

COMPTE RENDU

Bases de Données Avancées - Lab No. 5
3e année Cybersécurité - École Supérieure d'Informatique et du
Numérique (ESIN)
Collège d'Ingénierie & d'Architecture (CIA)

Étudiant : HATHOUTI Mohammed Taha
Filière : Cybersecurité
Année : 2025/2026
Enseignants : Mme.ELHAJI & M.HJJAMI
Date : 12 novembre 2025

Table des matières

1	Exercice 1 : BEFORE INSERT Trigger	2
2	Exercice 2 : AFTER UPDATE Trigger	3

1 Exercice 1 : BEFORE INSERT Trigger

Vous souhaitez définir automatiquement la date d'embauche de tout nouvel employé à la date actuelle si elle n'est pas fournie.

Tâches :

- Créer un trigger sur la table *EMPLOYEES*;
- Le trigger doit être un trigger *BEFORE INSERT*;
- Si *HIRE_DATE* est NULL, le définir à SYSDATE;
- Insérer un nouvel employé sans spécifier la date d'embauche et vérifier que le trigger la définit;

Code PL/SQL :

```
1 SQL> CREATE OR REPLACE TRIGGER date_embauche
2   BEFORE INSERT
3   ON employees
4   FOR EACH ROW
5   BEGIN
6     IF :NEW.HIRE_DATE IS NULL THEN
7       :NEW.HIRE_DATE := SYSDATE;
8     END IF;
9   END;
10* /
11
12 Elément Trigger DATE_EMBAUCHE compilé
13
14 Aucune erreur.
```

Résultat :

```
SQL> INSERT INTO EMPLOYEES (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL,
  PHONE_NUMBER, JOB_ID, SALARY, DEPARTMENT_ID)
  2* VALUES (999, 'Mohammed Taha', 'HATHOUTI', 'hathoutm', '0639860991',
    'IT_PROG', 500000, 60);

1 ligne inséré.

SQL> SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, HIRE_DATE
  2 FROM EMPLOYEES
  3* WHERE EMPLOYEE_ID = 999;

EMPLOYEE_ID FIRST_NAME      LAST_NAME HIRE_DATE
-----
          999 Mohammed Taha HATHOUTI  12/11/25
```

2 Exercice 2 : AFTER UPDATE Trigger

Vous souhaitez garder une trace des changements de salaire dans une table d'audit séparée *SALARY_AUDIT*.

Tâches :

- Créer une table d'audit *SALARY_AUDIT* avec les colonnes : *EMPLOYEE_ID*, *OLD_SALARY*, *NEW_SALARY*, *CHANGE_DATE* ;
- Créer un trigger *AFTER UPDATE* sur *EMPLOYEES* pour la colonne *SALARY* ;
- Lorsque le salaire d'un employé change, insérer un enregistrement dans *SALARY_AUDIT* avec l'ancien et le nouveau salaire ainsi que la date actuelle ;
- Mettre à jour le salaire d'un employé et vérifier que la table d'audit est remplie ;

Création de la table d'audit *SALARY_AUDIT* :

```
1 SQL> CREATE TABLE SALARY_AUDIT (  
2     EMPLOYEE_ID NUMBER(6),  
3     OLD_SALARY  NUMBER(8,2),  
4     NEW_SALARY  NUMBER(8,2),  
5     CHANGE_DATE DATE  
6* );  
7  
8 Table SALARY_AUDIT créé(e).
```

Code PL/SQL :

```
1 SQL> CREATE OR REPLACE TRIGGER trigger_audit_salaire  
2   AFTER UPDATE OF SALARY  
3   ON EMPLOYEES  
4   FOR EACH ROW  
5   BEGIN  
6     INSERT INTO SALARY_AUDIT (  
7       EMPLOYEE_ID, OLD_SALARY,  
8       NEW_SALARY, CHANGE_DATE  
9     ) VALUES (  
10      :OLD.EMPLOYEE_ID, :OLD.SALARY,  
11      :NEW.SALARY, SYSDATE);  
12   END;  
13  
14  
15 Elément Trigger TRIGGER_AUDIT_SALAIRE compilé  
16  
17 Aucune erreur.
```

Résultat :

```
SQL> SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY  
2 FROM EMPLOYEES  
3* WHERE EMPLOYEE_ID = 103;  
  
EMPLOYEE_ID FIRST_NAME LAST_NAME SALARY  
-----  
103 Alexander Hunold 18000
```

```
SQL> UPDATE EMPLOYEES
  2  SET SALARY = 20000
  3* WHERE EMPLOYEE_ID = 103;
```

1 ligne mis jour.

```
SQL> SELECT * FROM SALARY_AUDIT;
```

EMPLOYEE_ID	OLD_SALARY	NEW_SALARY	CHANGE_DATE
103	9000	18000	12/11/25
103	18000	20000	12/11/25