Nom / Prénom : Castellanos Alejandro

Questions SQL

Ecrire une requête permettant d’ajouter toutes les données de la table ENTREES dans DOSSIER avec la date d’entrée (DATE\_ENTREE) = 30/01/2014.

|  |  |  |  |
| --- | --- | --- | --- |
| **DOSSIER** |  | **ENTREES** | |
| IDDOSSIER : varchar(50) NOT NULL |  | IDDOSSIER : varchar(50) NOT NULL | |
| IDCLIENT : varchar(50) NOT NULL |  | IDCLIENT : varchar(50) NOT NULL | |
| DATE\_ENTREE : date NOT NULL |  | |
| IDAGENT : varchar(50) NULL |  | |
|  |  | |

Réponse : Opération invalide on ne peut pas filtrer aucun élément de la table Entrées car le champ Entrees.DATE\_ENTREE n’existe pas.

WITH helper AS (SELECT \* FROM ENTREES) INSERT into DOSSIER VALUES(

SELECT h.IDDOSSIER, h.IDCLIENT,”30/01/2014”, NULL FROM helper h);

Ecrire une procédure stockée appelée set\_Date qui prend en entrée une date et qui modifie la DATE\_ENTREE de tous les dossiers qui ont un ID AGENT non nul.

|  |
| --- |
| **DOSSIER** |
| IDDOSSIER : varchar(50) |
| IDCLIENT : varchar(50) |
| DATE\_ENTREE : date |
| IDAGENT : varchar(50) |

Réponse :

Create PROCEDURE set\_Date (IN @new\_date DATE)

BEGIN

update DOSSIER set DOSSIER.DATE\_ENTREE = @new\_date

where IDAGENT is not null;

END;

Quelle est la valeur de la colonne ENCOURS après l’exécution de cette requête ?

* **SELECT ID, IDAgent, ENCOURS FROM DOSSIER ;**

ID IDAgent ENCOURS

-------- --------- ---------

25D12 AGENT1 6000.00

* **UPDATE DOSSIER SET ENCOURS = 800 WHERE ID=’25D12’;**

1 ligne mise à jour.

* **ROLLBACK;**
* **SELECT ID, IDAgent, ENCOURS FROM DOSSIER ;**

ID IDAgent **ENCOURS**

-------- --------- ---------

25D12 AGENT1 6000.00

|  |
| --- |
| **DOSSIER** |
| ID : int NOT NULL |
| IDCLIENT : int NOT NULL |
| ENCOURS : float NULL |

Donner un aperçu de la table DOSSIER après cette requête.

* **INSERT INTO DOSSIER (ID, ENCOURS) VALUES (12,256.02),(23,454.10);**
* **UPDATE DOSSIER SET IDCLIENT = 1 WHERE IDCLIENT IS NULL;**

Réponse :

|  |  |  |
| --- | --- | --- |
| **ID** | **IDCLIENT** | **ENCOURS** |
| ? |  |  |
|  | ? |  |

Les dossiers d’id 12 et 23 ne seront pas enregistré car ils n’ont pas de idclient, ce champ est *not null* et n’a pas de paramètre *default* donc la requête insert retournera une erreur.

Quelles sont les commandes SQL qui peuvent être annulées dans une transaction ?

1. INSERT
2. ALTER
3. CREATE
4. DROP
5. TRUNCATE
6. DELETE
7. UPDATE

Créer une vue (vueAgent) qui affiche pour chaque agent le nombre de dossiers entrants + le montant total de l’encours.

|  |  |  |  |
| --- | --- | --- | --- |
| **DOSSIER** |  | **AGENT** | |
| IDDOSSIER : int NOT NULL |  | IDAGENT : int NOT NULL | |
| ENCOURS : float NOT NULL | 1..n 1..1 | NOM : varchar(32) NOT NULL | |
| IDAGENT : int NOT NULL | Est géré par | |  | |

**Vue demandée :**

|  |  |  |
| --- | --- | --- |
| **IDAGENT ????** | **NB\_DOSS** | **MT\_DOSS** |
| Agent1 | 5 | 555.22 |
| Agent2 | 45 | 112.55 |
| Agent3 | NULL ??? | NULL ?????? |
| Agent4 | 454 | 55523.22 |

Réponse :

Create view vueAgent (idagent, nb\_doss, mt\_doss) as

Select dossier.**IDAGENT, count**(**distinct** dossier.**IDDOSSIER**) as **NB\_DOSS, SUM(**dossier.**ENCOURS) AS MT\_DOSS**

from dossier , agent

Where dossier.IDAGENT = **AGENT.**IDAGENT

;

Qu’est-ce qu’un déclencheur (*trigger*) ? A quelle(s) commande(s) SQL peut-il être associé ?

Réponse :

Un trigger spécifie que la base de données doit exécuter automatiquement une fonction donnée chaque fois qu'un certain type d'opération est exécuté. Les fonctions déclencheurs peuvent être définies pour s'exécuter avant ou après une commande INSERT, UPDATE ou DELETE, soit une fois par ligne modifiée, soit une fois par expression SQL

Entourer la (les) bonne(s) réponse(s)

1. INSERT
2. ALTER
3. CREATE
4. DROP
5. TRUNCATE
6. DELETE
7. UPDATE

|  |  |  |  |
| --- | --- | --- | --- |
| **DOSSIER** |  | **AGENT** | |
| ID : int NOT NULL |  | ID : int NOT NULL | |
| MT\_RETARD : float NOT NULL | 1..n 1..1 | RETARD\_RESTANT : float NULL | |
| MT\_ENCAISSEMENT : float NOT NULL | Est géré par | |  | |
| IDAgent : int NOT NULL |  | |  | |

Remplir la colonne RETARD\_RESTANT de la table AGENT avec les données de la table DOSSIER.

Remarque : RETARD\_RESTANT = Somme (MT\_RETARD - MT\_ENCAISSEMENT)

Réponse :

UPDATE AGENT

SET AGENT.RETARD\_RESTANT = **DOSSIER**.MT\_RETARD - **DOSSIER.**MT\_ENCAISSEMENT

FROM dossier, agent

WHERE **DOSSIER.**IDAgent = AGENT.**ID;**