les bungalows de la table *Bungalows*, toutes les informations sur le bungalow ainsi que son nombre de locations, son chiffre d'affaires CA, et son chiffre d'affaires moyen par location CAML (correspondant à $CA / nbLocations$). Cette vue doit avoir la structure suivante :
LOCATIONSBUNGALOWS (idBungalow, nomBungalow, idCamping, nomCamping, superficieBungalow, nbLocations, CA, CAML)

```
SELECT * FROM LocationsBungalows
WHERE idCamping = 'CAMP2';
```

IDBUNGALOW	NOMBUNGALOW	IDCAMPING	NOMCAMPING	SUPERFI	NBLOCATIONS	CA	CAML
B1	Le Caniveau	CAMP2	La Décharge Monochrome	30	5	4200	840
B12	Le Polymorphe	CAMP2	La Décharge Monochrome	40	0	0	
B5	La Poubelle	CAMP2	La Décharge Monochrome	35	4	2910	727,5

4) Vue *ServicesBungalows*

- Créer une vue *ServicesBungalows* qui contient, pour tous les bungalows de la table *Bungalows*, toutes les informations sur le bungalow ainsi que le nombre de services et de catégories de services qu'il propose.
SERVICESBUNGALOWS (idBungalow, nomBungalow, idCamping, nomCamping, superficieBungalow, nbServices, nbCategoriesServices)

```
SELECT * FROM ServicesBungalows
WHERE superficieBungalow < 40;
```

IDBUNGALOW	NOMBUNGALOW	IDCAMPING	NOMCAMPING	SUPERFI	NBSERVICES	NBCATEGORIESSERV
B4	Le Titanic	CAMP1	Les Flots Bleus	35	3	2
B1	Le Caniveau	CAMP2	La Décharge Monochrome	30	0	0
B5	La Poubelle	CAMP2	La Décharge Monochrome	35	2	2

5) Vue *DonneesBungalows*

- Créer une vue *DonneesBungalows* contenant les données des deux précédentes vues.

```
SELECT * FROM DonneesBungalows
WHERE idCamping = 'CAMP2';
```

IDBUNG	NOMBUNGALOW	IDCAMPING	NOMCAMPING	SUP	NBLOC	CA	CAML	NBSERVICES	NBCAT
B1	Le Caniveau	CAMP2	La Décharge Monochrome	30	5	4200	840	0	0
B12	Le Polymorphe	CAMP2	La Décharge Monochrome	40	0	0		0	0
B5	La Poubelle	CAMP2	La Décharge Monochrome	35	4	2910	727,5	2	2

- En utilisant la vue précédente, écrire une requête qui retourne le nom et la superficie des bungalows qui proposent plus de deux services mais qui n'ont pas été loués.
- Ecrire une requête qui retourne le nom du bungalow qui a le plus de locations.
- Savez-vous toujours faire une division ? Ecrire une requête qui retourne le nom des bungalows qui proposent tous les services.

6) Vue *DonneesCampings*

- Créer une vue *DonneesCampings* contenant des données sur les services et les locations des différents campings. Cette vue doit avoir la structure suivante :
DONNESCAMPINGS (idCamping, nomCamping, villeCamping, nbEtoilesCamping, nbBungalows, superficieMoyenneBungalows, nbLocations, CA, CAML, nbServices, nbCategoriesServices)

Pour réaliser cette vue, on pourra créer des vues intermédiaires. Et on arrondira le CAML.

```
SELECT * FROM DonneesCampings;
```

IDCAMP	NOMCAMPING	VILLECAMPING	NBETOILES	NBBUNG	SUPMOYBUNG	NBLOC	CA	CAML	NBSERV	NBCAT
CAMP1	Les Flots Bleus	Palavas	5	5	49	7	10060	1437	6	3
CAMP2	La Décharge ...	Palavas	1	3	35	9	7110	790	2	2
CAMP3	The White ...	La Grande Motte	5	4	72,5	2	2030	1015	4	3

7) Vue *DonneesCampingsPlus*

- Réaliser une vue *DonneesCampingsPlus* contenant toutes les données de la vue *DonneesCampings* auxquelles vous rajouterez deux colonnes indiquant le nombre de bungalows loués et le nombre de bungalows non loués (le camping CAMP1 a 5 bungalows loués et 0 non-loué ; 2 et 1 pour le CAMP2 et 1 et 3 pour le CAMP3).

Deuxième partie – Insérer des données dans les tables à travers les vues :

Vues Monotables

8) Vue monotable *EmployesSansCamping* avec identifiant.

- Créer une vue *EmployesSansCamping* qui contient toutes les informations qui concernent les employés qui ne sont pas affectés à un camping. Cette vue doit avoir la structure suivante :
EMPLOYESANSCLAMPING (idEmploye, nomEmploye, prenomEmploye, salaireEmploye, idEmployeChef)
- En justifiant vos réponses, indiquez ce qu'il est possible de faire à travers cette vue :
 - **d'insérer des lignes dans la table *Employés* ?** ☐ OUI ☐ NON
 - **de modifier/supprimer des lignes de la table *Employés* ?** ☐ OUI ☐ NON
- Pour savoir si les réponses aux questions précédentes sont exactes, essayer :
 - d'insérer un nouvel employé dans la table *Employés* à travers la vue. *Par exemple, l'employé Judas Stiko, qui a l'identifiant E100, qui a un salaire de 3000 € et qui n'a pas de chef.*
 - de modifier un employé de la table *Employés* à travers la vue. *Par exemple, changer le nom (par exemple Nana) de l'employé dont l'identifiant est E100.*
 - de supprimer un employé de la table *Employés* à travers la vue. *Par exemple, l'employé dont l'identifiant est E100.*

9) Vue monotable *EmployesAvecCamping* sans identifiant.

- Créer une vue *EmployesAvecCamping* qui contient toutes les informations qui concernent les employés qui sont affectés à un camping (n'importe lequel). Cette vue doit avoir la structure suivante :
EMPLOYESAVECCLAMPING (nomEmploye, prenomEmploye, salaireEmploye)
- En justifiant vos réponses, indiquez ce qu'il est possible de faire à travers cette vue :
 - **d'insérer des lignes dans la table *Employés* ?** ☐ OUI ☐ NON
 - **de modifier/supprimer des lignes de la table *Employés* ?** ☐ OUI ☐ NON
- Pour savoir si les réponses aux questions précédentes sont exactes, essayer :
 - d'insérer un nouvel employé dans la table *Employés* à travers la vue. *Par exemple, l'employé Judas Nana, dont le salaire est de 5000 €.*
 - de modifier un employé de la table *Employés* à travers la vue. *Par exemple, changer le nom (par exemple en Javel) des employés dont le prénom est Aude.*
 - de supprimer un employé de la table *Employés* à travers la vue. *Par exemple, les employés dont le prénom est Aude.*

10) Vue monotable *ClientsParVille* avec regroupement.

- Créer une vue *ClientsParVille* qui contient le nombre de clients de chaque ville. Cette vue doit avoir la structure suivante :
CLIENTSPARVILLES (ville, nbClients)
- En justifiant vos réponses, indiquez ce qu'il est possible de faire à travers cette vue :
 - **d'insérer des lignes ?** ☐ OUI ☐ NON
 - **de modifier/supprimer des lignes ?** ☐ OUI ☐ NON
- Pour savoir si les réponses aux questions précédentes sont exactes, essayer :
 - d'insérer une nouvelle ville à travers la vue. *Par exemple, la ville 'Rodez' dans laquelle on aurait 3 habitants.*
 - de modifier une ville à travers la vue. *Par exemple, changer le nom de la ville Montpellier.*
 - de supprimer une ville à travers la vue. *Par exemple, supprimer les villes pour lesquelles on a deux clients.*

Vues Multitables

11) Vue multitable *BungalowsEtCampings* (avec table *Bungalows* préservée par sa clé).

- Créer une vue *BungalowsEtCampings* qui indique pour chaque bungalow le camping dans lequel il se trouve. Cette vue doit avoir la structure suivante :
BUNGALOWSETCAMPINGS (idBungalow, nomBungalow, superficieBungalow, idCamping, nomCamping)
- En justifiant vos réponses, indiquez ce qu'il est possible de faire à travers cette vue :
 - **d'insérer des lignes simultanément dans les tables *Bungalows* et *Campings* ?** ☐ OUI ☐ NON
 - **d'insérer des lignes dans la table *Bungalows* ?** ☐ OUI ☐ NON
 - **d'insérer des lignes dans la table *Campings* ?** ☐ OUI ☐ NON
 - **de modifier/supprimer des lignes de la table *Bungalows* ?** ☐ OUI ☐ NON
 - **de modifier/supprimer des lignes de la table *Campings* ?** ☐ OUI ☐ NON
- Pour savoir si les réponses aux questions précédentes sont exactes, essayer :
 - d'insérer en même temps un nouveau bungalow et un nouveau Camping à travers la vue. *Par exemple, le bungalow B13, 'Le Souterrain' de 75 m² qui se trouve dans le camping CAMP10 'Yellow Shark'.*
 - d'insérer uniquement un nouveau bungalow à travers la vue. *Par exemple, le bungalow B14 'Le Dépotoir' de 25m².* Si l'insertion n'est pas possible, essayer de renseigner le nom des colonnes (et uniquement celles-là) dans l'INSERT.
 - d'insérer uniquement un nouveau camping à travers la vue. *Par exemple, le camping CAMP11 'Apelsin Mollusk'.*
 - de modifier un bungalow de la table *Bungalows* à travers la vue. *Par exemple, changer la superficie du bungalow 'Le Palace'.*
 - de modifier un camping de la table *Campings* à travers la vue. *Par exemple, changer le nom du camping 'The White Majestic' et l'appeler 'Le Majestique Blanc'.*

Options WITH dans la déclaration d'une vue

12) Vue *CampingsPalavas*.

- Créer une vue *CampingsPalavas* qui contient toutes les informations qui concernent les campings qui se trouvent dans la ville de Palavas. Cette vue doit avoir la structure suivante :
CAMPINGSPALAVAS (idCamping, nomCamping, villeCamping, nbEtoilesCamping)
- En justifiant vos réponses, indiquez ce qu'il est possible de faire à travers cette vue :
 - **d'insérer dans la table *Campings* des campings qui se trouvent dans la ville de Palavas ?** ☐ OUI ☐ NON
 - **d'insérer dans la table *Campings* des campings qui ne se trouvent pas dans la ville de Palavas ?** ☐ OUI ☐ NON
- Pour savoir si les réponses aux questions précédentes sont exactes, essayer d'insérer à travers la vue, un nouveau camping qui se trouve dans la ville de Carnon. *Par exemple, le camping CAMP4, qui se nomme 'El Delfin Azul' qui se trouve à Carnon et possède 3 étoiles.*
 Vérifier ensuite s'il est possible de voir ce nouveau camping à travers la vue.
 Puis vérifier si le nouveau camping a été inséré ou non dans la table *Campings*.
- Modifier la vue *CampingPalavas* (CREATE OR REPLACE VIEW) afin qu'à travers cette vue, on ne puisse insérer que des campings qui se trouvent dans la ville de Palavas.
- Modifier à nouveau la vue pour qu'elle soit à présent "non modifiable" ; c'est-à-dire qu'il ne soit plus possible d'insérer, de modifier ou de supprimer des données à travers la vue.

Troisième partie – Assurer la confidentialité à travers les vues :

13) Gestion de la confidentialité des clients

- Permettre aux étudiants de la classe (et de la promotion) de voir toutes les données qui se trouvent dans votre table *Clients*. Pour cela :
 - attribuez à tous les étudiants de votre promotion (PUBLIC) le droit de visualiser les informations qui se trouvent dans votre table *Clients*.
 - quand votre binôme préféré en a fait de même, visualisez les données qui se trouvent dans sa table *Clients*.
 - demandez à votre binôme de rajouter un nouveau client dans sa table *Clients*. Pouvez-vous le voir ? Si ce n'est pas le cas, demandez à votre binôme de valider sa transaction (COMMIT) puis regardez à nouveau si vous pouvez voir le client ajouté.
 - vérifiez que vous ne pouvez pas insérer des données dans la table *Clients* de votre binôme.
 - créez un synonyme de la table *Clients* de votre binôme préféré (par exemple *ClientsDeMonBinome*). Puis visualisez la liste des clients de votre binôme à travers ce synonyme.
- Permettre à votre binôme de voir uniquement les clients montpelliérains de votre table *Clients*. Pour cela :
 - révoquez les droits que vous avez attribués sur votre table *Clients*.
 - attribuez les droits adéquats à votre binôme afin qu'il puisse voir uniquement les clients montpelliérains de votre table *Clients* (pour cela, vous pouvez utiliser une vue).
- Permettre à votre binôme d'insérer des nouveaux clients montpelliérains dans votre table *Clients* (les privilèges accordés à votre binôme ne doivent pas lui permettre d'insérer des clients qui ne sont pas montpelliérains). Une fois que votre binôme en a fait de même vérifiez que :
 - vous pouvez insérer un client montpelliérain dans la table *Clients* de votre binôme.
 - vous ne pouvez pas insérer un client qui n'est pas montpelliérain dans la table *Clients* de votre binôme.

14) Gestion de la confidentialité des locations

- Permettre à votre binôme favori de voir toutes les locations de votre schéma qui concernent le mois de juillet 2021. Pour chacune de ces locations, votre binôme doit pouvoir visualiser :
 - le numéro de la location
 - l'identifiant, le nom et le prénom du client
 - l'identifiant du bungalow, son nom et le nom de son camping
 - la date de début et la date de fin de la location
 - le montant de la location
- Permettre à votre binôme favori d'ajouter dans votre schéma des locations qui concernent le mois de juillet 2021 (il ne doit pas pouvoir insérer une location qui ne concerne pas ce mois-ci).

Quatrième partie – Simplifier une requête grâce aux vues :

15) En vous aidant de vues, réaliser les requêtes suivantes :

- **R22** : le nom et prénom de l'employé le plus mal payé du camping 'Les Flots Bleus'.
 - créer une vue *EmployesLFB* contenant l'identifiant, le nom, le prénom et le salaire des employés qui sont affectés au camping 'Les Flots Bleus'. Cette vue doit avoir la structure suivante :
EMPLOYESLFB (id, nom, prenom, salaire)
 - en utilisant la vue précédemment créée, réaliser la requête 22.
- **R117** : le nom du (ou des) camping qui a (ou ont) le plus d'employés.
 - créer une vue *EmployesParCamping* contenant le nombre d'employés de chaque camping. Cette vue doit avoir la structure suivante :
EMPLOYESPARCAMPING (nomCamping, nbEmployes)
 - en utilisant la vue précédemment créée, réaliser la requête 117.
- **R117B** : le nom du camping qui a le **moins** d'employés (attention, certains campings n'ont pas d'employé !).
 - pouvez-vous utiliser la vue *EmployesParCamping* pour réaliser cette requête ?
 - si ce n'est pas le cas, modifier la vue *EmployesParCamping* afin de pouvoir l'utiliser pour réaliser la requête 117B.
- **R122** : pour chacun des services, le nombre de bungalows qui proposent uniquement ce service (et pas un autre service).
 - créer une vue *BungalowsUnSeulService* contenant l'identifiant des bungalows qui ne proposent qu'un seul service. Cette vue doit avoir la structure suivante :
BUNGALOWSUNSEULSERVICE (idBungalow)
 - en utilisant la vue précédemment créée, réaliser la requête 122.
- **R123** : pour chacun des campings, indiquer le nom du bungalow qui propose le plus de services.
 - à vous de trouver une vue susceptible de simplifier cette requête ...
- **R125** : la liste des bungalows qui proposent exactement les mêmes services qu'un autre bungalow. Les noms des bungalows doivent être séparés par une virgule et les noms des services que proposent ces bungalows doivent aussi être séparés par une virgule.

Liste des Bungalows	Liste des Services
Le Caniveau, Le Polymorphe, Le Dépotoir	
La Suite Présidentielle, La Suite Régalienn	Lave Vaiselle, TV

- il est conseillé de faire une ou plusieurs vues pour réaliser cette requête (même si cela n'est pas obligatoire).
- pour concaténer des chaînes de caractères dans une requête, on pourra éventuellement utiliser les fonctions `WM_CONCAT()` ou `LISTAGG` qui sont disponibles depuis Oracle 11g.