# PLSQL

## TD2

### [BDD]

Premiere Partie

**Q1)**

DECLARE

v\_nbEtuQ1 NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_nbEtuQ1

FROM ETUDIANTS

WHERE IDGROUPE=‘Q1;

DBMS\_OUTPUT.PUT\_LINE('Il y a ' || v\_nbEtuQ1 || ' étudiants dans le groupe Q1');

END;

**Q2)**

**DECLARE**

V\_nbEtudiants NUMBER;

v\_idGroupe Groupes.idGroupe%TYPE := ‘Q1’;

**BEGIN**

SELECT COUNT(\*) INTO v\_nbEtudiants

FROM Etudiants

WHERE idGroupe = v\_idGroupe;

DBMS\_OUTPUT.PUT\_LINE(‘Il y a ‘ || v\_nbEtudiants || ‘ étudiant (s) dans le groupe ‘ || v\_idGroupe);

**END;**

**Q3)**

**DECLARE**

V\_nbEtudiants NUMBER;

v\_idGroupe Groupes.idGroupe%TYPE;

v\_idPromotion Groupes.idPromotion%TYPE;

**BEGIN**

v\_idGroupe := ‘Q1’;

SELECT idPromotion INTO v\_idPromotion

FROM Groupes

WHERE idGroupe = v\_idGroupe;

SELECT COUNT(\*) INTO v\_nbEtudiants

FROM Etudiants

WHERE idGroupe = v\_idGroupe;

DBMS\_OUTPUT.PUT\_LINE(‘Il y a ‘ || v\_nbEtudiants || ‘ étudiant (s) dans le groupe ‘ || v\_idGroupe);

**EXCEPTION**

WHEN **NO\_DATA\_FOUND** THEN

DBMS\_OUTPUT.PUT\_LINE(‘Il n’’y a pas de groupe ‘ || v\_idGroupe);

**END;**

**Q4)**

**DECLARE**

V\_nbEtudiants NUMBER;

v\_idGroupe Groupes.idGroupe%TYPE;

v\_nbGroupes NUMBER;

**BEGIN**

v\_idGroupe := ‘Q1’;

SELECT COUNT(\*) INTO v\_idGroupes

FROM Groupes

WHERE idGroupe = v\_idGroupe;

**IF** v\_nbGroupes = 1 THEN

SELECT COUNT(\*) INTO v\_nbEtudiants

FROM Etudiants

WHERE idGroupe = v\_idGroupe;

DBMS\_OUTPUT.PUT\_LINE(‘Il y a ‘ || v\_nbEtudiants || ‘ étudiant (s) dans le groupe ‘ || v\_idGroupe);

**ELSE**

DBMS\_OUTPUT.PUT\_LINE(‘Il n’’y a pas de groupe ‘ || v\_idGroupe);

**END IF;**

**END;**

**Q5)**

**DECLARE**

v\_idEtudiant Etudiants.idEtudiant%TYPE;

rty\_etudiant **Etudiants%ROWTYPE**;

**BEGIN**

DBMS\_OUTPUT.ENABLE;

v\_idEtudiant := ‘E1’;

SELECT \* INTO rty\_etudiant

FROM Etudiants

WHERE idEtudiant = v\_idEtudiant;

DBMS\_OUTPUT.PUT\_LINE(‘Identifiant étudiant : ‘ || rty\_etudiant.idEtudiant);

DBMS\_OUTPUT.PUT\_LINE(‘Nom étudiant : ‘ || rty\_etudiant.nomEtudiant);

DBMS\_OUTPUT.PUT\_LINE(‘Prénom étudiant : ‘ || rty\_etudiant.prenomEtudiant);

DBMS\_OUTPUT.PUT\_LINE(‘Sexe étudiant : ‘ || rty\_etudiant.sexeEtudiant);

DBMS\_OUTPUT.PUT\_LINE(‘Date naissance étudiant : ‘ || rty\_etudiant.dateNaissanceEtudiant);

DBMS\_OUTPUT.PUT\_LINE(‘Groupe étudiant : ‘ || rty\_etudiant.idGroupe);

**EXCEPTION**

WHEN **NO\_DATA\_FOUN** THEN

DBMS\_OUTPUT.PUT\_LINE(‘L’’étudiant : ‘ || v\_idEtudiant || ‘ n’’existe pas’);

**END;**

**Q6)**

CREATE OR REPLACE FUNCTION *nbEtudiantsParGroupe*(p\_idGroupe IN Groupes.idGroupe%TYPE) RETURN NUMBER IS

V\_nbEtudiants NUMBER;

v\_idPromotion Groupes.idPromotion%TYPE;

BEGIN

SELECT idPromotion INTO v\_idPromotion

FROM Groupes

WHERE idGroupe = p\_idGroupe;

SELECT *COUNT*(\*) INTO v\_nbEtudiants

FROM Etudiants

WHERE idGroupe = p\_idGroupe;

RETURN V\_nbEtudiants;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RETURN NULL;

END;

**Q7)**

CREATE OR REPLACE FUNCTION *nbEtudiantsParPromotion*(p\_idPromotion IN Promotions.idPromotion%TYPE) RETURN NUMBER IS

v\_nbEtu NUMBER;

BEGIN

SELECT *COUNT*(\*) INTO v\_nbEtu

FROM ETUDIANTS JOIN GROUPES G on ETUDIANTS.IDGROUPE = G.IDGROUPE

WHERE G.IDPROMOTION = p\_idPromotion;

RETURN v\_nbEtu;

end;

**Q8)**

CREATE OR REPLACE PROCEDURE *affichageInfosEtudiant*(p\_idEtudiant IN Etudiants.idEtudiant%TYPE) IS

rty\_etudiant Etudiants%ROWTYPE;

BEGIN

DBMS\_OUTPUT.*ENABLE*;

SELECT \* INTO rty\_etudiant

FROM Etudiants

WHERE idEtudiant = p\_idEtudiant;

DBMS\_OUTPUT.*PUT\_LINE*('Identifiant étudiant : ' || rty\_etudiant.idEtudiant);

DBMS\_OUTPUT.*PUT\_LINE*('Nom étudiant : ' || rty\_etudiant.nomEtudiant);

DBMS\_OUTPUT.*PUT\_LINE*('Prénom étudiant : ' || rty\_etudiant.prenomEtudiant);

DBMS\_OUTPUT.*PUT\_LINE*('Sexe étudiant : ' || rty\_etudiant.sexeEtudiant);

DBMS\_OUTPUT.*PUT\_LINE*('Date naissance étudiant : ' || rty\_etudiant.dateNaissanceEtudiant);

DBMS\_OUTPUT.*PUT\_LINE*('Groupe étudiant : ' || rty\_etudiant.idGroupe);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.*PUT\_LINE*('L''étudiant : ' || p\_idEtudiant || ' n''existe pas');

END;

**Q9)**

CREATE OR REPLACE PROCEDURE *miseAJourCoefficientModules* IS

BEGIN

UPDATE MODULES m SET m.COEFFICIENTMODULE = (SELECT *SUM*(COEFFICIENTMATIERE) FROM MATIERES WHERE MATIERES.IDMODULE=m.IDMODULE) WHERE IDMODULE = m.IDMODULE;

end;

**Q10)**

CREATE OR REPLACE PROCEDURE *affichageNotesEtudiant*(p\_idEtudiant IN ETUDIANTS.idEtudiant%TYPE) IS

CURSOR curs\_etudiant (v\_idEtudiants Etudiants.idEtudiant%TYPE) IS SELECT \* FROM NOTES WHERE IDETUDIANT=v\_idEtudiants;

v\_nomMatiere MATIERES.NOMMATIERE%TYPE;

BEGIN

FOR v\_ligne IN curs\_etudiant(p\_idEtudiant) LOOP

SELECT NOMMATIERE INTO v\_nomMatiere FROM MATIERES WHERE IDMATIERE=v\_ligne.IDMATIERE;

DBMS\_OUTPUT.*PUT\_LINE*(v\_nomMatiere || ' : ' || v\_ligne.NOTE);

end loop;

END;

**Q11)**

CREATE OR REPLACE PROCEDURE *affichageNotesEtudiantSemestre*(p\_idEtudiant IN Etudiants.idEtudiant%TYPE, p\_idSemestre IN Semestres.idSemestre%TYPE) IS

CURSOR curs\_etudiant (v\_idEtudiants Etudiants.idEtudiant%TYPE) IS SELECT \* FROM NOTES WHERE IDETUDIANT=v\_idEtudiants;

v\_nomMatiere MATIERES.NOMMATIERE%TYPE;

v\_idSemestre MODULES.IDSEMESTRE%TYPE;

BEGIN

FOR v\_ligne IN curs\_etudiant(p\_idEtudiant) LOOP

SELECT IDSEMESTRE INTO v\_idSemestre

FROM MATIERES M JOIN MODULES M2 on M2.IDMODULE = M.IDMODULE

WHERE IDMATIERE = v\_ligne.IDMATIERE;

IF v\_idSemestre = p\_idSemestre THEN

SELECT NOMMATIERE INTO v\_nomMatiere FROM MATIERES WHERE IDMATIERE=v\_ligne.IDMATIERE;

DBMS\_OUTPUT.*PUT\_LINE*(v\_nomMatiere || ' : ' || v\_ligne.NOTE);

end if;

end loop;

END;

**Q12)**

CREATE OR REPLACE PROCEDURE *affichageToutEtudiantSemestre*(p\_idEtudiant IN Etudiants.idEtudiant%TYPE, p\_idSemestre IN Semestres.idSemestre%TYPE) IS

BEGIN

*affichageInfosEtudiant*(p\_idEtudiant);

*affichageNotesEtudiantSemestre*(p\_idEtudiant, p\_idSemestre) ;

end;

**Q13)**

CREATE OR REPLACE PROCEDURE *affichageAbsencesParPromotion*(p\_idPromotion IN Promotions.idPromotion%TYPE) IS

CURSOR curs\_grp (v\_idPromo GROUPES.IDPROMOTION%TYPE) IS SELECT \* FROM GROUPES WHERE IDPROMOTION=v\_idPromo;

CURSOR curs\_etu (v\_idGroupes ETUDIANTS.IDGROUPE%TYPE) IS SELECT \* FROM ETUDIANTS WHERE IDGROUPE=v\_idGroupes ORDER BY ETUDIANTS.IDETUDIANT;

v\_nbAbs NUMBER;

BEGIN

FOR v\_grpLigne IN curs\_grp(p\_idPromotion) LOOP

DBMS\_OUTPUT.*PUT\_LINE*('Groupe: ' || v\_grpLigne.IDGROUPE);

FOR v\_etuLigne IN curs\_etu(v\_grpLigne.IDGROUPE) LOOP

SELECT *COUNT*(\*) INTO v\_nbAbs

FROM ABSENCES

WHERE ABSENCES.IDETUDIANT = v\_etuLigne.IDETUDIANT;

DBMS\_OUTPUT.*PUT\_LINE*(v\_etuLigne.NOMETUDIANT || ' ' || v\_etuLigne.PRENOMETUDIANT || ' à été absent ' || v\_nbAbs || ' fois');

end loop;

end loop;

end;

**Q14)**

CREATE OR REPLACE FUNCTION *moyenneEtudiantModule*(p\_idEtudiant IN Etudiants.idEtudiant%TYPE,p\_idModule IN Modules.idModule%TYPE) RETURN NUMBER IS

CURSOR curs\_etu(v\_idEtu ETUDIANTS.IDETUDIANT%TYPE) IS SELECT \* FROM NOTES WHERE IDETUDIANT=v\_idEtu;

v\_res NUMBER := 0.0;

v\_idMod MODULES.IDMODULE%TYPE;

v\_coef MATIERES.COEFFICIENTMATIERE%TYPE;

v\_coefCounter NUMBER := 0;

BEGIN

FOR v\_ligneEtu IN curs\_etu(p\_idEtudiant) LOOP

SELECT IDMODULE INTO v\_idMod

FROM MATIERES

WHERE MATIERES.IDMATIERE = v\_ligneEtu.IDMATIERE;

IF v\_idMod = p\_idModule THEN

SELECT COEFFICIENTMATIERE INTO v\_coef

FROM MATIERES

WHERE MATIERES.IDMATIERE = v\_ligneEtu.IDMATIERE;

v\_res := v\_res + (v\_coef\*v\_ligneEtu.NOTE);

v\_coefCounter := v\_coefCounter+v\_coef;

end if;

end loop;

v\_res := v\_res/v\_coefCounter;

RETURN v\_res;

END;

**Q15)**

CREATE OR REPLACE FUNCTION *valideEtudiantModule*(p\_idEtudiant IN Etudiants.idEtudiant%TYPE, p\_idModule IN Modules.idModule%TYPE) RETURN NUMBER IS

v\_Mod NUMBER;

BEGIN

v\_Mod := *MOYENNEETUDIANTMODULE*(p\_idEtudiant, p\_idModule);

IF v\_Mod > 8 THEN

RETURN 1;

ELSE

RETURN 0;

end if;

end;

**Q16)**

CREATE OR REPLACE FUNCTION moyenneEtudiantSemestreSansAbs(p\_idEtudiant IN Etudiants.idEtudiant%TYPE,p\_idSemestre IN SEMESTRES.idSemestre%TYPE) RETURN NUMBER IS

CURSOR curs\_module(v\_idModule MODULES.IDMODULE%TYPE) IS SELECT \* FROM MODULES WHERE idSemestre=p\_idSemestre;

v\_res DOUBLE PRECISION := 0;

v\_coefCounter NUMBER := 0;

BEGIN

FOR v\_index IN curs\_module(p\_idSemestre) LOOP

v\_res := v\_res + moyenneEtudiantModule(p\_idEtudiant, v\_index.IDMODULE)\*(v\_index.coefficientModule);

v\_coefCounter := v\_coefCounter + v\_index.coefficientModule;

END LOOP;

v\_res := v\_res/v\_coefCounter;

RETURN v\_res;

END;

**Q17)**