01 – Maitriser le langage de programmation procédurale sous MySQL

Les déclencheurs (Trigger)

Définition

- Un déclencheur est une procédure stockée, dans la base de données, qui s'appelle automatiquement à chaque fois qu'un événement spécial se produit sur la table à laquelle le déclencheur est attaché.
- Un déclencheur peut être utilisé pour valider les données, enregistrer les anciennes et les nouvelles valeurs dans une table d'audit (log) ou s'assurer que les règles métier sont respectées.
- Par exemple, un déclencheur peut être invoqué lorsqu'une ligne est insérée dans une table spécifique ou lorsque certaines colonnes de la table sont mises à jour.

Syntaxe

```
CREATE TRIGGER [nom_declencheur]
[before | after]
{insert | update | delete}
ON [nom_table]
[for each row]
[corps_declencheur]
```

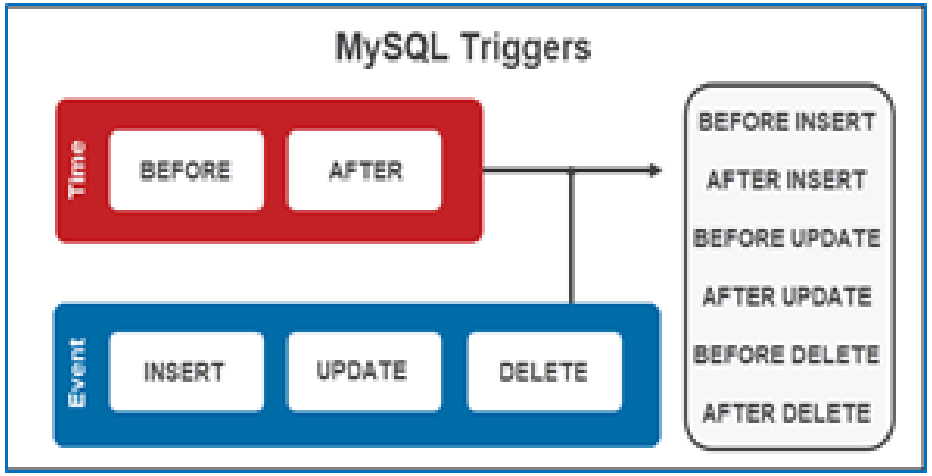
- [before | after] ([avant | apres]) : Ceci spécifie quand le déclencheur sera exécuté.
- {insert | update | delete} ({insérer | mise à jour | supprimer}) : ceci spécifie l'opération DML.
- ON [nom_table] : Ceci spécifie le nom de la table associée au déclencheur.
- [for each row] : Ceci spécifie un déclencheur au niveau de la ligne, c'est-à-dire que le déclencheur sera exécuté pour chaque ligne affectée.
- [corps_declencheur] : Ceci fournit les opérations à effectuer lorsque le déclencheur est déclenché.

01 – Maitriser le langage de programmation procédurale sous MySQL

Les déclencheurs (Trigger)

Types

- Les déclencheurs **BEFORE** exécutent l'action du déclencheur avant l'exécution de l'instruction de déclenchement(ÉVÉNEMENT).
- Les déclencheurs **AFTER** exécutent l'action du déclencheur après l'exécution de l'instruction de déclenchement
- Il y a trois ÉVÉNEMENTS possibles auxquels un déclencheur peut être assigné : **INSÉRER**, **METTRE À JOUR** et **SUPPRIMER**.



01 – Maitriser le langage de programmation procédurale sous MySQL

Les déclencheurs (Trigger)

Exemple

```
17      delimiter //
18      CREATE TRIGGER tr_containte_age BEFORE INSERT
19      ON clients
20      FOR EACH ROW
21      BEGIN
22      DECLARE MinAge INT;
23      SET MinAge:=18;
24      IF NEW.date_naissance >(SELECT DATE_SUB(curdate(), interval MinAge year)) THEN
25      SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'La personne doit etre age de plus de  18.';
26      END IF;
27      END//
28      delimiter ;
29      -- tester le declencheur par une operation insert sur clients
30      INSERT INTO clients(nom,prenom,date_naissance,adresse)
31      Values("Slami","saida","2010-5-22","casa")
```

Output

Action Output

#	Time	Action	Message
2	11:11:06	INSERT INTO clients(nom,prenom,date_naissance,adresse) Values("Slami","saida","2010-5...	Error Code: 1644. La personne doit etre age de plus de 18.

01 – Maitriser le langage de programmation procédurale sous MySQL

Les déclencheurs (Trigger)

Avantages

Les déclencheurs contribuent :

- À renforcer l'intégrité des données
- À la détection des erreurs
- Aux tâches qui peuvent être exécutées automatiquement lorsque le déclencheur se déclenche plutôt que d'être planifiées.
- À auditer les changements de données, enregistrer les événements et aider à prévenir les transactions invalides.

01 – Maitriser le langage de programmation procédurale sous MySQL

Les déclencheurs (Trigger)

Restrictions avec les déclenchers

On ne peut avoir qu'un seul déclencheur par événement ; On ne peut avoir qu'un seul déclencheur BEFORE UPDATE sur une table donnée, mais on peut y exécuter plusieurs instructions ;

Les déclencheurs ne renvoient pas de valeurs ;

Ils ne peuvent pas utiliser l'instruction CALL et ils ne peuvent pas créer de tables ou de vues temporaires ;

Les déclencheurs peuvent entraîner des résultats incohérents en fonction de plusieurs facteurs, notamment le type de moteur de base de données (InnoDB, MyISAM...) utilisé avec la table à laquelle le déclencheur a été attribué.

OLD/NEW

- le mot-clé OLD est utilisé pour accéder aux données de ligne qui sont remplacées par la mise à jour.
- Le mot-clé NEW permet d'accéder aux données de ligne entrantes qui remplaceront l'ancienne ligne, en cas de succès.
- Selon le type de déclencheur créé, les lignes OLD et NEW peuvent ne pas être disponibles :
  - INSERT TRIGGER : Accès possible a New uniquement.
  - UPDATE TRIGGER : Accès aux deux pseudo-lignes NEW et OLD
  - DELETE TRIGGER : Accès uniquement a OLD pseudo-lignes, c'est-à-dire qu'il n'y a pas de ligne OLD sur un déclencheur INSERT et pas de ligne NEW sur un déclencheur DELETE.

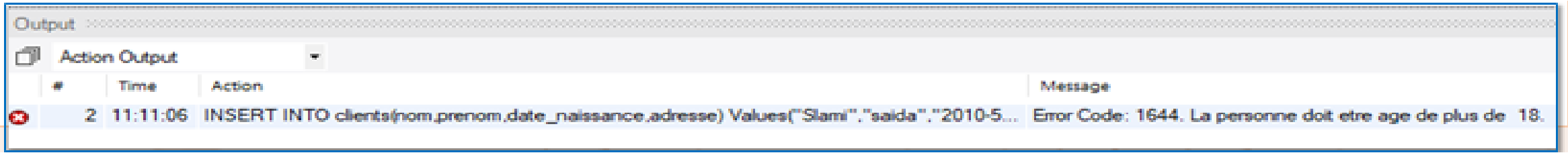
01 – Maitriser le langage de programmation procédurale sous MySQL

Les déclencheurs (Trigger)

Exemples : BEFORE INSERT

```
DELIMITER //

CREATE TRIGGER tr_contrainte_age BEFORE INSERT
FOR clients
BEGIN
        DECLARE MinAge INT ;
        SET MinAge := 18;
        IF NEW.date_naissance >(SELECT DATE_SUB (curdate(), interval MinAge year)) THEN
                SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'La personne doit etre agee de plus de 18 ans. ' ;
        END IF ;
END //
DELIMITER ;
- - tester le déclencheur par une opération insert sur la table clients
INSERT INTO clients (nom, prenom, date_naissance, adresse)
VALUES (''Slami'', ''Saida'',''2010-5-22'', ''casa'')
```



Exemples : AFTER INSERT

```
- - Initialisation de l age moyenne age
CREATE TABLE moyenne_age (moyenne double);
INSERT INTO moyenne_age SELECT (AVG (TIMESTAMPDIFF(YEAR, date_naissance, CURDATE())) FROM clients ;
SET SQL_SAFE_UPDATES = 0;


- - Creation du trigger
DROP TRIGGER IF EXISTS mise_ajour_moyenne_age ;
DELIMITER //
CREATE TRIGGER mise_ajour_moyenne_age AFTER INSERT
ON clients
FOR EACH ROW
UPDATE moyenne_age SET moyenne = age SELECT (AVG (TIMESTAMPDIFF(YEAR, date_naissance, CURDATE())) FROM clients ;
DELIMITER ;
- - déclencher le triger
INSERT INTO clients (nom, prenom, date_naissance, adresse)
VALUES (''Dadi'',''Hamza'',''2000-5-22'',''casa'');
- - afficher la nouvelle valeur de l'age moyen
SELECT * FROM moyenne_age;
```

Exemples : BEFORE UPDATE

```
DELIMITER //
CREATE TRIGGER tr_modification_age_controle BEFORE UPDATE
ON clients
FOR EACH ROW
BEGIN
        DECLARE MinAge INT ;
        SET MinAge := 18 ;
        IF NEW.date_naissace > (SELECT DATE_SUB(curdate(), interval MinAge year )) THEN
                SIGNAL SGLSTATE '50002' SET MESSAGE_TEXT = 'La personne doit etre agee de plus de 18 ans.';
        END IF;
END;
DELIMITER ;


- - tester le trigger
UPDATE Clients SET date_naissance = '2010-9-23'
WHERE id = 6;
UPDATE Clients SET date_naissance = '1999-9-23'
WHERE id = 7;
```

Exemples : AFTER UPDATE

- Ce déclencheur after_sales_update est automatiquement déclenché avant qu'un événement de mise à jour ne se produise pour chaque ligne de la table des ventes(sales).
- Si on met à jour la valeur dans la colonne de quantité à une nouvelle valeur, le déclencheur insère une nouvelle ligne pour consigner les modifications dans la table SalesChanges.

```
DELIMITER //

CREATE TRIGGER after_sales_update AFTER UPDATE
ON sales
FOR EACH ROW
BEGIN
        IF OLD.quantity <> NEW.quantity THEN
                INSERT INTO SalesChanges (salesId, beforeQuantity, afterQuantity)
                VALUES (OLD.id, OLD.quantity, NEW.quantity) ;
        END IF ;
END $$

DELIMITER ;
```

Exemples : BEFORE DELETE

Ce déclencheur permet avant de supprimer un salaire de le sauvegarder dans une table d'archivage SalaryArchives.

```
DELIMITER //

CREATE TRIGGER before_salaries_delete BEFORE DELETE
ON salaries
FOR EACH ROW
BEGIN
        INSERT INTO SalaryArchives (employeeNumber, validFrom, amount)
        VALUES (OLD.employeeNumber, OLD.validFrom, OLD.amount) ;
END $$

DELIMITER ;
```

Exemples : AFTER DELETE

Ce déclencheur permet après la suppression d'un salaire de mettre à jour le budget des salaires stockée dans la table SalaryBudgets.

```
CREATE TRIGGER after_salaries_delete AFTER DELETE
ON salaries
FOR EACH ROW
UPDATE SalaryBudgets
SET total = total - OLD.salary;
```

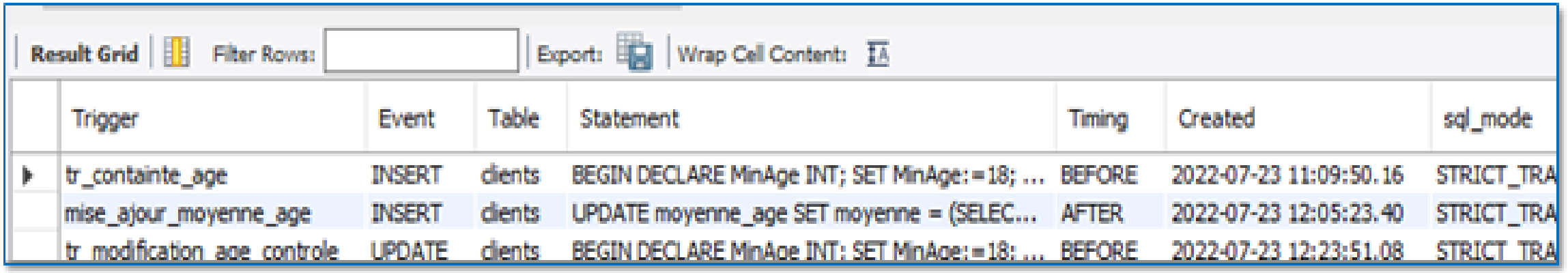
01 – Maitriser le langage de programmation procédurale sous MySQL

Les déclencheurs (Trigger)

LISTE des déclencheurs

Pour lister les déclencheur de la base de données, on utilise :

SHOW TRIGGERS;



Trigger	Event	Table	Statement	Timing	Created	sql_mode
tr_containte_age	INSERT	clients	BEGIN DECLARE MinAge INT; SET MinAge:=18; ...	BEFORE	2022-07-23 11:09:50.16	STRICT_TRA
mise_ajour_moyenne_age	INSERT	clients	UPDATE moyenne_age SET moyenne = (SELEC...	AFTER	2022-07-23 12:05:23.40	STRICT_TRA
tr_modification_age_controle	UPDATE	clients	BEGIN DECLARE MinAge INT; SET MinAge:=18; ...	BEFORE	2022-07-23 12:23:51.08	STRICT_TRA

Suppression d'un déclencheur

- Pour supprimer un trigger, on utilise :

  DROP TRIGGER [**IF EXISTS**] [nom_bd.nom_declencheur];

- **Remarque** : Si on supprime une table, MySQL supprimera automatiquement tous les déclencheurs associés à la table.

Modification d'un déclencheur

- Il n'existe pas une instruction pour modifier un trigger, on doit le supprimer et le recréer.
- **LOCK** permet aux sessions client d'acquérir explicitement des verrous de table dans le but de coopérer avec d'autres sessions pour accéder aux tables, ou d'empêcher d'autres sessions de modifier les tables pendant les périodes où une session nécessite un accès exclusif à celles-ci.
- **NB**: **MariaDB**, dans la version 10.1.4, a ajouté la prise en charge de **CREATE** OR **REPLACE TRIGGER** .

```
LOCK TABLES t1 WRITE ;
DROP TRIGGER t1_bi ;
DELIMITER $$
CREATE TRIGGER t1_bi BEFORE INSERT
ON t1
FOR EACH ROW
BEGIN
        …
END $$
DELIMITER ;
UNLOCK TABLES ;
```

CHAPITRE 2

Optimiser une base de données MySQL

Ce que vous allez apprendre dans ce chapitre :

- Optimiser les requêtes SQL ;
- Optimiser la structure de la base de données ;
- Optimiser la configuration de serveur MySQL.

CHAPITRE 2

Optimiser une base de données MySQL

1. **Optimiser les requêtes SQL ;**
2. Optimiser la structure de la base de données ;
3. Optimiser la configuration de serveur MySQL.