



# **ASGBD**

## **Rapport du TP N°4: « PL/SQL »**

**CHIKH Khadidja**

Master 1 SII  
Groupe 4

01/11/2018

**1) Ecrire un code PLSql qui permet d'afficher pour chaque chambre de chaque service le nombre de lits libres et occupés:**

La requête suivante permet d'afficher le nombre de lits occupés et celui des lits libres pour chaque chambre de n'importe quel service

```
SQL> select h.code_service,h.num_chambre,count(distinct(lit)) as lits_occupes,c.nb_lits
  from dbahopital.hospitalisation h,dbahopital.chambre c where h.code_service=c.code_ser
vice and h.num_chambre=c.num_chambre group by h.code_service,h.num_chambre,c.nb_lits or
der by h.code_service;
```

COD	NUM_CHAMBRE	LITS_OCCUPES	NB_LITS
CAR	106	1	1
CAR	102	2	2
CAR	105	1	2
CAR	104	2	3
CAR	101	2	3
CAR	103	1	1
CHG	405	1	1
CHG	202	3	4
CHG	401	2	4
CHG	301	2	2
CHG	404	1	2

COD	NUM_CHAMBRE	LITS_OCCUPES	NB_LITS
CHG	302	2	2
CHG	303	1	1
CHG	201	3	4
CHG	402	3	4
CHG	403	2	2
REA	102	1	1
REA	103	2	2
REA	101	1	1
REA	104	2	2
REA	107	2	2
REA	105	1	1

COD	NUM_CHAMBRE	LITS_OCCUPES	NB_LITS
REA	108	1	2

23 rows selected.

Nous utilisons alors cette requête pour écrire le code PLSql voulu: A chaque fois, le nombre de lits occupés est soustrait de celui du total des lits pour obtenir le nombre des lits libres ;  
Le code PLSql ainsi que son resultat sont montrés dans l'illustration suivante.

```

SQL> DECLARE
  2  cursor cr is select h.code_service,h.num_chambre,count(distinct(lit)) as lits_occu
pes,c.nb_lits from dbahopital.hospitalisation h,dbahopital.chambre c where h.code_servi
ce=c.code_service and h.num_chambre=c.num_chambre group by h.code_service,h.num_chambre
,c.nb_lits order by h.code_service;
  3  c_rec cr%rowtype;
  4  diff integer;
  5  begin
  6  for c_rec in cr loop
  7  diff:=c_rec.nb_lits-c_rec.lits_occupes;
  8  dbms_output.put_line('La chambre n° ' || c_rec.num_chambre || ' possede ' || c_rec
.lits_occupes || ' lit(s) occupe(s) et ' || diff || ' lit(s) libre(s)');
  9  exit when cr%notfound;
 10  end loop;
 11  end;
 12  /
La chambre n° 106 possede 1 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 102 possede 2 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 105 possede 1 lit(s) occupe(s) et 1 lit(s) libre(s)
La chambre n° 104 possede 2 lit(s) occupe(s) et 1 lit(s) libre(s)
La chambre n° 101 possede 2 lit(s) occupe(s) et 1 lit(s) libre(s)
La chambre n° 103 possede 1 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 405 possede 1 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 202 possede 3 lit(s) occupe(s) et 1 lit(s) libre(s)
La chambre n° 401 possede 2 lit(s) occupe(s) et 2 lit(s) libre(s)
La chambre n° 301 possede 2 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 404 possede 1 lit(s) occupe(s) et 1 lit(s) libre(s)
La chambre n° 302 possede 2 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 303 possede 1 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 201 possede 3 lit(s) occupe(s) et 1 lit(s) libre(s)
La chambre n° 402 possede 3 lit(s) occupe(s) et 1 lit(s) libre(s)
La chambre n° 403 possede 2 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 102 possede 1 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 103 possede 2 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 101 possede 1 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 104 possede 2 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 107 possede 2 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 105 possede 1 lit(s) occupe(s) et 0 lit(s) libre(s)
La chambre n° 108 possede 1 lit(s) occupe(s) et 1 lit(s) libre(s)

PL/SQL procedure successfully completed.

```

**2) Ecrire une fonction qui augmente le salaire de chaque infirmier et affiche pour chaque infirmier le nouveau salaire :**

La requête suivante permet d'afficher les informations (nécessaires) de chaque infirmier

```
SQL> select i.num_inf,e.nom_emp,e.prenom_emp,i.rotation,i.salaire from dbahopital.infirmier i,dbahopital.employe e where i.num_inf=e.num_emp;
```

NUM_INF	NOM_EMP	PRENOM_EMP	ROTA	SALAIRE
12	HADJ	Zouhir	jour	18842
15	OUSSEDIK	Hakim	jour	17670
22	ABAD	Abdelhamid	jour	22470
25	ABAYAHIA	Abdelkader	jour	23612
29	ABBOU	Mohamed	jour	20373
45	ABDELOUAHAB	OUAHIBA	jour	21980
49	ABDEMEZIANE	Madjid	jour	18849
57	ACHAIBOU	Rachid	jour	26481
71	AGGOUN	Khadidja	nuit	21373
73	AISSAT	Salima	nuit	23581
86	BABACI	Mourad	jour	17678

NUM_INF	NOM_EMP	PRENOM_EMP	ROTA	SALAIRE
95	BADI	Hatem	nuit	31154
97	BAKIR	ADEL	nuit	18944
98	BALI	Malika	jour	22476
116	BELABES	Abdelkader	jour	23621
127	BELHAMIDI	Mustapha	nuit	20251
130	BELKACEMI	Hocine	jour	20322
131	BELKOUT	Tayeb	jour	21983
139	FERAOUN	Houria	nuit	32600
151	CHAKER	Nadia	jour	26528
155	IGOUDJIL	Redouane	nuit	21338
162	GHEZALI	Lakhdar	nuit	22146

NUM_INF	NOM_EMP	PRENOM_EMP	ROTA	SALAIRE
163	KOULA	Brahim	nuit	23581
169	BELAID	Layachi	nuit	20717
176	CHALABI	Mourad	nuit	19494
189	SAIDOUNI	Wafa	nuit	21227
194	Yalaoui	Lamia	nuit	35256
195	AYATA	Samia	nuit	19810

28 rows selected.

Nous utilisons alors cette requête dans un curseur dans la fonction pour pouvoir parcourir tous les tuples à la fois et effectuer les modifications pour chacun, selon la rotation trouvée :

```

SQL> create or replace procedure aug_sal_inf is
  2  cursor cr is select i.num_inf,e.nom_emp,e.prenom_emp,i.rotation,i.salaire from dbah
opital.infirmier i,dbahopital.employe e where i.num_inf=e.num_emp;
  3  c cr%rowtype;
  4  ans number;
  5  ns number;
  6  vide exception;
  7  begin
  8  for c in cr loop
  9  ans:=c.salaire;
 10  if (c.rotation='jour') then
 11  update dbahopital.infirmier set salaire=salaire+salaire/2 where num_inf=c.num_inf;
 12  ns:=c.salaire+(c.salaire)/2;
 13  else update dbahopital.infirmier set salaire=salaire+salaire*3/5 where num_inf=c.nu
m_inf;
 14  ns:=c.salaire+(c.salaire)*3/5;
 15  end if;
 16  dbms_output.put_line('L infirmier n° '||c.num_inf||','||c.nom_emp || c.prenom_emp||
' de rotation '||c.rotation||' ,a comme ancien salaire '||ans||' et comme nouveau salair
e '|| ns);
 17  exit when cr%notfound;
 18  end loop;
 19  EXCEPTION when vide then dbms_output.put_line('Aucun infirmier existant!');
 20  end;
 21  /

Procedure created.

```

Nous cherchons d'abord la contrainte définie précédemment sur « Salaire » pour la désactiver

```

SQL> select constraint_name from all_constraints where owner='DBAHOPITAL' and table_name
='INFIRMIER';

CONSTRAINT_NAME
-----
CE2INFIRMIER
CHSALAIREINFIRMIER
SYS_C007034
CEINFIRMIER
CPINFIRMIER
CHINFIRMIER

6 rows selected.

SQL> alter table infirmier disable constraint chsalaireinfirmier;

Table altered.

```

En executant la procédure, nous obtenons le résultat suivant :

```

SQL> execute aug_sal_inf;
L infirmier n° 12,HADJ Zouhir de rotation jour ,a comme ancien salaire 12561 et
comme nouveau salaire 18841,5
L infirmier n° 22,ABAD Abdelhamid de rotation jour ,a comme ancien salaire
14980 et comme nouveau salaire 22470
L infirmier n° 49,ABDEMEZIANE Madjid de rotation jour ,a comme ancien salaire
12566 et comme nouveau salaire 18849
L infirmier n° 73,AISSAT Salima de rotation nuit ,a comme ancien salaire 14738
et comme nouveau salaire 23580,8
L infirmier n° 116,BELABES Abdelkader de rotation jour ,a comme ancien salaire
15747 et comme nouveau salaire 23620,5
L infirmier n° 127,BELHAMIDI Mustapha de rotation nuit ,a comme ancien salaire
12657 et comme nouveau salaire 20251,2
L infirmier n° 189,SAIDOUNI Wafa de rotation nuit ,a comme ancien salaire 13267
et comme nouveau salaire 21227,2
L infirmier n° 15,OUSSEDIK Hakim de rotation jour ,a comme ancien salaire 11780
et comme nouveau salaire 17670
L infirmier n° 25,ABAYAHIA Abdelkader de rotation jour ,a comme ancien salaire
15741 et comme nouveau salaire 23611,5
L infirmier n° 29,ABBOU Mohamed de rotation jour ,a comme ancien salaire 13582
et comme nouveau salaire 20373
L infirmier n° 45,ABDELOUAHAB OUAHIBA de rotation jour ,a comme ancien salaire
14653 et comme nouveau salaire 21979,5
L infirmier n° 57,ACHAIBOU Rachid de rotation jour ,a comme ancien salaire
17654 et comme nouveau salaire 26481
L infirmier n° 71,AGGOUN Khadidja de rotation nuit ,a comme ancien salaire
13358 et comme nouveau salaire 21372,8
L infirmier n° 86,BABACI Mourad de rotation jour ,a comme ancien salaire 11785
et comme nouveau salaire 17677,5
L infirmier n° 97,BAKIR ADEL de rotation nuit ,a comme ancien salaire 11840 et
comme nouveau salaire 18944
L infirmier n° 98,BALI Malika de rotation jour ,a comme ancien salaire 14984 et
comme nouveau salaire 22476
L infirmier n° 130,BELKACEMI Hocine de rotation jour ,a comme ancien salaire
13548 et comme nouveau salaire 20322
L infirmier n° 131,BELKOUT Tayeb de rotation jour ,a comme ancien salaire 14655
et comme nouveau salaire 21982,5
L infirmier n° 139,FERAOUN Houria de rotation nuit ,a comme ancien salaire
20375 et comme nouveau salaire 32600
L infirmier n° 151,CHAKER Nadia de rotation jour ,a comme ancien salaire 17685
et comme nouveau salaire 26527,5
L infirmier n° 155,IGOUDJIL Redouane de rotation nuit ,a comme ancien salaire
13336 et comme nouveau salaire 21337,6
L infirmier n° 162,GHEZALI Lakhdar de rotation nuit ,a comme ancien salaire
13841 et comme nouveau salaire 22145,6
L infirmier n° 163,KOULA Brahim de rotation nuit ,a comme ancien salaire 14738
et comme nouveau salaire 23580,8
L infirmier n° 169,BELAID Layachi de rotation nuit ,a comme ancien salaire
12948 et comme nouveau salaire 20716,8
L infirmier n° 176,CHALABI Mourad de rotation nuit ,a comme ancien salaire
12184 et comme nouveau salaire 19494,4
L infirmier n° 194,Yalaoui Lamia de rotation nuit ,a comme ancien salaire 22035
et comme nouveau salaire 35256
L infirmier n° 195,AYATA Samia de rotation nuit ,a comme ancien salaire 12381
et comme nouveau salaire 19809,6
L infirmier n° 95,BADI Hatem de rotation nuit ,a comme ancien salaire 19471 et
comme nouveau salaire 31153,6

```

PL/SQL procedure successfully completed.

3) Ecrire une fonction *Vérification (Salaire)* qui affiche « vérification positive » si le salaire de l'infirmier respecte la contrainte d'intégrité, et affiche « Vérification négative » sinon. Tester la procédure pour tous les infirmiers :

Pour cela nous écrivons d'abord une fonction *Vérification (Salaire)* qui retourne, par exemple : 1 si le salaire respecte les conditions de la contrainte et 0 sinon

Son test est montré sur l'illustration suivante :

```
SQL> create or replace function verification(salaire infirmier.salaire%type)
  2  return number is
  3  begin
  4  if(salaire>30000 or salaire<10000) then return 0;
  5  else return 1;
  6  end if;
  7  end;
  8  /
```

Function created.

```
SQL> select verification(900) from dual;
```

```
VERIFICATION(900)
-----
                0
```

Nous écrivons ensuite le code de la procédure qui appelle la fonction *Vérification* pour chaque tuple de Infirmier, et affiche un message selon son résultat

```
SQL> create or replace procedure verif_sal is
  2  cursor cr is select num_inf,salaire from infirmier order by num_inf;
  3  c cr%rowtype;
  4  vide exception;
  5  begin
  6  for c in cr loop
  7  if(verification(c.salaire)=1) then dbms_output.put_line('Infirmier n° '|| c.num_inf
|| ' salaire: '||c.salaire||' DA Verification: Positive');
  8  else dbms_output.put_line('Infirmier n° '|| c.num_inf||' salaire: '||c.salaire||' D
A Verification: Negative');
  9  end if;
 10  end loop;
 11  Exception when vide then dbms_output.put_line('Aucun infirmier enregistré!');
 12  end;
 13  /
```

Procedure created.

En executant la procédure *Verif\_sal* nous obtenons le résultat pour tous les infirmiers :



```

SQL> execute verif_sal;
Infirmier n° 12  salaire: 18842 DA Verification: Positive
Infirmier n° 15  salaire: 17670 DA Verification: Positive
Infirmier n° 22  salaire: 22470 DA Verification: Positive
Infirmier n° 25  salaire: 23612 DA Verification: Positive
Infirmier n° 29  salaire: 20373 DA Verification: Positive
Infirmier n° 45  salaire: 21980 DA Verification: Positive
Infirmier n° 49  salaire: 18849 DA Verification: Positive
Infirmier n° 57  salaire: 26481 DA Verification: Positive
Infirmier n° 71  salaire: 21373 DA Verification: Positive
Infirmier n° 73  salaire: 23581 DA Verification: Positive
Infirmier n° 86  salaire: 17678 DA Verification: Positive
Infirmier n° 95  salaire: 31154 DA Verification: Negative
Infirmier n° 97  salaire: 18944 DA Verification: Positive
Infirmier n° 98  salaire: 22476 DA Verification: Positive
Infirmier n° 116 salaire: 23621 DA Verification: Positive
Infirmier n° 127 salaire: 20251 DA Verification: Positive
Infirmier n° 130 salaire: 20322 DA Verification: Positive
Infirmier n° 131 salaire: 21983 DA Verification: Positive
Infirmier n° 139 salaire: 32600 DA Verification: Negative
Infirmier n° 151 salaire: 26528 DA Verification: Positive
Infirmier n° 155 salaire: 21338 DA Verification: Positive
Infirmier n° 162 salaire: 22146 DA Verification: Positive
Infirmier n° 163 salaire: 23581 DA Verification: Positive
Infirmier n° 169 salaire: 20717 DA Verification: Positive
Infirmier n° 176 salaire: 19494 DA Verification: Positive
Infirmier n° 189 salaire: 21227 DA Verification: Positive
Infirmier n° 194 salaire: 35256 DA Verification: Negative
Infirmier n° 195 salaire: 19810 DA Verification: Positive

PL/SQL procedure successfully completed.

```

4) Ecrire une fonction qui retourne, pour chaque spécialité donnée, le nombre de médecins affectés.  
Exécuter la fonction pour plusieurs spécialités :

Le code de la fonction :

```

SQL> create or replace function med_spec(s medecin.specialite%type) return number is
2  nbr number(3);
3  begin
4  select count(num_med) into nbr from medecin where specialite=s;
5  return nbr;
6  end;
7  /

Function created.

```

Le résultat de son exécution pour toutes les spécialités :



```

SQL> select med_spec('anesthesiste') from dual;

MED_SPEC('ANESTHESISTE')
-----
                        5

SQL> select med_spec('orthopediste') from dual;

MED_SPEC('ORTHOPEDISTE')
-----
                        5

SQL> select med_spec('cardiologue') from dual;

MED_SPEC('CARDIOLOGUE')
-----
                        8

SQL> select med_spec('traumatologue') from dual;

MED_SPEC('TRAUMATOLOGUE')
-----
                        5

SQL> select med_spec('pneumologue') from dual;

MED_SPEC('PNEUMOLOGUE')
-----
                        5

SQL> select med_spec('radiologue') from dual;

MED_SPEC('RADIOLOGUE')
-----
                        4

```

**5) Créer une procédure qui permet d'ajouter un employé de type infirmier à partir de tous les attributs nécessaires. Vérifier l'unicité de la clé et l'existence des clés étrangères vers EMPLOYE et SERVICE. Afficher les messages d'erreurs en cas de problèmes :**

La procedure devra effectuer la verification de l'existence de l'employé (la table Employe) en entrée, de son existence étant infirmier (la table Infirmier), et de l'existence du service en entrée (la table service). Nous utilisons des variables qui récupèrent le resultat (ici nombre de tuples) de chaque vérification, et selon leurs valeurs nous insérons les informations en entrée ou bien nous générons des execeptions qui affichent des messages d'erreurs,

Le code de la procedure est le suivant :

```

SQL> create or replace procedure ajout_emp_inf(ni infirmier.num_inf%type,ne employe.nom
_emp%type,pe employe.prenom_emp%type,cs infirmier.code_service%type) is
  2  i number(3);
  3  e number(3);
  4  s number(3);
  5  e1 exception;
  6  e2 exception;
  7  begin
  8  select count(*) into e from employe where num_emp=ni;
  9  select count(*) into i from infirmier where num_inf=ni;
10  select count(*) into s from service where code_service=cs;
11  if (e!=0) then
12  if (i!=0) then raise e1;
13  else
14  if (s=0) then raise e2;
15  else insert into infirmier(num_inf,code_service) values(ni,cs);
16  dbms_output.put_line('Enregistrement effectuee!');
17  end if;
18  end if;
19  else
20  if (s=0) then raise e2;
21  else
22  insert into employe(num_emp,nom_emp,prenom_emp) values(ni,ne,pe);
23  insert into infirmier(num_inf,code_service) values(ni,cs);
24  dbms_output.put_line('Enregistrement effectuee!');
25  end if;
26  end if;
27  Exception
28  when e1 then dbms_output.put_line('Infirmier deja enregistre!');
29  when e2 then dbms_output.put_line('Code de service errone!');
30  end;
31  /

```

Procedure created.

Nous testons alors la procédure *ajout\_emp\_inf()* et obtenons les résultats suivants :

```

SQL> execute ajout_emp_inf(200,'chikh','khadidja','chg');
Infirmier deja enregistre!

PL/SQL procedure successfully completed.

SQL> execute ajout_emp_inf(154,'chikh','khadidja','chg');
Code de service errone!

PL/SQL procedure successfully completed.

SQL> execute ajout_emp_inf(154,'chikh','khadidja','CHG');
Enregistrement effectuee!

PL/SQL procedure successfully completed.

```