



SOCIAL PROFILES



Search

[Home](#) » [SQL](#) » Série 2 Exercices SQL

Série 2 Exercices SQL

0

By [X-Proof](#) 06:36

Série Exercices SQL



Exercice 1

On suppose qu'une bibliothèque gère une base de données dont le schéma est le suivant (les clés primaires des relations sont soulignées)

Emprunt (Personne, Livre, DateEmprunt, DateRetourPrevue, DateRetourEffective)

Retard (Personne, Livre, DateEmprunt, PenalitéRetard)

Exprimer, lorsque cela est possible, les requêtes suivantes

en **algèbre relationnelle**, en **calcul à variable nuplet** et en **SQL**

Quelles sont les personnes ayant emprunté le livre "Recueil Examens BD" ?

Quelles sont les personnes n'ayant jamais rendu de livre en retard ?

Quelles sont les personnes ayant emprunté tous les livres (empruntés au moins une fois) ?

Quels sont les livres ayant été empruntés par tout le monde (i.e. tous les emprunteurs) ?

Quelles sont les personnes ayant toujours rendu en retard les livres qu'elles ont empruntés ?

Exercice 2

Soit le modèle relationnel suivant relatif à la gestion simplifiée des étapes du Tour de France 97, dont une des étapes de type "contre la montre individuel" se déroula à Saint-Etienne :

EQUIPE(CodeEquipe, NomEquipe, DirecteurSportif)

COUREUR(NuméroCoureur, NomCoureur, CodeEquipe*, CodePays*)

PAYS(CodePays, NomPays)

TYPE_ETAPE(CodeType, LibelléType)

ETAPE(NuméroEtape, DateEtape, VilleDép, VilleArr, NbKm, CodeType*)

PARTICIPER(NuméroCoureur*, NuméroEtape*, TempsRéalisé)

ATTRIBUER_BONIFICATION(NuméroEtape*, km, Rang, NbSecondes, NuméroCoureur*)

Remarque : les clés primaires sont soulignées et les clés étrangères sont marquées par *

Requête 1 - Quelle est la composition de l'équipe Festina (Numéro, nom et pays des coureurs) ?

Requête 2 - Quel est le nombre de kilomètres total du Tour de France 97 ?

Requête 3 - Quel est le nombre de kilomètres total des étapes de type "Haute Montagne" ?

Requête 4 - Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?

Requête 5 - Quels sont les noms des coureurs qui ont participé à toutes les étapes ?

Requête 6 - Quel est le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l'issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps réalisés à chaque étape ?

Requête 7 - Quel est le classement par équipe à l'issue des 13 premières étapes (nom et temps des équipes) ?

Exercice 3

Un organisme de gestion de spectacles, de salles de concert et de vente de billets de spectacles gère une base de données dont le schéma relationnel est le suivant :

Spectacle (Spectacle_ID, Titre, DateDéb, Durée, Salle_ID, Chanteur)

Concert (Concert_ID, Date, Heure, Spectacle_ID)

Salle (Salle_ID, Nom, Adresse, Capacité)

Billet (Billet_ID, Concert_ID, Num_Place, Catégorie, Prix)

Vente (Vente_ID, Date_Vente, Billet_ID, MoyenPaiement)

Popular

Tags

Blog Archives



Série 2 Exercices UML

Série 2 Exercices UML Exercice 1 : DAB Le système à étudier est celui d'un distributeur de billets (DAB). • Un client peut effe...

$$3) = \overline{A \cdot B}$$

A

$$A + \overline{B} + A \cdot \overline{C} =$$

Exercices Algèbre de boole

Exercice 1 – Algèbre de Boole
Exercice 2 – Analyse et synthèse de circuits L'implémentation d'une fonction logique Z relativement ...



Série 2 Exercices Langage C

Série 2 Exercices Langage C
Exercice1 On appelle palindrome une suite de caractères qui se lit de la même façon dans les deux sen...

Exercices algèbre de boole et circuits logiques

Sachant que A, B, C et D sont des variables booléennes. a) En utilisant exclusivement

l'algèbre booléenne, démontrez que
: b) En u...

=



Exercice Corrigé base de données relationnel SGBD Access

Énoncé 1 Un fanatique de cinéma et de festivals de films décide de réaliser sa base de données personnelle en y stockant les diffé...

Fourni par Blogger.

FOLLOW BY EMAIL

Les attributs soulignés sont les attributs appartenant à la clé primaire. Ils sont de type entier. L'attribut Salle_ID de la relation Spectacle est une clé étrangère qui fait référence à l'attribut de même nom de la relation Salle. L'attribut Spectacle_ID de la relation Concert est une clé étrangère qui fait référence à l'attribut de même nom de la relation Spectacle. L'attribut Concert_ID de la relation Billet est une clé étrangère qui fait référence à l'attribut de même nom de la relation Concert. L'attribut Billet_ID de la relation Vente est une clé étrangère qui fait référence à l'attribut de même nom de la relation Billet.

Exprimez, lorsque cela est possible, les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variable nuplet et en [SQL](#).

Quelles sont les dates du concert de Corneille au Zenith ?

Quels sont les noms des salles ayant la plus grande capacité ?

Quels sont les chanteurs n'ayant jamais réalisé de concert à la Cygale ?

Quels sont les chanteurs ayant réalisé au moins un concert dans toutes les salles ?

Quels sont les dates et les identificateurs des concerts pour lesquels il ne reste aucun billet invendu ?

Exercice 4

Soit le schéma de base de données relationnel suivant :

AGENCE (Num_Agence, Nom, Ville, Actif)

CLIENT (Num_Client, Nom, Ville)

COMPTE (Num_Compte, Num_Agence, Num_Client, Solde)

EMPRUNT (Num_Emprunt, Num_Agence, Num_Client, Montant)

Ecrire les requêtes suivantes en [SQL](#) :

Liste des agences ayant des comptes-clients

Clients ayant un compte à "Agadir"

Clients ayant un compte ou un emprunt à "Agadir"

Clients ayant un compte et un emprunt à "Agadir"

Clients ayant un compte et pas d'emprunt à "Agadir"

Clients ayant un compte et nom de la ville où ils habitent

Clients ayant un compte à "Paris-Etoile" et nom de la ville où ils habitent

Clients ayant un compte dans une agence où "Ahmed" a un compte

Agences ayant un actif plus élevé que toute agence d'"AAAA"

Clients ayant un compte dans chaque agence d'"AAAA"

Clients ayant un compte dans au-moins une agence d'"AAAA"

Emprunteurs de l'agence "Crédit-Agadir" classés par ordre alphabétique

Solde moyen des comptes-clients de chaque agence

Solde moyen des comptes-clients des agences dont le solde moyen est > "10 000"

Nombre de clients habitant "Safi"

Nombre de clients de l'agence "CETELEM" n'ayant pas leur adresse dans la relation CLIENT

Insérer le n-uplet dans la relation CLIENT

Diminuer l'emprunt de tous les clients habitant "Tanger" de "5%"

Fermer les comptes de "Ahmed"

Supprimer de la relation AGENCE toutes les agences sans client

Exercice 5

Soit le modèle relationnel suivant relatif à la gestion des notes annuelles d'une promotion d'étudiants :

ETUDIANT(N°Etudiant, Nom, Prénom)

MATIERE(CodeMat, LibelléMat, CoeffMat)

EVALUER(N°Etudiant*, CodeMat*, Date, Note)

Remarque : les clés primaires sont soulignées et les clés étrangères sont marquées par *

Requête 1 - Quel est le nombre total d'étudiants ?

Requête 2 - Quelles sont, parmi l'ensemble des notes, la note la plus haute et la note la plus basse ?

Requête 3 - Quelles sont les moyennes de chaque étudiant dans chacune des matières ?

Requête 4 - Quelles sont les moyennes par matière ? On utilisera la requête de la question 3 comme table source

Requête 5 - Quelle est la moyenne générale de chaque étudiant ?

On utilisera la requête de la question 3 comme table source

Requête 6 - Quelle est la moyenne générale de la promotion ?

On utilisera la requête de la question 5 comme table source

Requête 7 - Quels sont les étudiants qui ont une moyenne générale supérieure ou égale à la moyenne générale de la promotion ?

On utilisera la requête de la question 5 comme table source

Correction

| Exercice 1 | Exercice 2 | Exercice 5 |
|--|---|---|
| <p>1- Quelles sont les personnes ayant emprunté le livre "Recueil Examens BD" ?</p> <pre>SELECT Personne FROM Emprunt WHERE Livre = 'Recueil...</pre> <p>2- Quelles sont les personnes n'ayant jamais rendu de livre en retard ?</p> <pre>SELECT t.Personne FROM Emprunt t WHERE NOT EXISTS SELECT * FROM Retard u WHERE u.Personne=t.Personne)</pre> <p>SELECT Personne FROM Emprunt WHERE Personne NOT IN (SELECT Personne FROM Retard)</p> <p>3- Quelles sont les personnes ayant emprunté tous les livres (empruntés au moins une fois) ?</p> <pre>SELECT t.Personne FROM Emprunt t WHERE NOT EXISTS (SELECT * FROM Emprunt u WHERE NOT EXISTS (SELECT * FROM Emprunt v WHERE v.Personne=t.Personne AND v.Livre=u.Livre))</pre> <p>4- Quels sont les livres ayant été empruntés par tout le monde (i.e. tous les emprunteurs)?</p> <pre>SELECT t.Livre FROM Emprunt t WHERE NOT EXISTS (SELECT * FROM Emprunt u WHERE NOT EXISTS (SELECT * FROM Emprunt v WHERE u.Livre=t.Livre AND v.Personne=u.Personne))</pre> <p>5- Quelles sont les personnes ayant toujours rendu en retard les livres qu'elles ont empruntés ?</p> <pre>SELECT t.Personne FROM Emprunt t WHERE NOT EXISTS (SELECT * FROM Emprunt u WHERE u.Personne=t.Personne AND NOT EXISTS (SELECT * FROM Retard v WHERE v.Personne=u.Personne</pre> | <p>1 - Quelle est la composition de l'équipe FESTINA (Numéro, nom et pays des coureurs) ?</p> <pre>SELECT NuméroCoureur, NomCoureur, NomPays FROM EQUIPE A, COUREUR B, PAYS C WHERE A.CodeEquipe=B.CodeEquipe And B.CodePays=C.CodePays And NomEquipe="FESTINA" ;</pre> <p>2 - Quel est le nombre de kilomètres total du Tour de France 97 ?</p> <pre>SELECT SUM(Nbkm) FROM ETAPE ;</pre> <p>3 - Quel est le nombre de kilomètres total des étapes de type HAUTE MONTAGNE ?</p> <pre>SELECT SUM(Nbkm) FROM ETAPE A, TYPE_ETAPE B WHERE A.CodeType=B.CodeType And LibelléType="HAUTE MONTAGNE" ;</pre> <p>4 - Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?</p> <pre>SELECT NomCoureur FROM COUREUR WHERE NuméroCoureur NOT IN (SELECT NuméroCoureur FROM ATTRIBUER_BONIFICATION) ;</pre> <p>5 - Quels sont les noms des coureurs qui ont participé à toutes les étapes ?</p> <pre>SELECT NomCoureur FROM PARTICIPER A, COUREUR B WHERE A.NuméroCoureur=B.NuméroCoureur GROUP BY NuméroCoureur, NomCoureur HAVING COUNT(*)=(SELECT COUNT(*) FROM ETAPE) ;</pre> <p>6 - Quel est le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l'issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps réalisés à chaque étape ?</p> <pre>SELECT NomCoureur, CodeEquipe, CodePays, SUM(TempsRéalisé) AS Total FROM PARTICIPER A, COUREUR B WHERE A.NuméroCoureur=B.NuméroCoureur and NuméroEtape<=13 GROUP BY A.NuméroCoureur, NomCoureur, CodeEquipe, CodePays</pre> | <p>1 - Quel est le nombre total d'étudiants ?</p> <pre>SELECT COUNT(*) FROM ETUDIANT ;</pre> <p>2 - Quelles sont, parmi l'ensemble des notes, la note la plus haute et la note la plus basse ?</p> <pre>SELECT MIN(Note), MAX(Note) FROM EVALUER ;</pre> <p>3 - Quelles sont les moyennes de chaque étudiant dans chacune des matières?</p> <pre>SELECT ETUDIANT.N°Etudiant, Nom, Prénom, LibelléMat, CoeffMat, AVG(Note) AS MoyEtuMat FROM EVALUER, MATIERE, ETUDIANT WHERE EVALUER.CodeMat = MATIERE.CodeMat AND EVALUER.N°Etudiant = ETUDIANT.N°Etudiant</pre> <p>GROUP BY ETUDIANT.N°Etudiant, Nom, Prénom, LibelléMat, CoeffMat;</p> <p>4 - Quelles sont les moyennes par matière ?</p> <pre>SELECT LibelléMat, AVG(MoyEtuMat) FROM MOYETUMAT GROUP BY LibelléMat ;</pre> <p>5 - Quelle est la moyenne générale de chaque étudiant ?</p> <pre>SELECT N°Etudiant, Nom, Prénom, SUM(MoyEtuMat*CoeffMat)/SUM(CoeffMat) AS MgEtu FROM MOYETUMAT GROUP BY N°Etudiant, Nom, Prénom ;</pre> <p>6 - Quelle est la moyenne générale de la promotion ? Avec la vue MGETU de la question 5 :</p> <pre>SELECT AVG(MgEtu) FROM MGETU ;</pre> <p>7 - Quels sont les étudiants qui ont une moyenne générale supérieure ou égale à la moyenne générale de la promotion ? Avec la vue MGETU de la question 5 :</p> <pre>SELECT N°Etudiant, Nom, Prénom, MgEtu FROM MGETU WHERE MgEtu >= (SELECT AVG(MgEtu) FROM MGETU) ;</pre> |

| | | |
|--|---|--|
| <p>AND</p> <p>v.Livre=u.Livre</p> <p>)</p> <p>)</p> | <p>ORDER BY Total;</p> <p>7 - Quel est le classement par équipe à l'issue des 13 premières étapes (nom et temps des équipes) ?</p> <p>SELECT NomEquipe,</p> <p>SUM(TempsRéalisé) AS Total</p> <p>FROM PARTICIPER A, COUREUR B,</p> <p>EQUIPE C</p> <p>WHERE</p> <p>A.NuméroCoureur=B.NuméroCoureur</p> <p>And B.CodeEquipe=C.CodeEquipe</p> <p>And NuméroEtape<=13</p> <p>GROUP BY B.CodeEquipe,</p> <p>NomEquipe</p> <p>ORDER BY Total;</p> | |
| <p>Exercice3</p> <p>1- Quelles sont les dates du concert de Corneille au Zenith ?</p> <p>SELECT Date</p> <p>FROM Concert t, Spectacle u, Salle v</p> <p>WHERE t.Spectacle_ID =</p> <p>u.Spectacle_ID</p> <p>AND u.Chanteur = 'Corneille'</p> <p>AND u.Salle_ID = v.Salle_ID</p> <p>AND v.Nom = 'Zenith'</p> <p>2- Quels sont les noms des salles ayant la plus grande capacité ?</p> <p>Il est possible de traduire directement la requête exprimée en calcul relationnel, comme ci-dessous.</p> <p>SELECT Nom</p> <p>FROM Salle t</p> <p>WHERE NOT EXISTS</p> <p>(SELECT * FROM Salle u</p> <p>WHERE u.Capacité >= t.</p> <p>Capacité</p> <p>)</p> <p>Il est également possible d'utiliser l'opérateur d'agrégation MAX, comme pour la requête suivante.</p> <p>SELECT Nom</p> <p>FROM Salle</p> <p>WHERE Capacité >= (SELECT</p> <p>(MAX(Capacité)</p> <p>FROM Salle</p> <p>)</p> <p>Il est également possible d'utiliser le mot-clé ALL :</p> <p>SELECT Nom</p> <p>FROM Salle</p> <p>WHERE Capacité >= ALL (SELECT</p> <p>Capacité</p> <p>FROM Salle</p> <p>)</p> <p>3- Quels sont les chanteurs n'ayant jamais réalisé de concert à la Cygale ?</p> <p>SELECT Chanteur</p> <p>FROM Spectacle</p> <p>WHERE Chanteur NOT IN</p> | <p>Exercice4</p> <p>1. Liste des agences ayant des comptes-clients</p> <p>SELECT DISTINCT Nom FROM</p> <p>AGENCE, COMPTE</p> <p>WHERE AGENCE.Num_Agence =</p> <p>COMPTE.Num_Agence</p> <p>2. Clients ayant un compte à "Agadir"</p> <p>SELECT CLIENT.Nom FROM CLIENT,</p> <p>AGENCE, COMPTE</p> <p>WHERE AGENCE.Num_Agence =</p> <p>COMPTE.Num_Agence</p> <p>AND CLIENT.Num_Client =</p> <p>COMPTE.Num_Client</p> <p>AND AGENCE.Ville = "Agadir"</p> <p>3. Clients ayant un compte ou un emprunt à "Agadir"</p> <p>SELECT CLIENT.Nom FROM CLIENT,</p> <p>AGENCE, COMPTE</p> <p>WHERE CLIENT.Num_Client =</p> <p>COMPTE.Num_Client</p> <p>AND AGENCE.Num_Agence =</p> <p>COMPTE.Num_Agence</p> <p>AND AGENCE.Ville = "Agadir"</p> <p>UNION</p> <p>SELECT CLIENT.Nom FROM CLIENT,</p> <p>AGENCE, EMPRUNT</p> <p>WHERE CLIENT.Num_Client =</p> <p>EMPRUNT.Num_Client</p> <p>AND AGENCE.Num_Agence =</p> <p>EMPRUNT.Num_Agence</p> <p>AND AGENCE.Ville = "Agadir"</p> <p>4. Clients ayant un compte et un emprunt à "Agadir"</p> <p>SELECT CLIENT.Nom FROM CLIENT,</p> <p>AGENCE, COMPTE</p> <p>WHERE CLIENT.Num_Client =</p> <p>COMPTE.Num_Client</p> <p>AND AGENCE.Num_Agence =</p> <p>COMPTE.Num_Agence</p> <p>AND AGENCE.Ville = "Agadir"</p> | |

| | |
|--|--|
| <pre> (SELECT Chanteur FROM Spectacle u, Salle v WHERE u.Salle_ID=v.Salle_ID AND v.Nom='Cygale') </pre> | <pre> intersect SELECT CLIENT.Nom FROM CLIENT, AGENCE, EMPRUNT WHERE CLIENT.Num_Client = EMPRUNT.Num_Client AND AGENCE.Num_Agence = EMPRUNT.Num_Agence AND AGENCE.Ville = "Agadir" </pre> |
| <p>Cette requête peut aussi s'exprimer avec un NOT EXISTS en utilisant une variable nuplet tdans le premier FROM, par une simple traduction du calcul relationnel :</p> <pre> SELECT Chanteur FROM Spectacle t WHERE Chanteur NOT EXISTS (SELECT * FROM Spectacle u, Salle v WHERE u.Salle_ID=v.Salle_ID AND v.Nom='Cygale' AND t.CHanteur=u.Chanteur) </pre> | <p>5. Clients ayant un compte et pas d'emprunt à "Agadir"</p> <pre> SELECT CLIENT.Nom FROM CLIENT, AGENCE, COMPTE WHERE CLIENT.Num_Client = COMPTE.Num_Client AND AGENCE.Num_Agence = COMPTE.Num_Agence AND AGENCE.Ville = "Agadir" </pre> |
| <p>4- Quels sont les chanteurs ayant réalisé au moins un concert dans toutes les salles?</p> <pre> SELECT Chanteur FROM Spectacle t WHERE NOT EXISTS (SELECT * FROM Salle u WHERE NOT EXISTS (SELECT * FROM Spectacle v WHERE v.Chanteur = t. Chanteur AND u.Salle_ID = v.Salle_ID)) </pre> | <p>minus</p> <pre> SELECT CLIENT.Nom FROM CLIENT, AGENCE, EMPRUNT WHERE CLIENT.Num_Client = EMPRUNT.Num_Client AND AGENCE.Num_Agence = EMPRUNT.Num_Agence AND AGENCE.Ville = "Agadir" </pre> |
| <p>5- Quels sont les dates et les identifiants des concerts pour lesquels il n'existe aucun billet invendu ?</p> <pre> SELECT Concert_ID, Date FROM Concert t WHERE NOT EXISTS (SELECT * FROM Billet u WHERE u.Concert_ID=t.Concert_ID AND NOT EXISTS (SELECT * FROM Vente v WHERE u.Billet_ID = v.Billet_ID)) </pre> | <p>6. Clients ayant un compte et nom de la ville où ils habitent</p> <p>Première solution :</p> <pre> SELECT Nom, Ville FROM CLIENT, COMPTE WHERE CLIENT.Num_Client = COMPTE. Num_Client </pre> <p>Deuxième solution :</p> <pre> SELECT Nom, Ville FROM CLIENT WHERE Num_Client IN (SELECT Num_Client FROM COMPTE) </pre> |
| <p>7. Clients ayant un compte à "wafa-agadir" et nom de la ville où ils habitent</p> <p>Première solution:</p> <pre> SELECT CLIENT.Nom, CLIENT.Ville FROM CLIENT, AGENCE, COMPTE WHERE CLIENT.Num_Client = COMPTE. Num_Client AND AGENCE.Num_Agence = COMPTE.Num_Agence AND AGENCE.Nom = "wafa-agadir" </pre> <p>Deuxième solution :</p> <pre> SELECT Nom, Ville FROM CLIENT WHERE Num_Client IN (SELECT Num_Client FROM COMPTE WHERE Num_Agence IN (SELECT Num_Agence FROM AGENCE WHERE Nom = "wafa- agadir")) </pre> | <p>7. Clients ayant un compte à "wafa-agadir" et nom de la ville où ils habitent</p> <p>Première solution:</p> <pre> SELECT CLIENT.Nom, CLIENT.Ville FROM CLIENT, AGENCE, COMPTE WHERE CLIENT.Num_Client = COMPTE. Num_Client AND AGENCE.Num_Agence = COMPTE.Num_Agence AND AGENCE.Nom = "wafa-agadir" </pre> <p>Deuxième solution :</p> <pre> SELECT Nom, Ville FROM CLIENT WHERE Num_Client IN (SELECT Num_Client FROM COMPTE WHERE Num_Agence IN (SELECT Num_Agence FROM AGENCE WHERE Nom = "wafa- agadir")) </pre> |

8. Clients ayant un compte dans une agence où "Ahmed" a un compte

Première solution :

```
SELECT Nom FROM CLIENT,
COMPTE
WHERE CLIENT.Num_Client =
COMPTE.Num_Client AND
Num_Agence IN (
SELECT Num_Agence FROM CLIENT,
COMPTE
WHERE CLIENT.Num_Client =
COMPTE.Num_Client AND Nom =
"Ahmed")
```

Deuxième solution :

```
SELECT Nom FROM CLIENT WHERE
Num_Client IN (
SELECT Num_Client FROM COMPTE
WHERE Num_Agence IN (
SELECT Num_Agence FROM CLIENT,
COMPTE
WHERE CLIENT.Num_Client =
COMPTE.Num_Client AND Nom =
"Ahmed"))
```

9. Agences ayant un actif plus élevé que toute agence de "AAAA"

```
SELECT Nom FROM AGENCE
WHERE Actif > ALL (
SELECT Actif FROM AGENCE
WHERE Ville = "AAAA")
```

11. Clients ayant un compte dans au-moins une agence d"AAAA"

Première solution :

```
SELECT Nom FROM CLIENT WHERE
Num_Client IN (
SELECT Num_Client FROM COMPTE
WHERE Num_Agence IN (
SELECT Num_Agence FROM
AGENCE WHERE Ville = "AAAA"))
```


Deuxième solution :

```
SELECT CLIENT.Nom FROM CLIENT,
COMPTE, AGENCE
WHERE CLIENT.Num_Client =
COMPTE.Num_Client
AND COMPTE.Num_Agence =
AGENCE.Num_Agence
AND AGENCE.Ville = "AAAA"
```

12. Emprunteurs de l'agence "Grédit-Agadir" classés par ordre alphabétique

```
SELECT Nom FROM CLIENT WHERE
Num_Client IN (
SELECT Num_Client FROM
EMPRUNT WHERE Num_Agence IN (
SELECT Num_Agence FROM
AGENCE WHERE Nom = "Grédit-
```

| | |
|--|--|
| | <p>Agadir"))</p> <p>ORDER BY Nom</p> <p>13. Solde moyen des comptes-clients de chaque agence</p> <p>SELECT Nom, avg(Solde) FROM</p> <p>AGENCE, COMPTE</p> <p>WHERE AGENCE.Num_Agence =</p> <p>COMPTE.Num_Agence</p> <p>GROUP BY Nom</p> <p>14. Solde moyen des comptes-clients des agences dont le solde moyen est > "10 000"</p> <p>SELECT Nom, avg(Solde) FROM</p> <p>AGENCE, COMPTE</p> <p>WHERE AGENCE.Num_Agence =</p> <p>COMPTE.Num_Agence</p> <p>GROUP BY Nom</p> <p>HAVING avg(Solde) > 10000</p> <p>15. Nombre de clients habitant " Safi "</p> <p>SELECT count(*) FROM CLIENT</p> <p>WHERE Ville = "Safi"</p> <p>16. Nombre de clients de l'agence "CETEM " n'ayant pas leur adresse dans la relation CLIE</p> <p>Première solution :</p> <p>SELECT count(*) FROM CLIENT</p> <p>WHERE Ville = NULL AND</p> <p>Num_Client IN (</p> <p>SELECT Num_Client FROM COMPTE</p> <p>WHERE Num_Agence IN (</p> <p>SELECT Num_Agence FROM</p> <p>AGENCE WHERE Nom = "CETEM</p> <p>")</p> <p>Deuxième solution :</p> <p>SELECT count(*) FROM CLIENT,</p> <p>COMPTE, AGENCE</p> <p>WHERE Ville = NULL</p> <p>AND CLIENT.Num_Client =</p> <p>COMPTE.Num_Client</p> <p>AND COMPTE.Num_Agence =</p> <p>AGENCE.Num_Agence</p> <p>AND AGENCE.Nom = "CETEM "</p> <p>17. Insérer le n-uplet < leila, Agadir > dans la relation CLIENT</p> <p>INSERT INTO CLIENT VALUES</p> <p>(130765, "leila", "Agadir")</p> <p>18. Diminuer l'emprunt de tous les clients habitant "Tanger" de "5%"</p> |
|--|--|

 Recommander ce contenu sur Google

Related Posts:



Série Exercices SQL

Série Exercices SQL Exercice 1 Faire les requêtes SQL permettant de répondre aux demandes sur la base de données suivante: Servir (Café, Boisson) Fréquenter (Client, Café) Apprécier (Client, Boisson) Exprimer l...

[Read More](#)

Série 2 Exercices SQL

Série Exercices SQL Exercice 1 On suppose qu'une bibliothèque gère une base de données dont le schéma est le suivant (les clés primaires des relations sont soulignées) Emprunt (Personne, Livre, DateEmprunt, Date...

[Read More](#)[← Article plus récent](#)[Accueil](#)

0 commentaires :

Enregistrer un commentaire

Saisissez votre commentaire...

Commentaire : Sélectionner le pro

Publier

Aperçu

CONTACT

Votre nom

Votre adresse email

Votre message

Envoyer

À PROPOS

X-PROOF

Puisque les langages de programmation ont devenu un outil nécessaire dans le développement informatique, j'ai opté la réalisation d'un petit Blog .. L'objectif de ce Blog est de devenir un outil familier , utile pour les newbies du développement et aussi pour ceux préparant le concours Il rassemble de nombreux exercices, quelques problèmes ,sujets ,et des applications J'espère que vous avez trouvé au moins ce que vous cherchiez :)

[AFFICHER MON PROFIL COMPLET](#)

LABELS

Algorithmique
Bureautique

Réseau
Visual Basic (VB)

Système de numération

Copyright © 2015 - Tous droits réservés | <http://devinfos.blogspot.com>

Theme by Lasantha - Politique de confidentialité