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 : 09/03/98
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 sans les absences : 16.6
Nombre de demi-journées d'absence non justifiées : 0
Moyenne avec les absences : 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 sans les absences est calculée avec les moyennes des modules (en prenant en compte les coefficients des modules).

La moyenne avec les absences est calculée à partir de la moyenne sans les absences à laquelle on retranche 0,1 point à chaque demi-journée d'absence non justifiée. Il est à noter que si l'étudiant n'a qu'une seule demi-journée d'absence non justifiée, on ne lui retire rien.

Le résultat indique si l'étudiant valide son semestre (O/N). Pour valider un semestre, il faut avoir au moins 10 de moyenne générale (moyenne générale avec les absences), 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 S1 de l'étudiant E18.

```
Identifiant étudiant : E18
Nom étudiant : Ouzy
Prénom étudiant : Jacques
Sexe étudiant : M
Date naissance étudiant : 30/10/03
Groupe étudiant : Q3

Prog ADA : 8
BD : 11
Moyenne module Initiation Informatique : 9.5

Math : 10
Gestion : 11
Communication : 9
Moyenne module Culture Générale : 10

Projet : 12
Initiation Joomla! : 18
Moyenne module Mise en Pratique : 14

Moyenne semestre sans les absences : 10.6
Nombre de demi-journées d'absence non justifiées : 12
Moyenne avec les absences : 9.4

Resultat : N
Classement : 13
```

L'étudiant précédent ne valide pas son semestre car il n'a pas la moyenne générale. Il avait pourtant 10,6 de moyenne sans les absences mais à cause de ses 12 demi-journées d'absence non justifiées, sa moyenne a chuté à 9,4 et il ne valide donc pas son semestre. C'est d'ailleurs ce qui risque de vous arriver ce semestre si vous ne venez pas à tous les cours et TD de BD.

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 : 08/04/01
Groupe étudiant : Q2

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 sans les absences : 11.3
Nombre de demi-journées d'absence non justifiées : 1
Moyenne avec les absences : 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 8ème 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" (à cause de la matière Joomla! où il n'a eu que 6,5 – tant pis pour lui, il n'avait qu'à apprendre à programmer sous Joomla! Ou un autre CMS ce qui est planifié dans son Product Backlog de sprint).

Il est à noter que comme l'étudiant n'a qu'une seule demi-journée d'absence, sa moyenne avec les absences n'est pas impactée par les absences.

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 gèrent les absences. Enfin nous terminerons par coder 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 moyenneEtudiantSemestreSansAbs.

- Ecrire une fonction stockée `moyenneEtudiantSemestreSansAbs` 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 moyenneEtudiantSemestreSansAbs (
                    p_idEtudiant IN Etudiants.idEtudiant%TYPE,
                    p_idSemestre IN Semestres.idSemestre%TYPE)
                                                    RETURN NUMBER
```

- Résultats attendus :

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

moyenneEtudiantSemestreSansAbs('E1', 'S1')
------------------------------------------
11.8

SELECT moyenneEtudiantSemestreSansAbs('E3', 'S2')
FROM DUAL ;

moyenneEtudiantSemestreSansAbs('E3', 'S2')
------------------------------------------
16.4

SELECT moyenneEtudiantSemestreSansAbs('E3', 'S3')
FROM DUAL ;

moyenneEtudiantSemestreSansAbs('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 : 09/03/98
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 sans les absences : 16.6
```

B. Gestion simpliste des absences

18) Fonction stockée typeAbsence

- Pour pouvoir comptabiliser les absences dans la moyenne générale, il faut savoir quelles sont les absences de la table *Absences* qui sont excusées par un justificatif d'absence de la table *JustificatifsAbsences* et celles qui ne le sont pas.

  Pour cela, écrire une fonction stockée `typeAbsence` qui retourne 'E' si l'absence `p_idAbsence` est excusée par un justificatif d'absence, et 'A' dans le cas contraire. Pour savoir si un justificatif peut excuser une absence, il faut s'assurer que l'absence est incluse dans la période du justificatif et que le justificatif et l'absence concernent le même étudiant.

  Cette fonction doit avoir la signature suivante :

```
FUNCTION typeAbsence (p_idAbsence IN Absences.idAbsence%TYPE)
                                                    RETURN VARCHAR
```

- Résultats attendus :

```
SELECT typeAbsence('A92')
FROM DUAL ;

typeAbsence('A92')
------------------
E

SELECT typeAbsence('A96')
FROM DUAL ;

typeAbsence('A96')
------------------
E

SELECT typeAbsence('A100')
FROM DUAL ;

typeAbsence('A100')
-------------------
A
```

19) Fonction stockée nbAbsencesNonJustifiees

- Ecrire une fonction stockée `nbAbsencesNonJustifiees` qui retourne le nombre d'absences non justifiées de l'étudiant `p_idEtudiant` pour le semestre `p_idSemestre`. Cette fonction doit avoir la signature suivante :

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

- Résultats attendus :

```
SELECT nbAbsencesNonJustifiees('E22', 'S1')
FROM DUAL ;

nbAbsencesNonJustifiees('E22', 'S1')
------------------------------------
3

SELECT nbAbsencesNonJustifiees('E11', 'S2')
FROM DUAL ;

nbAbsencesNonJustifiees('E11', 'S2')
------------------------------------
5
```

20) Fonction stockée moyenneEtudiantSemestreAvecAbs.

- Ecrire une fonction stockée `moyenneEtudiantSemestreAvecAbs` qui retourne la véritable moyenne qu'a obtenue un étudiant `p_idEtudiant` au semestre `p_idSemestre` après avoir retiré de la moyenne générale de l'étudiant les points relatifs aux absences non justifiées.

  Si l'étudiant n'a pas d'absence ou bien s'il n'a qu'une seule absence non justifiée dans le semestre, on ne lui retranche pas de point (sa moyenne avec les absences est identique à la moyenne sans les absences). Sinon, on retire à la moyenne 0,1 point par absence non justifiée. Cette fonction doit avoir la signature suivante :

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

- Résultats attendus :

```
SELECT moyenneEtudiantSemestreAvecAbs('E12', 'S1')
FROM DUAL ;

moyenneEtudiantSemestreAvecAbs('E12', 'S1')
-------------------------------------------
9

SELECT moyenneEtudiantSemestreAvecAbs('E11', 'S2')
FROM DUAL ;

moyenneEtudiantSemestreAvecAbs('E11', 'S2')
-------------------------------------------
4.9
```

21) Procédure stockée affichageAbsEtudiantSemestre.

- Ecrire une procédure stockée `affichageAbsEtudiantSemestre` qui affiche le nombre d'absences non justifiées ainsi que la véritable moyenne générale (absences comprises) de l'étudiant `p_idEtudiant` au cours du semestre `p_idSemestre`. Cette procédure doit avoir la signature suivante :

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

- Résultats attendus :

```
CALL affichageAbsEtudiantSemestre('E13', 'S2') ;

Nombre d'absences non justifiées : 5
Moyenne avec les absences : 7.7
```

C. Gestion du résultat et du classement

22) 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 (absences comprises) 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')
---------------------------
N


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

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

23) 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, absences comprises. 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')
---------------------------------------
6
```

24) 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 : O
Classement : 1
```

25) 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 : 09/03/98
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 sans les absences : 18
Nombre d'absences non justifiées : 0
Moyenne avec les absences : 18

Resultat : O
Classement : 1


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

Identifiant étudiant : E9
Nom étudiant : Croque
Prénom étudiant : Odile
Sexe étudiant : F
Date naissance étudiant : 14/07/00
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 sans les absences : 11
Nombre d'absences non justifiées : 1
Moyenne avec les absences : 11

Resultat : O
Classement : 9
```

D. Gestion des absences par demi-journées

En réalité, sur le relevé de notes semestriel d'un étudiant, on ne compte pas le nombre d'absences non justifiées mais plutôt le nombre de demi-journées d'absence non justifiées. Ainsi, si un étudiant a deux absences non justifiées lors de la même demi-journée (par exemple, une de 8h à 10h et l'autre de 10h à 12h), cela ne lui comptera qu'une seule demi-journée d'absence. Toutefois, si dans la même journée, il a été absent de 8h à 10h puis de 14h à 16h, on lui comptera deux demi-journées d'absence.

Nous allons donc légèrement modifier notre application afin que le relevé de notes affiche non pas le nombre d'absences non justifiées mais le nombre de demi-journées d'absence non justifiées. Pour cela nous allons commencer par écrire une fonction qui permettra de savoir si une absence a eu lieu le matin ou l'après-midi. Ensuite nous écrirons une fonction qui calcule le nombre de demi-journées d'absence d'un étudiant pour un semestre donné. Enfin, nous modifierons légèrement l'application pour prendre en compte les fonctions nouvellement ajoutées.

26) Fonction stockée momentAbsence

- Ecrire une fonction stockée `momentAbsence` qui retourne `'MATIN'` si l'absence `p_idAbsence` concerne un cours qui a commencé avant midi. Si par contre, le cours a commencé après midi, la fonction doit retourner `'APREM'`. Cette fonction doit avoir la signature suivante :

```
FUNCTION momentAbsence(p_idAbsence IN Absences.idAbsence%TYPE)
                                                    RETURN VARCHAR
```

  Sous Oracle pour extraire l'heure d'un attribut de type `DATE`, il est possible d'utiliser la fonction `TO_CHAR`. Par exemple, `TO_CHAR(maDate, 'HH24').`

- Résultats attendus :

```
SELECT momentAbsence('A1')
FROM DUAL ;

momentAbsence('A1')
--------------------
MATIN


SELECT momentAbsence('A3')
FROM DUAL ;

momentAbsence('A3')
--------------------
APREM
```

27) Fonction stockée nbDemiJourneesNonJustifiees

- Ecrire une fonction stockée `nbDemiJourneesNonJustifiees` qui retourne le nombre de demi-journées d'absence non justifiées de l'étudiant `p_idEtudiant` pour le semestre `p_idSemestre`. Cette fonction doit avoir la signature suivante :

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

  Sous Oracle pour extraire uniquement la date (sans l'heure) d'un attribut de type `DATE`, il est possible d'utiliser la fonction `TO_CHAR`. Par exemple, `TO_CHAR(maDate, 'DD/MON/YYYY')`

- Résultats attendus :

```
SELECT nbDemiJourneesNonJustifiees('E14', 'S3')
FROM DUAL ;

nbDemiJourneesNonJustifiees('E14', 'S3')
----------------------------------------
1
```

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

nbDemiJourneesNonJustifiees('E1', 'S1')
---------------------------------------
4

SELECT nbDemiJourneesNonJustifiees('E5', 'S3')
FROM DUAL ;

nbDemiJourneesNonJustifiees('E5', 'S3')
---------------------------------------
3

SELECT nbDemiJourneesNonJustifiees('E18', 'S1')
FROM DUAL ;

nbDemiJourneesNonJustifiees('E18', 'S1')
----------------------------------------
12
```

28) Fonction stockée moyenneEtudiantSemestreAvecAbs.

- Modifier la méthode `moyenneEtudiantSemestreAvecAbs` afin que la véritable moyenne (absences comprises) ne soit pas calculée en retranchant 0,1 point par absence non justifiée, mais 0,1 point par demi-journée d'absence non justifiée.
- Résultats attendus :

```
SELECT moyenneEtudiantSemestreAvecAbs('E5', 'S3')
FROM DUAL ;

moyenneEtudiantSemestreAvecAbs('E5', 'S3')
------------------------------------------
8.1


SELECT moyenneEtudiantSemestreAvecAbs('E18', 'S1')
FROM DUAL ;

moyenneEtudiantSemestreAvecAbs('E18', 'S2')
-------------------------------------------
9.4
```

29) Procédure stockée affichageAbsEtudiantSemestre.

- Modifier la procédure stockée `affichageAbsEtudiantSemestre` pour que soit affiché non pas le nombre d'absences non justifiées mais le nombre de demi-journées d'absence non justifiées.
- Résultat attendu :

```
CALL affichageAbsEtudiantSemestre('E1', 'S2') ;

Nombre de demi-journées d'absence non justifiées : 8
Moyenne avec les absences : 6.8

CALL affichageAbsEtudiantSemestre('E17', 'S2') ;

Nombre de demi-journées d'absence non justifiées : 1
Moyenne avec les absences : 11.6
```

30) Procédure stockée affichageReleveNotes.

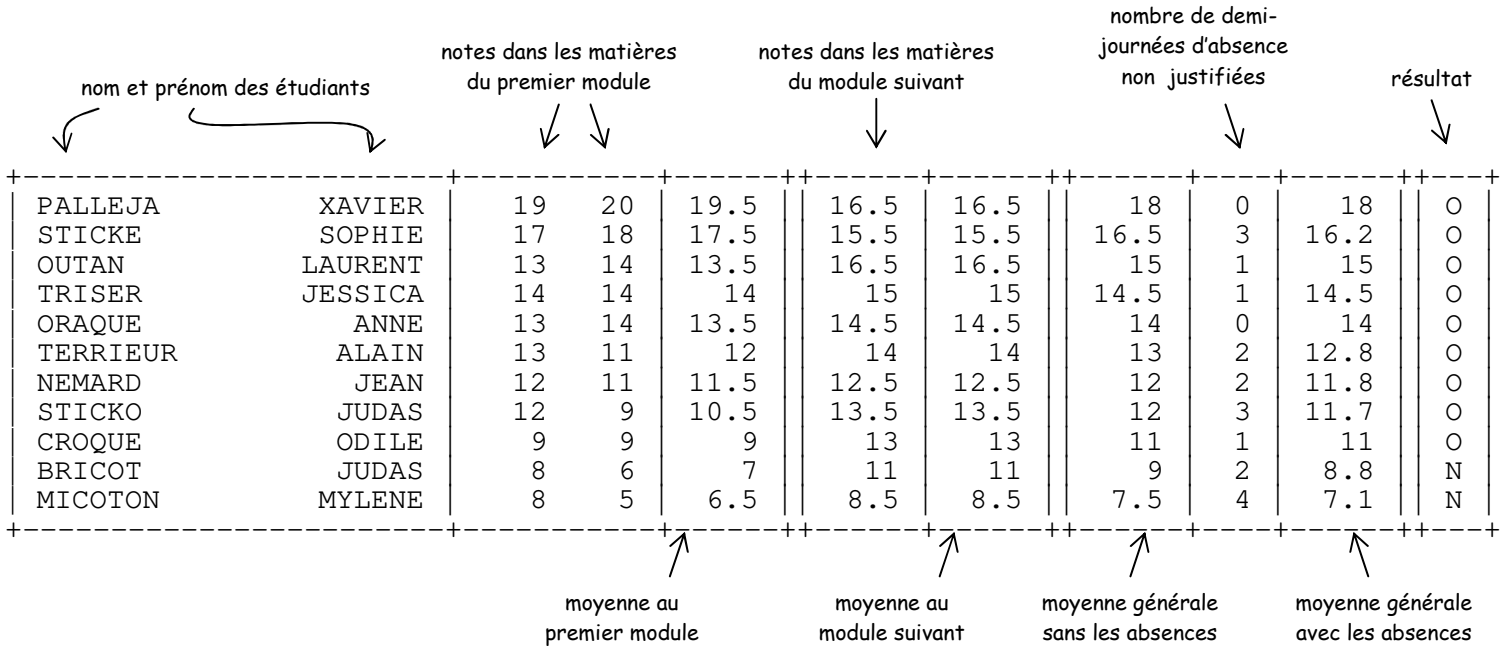
- Vérifier que la procédure `affichageReleveNotes` retourne les résultats attendus sans qu'il soit utile de la modifier.

- Résultats attendus :

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

Identifiant étudiant : E10
Nom étudiant : Palleja
Prénom étudiant : Xavier
Sexe étudiant : M
Date naissance étudiant : 09/03/98
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 sans les absences : 18
Nombre de demi-journées d'absence non justifiées : 0
Moyenne avec les absences : 18

Resultat : O
Classement : 1


CALL affichageReleveNotes('E12', 'S2') ;

Identifiant étudiant : E12
Nom étudiant : Vesselle
Prénom étudiant : Aude
Sexe étudiant : F
Date naissance étudiant : 31/10/01
Groupe étudiant : Q2

Prog Objet : 10
BD Avancées : 10
Moyenne module Informatique : 10

Statistiques : 8
Anglais : 6
Moyenne module Culture Générale : 7

Gestion de Projet Agile : 10
Moyenne module Mise en Pratique : 10

Moyenne semestre sans les absences : 8.8
Nombre de demi-journées d'absence non justifiées : 3
Moyenne avec les absences : 8.5

Resultat : N
Classement : 13


CALL affichageReleveNotes('E17', 'S2') ;

Identifiant étudiant : E17
Nom étudiant : Tarembois
Prénom étudiant : Guy
Sexe étudiant : M
Date naissance étudiant : 08/04/01
Groupe étudiant : Q2

Prog Objet : 14
BD Avancées : 13
Moyenne module Informatique : 13.5

Statistiques : 12
Anglais : 12
Moyenne module Culture Générale : 12

Gestion de Projet Agile : 7
Moyenne module Mise en Pratique : 7

Moyenne semestre sans les absences : 11.6
Nombre de demi-journées d'absence non justifiées : 1
Moyenne avec les absences : 11.6

Resultat : N
Classement : 8
```

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 – sans et avec les absences), le nombre de demi-journées d'absence qui leur ont été comptabilisées et le résultat.
Par exemple, pour le semestre S3 de la promotion A2, on souhaite éditer le PV suivant :



PALLEJA	XAVIER	19	20	19.5	16.5	16.5	18	0	18	O
STICKE	SOPHIE	17	18	17.5	15.5	15.5	16.5	3	16.2	O
OUTAN	LAURENT	13	14	13.5	16.5	16.5	15	1	15	O
TRISER	JESSICA	14	14	14	15	15	14.5	1	14.5	O
ORAQUE	ANNE	13	14	13.5	14.5	14.5	14	0	14	O
TERRIEUR	ALAIN	13	11	12	14	14	13	2	12.8	O
NEMARD	JEAN	12	11	11.5	12.5	12.5	12	2	11.8	O
STICKO	JUDAS	12	9	10.5	13.5	13.5	12	3	11.7	O
CROQUE	ODILE	9	9	9	13	13	11	1	11	O
BRICOT	JUDAS	8	6	7	11	11	9	2	8.8	N
MICOTON	MYLENE	8	5	6.5	8.5	8.5	7.5	4	7.1	N

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 avec les absences. 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.

31) 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 qui 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 `<BR>` :
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	1	16.4	O
TIMETTRE	VINCENT	16	17	16.5	13	12	14	13	15.5	14	15	14.8	2	14.6	O
KAECOUTE	XAVIER	10	18	14	14	14	17	15	16.5	15	16	14.8	3	14.5	O
NANAS	JUDAS	14	14	14	12	9	12	11	13	14.5	13.5	12.7	3	12.4	O
GATOR	ALI	8	14	11	14	12	13	13	14	12.5	13.5	12.3	0	12.3	O
ZETOFRAIS	MELANIE	9	10	9.5	14	13	12	13	15	15	15	12	2	11.8	O
ALIZAN	GASPARD	12	11	11.5	9	11	13	11	14	14	14	11.8	4	11.4	O
TERRIEUR	ALEX	13	12	12.5	10	9	8	9	13	11.5	12.5	11.1	0	11.1	O
BONO	JEAN	6	13	9.5	9	10	14	11	13	13	13	10.8	1	10.8	O
JAVEL	AUDE	13	12	12.5	10	9	8	9	11	14	12	11	3	10.7	O
TAREMBOIS	GUY	12	13	12.5	10	9	17	12	8	6.5	7.5	11.3	1	11.3	N
ASSIN	MARC	13	14	13.5	9	5	8.5	7.5	12	15	13	11	3	10.7	N
OUZY	JACQUES	8	11	9.5	10	11	9	10	12	18	14	10.6	12	9.4	N
VESSELLE	AUDE	9	11	10	9	7	5	7	11	14	12	9.2	2	9	N
ZEBLOUSE	AGATHE	2	1	1.5	15	13	11	13	15	15	15	8.8	1	8.8	N
NIOHRANGIN	NICOLAS	1	9	5	9	11	7	9	13	8.5	11.5	7.9	4	7.5	N

```
CALL affichagePV('S2');
```

KAECOUTE	XAVIER	16	14	15	17	18	17.5	17	17	16.4	1	16.4	O
DEUF	JOHN	14	14	14	15	15	15	14	14	14.4	2	14.2	O
TIMETTRE	VINCENT	16	18	17	11	12	11.5	14	14	14.2	3	13.9	O
GATOR	ALI	14	14	14	10	11	10.5	13	13	12.4	1	12.4	O
NANAS	JUDAS	13	12	12.5	10	14	12	14	14	12.6	3	12.3	O
TERRIEUR	ALEX	12	12	12	11	13	12	12	12	12	0	12	O
JAVEL	AUDE	12	12	12	11	13	12	12	12	12	1	12	O
OUZY	JACQUES	13	12	12.5	11	14	12.5	12	12	12.4	8	11.6	O
ZETOFRAIS	MELANIE	9	9	9	15	14	14.5	12	12	11.8	2	11.6	O
BONO	JEAN	12	11	11.5	10	13	11.5	12	12	11.6	3	11.3	O
ASSIN	MARC	12	11	11.5	10	9	9.5	15	15	11.4	3	11.1	O
TAREMBOIS	GUY	14	13	13.5	12	12	12	7	7	11.6	1	11.6	N
VESSELLE	AUDE	10	10	10	8	6	7	10	10	8.8	3	8.5	N
NIOHRANGIN	NICOLAS	5	4	4.5	10	11	10.5	11	11	8.2	5	7.7	N
ALIZAN	GASPARD	9	6	7.5	10	9	9.5	4	4	7.6	8	6.8	N
ZEBLOUSE	AGATHE	1	1	1	10	10	10	5	5	5.4	5	4.9	N

```
CALL affichagePV('S3');
```

PALLEJA	XAVIER	18	17	17.5	14	16	18	16	16	16	16.6	0	16.6	O
OUTAN	LAURENT	13	17	15	14	17	11	14	14	14	14.4	1	14.4	O
TRISER	JESSICA	14	16	15	14	13	12	13	15	15	14.2	1	14.2	O
TERRIEUR	ALAIN	13	14	13.5	12	12	10.5	11.5	15	15	13	1	13	O
ORAQUE	ANNE	12	13	12.5	14	13	12	13	14	14	13	0	13	O
NEMARD	JEAN	12	10	11	11	12	8.5	10.5	14	14	11.4	2	11.2	O
CROQUE	ODILE	12	14	13	9	10	8	9	12	12	11.2	7	10.5	O
STICKE	SOPHIE	17	19	18	8	9	4	7	14	14	12.8	5	12.3	N
STICKO	JUDAS	10	5	7.5	14	16	6	12	12	12	10.2	2	10	N
MICOTON	MYLENE	9	8	8.5	11	10	4.5	8.5	10	10	8.8	7	8.1	N
BRICOT	JUDAS	8	6	7	13	12	.5	8.5	11	11	8.4	3	8.1	N

```
CALL affichagePV('S4');
```

PALLEJA	XAVIER	19	20	19.5	16.5	16.5	18	0	18	O
STICKE	SOPHIE	17	18	17.5	15.5	15.5	16.5	3	16.2	O
OUTAN	LAURENT	13	14	13.5	16.5	16.5	15	1	15	O
TRISER	JESSICA	14	14	14	15	15	14.5	1	14.5	O
ORAQUE	ANNE	13	14	13.5	14.5	14.5	14	0	14	O
TERRIEUR	ALAIN	13	11	12	14	14	13	2	12.8	O
NEMARD	JEAN	12	11	11.5	12.5	12.5	12	2	11.8	O
STICKO	JUDAS	12	9	10.5	13.5	13.5	12	3	11.7	O
CROQUE	ODILE	9	9	9	13	13	11	1	11	O
BRICOT	JUDAS	8	6	7	11	11	9	2	8.8	N
MICOTON	MYLENE	8	5	6.5	8.5	8.5	7.5	4	7.1	N