



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**

Membre de 
HONORIS UNITED UNIVERSITIES

École Marocaine des Sciences de l'Ingénieur

Filière: IIR

Les Bases de Données

Prof. Zakaria KHATAR

Les exceptions en PL/SQL

La Gestion des Exceptions

1) Définition des exceptions :

En PL/SQL, les exceptions sont des événements qui surviennent pendant l'exécution d'un bloc de programme et qui perturbent son fonctionnement normal. Elles sont utilisées pour gérer des erreurs ou des comportements inattendus dans les programmes.

2) Types d'exceptions :

- **Exceptions prédéfinies** : Ce sont des exceptions fournies par Oracle, **chacune associée à un code d'erreur spécifique**. Par exemple, **NO_DATA_FOUND** est déclenchée lorsqu'une requête ne retourne pas de résultats, et **TOO_MANY_ROWS** est déclenchée lorsqu'une requête retourne plus d'une ligne alors qu'une seule était attendue.
- **Exceptions non prédéfinies** : Les utilisateurs **peuvent déclarer leurs propres exceptions** dans Oracle. Ces exceptions **sont levées manuellement** dans le code à l'aide de la commande **RAISE**.

3) Exceptions prédéfinies par Oracle :

Ces exceptions sont déjà définies par Oracle., pour les gérer on utilise la syntaxe suivante :

BEGIN

Instructions de traitement du bloc PL/SQL

....

EXCEPTION

WHEN nom_exception **THEN** traitement de l'exception;
... etc

END;

la section dédiée
au traitement des
exceptions.

3) Exceptions prédéfinies par Oracle :

Voici la liste courantes des exceptions prédéfinies par Oracle, Ces exceptions se déclenchent d'une manière **automatique**.

Exception	Description
NO_DATA_FOUND	Déclenchée lorsqu'une requête SELECT ... INTO ne trouve pas de lignes correspondantes.
TOO_MANY_ROWS	Déclenchée lorsqu'une requête SELECT INTO trouve plus d'une ligne correspondante.
ZERO_DIVIDE	Déclenchée lors d'une tentative de division par zéro.
DUP_VAL_ON_INDEX	Déclenchée lorsqu'une tentative d'insertion ou de mise à jour viole une contrainte d'unicité d'index.
VALUE_ERROR	Déclenchée lorsqu'une opération de conversion de type, telle que l'assignation d'une chaîne de caractères à une variable numérique, échoue.
CURSOR_ALREADY_OPEN	Déclenchée lorsqu'une tentative est faite pour ouvrir un curseur qui est déjà ouvert.

3) Exceptions prédéfinies par Oracle :

Exemple :

DECLARE

v_last_name employees.last_name%type;

v_first_name employees.first_name%type;

BEGIN

SELECT last_name, first_name **INTO** v_last_name, v_first_name

FROM employees **WHERE** employee_id = 30;

DBMS_OUTPUT.PUT_LINE(v_last_name || ' - ' || v_first_name);

EXCEPTION

WHEN NO_DATA_FOUND THEN

DBMS_OUTPUT.PUT_LINE('Le nom d'employé demandé n'existe pas');

END;

3) Exceptions prédéfinies par Oracle :

Quand une exception se déclenche, on peut identifier le code et le message de l'erreur, en utilisant les deux fonctions **SQLCODE** et **SQLERRM**.

❖ **SQLCODE** : retourne le numéro de l'erreur.

❖ **SQLERRM** : retourne le message associé à l'erreur.

Exemple de quelques codes d'erreur avec les messages associés

SQLCODE	Message SQLERRM
-1	ORA-00001: unique constraint (%s.%s) violated
-904	ORA-00904: "%s": invalid identifier
-918	ORA-00918: column ambiguously defined
-936	ORA-00936: missing expression
-1400	ORA-01400: cannot insert NULL into (%s)
-2290	ORA-02290: check constraint (%s.%s) violated

3) Exceptions prédéfinies par Oracle :

Exemple :

DECLARE

v_last_name employees.last_name%type;

v_first_name employees.first_name%type;

BEGIN

SELECT last_name, first_name **INTO** v_last_name, v_first_name
FROM employees **WHERE** employee_id = 30;

DBMS_OUTPUT.PUT_LINE(v_last_name || ' - ' || v_first_name);

EXCEPTION

WHEN NO_DATA_FOUND THEN

DBMS_OUTPUT.PUT_LINE('Le nom d"employé n"existe pas');

DBMS_OUTPUT.PUT_LINE('Le code d"erreur : ' || **SQLCODE**);

DBMS_OUTPUT.PUT_LINE('Le nom de l"erreur: ' || **SQLERRM**);

END;

4) Exceptions non prédéfinies :

Les Exceptions non prédéfinies sont des exceptions que les utilisateurs peuvent déclarer et gérer dans Oracle. Elles sont levées manuellement dans le code à l'aide de la commande **RAISE**.

Pour déclarer une exception on utilise la syntaxe dans le diapo suivant :

4) Exceptions non prédéfinies :

Syntaxe :

```
DECLARE
    nom_erreur EXCEPTION;
    ...
BEGIN
    ...
    IF condition THEN
        RAISE nom_erreur;      -- On déclenche l'erreur
    ...
    EXCEPTION
        WHEN nom_erreur THEN traitement de l'erreur;
    ... etc
END;
```

4) Exceptions non prédéfinies :

Exemple 1 : Exception pour une vérification d'âge

DECLARE

too_young **EXCEPTION**;

v_age NUMBER := &v_age;

BEGIN

IF v_age < 18 **THEN**

RAISE too_young; *-- Lève l'exception si la condition est vraie*

END IF;

DBMS_OUTPUT.PUT_LINE('Age is OK.');

-- Gestion de l'exception levée

EXCEPTION

WHEN too_young **THEN**

DBMS_OUTPUT.PUT_LINE('The person is too young.');

END;

4) Exceptions non prédéfinies :

Comme les curseurs, on peut exploiter Les variables de condition suivantes :

Variable	Description
SQL%ROWCOUNT	Nombre de lignes affectées par la dernière opération.
SQL%ISOPEN	Indique si un curseur est ouvert ou fermé.
SQL%FOUND	Indique si la dernière opération a affecté au moins une ligne.
SQL%NOTFOUND	Indique si la dernière opération de sélection de données (SELECT... INTO) n'a retourné aucune ligne

4) Exceptions non prédéfinies :

Exemple d'utilisation:

La variable de condition **SQL%FOUND** :

Cette variable est un booléen (**vrai ou faux**) qui indique si la dernière opération effectuée par une commande **SELECT INTO**, **FETCH**, ou **INSERT**, **UPDATE**, **DELETE** a été un succès.

La variable est mise à jour chaque fois qu'une de ces commandes est exécutée. Elle est l'opposé de **SQL%NOTFOUND**.

Exemple 1 : **BEGIN**

```
-- code qui exécute une commande SELECT... INTO  
IF SQL%NOTFOUND THEN  
    RAISE v_variable_exception;  
END IF;  
END;
```

4) Exceptions non prédéfinies :

Exemple 2 :

```
DECLARE
    v_except EXCEPTION;
BEGIN
    UPDATE employess SET salary = 4000 WHERE employee_id = 150;
    IF SQL%NOTFOUND THEN
        RAISE v_except;
    ENDIF;
    COMMIT;
EXCEPTION
    WHEN v_except THEN
        DBMS_OUTPUT.PUT_LINE('L'id de l'employé n'existe pas ...');
        DBMS_OUTPUT.PUT_LINE('Le no de l'erreur : ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('Le nom de l'erreur: ' || SQLERRM);
END;
```

4) Exceptions non prédéfinies :

Exemple 2 :

```
DECLARE
    v_except EXCEPTION;
BEGIN
    UPDATE employess SET salary = 4000 WHERE employee_id = 150;
    IF SQL%NOTFOUND THEN
        RAISE v_except;
    ENDIF;
    COMMIT;
EXCEPTION
    WHEN v_except THEN
        DBMS_OUTPUT.PUT_LINE('L'id de l'employé n'existe pas ...');
        DBMS_OUTPUT.PUT_LINE('Le no de l'erreur : ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('Le nom de l'erreur: ' || SQLERRM);
END;
```