

Quatrième partie – Edition des relevés de notes individuels :

Dans cette partie, on souhaite pouvoir éditer le relevé de notes semestriel d'un étudiant donné. Par exemple, le relevé de notes du semestre S3 de l'éblouissant étudiant E10 doit avoir la forme suivante :

Identifiant étudiant : E10
Nom étudiant : Palleja
Prénom étudiant : Xavier
Sexe étudiant : M
Date naissance étudiant : 09.03.1994
Groupe étudiant : D2

Prog PHP : 18
Systèmes et Réseaux : 17
Moyenne module Informatique Web : 17.5

Refactoring Agile : 14
Qualité et Tests : 16
eXtreme Programming : 18
Moyenne module Culture Générale Info : 16

Projet Programmation : 16
Moyenne module Projet Final Pratique : 16

Moyenne semestre : 16.6

Resultat : O
Classement : 1

La moyenne d'un module est calculée avec les notes des différentes matières qui sont incluses dans le module (en prenant en compte les coefficients des matières). La moyenne d'un semestre est calculée avec les moyennes des modules (en prenant en compte les coefficients des modules).

Le résultat indique si l'étudiant valide son semestre (O/N). Pour valider un semestre, il faut avoir au moins 10 de moyenne générale, et il faut également avoir au moins 8 dans chacun des modules. Le classement indique la place de l'étudiant dans sa promotion pour le semestre donné (naturellement, l'étudiant précédent est major de sa promotion).

Ci-dessous, l'exemple du relevé de notes du semestre S3 de l'étudiant E19.

Identifiant étudiant : E19
Nom étudiant : Micoton
Prénom étudiant : Mylène
Sexe étudiant : F
Date naissance étudiant : 08.04.1995
Groupe étudiant : D1

Prog PHP : 9
Systèmes et Réseaux : 8
Moyenne module Informatique Web : 8,5

Refactoring Agile : 11
Qualité et Tests : 10
eXtreme Programming : 4,5
Moyenne module Culture Générale Info : 8,5

Projet Programmation : 10
Moyenne module Projet Final Pratique : 10

Moyenne semestre : 8,8

Resultat : N
Classement : 10

L'étudiant précédent ne valide pas son semestre car, bien qu'ayant plus de 8 à tous les modules, il n'a pas la moyenne générale.

Enfin, un dernier exemple de relevé de notes ; celui du semestre S1 de l'étudiant E17.

```

Identifiant étudiant : E17
Nom étudiant : Tarembois
Prénom étudiant : Guy
Sexe étudiant : M
Date naissance étudiant : 08.04.1997
Groupe étudiant : T2

Prog ADA : 12
BD : 13
Moyenne module Initiation Informatique : 12.5

Math : 10
Gestion : 9
Communication : 17
Moyenne module Culture Générale : 12

Projet : 8
Initiation Joomla! : 6.5
Moyenne module Mise en Pratique : 7.5

Moyenne semestre : 11.3

Resultat : N
Classement : 8

```

L'étudiant précédent ne valide pas son semestre 1, bien qu'il ait la moyenne générale et qu'il soit classé 8^{ème} de promo, car il n'a pas au moins 8 dans tous les modules. En effet il n'a que 7,5 dans le module "Mise en Pratique".

Pour éditer ces relevés de notes, nous allons commencer par écrire les fonctions qui permettent de calculer les moyennes intermédiaires. Puis nous écrirons les fonctions qui calculent le résultat et le classement de l'étudiant.

A. Gestions des moyennes

14) Fonction stockée moyenneEtudiantModule.

- Ecrire une fonction stockée moyenneEtudiantModule qui retourne la moyenne qu'a obtenue un étudiant p_idEtudiant à un module p_idModule.

Pour calculer la moyenne au module, il faut prendre en compte les coefficients des matières incluses dans le module.

Par exemple, si dans un module, un étudiant a obtenu 8 à une matière qui a un coefficient 2 et 11 à une matière qui a un coefficient 1, il aura 9 de moyenne au module :

$$(8*2 + 11*1) / (2+1) = 9$$

Autre exemple, si à un module, un étudiant a obtenu 8 à une matière qui a un coefficient 2 et 11 à une matière qui a un coefficient 2, il aura 9,5 de moyenne au module :

$$(8*2 + 11*2) / (2+2) = 9,5$$

La fonction doit avoir la signature suivante :

```

FUNCTION moyenneEtudiantModule(
                                p_idEtudiant IN Etudiants.idEtudiant%TYPE,
                                p_idModule IN Modules.idModule%TYPE)
                                RETURN NUMBER

```

- Résultats attendus :

```

SELECT moyenneEtudiantModule('E6', 'M112')
FROM DUAL ;

moyenneEtudiantModule('E6', 'M112')
-----
7.5

```

```

SELECT moyenneEtudiantModule('E8', 'M113')
FROM DUAL ;

moyenneEtudiantModule('E8', 'M113')
-----
13.5

```

15) Fonction stockée valideEtudiantModule.

- Ecrire une fonction stockée valideEtudiantModule qui retourne 1 si un étudiant p_idEtudiant valide un module p_idModule ; 0 sinon. On rappelle qu'un étudiant valide un module si il a au moins 8 de moyenne à ce module. Cette fonction doit avoir la signature suivante :

```
FUNCTION valideEtudiantModule (
    p_idEtudiant IN Etudiants.idEtudiant%TYPE,
    p_idModule IN Modules.idModule%TYPE)
    RETURN NUMBER
```

- Résultats attendus :

```
SELECT valideEtudiantModule('E6', 'M112')
FROM DUAL ;
```

```
valideEtudiantModule('E6', 'M112')
-----
0
```

```
SELECT valideEtudiantModule('E19', 'M211')
FROM DUAL ;
```

```
valideEtudiantModule('E19', 'M211')
-----
1
```

16) Fonction stockée moyenneEtudiantSemestre.

- Ecrire une fonction stockée moyenneEtudiantSemestre qui retourne la moyenne qu'a obtenue un étudiant p_idEtudiant au semestre p_idSemestre.

Pour calculer cette moyenne, il faut prendre en compte les coefficients des différents modules qu'a suivis l'étudiant au cours du semestre.

Cette fonction doit avoir la signature suivante :

```
FUNCTION moyenneEtudiantSemestre (
    p_idEtudiant IN Etudiants.idEtudiant%TYPE,
    p_idSemestre IN Semestres.idSemestre%TYPE)
    RETURN NUMBER
```

- Résultats attendus :

```
SELECT moyenneEtudiantSemestre('E1', 'S1')
FROM DUAL ;
```

```
moyenneEtudiantSemestre('E1', 'S1')
-----
11.8
```

```
SELECT moyenneEtudiantSemestre('E3', 'S2')
FROM DUAL ;
```

```
moyenneEtudiantSemestre('E3', 'S2')
-----
16.4
```

```
SELECT moyenneEtudiantSemestre('E3', 'S3')
FROM DUAL ;
```

```
moyenneEtudiantSemestre('E3', 'S3')
-----
```

17) Procédure stockée affichageMoyEtudiantSemestre.

- Ecrire une procédure stockée affichageMoyEtudiantSemestre qui affiche toutes les notes et les moyennes obtenues par un étudiant p_idEtudiant au cours du semestre p_idSemestre. Cette procédure doit avoir la signature suivante :

```
PROCEDURE affichageMoyEtudiantSemestre (
    p_idEtudiant IN Etudiants.idEtudiant%TYPE,
    p_idSemestre IN Semestres.idSemestre%TYPE)
```

- Résultat attendu :

```
CALL affichageMoyEtudiantSemestre('E10', 'S3') ;
```

```
Identifiant étudiant : E10
Nom étudiant : Palleja
Prénom étudiant : Xavier
Sexe étudiant : M
Date naissance étudiant : 09.03.1994
Groupe étudiant : D2

Prog PHP : 18
Systèmes et Réseaux : 17
Moyenne module Informatique Web : 17.5

Refactoring Agile : 14
Qualité et Tests : 16
eXtreme Programming : 18
Moyenne module Culture Générale Info : 16

Projet Programmation : 16
Moyenne module Projet Final Pratique : 16

Moyenne semestre : 16.6
```

B. Gestion du résultat et du classement

18) Fonction stockée valideSemestre

- Ecrire une fonction stockée valideSemestre qui retourne 'O' si l'étudiant p_idEtudiant valide le semestre p_idSemestre ; 'N' sinon.

Un étudiant valide un semestre s'il a au moins 10 de moyenne générale ET s'il a également au moins 8 de moyenne dans chacun des modules qu'il a suivi. Cette fonction doit avoir la signature suivante :

```
FUNCTION valideSemestre(p_idEtudiant IN Etudiants.idEtudiant%TYPE,
    p_idSemestre IN Semestres.idSemestre%TYPE)
    RETURN VARCHAR
```

- Résultats attendus :

```
SELECT valideSemestre('E17', 'S2')
FROM DUAL ;
```

```
valideSemestre('E17', 'S2')
-----
N
```

```
SELECT valideSemestre('E18', 'S1')
FROM DUAL ;
```

```
valideSemestre('E18', 'S1')
-----
O
```

```
SELECT valideSemestre('E9', 'S3')
FROM DUAL ;
```

```
valideSemestre('E9', 'S3')
-----
O
```

19) Fonction stockée classementEtudiantSemestre

- Ecrire une fonction stockée classementEtudiantSemestre qui retourne le classement de l'étudiant p_idEtudiant dans sa promotion pour le semestre p_idSemestre.

Le classement est calculé à partir de la moyenne générale des étudiants. L'étudiant qui a la plus forte moyenne de la promotion a le classement 1, le suivant le classement 2 et ainsi de suite. Si deux étudiants ont la même moyenne ils doivent avoir le même classement.

Cette fonction doit avoir la signature suivante :

```
FUNCTION classementEtudiantSemestre(
    p_idEtudiant IN Etudiants.idEtudiant%TYPE,
    p_idSemestre IN Semestres.idSemestre%TYPE)
    RETURN NUMBER
```

- Il est à noter qu'il est possible d'écrire cette fonction de façon extrêmement simple sans avoir à utiliser un curseur !

- Résultats attendus :

```
SELECT classementEtudiantSemestre('E10', 'S3')
FROM DUAL ;
```

```
classementEtudiantSemestre('E10', 'S3')
-----
1
```

```
SELECT classementEtudiantSemestre('E21', 'S2')
FROM DUAL ;
```

```
classementEtudiantSemestre('E21', 'S2')
-----
7
```

20) Procédure stockée affichageResEtudiantSemestre.

- Ecrire une procédure stockée affichageResEtudiantSemestre qui affiche le résultat et le classement de l'étudiant p_idEtudiant au semestre p_idSemestre.

Cette procédure doit avoir la signature suivante :

```
PROCEDURE affichageResEtudiantSemestre(
    p_idEtudiant IN Etudiants.idEtudiant%TYPE,
    p_idSemestre IN Semestres.idSemestre%TYPE)
```

- Résultat attendu :

```
CALL affichageResEtudiantSemestre('E10', 'S3') ;
```

```
Resultat : 0
Classement : 1
```

21) Procédure stockée affichageReleveNotes.

- Ecrire une procédure stockée affichageReleveNotes qui affiche le relevé de notes de l'étudiant p_idEtudiant au semestre p_idSemestre. Cette procédure doit avoir la signature suivante :

```
PROCEDURE affichageReleveNotes(
                                p_idEtudiant IN Etudiants.idEtudiant%TYPE,
                                p_idSemestre IN Semestres.idSemestre%TYPE)
```

- Résultats attendus :

```
CALL affichageReleveNotes('E10', 'S4') ;
```

```

Identifiant étudiant : E10
Nom étudiant : Palleja
Prénom étudiant : Xavier
Sexe étudiant : M
Date naissance étudiant : 09.03.1994
Groupe étudiant : D2

BD Objet : 19
UML et Design Patterns : 20
Moyenne module Informatique Compl. : 19.5

Stage : 16.5
Moyenne module Stage Pratique : 16.5

Moyenne semestre : 18

Resultat : 0
Classement : 1
```

```
CALL affichageReleveNotes('E9', 'S4') ;
```

```

Identifiant étudiant : E9
Nom étudiant : Croque
Prénom étudiant : Odile
Sexe étudiant : F
Date naissance étudiant : 14.07.1996
Groupe étudiant : D1

BD Objet : 9
UML et Design Patterns : 9
Moyenne module Informatique Compl. : 9

Stage : 13
Moyenne module Stage Pratique : 13

Moyenne semestre : 11

Resultat : 0
Classement : 9
```

Cinquième partie – Edition du PV de fin de semestre :

Pour faciliter le déroulement des jurys de fin de semestre, on souhaite pouvoir éditer le Procès Verbal (PV) d'une promotion pour un semestre donné. Un PV doit afficher dans un même tableau toutes les notes obtenues par les étudiants d'une promotion au cours d'un semestre. Il doit également indiquer toutes les moyennes de ces étudiants (moyennes des modules, moyennes générales), et le résultat.

Par exemple, pour le semestre S4 de la promotion A2, on souhaite éditer le PV suivant :

nom et prénom des étudiants		notes dans les matières du premier module		notes dans les matières du module suivant			résultat	
PALLEJA	XAVIER	19	20	19,5	16,5	16,5	18	O
STICKE	SOPHIE	17	18	17,5	15,5	15,5	16,5	O
OUTAN	LAURENT	13	14	13,5	16,5	16,5	15	O
TRISER	JESSICA	14	14	14	15	15	14,5	O
ORAQUE	ANNE	13	14	13,5	14,5	14,5	14	O
TERRIEUR	ALAIN	13	11	12	14	14	13	O
STICKO	JUDAS	12	9	10,5	13,5	13,5	12	O
NEMARD	JEAN	12	11	11,5	12,5	12,5	12	O
CROQUE	ODILE	9	9	9	13	13	11	O
BRICOT	JUDAS	8	6	7	11	11	9	N
MICOTON	MYLENE	8	5	6,5	8,5	8,5	7,5	N
		moyenne au premier module		moyenne au module suivant			moyenne générale	

Il est à noter que sur le PV, sont d'abord affichés tous les étudiants qui ont validé le semestre (c'est-à-dire pour qui le résultat est O) ; puis en dessous, tous ceux qui n'ont pas validé le semestre (résultat = N). Les étudiants qui valident leur semestre sont classés par rapport à leur moyenne générale. Même chose pour les étudiants qui ne valident par leur semestre.

On veut que le nom et le prénom des étudiants soient écrits en lettres majuscules. Si un nom ou un prénom fait plus de 10 lettres, il pourra être tronqué.

Pour les notes et les moyennes, on n'affichera pas plus d'une décimale. Si une note ou une moyenne possède plusieurs décimales, elle pourra être tronquée.

22) Procédure stockée affichagePV.

- Ecrire une procédure stockée `affichagePV` qui affiche le Procès Verbal de la promotion au semestre `p_idSemestre`. Cette procédure doit avoir la signature suivante :

```
PROCEDURE affichagePV( p_idSemestre IN Semestres.idSemestre%TYPE)
```

Pour améliorer la lisibilité de votre code et faciliter sa maintenance, il est fortement conseillé de créer d'autres procédures stockées qui seront appelées à l'intérieur du corps de procédure `affichagePV`.

Afin d'améliorer les performances, vous avez la possibilité, si vous le souhaitez, de rajouter des champs calculés dans votre Schéma Relationnel.

Pour mettre en forme votre texte, vous pourrez utiliser les fonctions Oracle suivantes :

- `LPAD()` et `RPAD()` qui permettent d'aligner un texte sur la gauche ou sur la droite et qui permet également de le tronquer si il est trop long.
- `UPPER()` qui permet de mettre en majuscules toutes les lettres d'un texte.
- `TRANSLATE()` qui peut être utilisé pour remplacer les lettres accentuées d'un texte par des lettres non accentuées (par exemple, remplacer É par E).

- Il est à noter que si on affiche le résultat directement dans le navigateur, on risque de perdre la mise en forme (les espaces consécutifs vont disparaître). Pour cette raison, demandez à `iSQL*Plus` de ne pas générer le résultat dans le navigateur mais dans un fichier externe que vous pourrez ouvrir avec votre éditeur de texte préféré (par exemple Notepad++) :

Préférences – Emplacement de sortie – Enregistrer dans un fichier HTML.

Comme votre fichier externe sera ouvert avec un éditeur texte, demandez également à `iSQL*Plus`, lors de la génération du fichier, de ne pas remplacer les sauts de lignes par des balises `
` :

Préférences – Formatage du script – Sortie préformatée – Activer (On).

- Résultats attendus : voir pages suivantes.

CALL affichagePV('S1');

DEUF	JOHN	15	15	15	19	18	17	18	16,5	15	16	16,4	O
KAECOUTE	XAVIER	10	18	14	14	14	17	15	16,5	15	16	14,8	O
TIMETTRE	VINCENT	16	17	16,5	13	12	14	13	15,5	14	15	14,8	O
NANAS	JUDAS	14	14	14	12	9	12	11	13	14,5	13,5	12,7	O
GATOR	ALI	8	14	11	14	12	13	13	14	12,5	13,5	12,3	O
ZETOFRAIS	MELANIE	9	10	9,5	14	13	12	13	15	15	15	12	O
ALIZAN	GASPARD	12	11	11,5	9	11	13	11	14	14	14	11,8	O
TERRIEUR	ALEX	13	12	12,5	10	9	8	9	13	11,5	12,5	11,1	O
JAVEL	AUDE	13	12	12,5	10	9	8	9	11	14	12	11	O
BONO	JEAN	6	13	9,5	9	10	14	11	13	13	13	10,8	O
OUZY	JACQUES	8	11	9,5	10	11	9	10	12	18	14	10,6	O
TAREMBOIS	GUY	12	13	12,5	10	9	17	12	8	6,5	7,5	11,3	N
ASSIN	MARC	13	14	13,5	9	5	8,5	7,5	12	15	13	11	N
VESSELLE	AUDE	9	11	10	9	7	5	7	11	14	12	9,2	N
ZEBLOUSE	AGATHE	2	1	1,5	15	13	11	13	15	15	15	8,8	N
NIOHRANGIN	NICOLAS	1	9	5	9	11	7	9	13	8,5	11,5	7,9	N

CALL affichagePV('S2');

KAECOUTE	XAVIER	16	14	15	17	18	17,5	17	17	16,4	O
DEUF	JOHN	14	14	14	15	15	15	14	14	14,4	O
TIMETTRE	VINCENT	16	18	17	11	12	11,5	14	14	14,2	O
NANAS	JUDAS	13	12	12,5	10	14	12	14	14	12,6	O
GATOR	ALI	14	14	14	10	11	10,5	13	13	12,4	O
OUZY	JACQUES	13	12	12,5	11	14	12,5	12	12	12,4	O
TERRIEUR	ALEX	12	12	12	11	13	12	12	12	12	O
JAVEL	AUDE	12	12	12	11	13	12	12	12	12	O
ZETOFRAIS	MELANIE	9	9	9	15	14	14,5	12	12	11,8	O
BONO	JEAN	12	11	11,5	10	13	11,5	12	12	11,6	O
ASSIN	MARC	12	11	11,5	10	9	9,5	15	15	11,4	O
TAREMBOIS	GUY	14	13	13,5	12	12	12	7	7	11,6	N
VESSELLE	AUDE	10	10	10	8	6	7	10	10	8,8	N
NIOHRANGIN	NICOLAS	5	4	4,5	10	11	10,5	11	11	8,2	N
ALIZAN	GASPARD	9	6	7,5	10	9	9,5	4	4	7,6	N
ZEBLOUSE	AGATHE	1	1	1	10	10	10	5	5	5,4	N

CALL affichagePV('S3');

PALLEJA	XAVIER	18	17	17,5	14	16	18	16	16	16	16,6	O
OUTAN	LAURENT	13	17	15	14	17	11	14	14	14	14,4	O
TRISER	JESSICA	14	16	15	14	13	12	13	15	15	14,2	O
TERRIEUR	ALAIN	13	14	13,5	12	12	10,5	11,5	15	15	13	O
ORAQUE	ANNE	12	13	12,5	14	13	12	13	14	14	13	O
NEMARD	JEAN	12	10	11	11	12	8,5	10,5	14	14	11,4	O
CROQUE	ODILE	12	14	13	9	10	8	9	12	12	11,2	O
STICKE	SOPHIE	17	19	18	8	9	4	7	14	14	12,8	N
STICKO	JUDAS	10	5	7,5	14	16	6	12	12	12	10,2	N
MICOTON	MYLENE	9	8	8,5	11	10	4,5	8,5	10	10	8,8	N
BRICOT	JUDAS	8	6	7	13	12	,5	8,5	11	11	8,4	N

CALL affichagePV('S4');

PALLEJA	XAVIER	19	20	19,5	16,5	16,5	18	O
STICKE	SOPHIE	17	18	17,5	15,5	15,5	16,5	O
OUTAN	LAURENT	13	14	13,5	16,5	16,5	15	O
TRISER	JESSICA	14	14	14	15	15	14,5	O
ORAQUE	ANNE	13	14	13,5	14,5	14,5	14	O
TERRIEUR	ALAIN	13	11	12	14	14	13	O
STICKO	JUDAS	12	9	10,5	13,5	13,5	12	O
NEMARD	JEAN	12	11	11,5	12,5	12,5	12	O
CROQUE	ODILE	9	9	9	13	13	11	O
BRICOT	JUDAS	8	6	7	11	11	9	N
MICOTON	MYLENE	8	5	6,5	8,5	8,5	7,5	N