

DS08

Bases de données

Sources :

Proposition de corrigé

Exercice 1 : Base de données de gestion d'un hotel

Q 1 : Donner la requête permettant de lister tous les noms, prénoms et les titres (M. Mme. ou Melle.) des clients.

```
SELECT CLI_NOM, CLI_PRENOM, TIT_CODE FROM T_CLIENT;
```

Q 2 : Donner le nombre de clients enregistrés dans la base.

```
SELECT COUNT(*) FROM T_CLIENT ;
```

Q 3 : Trouver les noms et prénoms des clients dont le titre est 'Mme.' (madame).

```
SELECT CLI_NOM, CLI_PRENOM, TIT_CODE FROM T_CLIENT WHERE TIT_CODE= 'Mme. ';
```

Q 4 : Donner les noms , prénoms et les titres (Mme. ou Melle.) des clientes.

```
SELECT CLI_NOM, CLI_PRENOM, TIT_CODE FROM T_CLIENT WHERE TIT_CODE= 'Mme.' OR TIT_CODE= 'Melle.';
```

Q 5 : Donner le nombre de clientes.

```
SELECT count(*) FROM T_CLIENT WHERE TIT_CODE= 'Mme.' OR TIT_CODE= 'Melle.';
```

Q 6 : Classer par ordre alphabétique les clients de sexe féminin. On ne demande que les noms et les prénoms qui devront être appelés Noms et Prénoms.

```
SELECT CLI_NOM as noms, CLI_PRENOM as prenom FROM T_CLIENT WHERE TIT_CODE= 'Mme.' OR TIT_CODE= 'Melle.' ORDER BY noms, prenom;
```

Q 7 : Établir une nouvelle relation (table) faisant apparaître les noms des clients et leurs numéros de téléphone.

```
SELECT T_CLIENT.CLI_NOM, T_TELEPHONE.TEL_NUMERO FROM T_CLIENT JOIN T_TELEPHONE ON T_CLIENT.CLI_ID=T_TELEPHONE.CLI_ID WHERE T_TELEPHONE.TYP_CODE="TEL";
```

Q 8 : Donner la liste des clients ayant le même nom

```
SELECT NOM FROM (SELECT CLI_NOM as Nom, CLI_PRENOM as prenom, COUNT(*) as nb_nom FROM T_CLIENT GROUP BY Nom) WHERE nb_nom>1;
```

Q 9 : Donner le nombre d'occurrences de chacun de ces noms.

```
SELECT * FROM (SELECT CLI_NOM, COUNT(CLI_NOM) AS ct FROM T_CLIENT GROUP BY CLI_NOM
ORDER BY CLI_NOM) WHERE ct>1;
```

Q 10 : Donner la valeur moyenne des remises en pourcentage et en montant.

```
SELECT AVG(LIF_REMISE_MONTANT), AVG(LIF_REMISE_POURCENT) FROM T_LIGNE_FACTURE;
```

Q 11 : Donner la valeur maximale des remises en pourcentage et en montant.

```
SELECT MAX(LIF_REMISE_MONTANT), AVG(LIF_REMISE_POURCENT) FROM T_LIGNE_FACTURE;
```

Q 12 : Donner les identifiants facture (FAC_ID que l'on renomera fac_id1) qui ont bénéficié de remise (soit en pourcentage soit en montant).

```
SELECT DISTINCT FAC_ID FROM T_LIGNE_FACTURE WHERE LIF_REMISE_POURCENT>0 OR
LIF_REMISE_MONTANT>0 ;
```

Q 13 : Donner les identifiants clients (CLI_ID) qui ont bénéficié d'une remise (soit en pourcentage soit en montant).

```
SELECT DISTINCT CLI_ID FROM T_FACTURE JOIN (SELECT T_LIGNE_FACTURE.FAC_ID as facid1
FROM T_LIGNE_FACTURE WHERE LIF_REMISE_POURCENT>0 OR LIF_REMISE_MONTANT>0) ON
facid1=T_FACTURE.FAC_ID;
```

Q 14 : Donner les identifiants clients (CLI_ID) qui n'ont pas bénéficié d'une remise.

```
SELECT CLI_ID FROM T_CLIENT EXCEPT SELECT DISTINCT CLI_ID FROM T_FACTURE JOIN (
SELECT T_LIGNE_FACTURE.FAC_ID as facid1 FROM T_LIGNE_FACTURE WHERE
LIF_REMISE_POURCENT>0 OR LIF_REMISE_MONTANT>0) ON facid1=T_FACTURE.FAC_ID;
```

Q 15 : Donner le nom et prénom du client qui a le plus bénéficié de remises. Donner le montant de cette remise totale.

```
SELECT T_CLIENT.CLI_NOM, T_CLIENT.CLI_PRENOM, max(montant_total) FROM T_CLIENT JOIN
(SELECT CLI_ID as id_client, sum(montant) as montant_total FROM T_FACTURE JOIN (
SELECT T_LIGNE_FACTURE.FAC_ID as facid1, T_LIGNE_FACTURE.LIF_REMISE_MONTANT as
montant FROM T_LIGNE_FACTURE WHERE LIF_REMISE_POURCENT>0 OR LIF_REMISE_MONTANT
>0) ON facid1=T_FACTURE.FAC_ID GROUP BY CLI_ID) ON T_CLIENT.CLI_ID=id_client;;
```