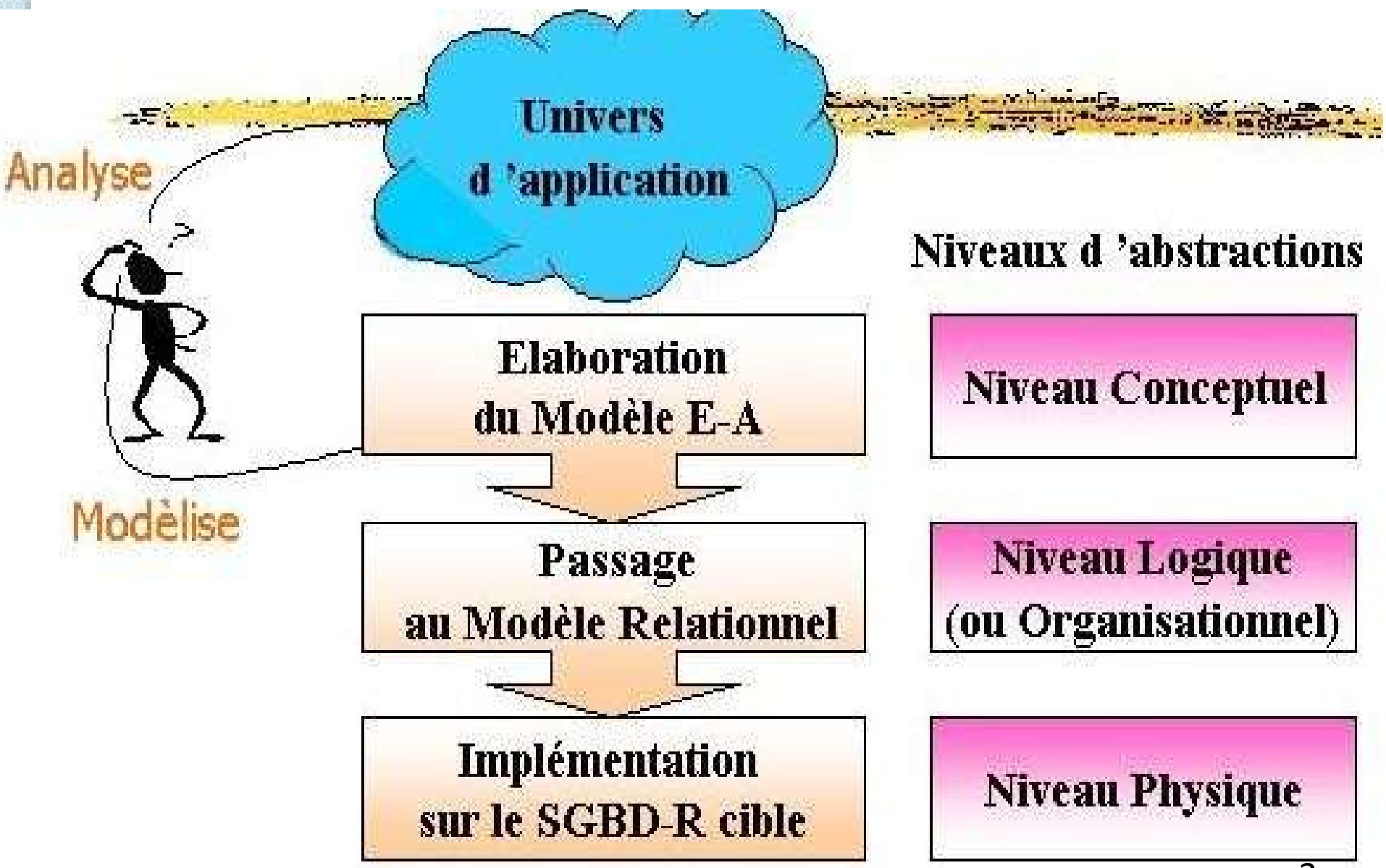
Chapitre I : Introduction à PL/SQL

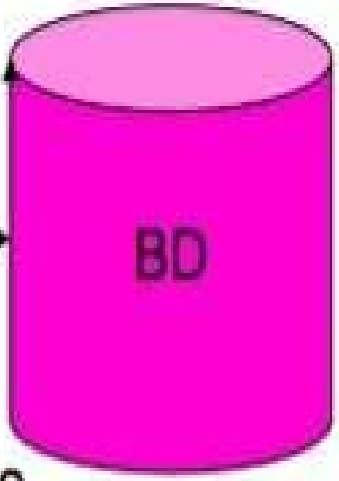
SGBD et BD

- Une BD (Base de données) est une collection de données stockées dans des fichiers sur disques et offrant à l'utilisateur une grande variété d'utilisation (création, modification, consultation, suppression).
- Un SGDB (Système de Gestion de Bases de Données), apparaît donc comme un ensemble de moyens logiciels permettant de stocker, retrouver, traiter et communiquer cette collection de données. Il est la coquille autour de la BD qu'il gère. Il en assure l'accès, le partage, l'intégrité et la sécurité.
- Le langage SQL est un langage de définition, de contrôle et de manipulation des données à travers des requêtes.
- Les SGBD relationnels supporte le langage SQL en tant que standard d'implantation, de manipulation et d'administration des BD.

SGBD et BD


Univers
d 'application
Analyse
Modèlise
Niveaux d 'abstractions
Elaboration
du Modèle E-A
Niveau Conceptuel
Passage
au Modèle Relationnel
Niveau Logique
(ou Organisationnel)
Implémentation
sur le SGBD-R cible
Niveau Physique

SGBD et BD



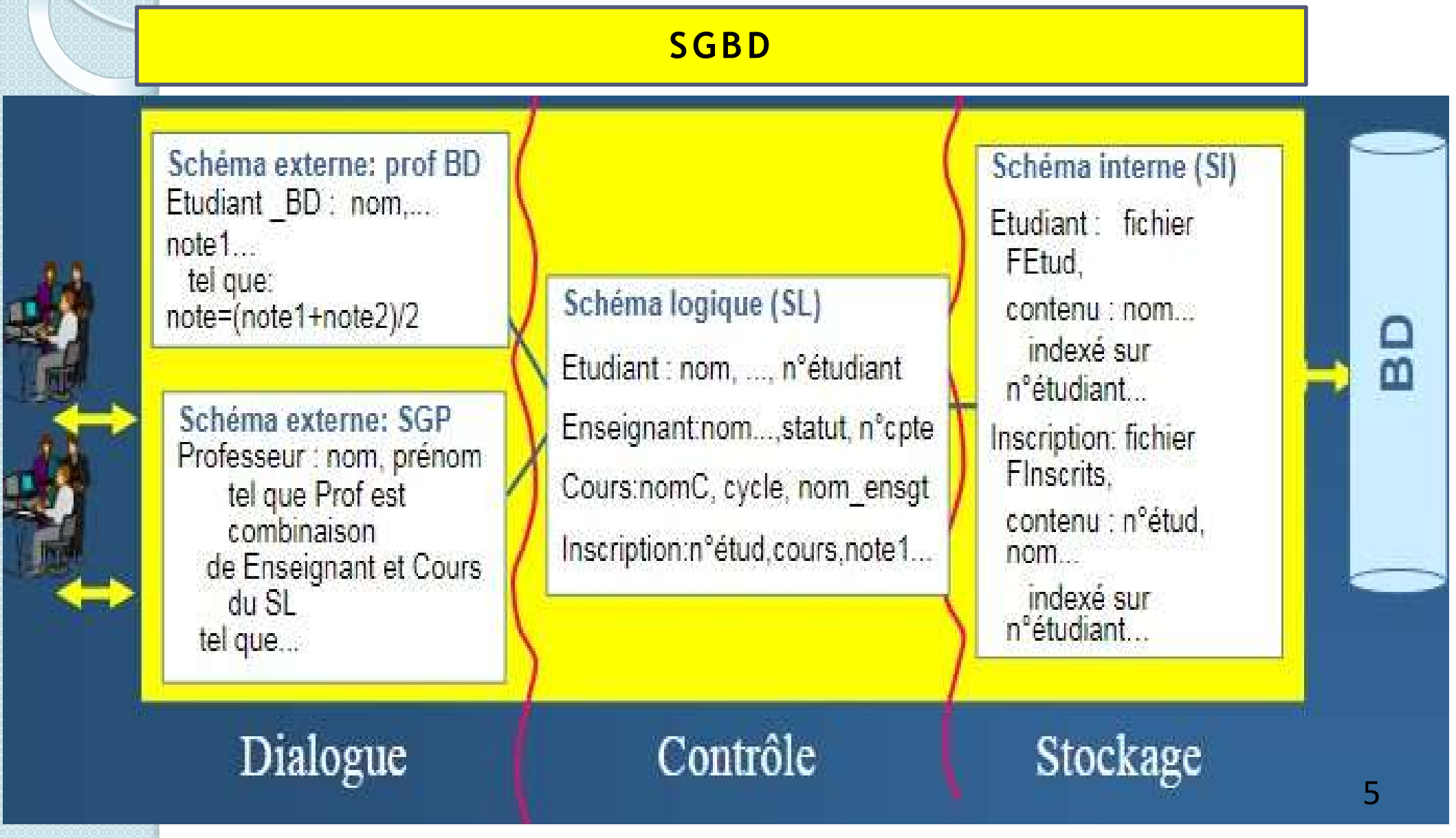
Interface utilisateur

Interface d'accès physique

- Convivialité de l'interface
- Analyse, vérification de requêtes

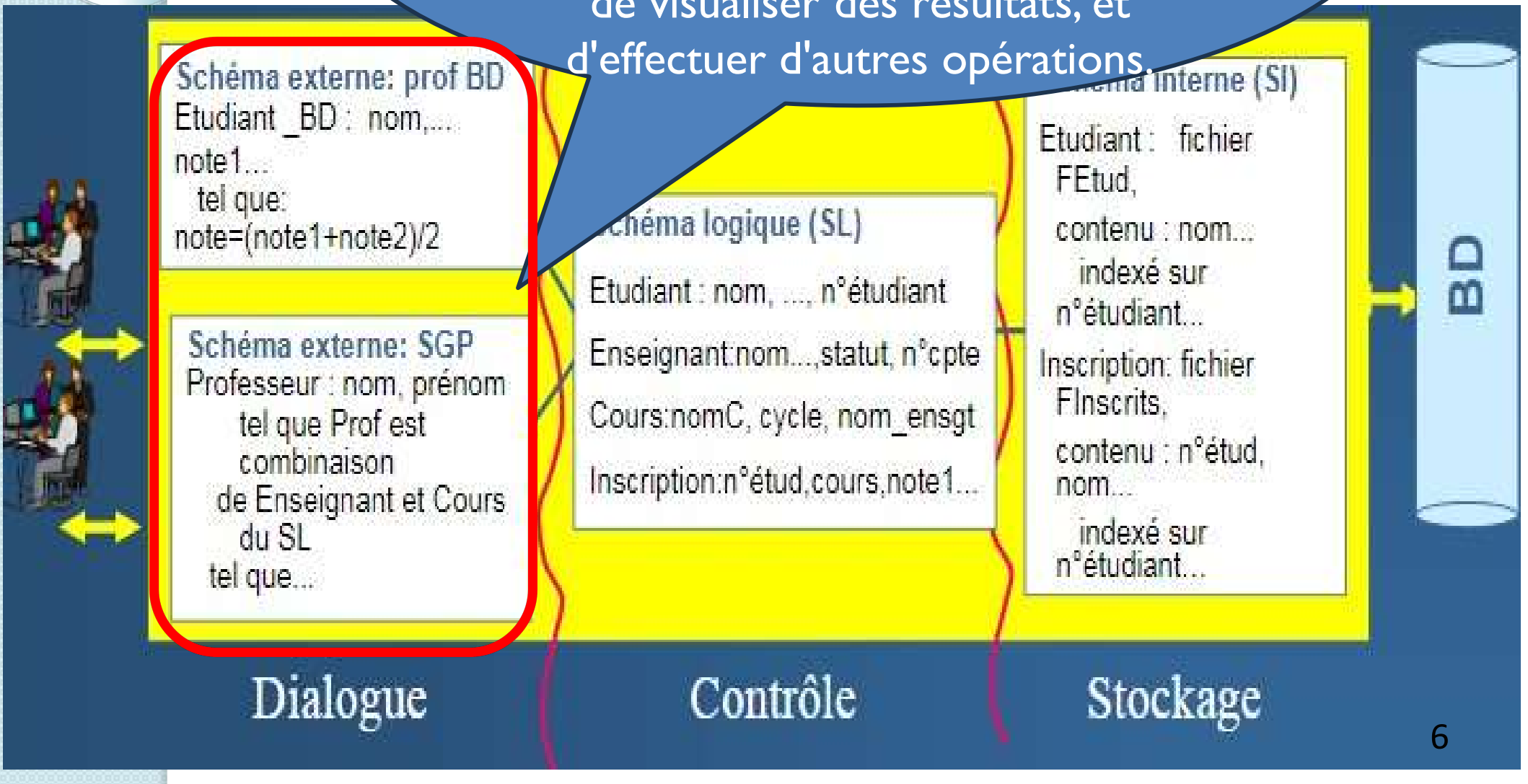
- Stockage / accès aux données
- Optimisation des performances

SGBD et BD


SGBD
Schéma externe: prof BD
Etudiant _BD : nom,....
note1...
tel que:
note=(note1+note2)/2
Schéma externe: SGP
Professeur : nom, prénom
tel que Prof est
combinaison
de Enseignant et Cours
du SL
tel que...
Schéma logique (SL)
Etudiant : nom, ..., n°étudiant
Enseignant:nom...,statut, n°cpte
Cours:nomC, cycle, nom_ensgt
Inscription:n°étud,cours,note1...
Schéma interne (SI)
Etudiant : fichier
FEtud,
contenu : nom...
indexé sur
n°étudiant...
Inscription: fichier
FInscrits,
contenu : n°étud,
nom...
indexé sur
n°étudiant...
BD
Dialogue
Contrôle
Stockage

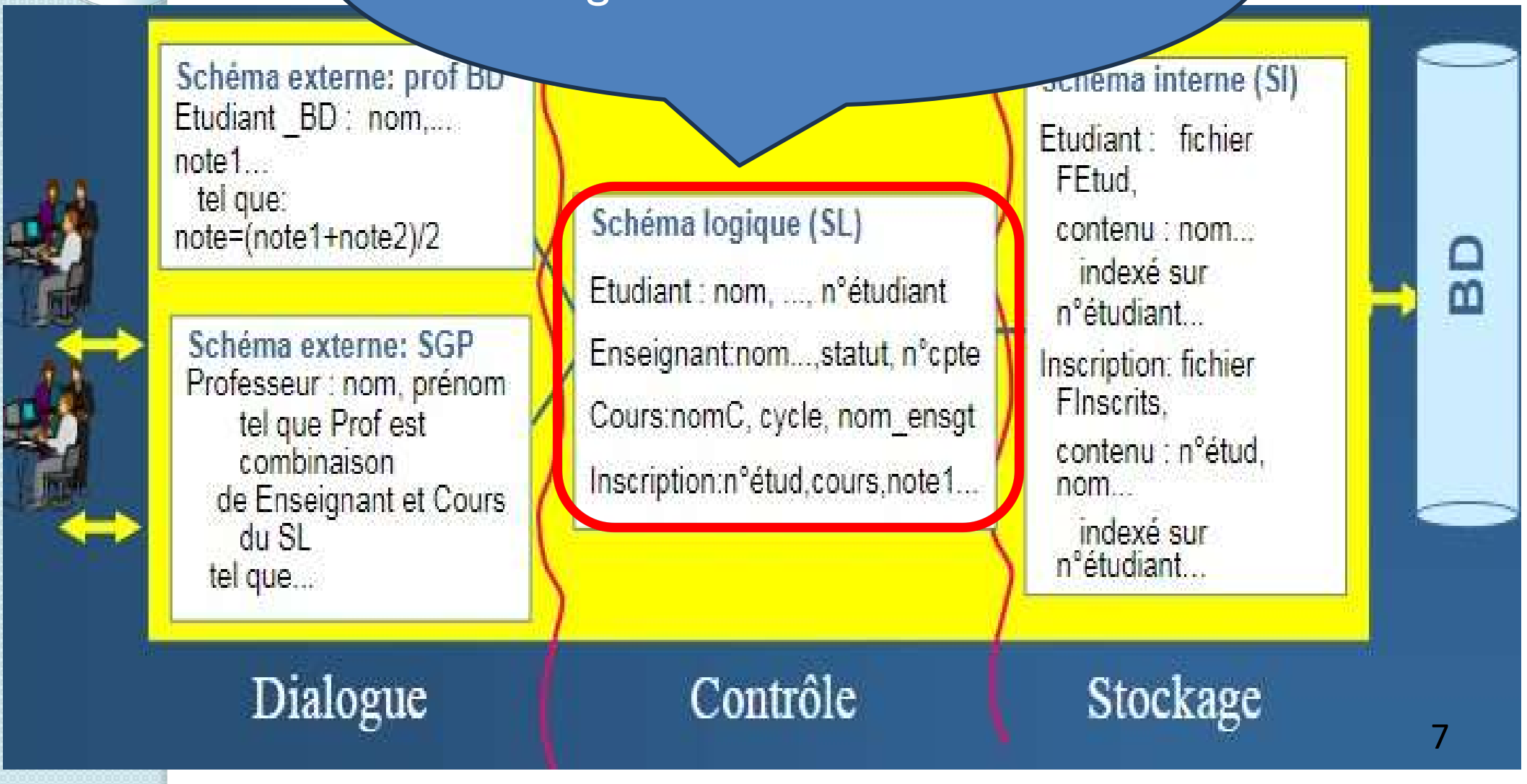
SGB

Le dialogue fait référence à l'interaction entre les utilisateurs et le SGBD. l'interface utilisateur permettant aux utilisateurs d'interagir avec le système, de saisir des requêtes, de visualiser des résultats, et d'effectuer d'autres opérations.

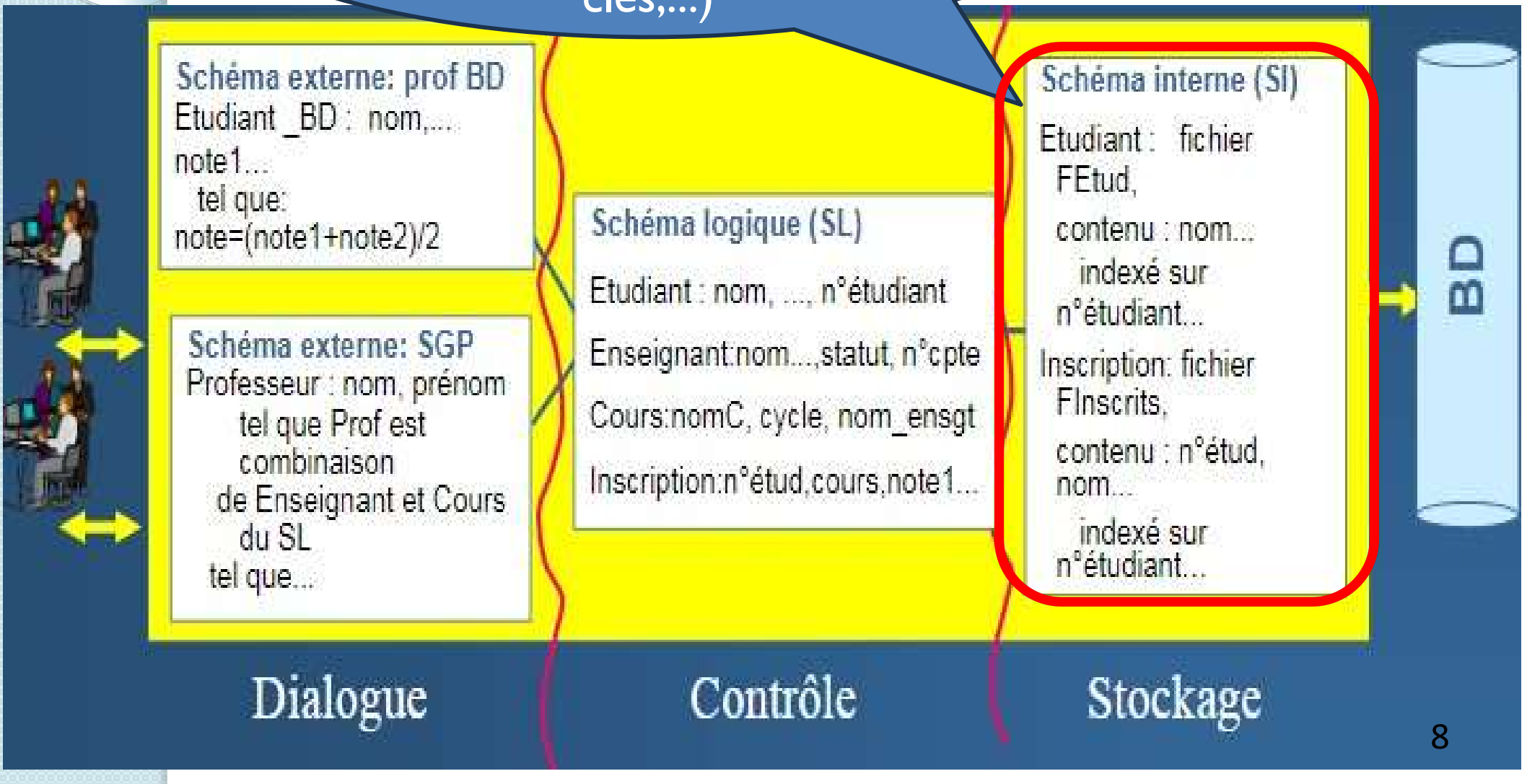


SGBD et BD

contrôle global et structure globale des données



Stockage des données sur des supports physiques, gestion des structures de mémorisation (fichiers) et accès (gestion des clés,...)



Langage SQL

- Langage standard de requête et de mise à jour, présente l'avantage d'être assertionnel permettant de décrire, dans une syntaxe simple, le traitement qu'on veut réaliser sur la BD sans se préoccuper de la façon algorithmique pour le faire
- SGBD accomplisse cette tache.
- Mixage des ordres SQL avec les instructions procédurales
- Oracle propose une extension de SQL pour permettre des traitements algorithmiques sur des ensembles de n-uplets issus de la BD
- Naissance de PL/SQL

Langage PL/SQL

- Le langage PL/SQL (Procedural language/SQL) est une extension du langage SQL qui offre un environnement procédural au langage SQL.
- Il supporte les commandes LMD (SELECT, INSERT, UPDATE, DELETE) et une collection de constructions algorithmiques: tests, boucles, affectations
- Il supporte aussi des instructions déclaratives de variables, records, curseurs ainsi que des fonctions intégrées et offre des bibliothèques de programmes et créer d'autres
- PL/SQL est une extension procédurale à SQL: un langage structuré de développement de programmes modulaires
- PL/SQL est utilisé à partir de SQL*Plus pour tester et déboguer des séquences et des sous-programmes PL/SQL.

Langage PL/SQL

- Dans l'environnement PL/SQL, les ordres SQL et PL/SQL sont regroupés en blocs.

Dans PL/SQL, les commandes SQL (telles que SELECT, INSERT, UPDATE, DELETE) et les commandes PL/SQL (utilisées pour le traitement procédural) sont combinées dans des blocs de code. Un bloc de code PL/SQL commence par les mots-clés DECLARE, BEGIN et se termine par END.

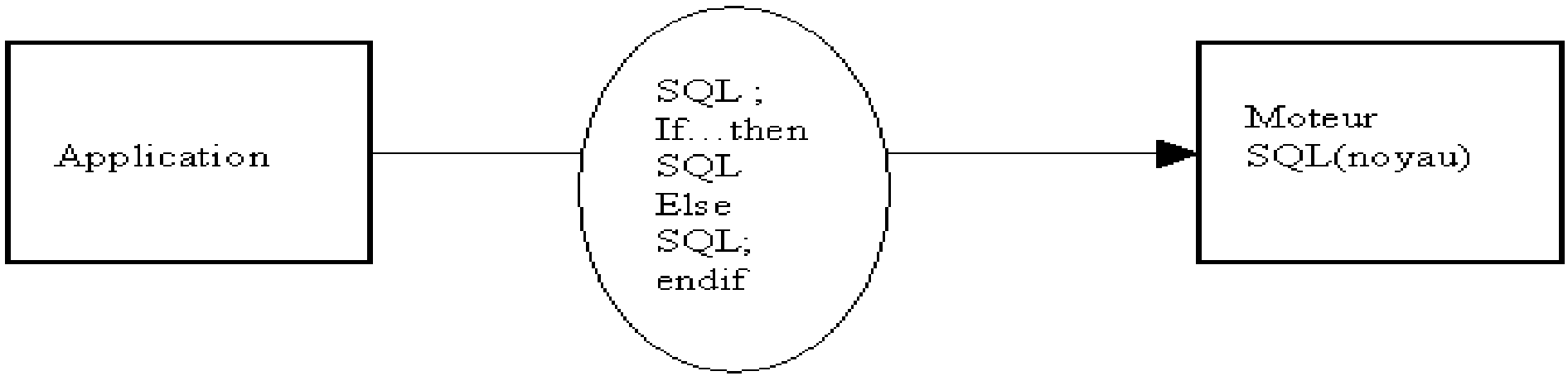
Langage PL/SQL

- Dans l'environnement PL/SQL, les ordres SQL et PL/SQL sont regroupés en blocs.
- Un bloc ne demande qu'un seul transfert vers le moteur PL/SQL qui interprète en une seule fois l'ensemble des commandes contenues dans le bloc.

Lorsqu'un bloc PL/SQL est exécuté, il est envoyé au moteur PL/SQL pour interprétation et exécution. Cela signifie que toutes les commandes SQL et PL/SQL à l'intérieur du bloc sont envoyées et interprétées en une seule opération

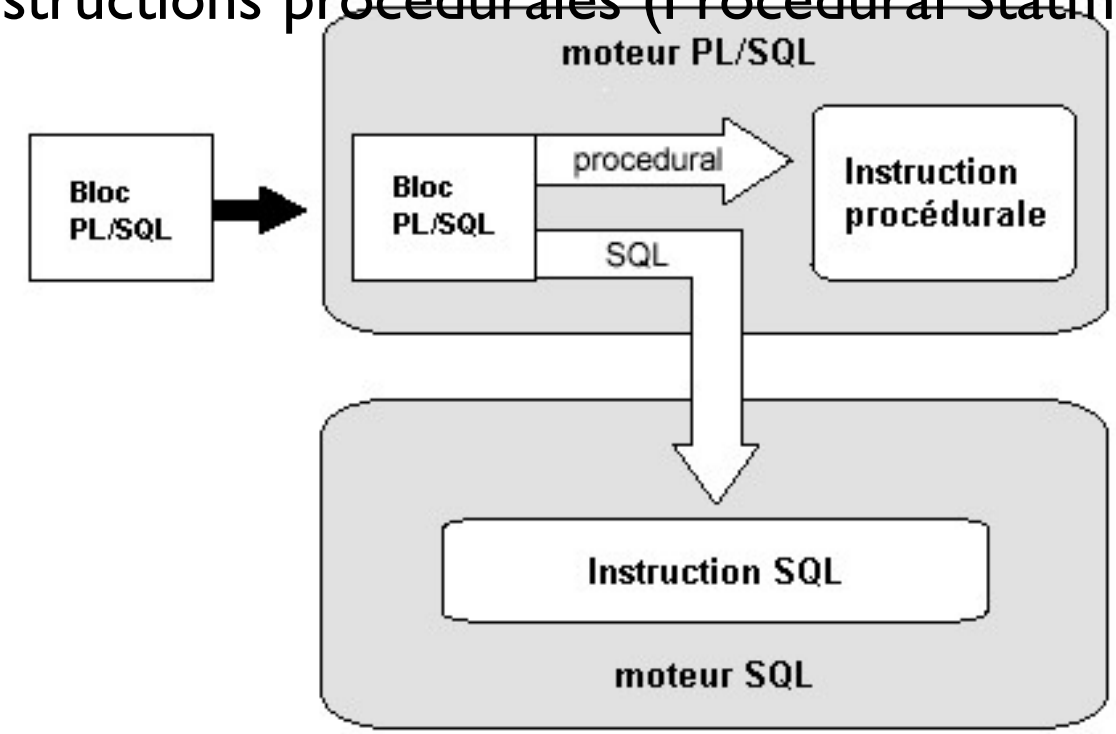
Langage PL/SQL

- Dans l'environnement PL/SQL, les ordres SQL et PL/SQL sont regroupés en blocs.
- Un bloc ne demande qu'un seul transfert vers le moteur PL/SQL qui interprète en une seule fois l'ensemble des commandes contenues dans le bloc.
- L'interet de présence du moteur dans l'outil est d'alléger les transferts client/serveur et donc d'améliorer les performances.



Langage PL/SQL

- Lorsque le moteur PL/SQL reçoit un bloc pour exécution, il effectue les opérations suivantes:
  - Séparation des ordres SQL et PL/SQL
  - Passage des commandes SQL au processeur SQL (SQL Statment Executor)
  - Passage des instructions procédurales au processeur d'instructions procédurales (Procedural Statment Executor)



Structure d'un bloc PL/SQL

```
DECLARE
   -- definition des variables
BEGIN
   -- code du programme
EXCEPTION

-- code de gestion des erreurs
END;
```

Section optionnelle: pour chaque erreur prévisible un gestionnaire d'exception approprié est souhaitable (une ou plusieurs instructions ou même un sous-bloc PL/SQL. Les erreurs à prévoir sont liées aux instructions qu'il contient

Section exécutable: obligatoire: contient des instructions du programme et la section de gestion des erreurs si présente. Les instructions peuvent etre:
-Instructions d'affectation
-Instructions de contrôle de flux
-Instructions SQL (select, insert, update, delete, commit, rollback, etc)
-Instructions de gestion des curseurs
-Instructions de gestion des erreurs

Types PL/SQL et déclarations

- Pour réaliser des traitements, il faut utiliser des variables.
- Ces variables servent notamment au stockage des données obtenues depuis la base par exécution d'une requête SQL ou utilisées comme paramètres dans les ordres de LMD.
- Toute variable utilisée dans un bloc doit être déclarée dans une section DECLARE.

Types scalaires	Types composés
• BOOLEAN • BINARY_INTEGER, NUMBER • INT, INTEGER, SMALLINT, DEC, REAL, DECIMAL • NATURAL, POSITIVE, ROWID	• Type enregistrement (RECORD) • Type Table (TABLE)

Types PL/SQL et déclarations

- Pour une meilleure lisibilité des programmes, il est recommandé de préfixer les identificateurs selon les conventions ci-dessous:

Identifiant	Préfix	Exemple
Variables	V_	V_sal
Exception	E_	E_RupStock
Variables Hôtes	H_	H_Ename
Curseurs	C_	C_EMPDEPT10
Paramètre	P_	P_EMPNO

heureDepart
getNom() C[illegible]Nom()

Types scalaires

Outre les types CHAR, VARCHAR2, NUMBER et DATE, disponibles dans le langage SQL, le langage PL/SQL offre les types supplémentaires suivants :

- BOOLEAN
- BINARY_INTEGER
- DECIMAL
- FLOAT
- INTEGER
- REAL.

Types scalaires

- La déclaration d'une variable se fait par association du nom de la variable à un type sous la forme :

  nom_variable type [NOT NULL] [{:=|DEFAULT} expression];

  Exemple: v_nompilote varchar2(25);

  [:=|DEFAULT expression]initialise une variable dés sa déclaration. NOT NULL interdit d'affecter une valeur nulle dans la variable durant toute sa vie et exige obligatoirement son initialisation à la déclaration.

- Il est aussi possible de déclarer une variable par référence à une colonne d'une table, par la notation %type :

  nom_variable nom_table.nom_colonne%type;

  Exemple: v_nompilote pilote.nom%type;

Types scalaires

Type	Explication
BOOLEAN	Une variable booléenne peut recevoir les valeurs TRUE, FALSE ou NULL
BINARY_INTEGER	Entiers signés -2147483647 à + 2147483647, stockés en binaire en format complément à 2, directement utilisables sans conversion et sont donc convenables pour les indices de boucles ou de tables PL/SQL.
NUMBER (n,m)	Valeur numérique décimale ou entière identique au type NUMBER des colonnes. Le maximum de n est 38. m varie entre -84 et 127.
INT, INTEGER, SMALLINT, DEC, REAL, DECIMAL	Sont des sous-types du type NUMBER
NATURAL	Sous-ensemble de BINARY_INTEGER des entiers de 0 à 2147483647
POSITIVE	Sous-ensemble de BINARY_INTEGER des entiers de 1 à 2147483647
ROWID	Identique aux pseudos colonnes de type ROWID. Un ROWID est un identificateur interne unique de chaque n-uplet dans la base. Il est stocké en format binaire de longueur fixe.

Types scalaires

- Une constante est un nom attribué à une valeur qui ne change pas durant toute sa durée de vie dans le programme.
- La déclaration d'une variable se fait par association du nom de la variable à un type sous la forme :

nom_constante CONSTANT nom_de_type [{:=DEFAULT}expression];

Exemple: pie CONSTANT NUMBER(8,5):=3.14159;