

TP sur la programmation d'évènements en SQL :

Timothé VAQUIÉ




Problématique :

On souhaite pouvoir enregistrer les changements d'une base de données. Dans un premier temps dans une table puis par la suite dans un fichier qui sera créer. La sauvegarde pourra être effectuée automatiquement de manière périodique.

Triggers de la table "users"

- add_users_sauv :




Après chaque insertion dans la table users, on ajoute dans la table users_sauv une nouvelle occurrence avec un attribut action en plus. Il aura comme valeur 'C' pour create.

☐ add_users_sauv  Éditer  Exporter  Supprimer AFTER INSERT

```
BEGIN
INSERT INTO users_sauv (IdUser, UsernameUser, MdpHashUser, action) VALUES (NEW.IdUser,
NEW.UsernameUser, NEW.MdpHashUser, 'C');
END
```

- update_users_sauv :




Après chaque insertion dans la table users, on ajoute dans la table users_sauv une nouvelle occurrence avec un attribut action en plus. Il aura comme valeur 'U' pour update.

☐ update_user_sauv  Éditer  Exporter  Supprimer AFTER UPDATE

```
BEGIN
INSERT INTO users_sauv (IdUser, UsernameUser, MdpHashUser, action) VALUES (OLD.IdUser,
OLD.UsernameUser, OLD.MdpHashUser, 'U');
END
```

- delete_users_sauv :

Après chaque insertion dans la table users, on ajoute dans la table users_sauv une nouvelle occurrence avec un attribut action en plus. Il aura comme valeur 'D' pour delete.

☐ delete_user_sauv  Éditer  Exporter  Supprimer AFTER DELETE

```
BEGIN
INSERT INTO users_sauv (IdUser, UsernameUser, MdpHashUser, action) VALUES (OLD.IdUser,
OLD.UsernameUser, OLD.MdpHashUser, 'D');
END
```

Procédure de la base de données

- sauvegarde_users :

Lorsqu'elle est lancée, il nous crée un fichier dans un répertoire défini et sauvegarde toute la table users_sauv. On utilise la sauvegarde complète.

```
BEGIN
SET @fileName = CONCAT('C:/Users/Wallans/Desktop/sql/sauvegarde_users_',CURRENT_DATE(),'.sauv');
SET @requete = CONCAT("SELECT IdUser, UsernameUser, MdpHashUser INTO OUTFILE '", @fileName,""
FROM test.users_sauv");

PREPARE s1 FROM @requete;
EXECUTE s1;
DEALLOCATE PREPARE s1;
END
```

Pour finir, il nous faut activer le programmeur 'event_scheduler'

```
SET GLOBAL event_scheduler=1;
```

On ajoute un événement qui lancera la procédure sauvegarde_users tout les jours.

```
DROP EVENT IF EXISTS sauvegarde_complete;
CREATE EVENT IF NOT EXISTS `sauvegarde_complete`
ON SCHEDULE EVERY 1 DAY
DO CALL bdd.sauvegarde_users;
```