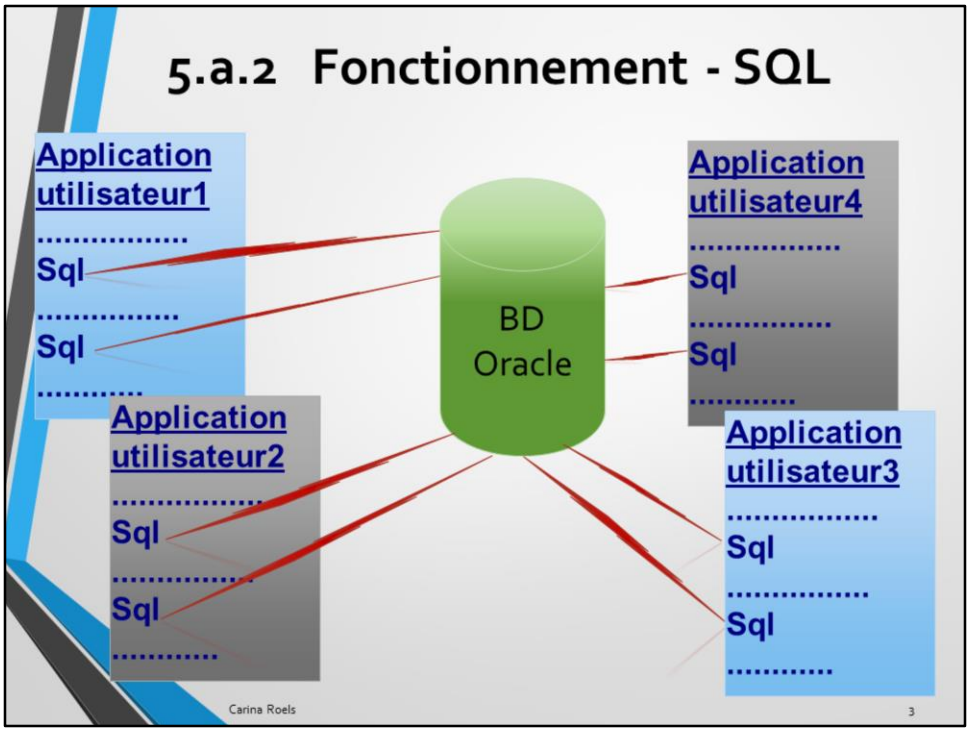
ORAE505
Conception et administration de B.D.

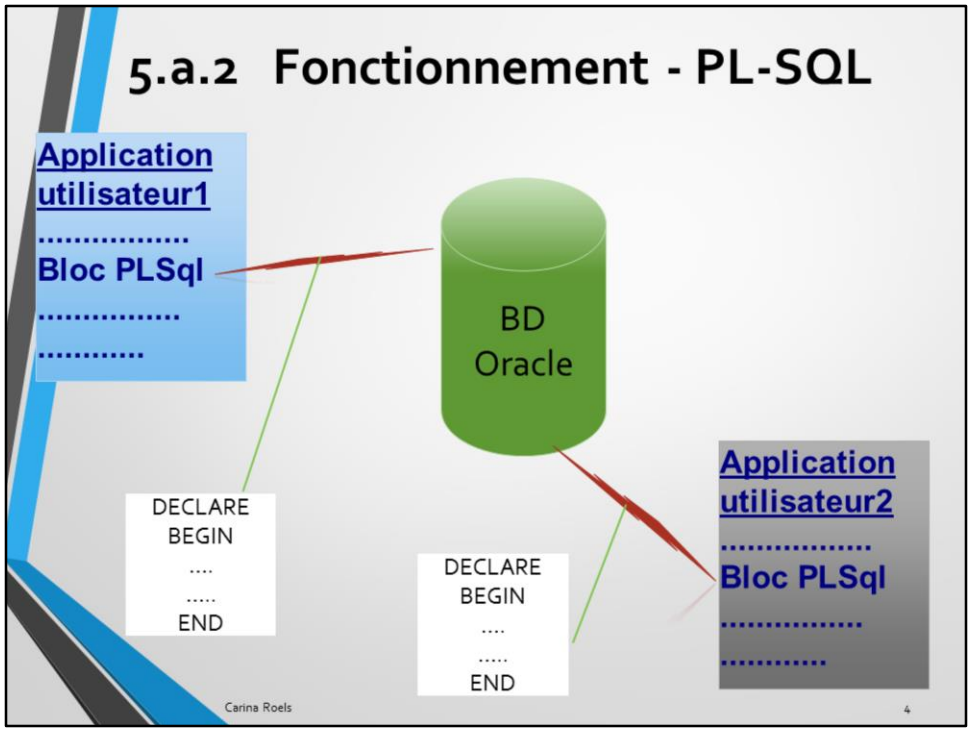
5. Le langage PL/SQL
a. Les bases

Carina Roels

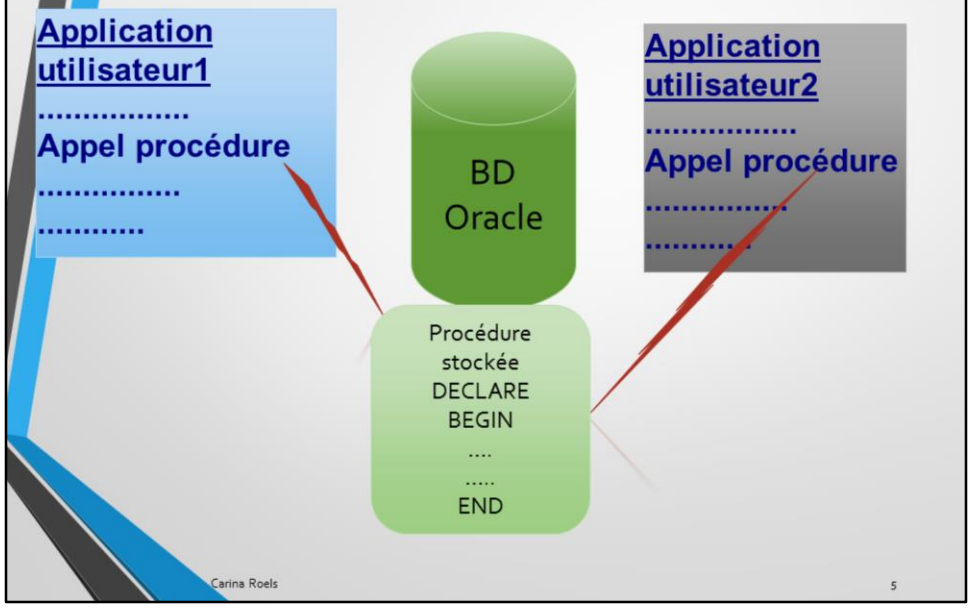
5.a.1 Généralités

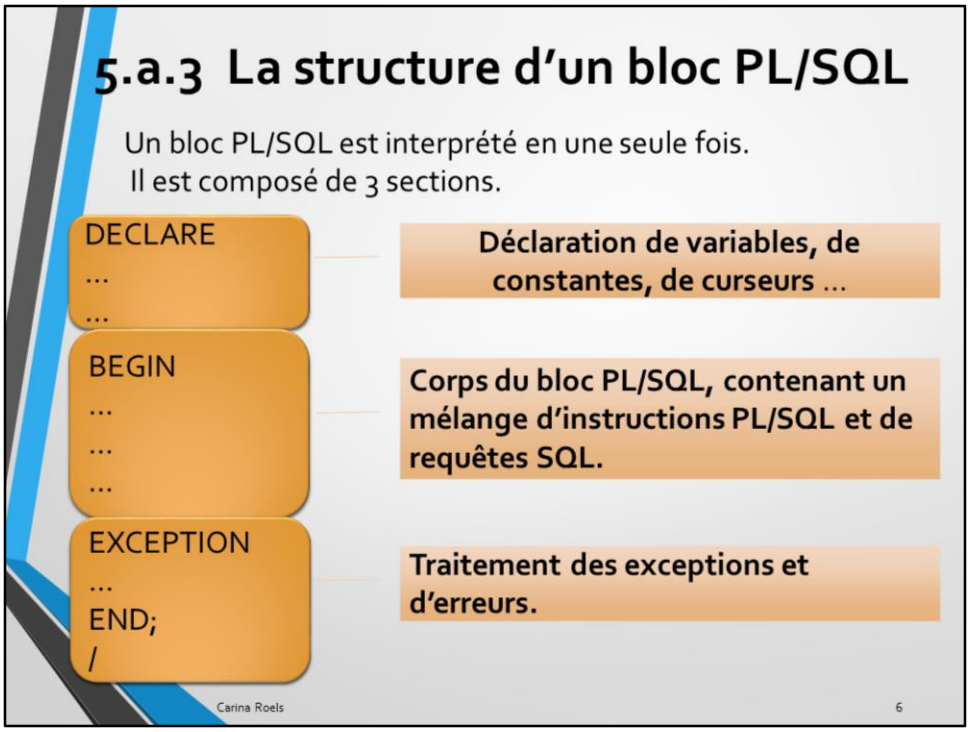
SQL Langage NON procédural	PL-SQL Langage procédural
SELECT	Déclaration de variables
INSERT	Traitements conditionnels
UPDATE	Traitements répétitifs
DELETE	Gestion de curseurs
COMMIT	Gestion d'exceptions
ROLLBACK	Intégration d'instructions SQL
...	


5.a.2 Fonctionnement - SQL
Application utilisateur1
.....
Sql
.....
Sql
.....
Application utilisateur2
.....
Sql
.....
Sql
.....
BD Oracle
Application utilisateur4
.....
Sql
.....
Sql
.....
Application utilisateur3
.....
Sql
.....
Sql
.....
Carina Roels
3


5.a.2 Fonctionnement - PL-SQL
Application utilisateur1
.....
Bloc PLSql
.....
.....
BD Oracle
Application utilisateur2
.....
Bloc PLSql
.....
.....
DECLARE
BEGIN
....
.....
END
DECLARE
BEGIN
....
.....
END
Carina Roels
4

5.a.2 Fonctionnement - objets stockés





Les sections DECLARE et EXCEPTION sont facultatives.

Chaque instruction doit se terminer par un **;**

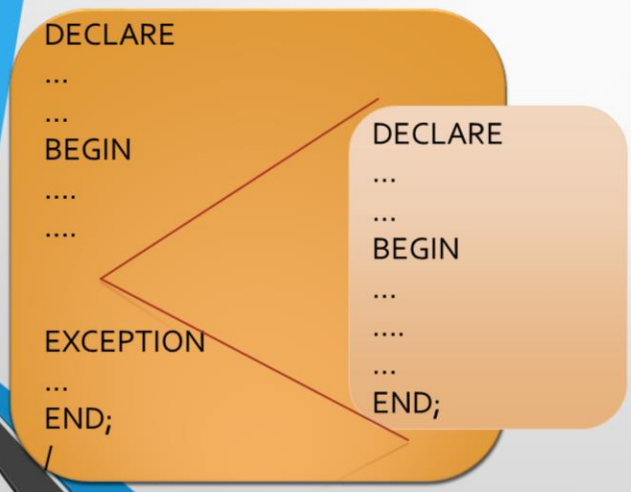
Il est possible d'introduire des commentaires dans un bloc PL/SQL

-- sur une ligne
/* sur plusieurs lignes ***/**

Le bloc PL/SQL est exécuté par un / se trouvant à fin du code (en première position d'une ligne).

5.a.3 La structure d'un bloc PL/SQL (suite)

Il est possible d'imbriquer les blocs PL/SQL.
Cela sera particulièrement utile lors de la gestion des exceptions.



5.a.4 La déclaration de variables

Les variables locales au bloc PL/SQL

- variables de type ORACLE (char, varchar, number, date ...)
- variables de type BOOLEAN
- variables faisant référence au dictionnaire de données.

Les variables externes au bloc PL/SQL

les champs d'écran (SQL*Forms)
les variables en lagage hôte (Pro*) — préfixées par **:**

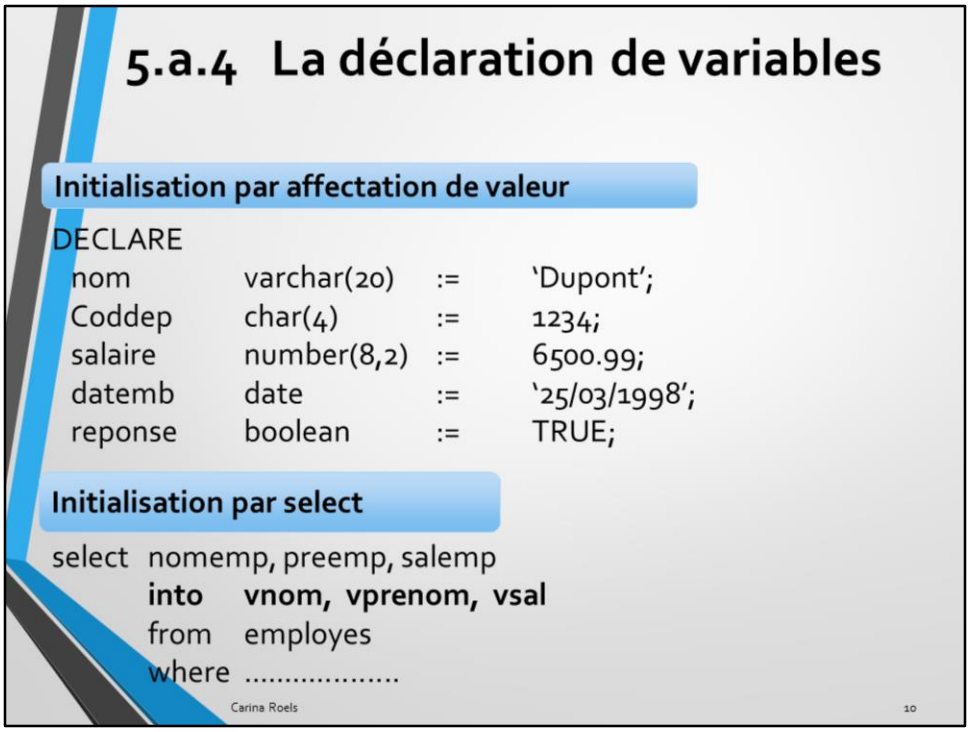
les variables SQL*Plus — préfixées par **&**

5.a.4 La déclaration de variables

```
DECLARE
  nom          varchar(20);
  coddep       char(4);
  salaire      number(8,2);
  datemb       date;
  reponse      boolean;

  nomemp       employe.nomemp%TYPE;
  salemp       employe.salemp%TYPE;

  vdept        dept%ROWTYPE ;
```

Référence au dictionnaire de donmême format que la colonne nomemp dans la table employenées: nomemp prendra le.

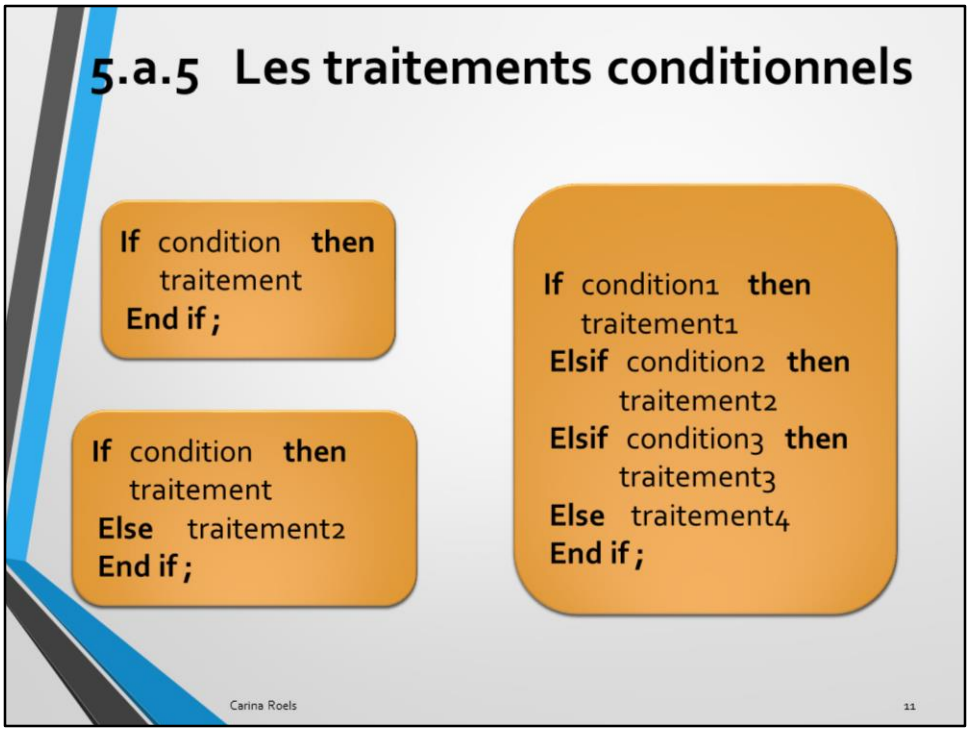
Référence au dictionnaire de données: vdept sera une structure qui comportera des variables ayant le même nom et le même format que les colonnes de la table dept.



La clause **INTO** est obligatoire !

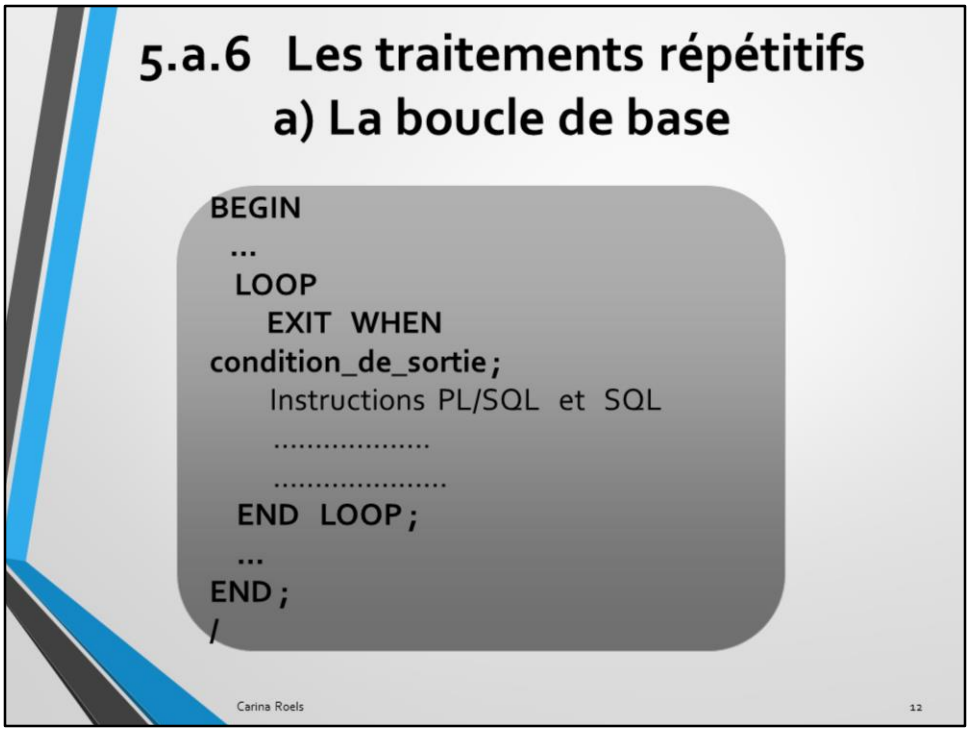
Le select doit ramener **1 seule** ligne de résultat

(si non, il faut utiliser un curseur, cf. chapitre suivant).



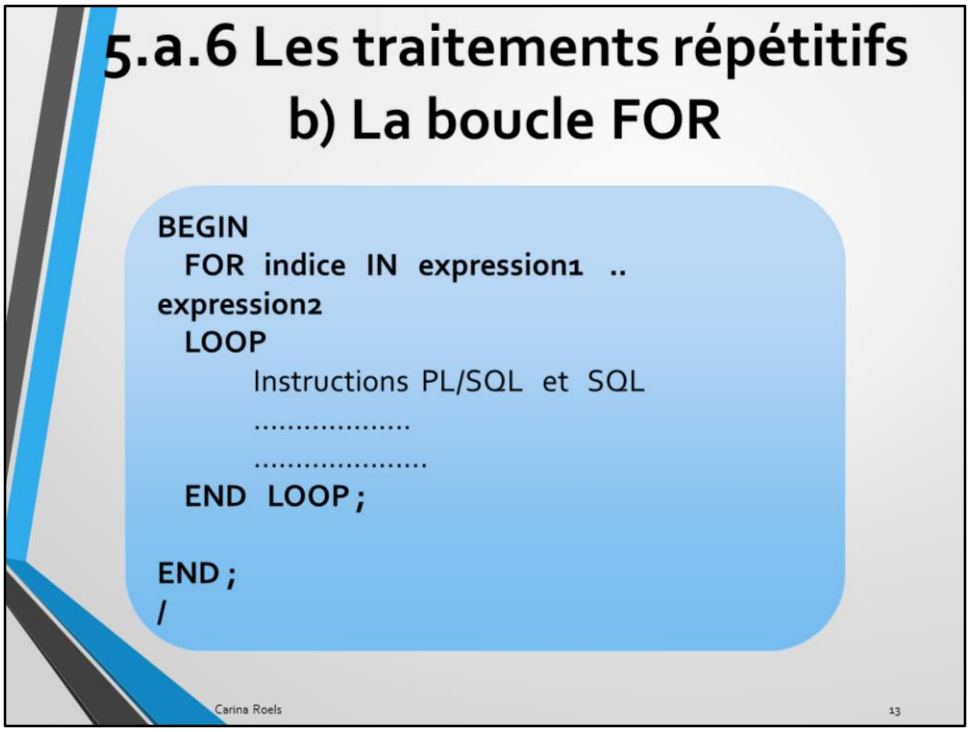
Les opérateurs utilisés dans les conditions d'un traitement conditionnel sont les mêmes que dans les requêtes SQL :

=	**<**	**>**	**!=**	**>=**	**<=**
IS NULL	**IS NOT NULL**	**BETWEEN**			
LIKE	**AND**		**OR**		
. . .					



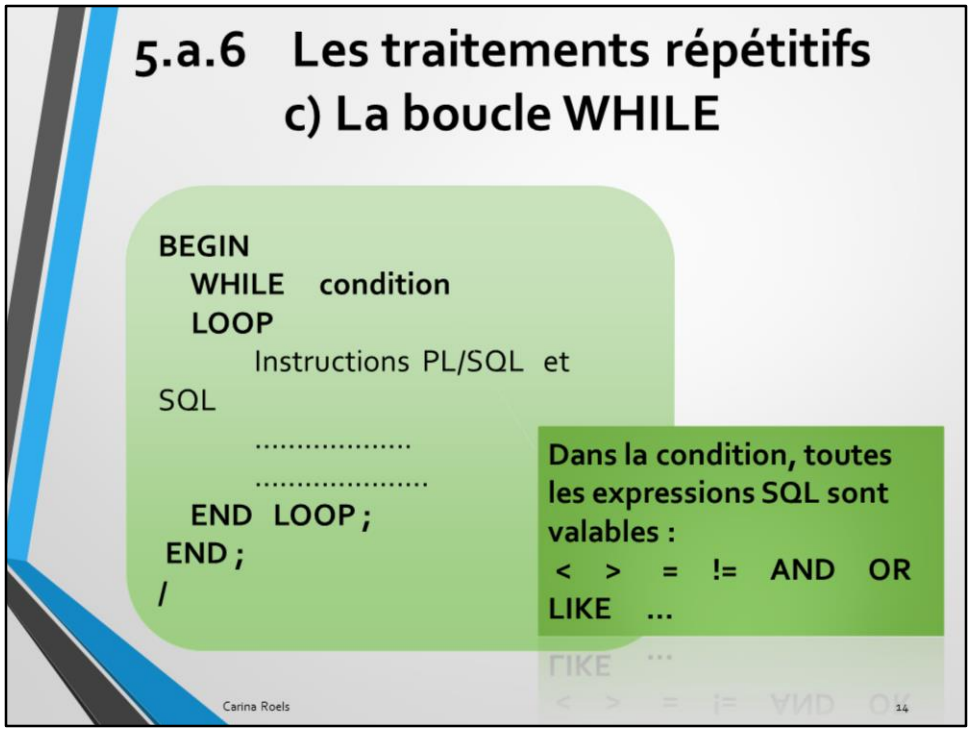
Exemple : Afficher la table de multiplication du chiffre 5.

```
DECLARE
   I            number(2)     :=  1 ;
   Chiffre      number(1)     :=  5 ;
   Result       number(2) ;

BEGIN

   LOOP
      Result  :=  Chiffre  *  I ;
      INSERT  INTO  RESULTAT ( chaine )
      VALUES  ( I  ||  ' x  '  ||  chiffre  ||  ' = '  ||  result  ) ;

      I  :=  I  +  1 ;
      EXIT  WHEN  I  >  10 ;

   END  LOOP ;
END ;
/

select * from resultat ;
```



Exemple : Afficher la table de multiplication du chiffre 5.

```
DELARE
   Chiffre        number(1)      :=  5 ;
   Result         number(2) ;
BEGIN
   FOR   I  IN  1  ..  10
   LOOP
      Result  :=  Chiffre  *  I ;

      INSERT  INTO  RESULTAT ( chaine )
      VALUES  ( I  ||  ' x  '  ||  chiffre  ||  ' = '  ||  result  ) ;

   END  LOOP ;
END ;
/

select * from resultat ;
```



Exemple : Afficher la table de multiplication du chiffre 5.

```
DELARE
   I            number(2)     :=  1 ;
   Chiffre      number(1)     :=  5 ;
   Result       number(2) ;
BEGIN

   WHILE  I  <=  10
   LOOP
      Result  :=  Chiffre  *  I ;

      INSERT  INTO  RESULTAT ( chaine )
      VALUES  ( I  ||  ' x  '  ||  chiffre  ||  ' = '  ||  result ) ;

      I  :=  I + 1 ;

   END  LOOP ;
END ;
/

select * from resultat ;
```