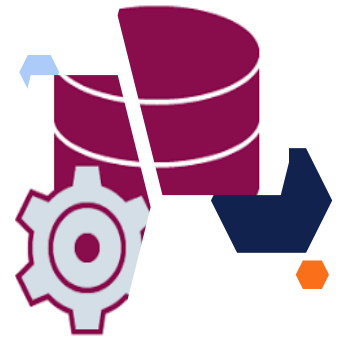




1

Plan du cours

1. INTRODUCTION
2. LES STRUCTURES DE CONTRÔLE
3. LES CURSEURS
4. LES EXCEPTIONS
5. LES SOUS-PROGRAMMES
6. LES DECLENCHEURS
7. LES PACKAGES



2

Les déclencheurs / Les triggers



Sonia LOULOU & Sameh CHAËBANE

24/04/2025

3

3

Les triggers : Définition



- Un **trigger** est un type spécial de procédure **stockée** qui est **implicitement (automatiquement)** exécuté (déclenché) suite à la production d'un **événement** déclencheur.
- Un événement peut être :
 - Une instruction du LMD exécutée sur une table : INSERT, UPDATE ou DELETE.
 - Une instruction du LDD : une instruction CREATE ou ALTER.
 - Un événement utilisateur tel que la connexion ou la déconnexion : LOGON ou LOGOFF.
 - ...

Sonia LOULOU & Sameh CHAËBANE

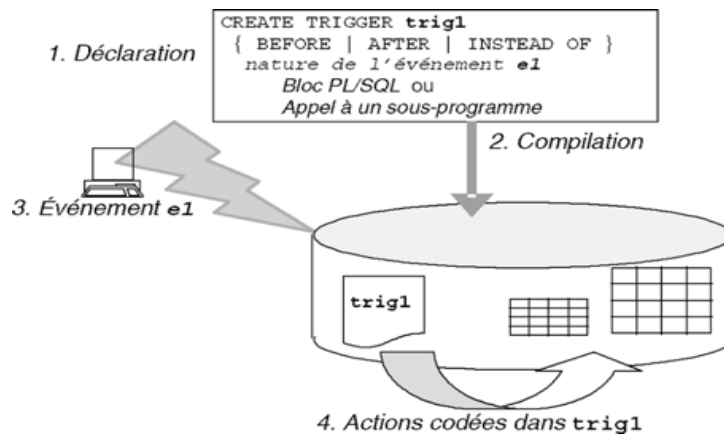
24/04/2025

4

4

Les triggers : Définition

- Un trigger est basé sur le principe ECA (Événement – [Condition -] Action).



Sonia LOULOU & Sameh CHAËBANE

24/04/2025

5

5

Les triggers : Avantages et Utilité

- Les triggers permettent de :
 - Appliquer des **règles** qui ne peuvent pas être établies à l'aide des **contraintes d'intégrité** du LDD (UNIQUE, NOT NULL, CHECK, etc.) tels que les multiplicités (cardinalités).
 - Générer automatiquement des valeurs pour des colonnes dérivées.
 - Alléger le développement d'applications via les traitements implicites.
 - ...

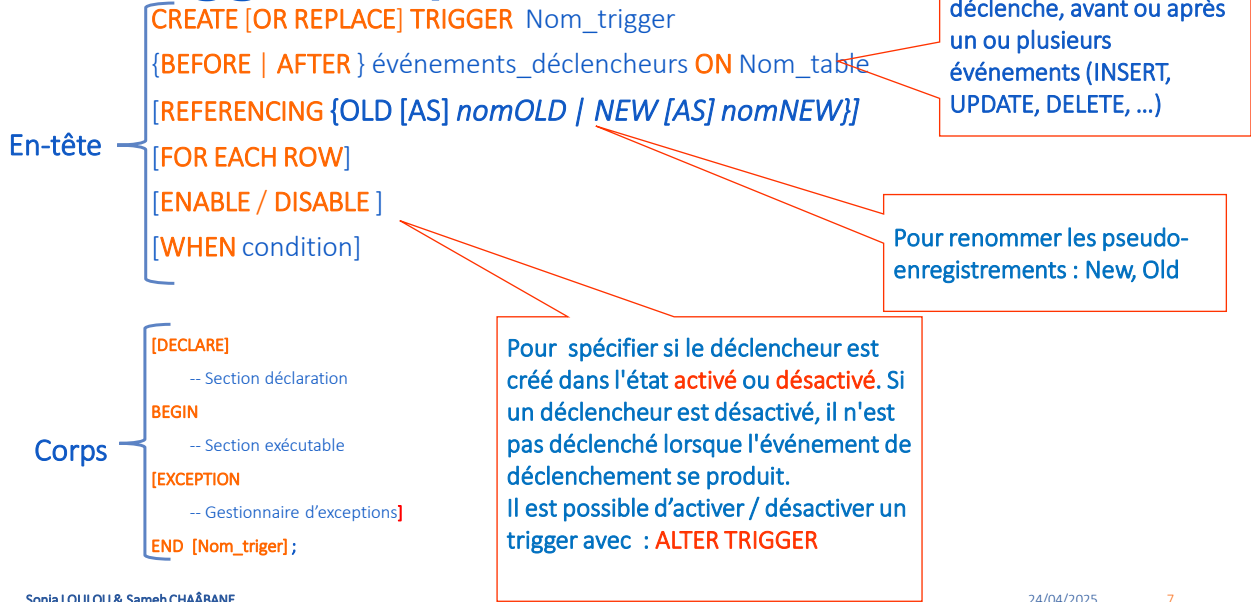
Sonia LOULOU & Sameh CHAËBANE

24/04/2025

6

6

Les triggers : Syntaxe



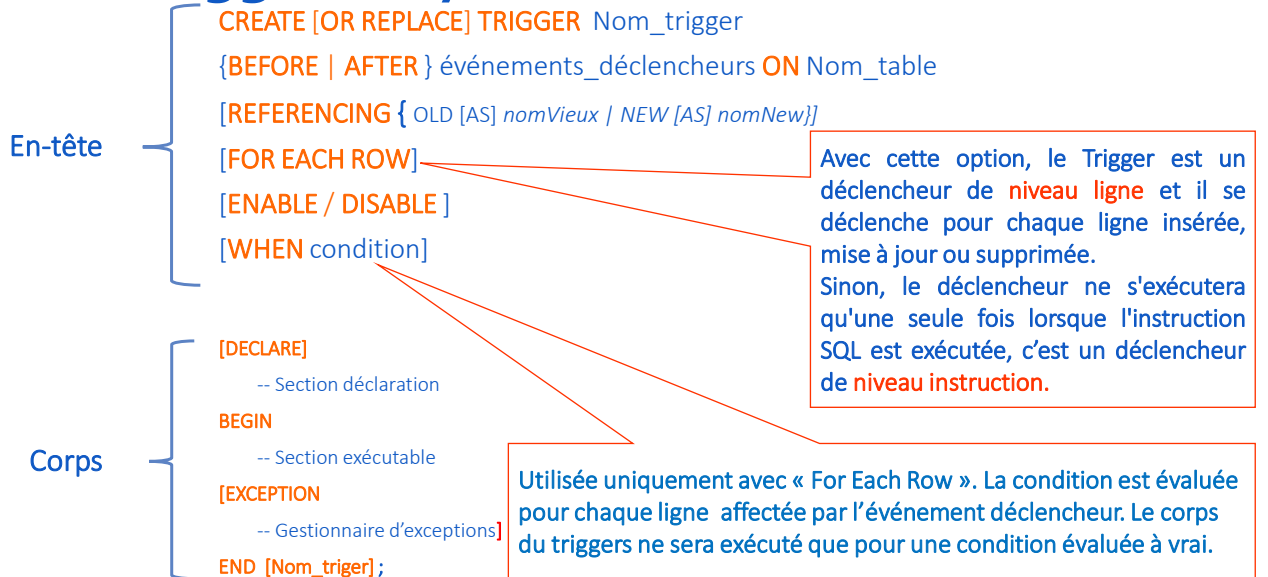
Sonia LOULOU & Sameh CHAËBANE

24/04/2025

7

7

Les triggers : Syntaxe



Sonia LOULOU & Sameh CHAËBANE

24/04/2025

8

8

Les triggers : Syntaxe



Elément	Valeurs possibles	Explication
Nom_Trigger	Choisi par le développeur	Identifiant du trigger.
Nom_Table	Nom d'une table	La table sur laquelle le trigger est défini. Si l'événement est exécuté sur cette table, le trigger sera déclenché.
Evènement	INSERT, UPDATE, DELETE	Evènement déclenchant l'exécution du trigger : commande LMD ou une combinaison utilisant OR
Séquencement (Timing)	BEFORE ou AFTER	Caractérise l'enchaînement du trigger par rapport à son événement. Avec BEFORE (AFTER), l'exécution du corps du déclencheur est réalisée avant (après) que l'évènement déclencheur ne sera exécuté.
Niveau de déclenchement	Présence ou non de FOR EACH ROW	Il existe deux niveaux de déclenchement : <ul style="list-style-type: none">- niveau ligne (row level trigger), et- niveau instruction (statement level trigger).

Sonia LOULOU & Sameh CHAËBANE

24/04/2025

9

9

Les triggers : Types



- Trigger **niveau ligne** : utilisation de l'option « For Each Row »
 - Exécuté pour chaque ligne concernée par l'événement déclencheur : insert, update ou delete. Si aucune ligne affectée par l'événement, le trigger ne sera pas déclenché.
- Trigger **niveau instruction** :
 - Exécuté **une seule fois** lors du déclenchement de l'événement même si ce dernier affecte plusieurs lignes à la fois.

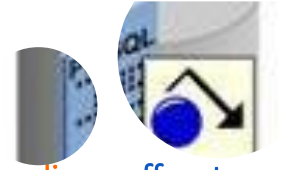
Sonia LOULOU & Sameh CHAËBANE

24/04/2025

10

10

Les triggers : Niveau ligne



- Contrairement aux triggers niveau instruction, les triggers **niveau ligne** offrent la possibilité d'accéder aux données des lignes en cours de manipulation par le biais des 2 pseudo-enregistrements **OLD** et **NEW** de **même structure** que la **table** sur laquelle le trigger est défini et contenant respectivement les **anciennes** et les **nouvelles** valeurs du **n-uplet affecté** par l'événement.
- Les champs des pseudo-enregistrements sont utilisés comme des variables dans les commandes SQL et PL/SQL. Ils sont référencés par :



:OLD.NomColonne

:NEW.NomColonne

- La clause WHEN du trigger peut utiliser les pseudo-enregistrements **old** et **new** sans les préfixer par les « : ».

Sonia LOULOU & Sameh CHAËBANE

24/04/2025

11

11

Les triggers : Niveau ligne



- Si l'instruction de déclenchement du trigger est **INSERT**, seul le pseudo-enregistrement (**NEW**) a un sens. Les colonnes de **OLD** sont à **null**.
- Si l'instruction de déclenchement du trigger est **DELETE**, seul le pseudo-enregistrement (**OLD**) a un sens. Les colonnes de **NEW** sont à **null**.
- Si l'instruction de déclenchement du trigger est **UPDATE**, les deux pseudo-enregistrements (**OLD ET NEW**) ont un sens.

Sonia LOULOU & Sameh CHAËBANE

24/04/2025

12

12

Les triggers : Exemples



- Considérons le nouvel attribut **NbEnf** dans la table **maman**, désignant le nombre d'enfants de chacune des mamans de la table :

`Alter Table Maman Add (NbEnf number(1));`

- Programmer le trigger **TrigInsEnf** qui surveille les insertions effectuées sur la table **Enfant** et incrémente de 1 la colonne **NbEnf** pour la Maman concernée.
- Programmer le trigger **TrigSupEnf** qui surveille les suppressions de la table **Enfant** et décrémente la colonne **NbEnf** pour la maman concernée.

Les triggers : Exemples



Create or Replace TRIGGER **TrigInsEnf**

After INSERT ON **Enfant**

FOR EACH ROW

Begin

Update Maman set NbEnf = NbEnf + 1 where CINMAM = :**New.CinMam**;

End;



Pour exécuter le trigger, il suffit d'insérer un nouvel enfant :

Insert into Enfant

Values (2, '02030408', 'Mohamed', 'TOUNSI', Sysdate, 3.5, 57) ;

Les triggers : Exemples



Create or Replace TRIGGER **TrigSupEnf**
After DELETE ON Enfant
FOR EACH ROW

Begin

Update Maman set NbEnf = NbEnf - 1 Where CINMAM=:Old.CinMam;

End;



Pour exécuter le trigger, il suffit de supprimer un ou plusieurs enfants :
Delete from Enfant where NumEnf=1 and CinMam= '02030408';

Les triggers : Exemples



- Programmer le trigger **TrigDecEnf** qui permet d'insérer dans une table DecEnf (Ayant la même structure que la table Enfant) l'enfant supprimé.

Create or replace Trigger TrigDecEnf

After DELETE ON Enfant

For Each Row

Begin

Insert into DecEnf Values (:old.NumEnf, :old.CinMam, :old.NomEnf,
:old.PrenEnf, :old.DateNaissEnf, :old.PoidsNaiss , :old.TailleNaiss);

End;

Les triggers : Exemples



- Programmer le trigger **TrigUpdVacc** qui permet d'insérer dans une table **HistVaccEnf** toute modification des périodes de vaccination.
HistVaccEnf(CodeVacc, Date Maj, AncAgeVacc , AncUniteAge, NovAgeVacc , NovUniteAge)

Les triggers : Exemples



Create or Replace trigger **TrigUpdVacc**
After Update of AGEVACC, UNITEAGE On Vaccin
For Each Row

Begin

insert into HistVaccEnf values (:old.CodeVacc , sysdate, :old.AgeVacc ,
:old.UniteAge , :new.AgeVacc , :new.UniteAge);

end;



Pour exécuter le trigger, il suffit de lancer la commande de modification des données d'un vaccin :

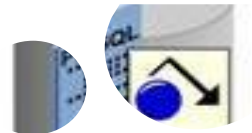
Update **Vaccin** set AgeVacc = 3 where CodeVacc= 'VHB-0';

Les triggers : Exceptions



- En cas d'une exception levée lors de l'exécution d'un trigger :
 - Si **présence** d'un gestionnaire approprié ou générique dans le trigger :
 - poursuite avec l'exécution du traitement de l'exception
 - Si **absence** d'un gestionnaire approprié ou générique dans le trigger :
 - Si besoin d'annulation de l'instruction ayant déclenché le trigger, il faut générer une erreur applicative utilisant la procédure stockée **Raise_Application_Error (Code_Erreur, 'Message personnalisé de l'erreur')**
Code_Erreur : -20001 .. -20999
- ➔ Annulation des opérations déjà effectuées dans le déclencheur.

Les triggers : Exceptions



- Programmer le trigger **JourneeVacc** qui interdit toute réalisation de vaccin le dimanche.

Create or Replace trigger JourneeVacc

Before INSERT Or UPDATE ON EffectuerVaccin

For Each Row

Begin

if TO_CHAR(:New.DateRealVacc, 'DAY') = 'DIMANCHE' Then

RAISE_APPLICATION_ERROR(-20100, 'Vérifiez date : pas de vaccination le Dimanche');

end if;

End;

Les triggers : Exceptions



- Le même trigger **JourneeVacc** peut être écrit autrement (en utilisant l'option When) :

Create or Replace trigger JourneeVacc

Before INSERT Or UPDATE ON EffectuerVaccin

For Each Row

When (TO_CHAR(New.DateRealVacc, 'DAY') = 'DIMANCHE')

Begin

RAISE_APPLICATION_ERROR(-20100,'Vérifiez date : pas de vaccination le Dimanche');

End;

Les triggers : Exceptions



Pour tester le trigger, il suffit d'exécuter l'une des commandes :

insert into EffectuerVaccin values (5, '02030408', 'VPI', '25/04/2021'); --Ou bien

Update EffectuerVaccin set DateRealVacc='18/04/2021'

where CinMam='02030408' and NumEnf=5;



```
insert into EffectuerVaccin values ('VPI',1,2030408,'25/04/2021')
Rapport d'erreur -
Erreur SQL : ORA-20100: Vérifiez date : pas de vaccination le Dimanche
ORA-06512: at "BDVACCIN.JOURNEEVACC", line 3
ORA-04088: error during execution of trigger 'BDVACCIN.JOURNEEVACC'
```

Les triggers : Regroupement d'évènements



- **Regroupements d'évènements** : Des événements (INSERT, UPDATE ou DELETE) peuvent être regroupés au sein d'un même déclencheur s'ils sont de même type (BEFORE ou AFTER).
- Un seul déclencheur est à coder et des instructions dans le corps du déclencheur permettent de retrouver la nature de l'évènement déclencheur :
 - IF (INSERTING) THEN ... bloc à exécuter dans le cas d'une insertion.
 - IF (UPDATING ('NomColonne')) THEN ... bloc à exécuter dans le cas d'une modification d'une colonne.
 - IF (DELETING) THEN ... bloc à exécuter en cas d'une suppression.

Les triggers : Regroupement d'évènements



- Programmer le déclencheur **TrigInsSupEnf** qui surveille :
 - les insertions effectuées sur la table **Enfant** et incrémente de 1 la colonne **NbEnf** pour la Maman concernée.
 - les suppressions de la table **Enfant** et décrémente de 1 la colonne **NbEnf** pour la Maman concernée.

Les triggers : Regroupement d'événements



Create or Replace trigger TrigInsSupEnf

After **INSERT** or **DELETE** ON Enfant

For Each Row

Begin

If **inserting** Then

 update Maman set NbEnf=NbEnf+1 where CINMAM=:New.CinMam;

Else -- **deleting**

 update Maman set NbEnf=NbEnf-1 where CINMAM=:Old.CinMam;

End if;

End;

Sonia LOULOU & Sameh CHAËBANE

24/04/2025

25

25

Gestion des triggers



- Un trigger ne doit pas **MAJ** la table sur laquelle l'évènement déclencheur est défini : risque d'incohérence de données.
- Un trigger de timing « After » ne doit pas manipuler la table sur laquelle l'évènement déclencheur est défini : risque d'incohérence de données.
- Suppression d'un trigger :
 DROP TRIGGER NomTrigger ;
- Activation / Désactivation d'un trigger : il est préférable de désactiver un trigger que de le supprimer en vue de le réactiver ultérieurement.
 ALTER TRIGGER NomTrigger {**ENABLE** | **DISABLE**} ;
- Un trigger peut provoquer le déclenchement d'un autre trigger (triggers en cascade).

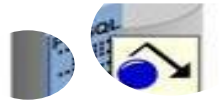
Sonia LOULOU & Sameh CHAËBANE

24/04/2025

26

26

Gestion des triggers



- Pour une table nous pouvons définir plusieurs déclencheurs en combinant :
 - Séquencement (Before ou After)
 - Évènement (Insert, Update ou Delete)
 - Niveau d'exécution (instruction ou ligne)
 - Maximum $2 \times 3 \times 2 = 12$ types de déclencheurs par table.
- Il est alors nécessaire de savoir l'ordre de déclenchement des triggers pour un évènement sur une table .

Gestion des triggers



- Ordre de déclenchement des triggers LMD :
 - Exécutez les triggers de niveau instruction « Before », s'ils existent.
 - Pour chaque ligne affectée par l'instruction :
 - Exécutez les triggers de niveau ligne « Before », s'ils existent.
 - Exécutez l'instruction elle-même.
 - Exécutez les triggers de niveau ligne « After », s'ils existent.
 - Exécutez les triggers de niveau instruction « After », s'ils existent.