

-- Génération automatique de classements aléatoires en production, pour les FNS --

=====

FUNCTION gen_class_alea_V1_relatif_grp(

o_g_ea_cod_ins IN VARCHAR2,

o_g_ti_cod IN NUMBER,

o_c_gp_cod IN NUMBER,

o_g_tg_cod IN NUMBER,

login IN VARCHAR2,

type_login IN NUMBER,

mode_dev IN NUMBER,

confirm IN NUMBER,

saio IN NUMBER,

nip IN VARCHAR2,

indic IN NUMBER,

mess_err OUT VARCHAR2,

mess_aff OUT VARCHAR2)

RETURN NUMBER IS

retour NUMBER;

X VARCHAR2(2);

dummy NUMBER;

dummy2 NUMBER;

l_c_gp_flg_sel c_grp.c_gp_flg_sel%TYPE;

l_g_tg_cod c_grp.g_tg_cod%TYPE;

l_c_gp_eta_cla c_grp.c_gp_eta_cla%TYPE;

l_g_flh_sel sp_g_tri_ins.g_flh_sel%TYPE;

```
l_g_fr_reg_for      g_for.g_fr_reg_for%TYPE;
l_g_ea_cod_ges      g_tri_ins.g_ea_cod_ges%TYPE;
l_c_ja_cod          c_jur_adm.c_ja_cod%TYPE;
l_c_tj_cod          c_jur_adm.c_tj_cod%TYPE;
```

```
l                      NUMBER;
IS_prod              NUMBER;
l_six_voe            NUMBER;
```

```
UNIQUE_CONSTRAINT EXCEPTION;
PRAGMA EXCEPTION_INIT (UNIQUE_CONSTRAINT, -00001);
```

```
--classement aléatoire sur voeu 1 groupé relatif
```

```
CURSOR classement_aleatoire_efe IS
```

```
-- on traite d'abord les candidats AEFE s'il y en a
```

```
SELECT c.g_cn_cod,
       a_ve_ord_vg_rel, -- Ordre du voeu avec voeux groupés relatifs
licence
       a_ve_ord_aff, -- Ordre du voeu avec voeux groupé relatif
licence et tous les autres voeux
       a_vg_ord, -- Ordre du sous-voeu dans le voeu
groupé
       DBMS_RANDOM.value(1, 999999),
       i.i_ep_cod
FROM g_can c, i_ins i, a_rec r, a_voe v
WHERE i.g_ti_cod=o_g_ti_cod
AND g_gf_cod=o_c_gp_cod
AND l.g_cn_cod=c.g_cn_cod
```

-- Génération automatique de classements aléatoires en production, pour les FNS --

=====

FUNCTION gen_class_alea_V1_relatif_grp(

o_g_ea_cod_ins IN VARCHAR2,

o_g_ti_cod IN NUMBER,

o_c_gp_cod IN NUMBER,

o_g_tg_cod IN NUMBER,

login IN VARCHAR2,

type_login IN NUMBER,

mode_dev IN NUMBER,

confirm IN NUMBER,

saio IN NUMBER,

nip IN VARCHAR2,

indic IN NUMBER,

mess_err OUT VARCHAR2,

mess_aff OUT VARCHAR2)

RETURN NUMBER IS

retour NUMBER;

X VARCHAR2(2);

dummy NUMBER;

dummy2 NUMBER;

l_c_gp_flg_sel c_grp.c_gp_flg_sel%TYPE;

l_g_tg_cod c_grp.g_tg_cod%TYPE;

l_c_gp_eta_cla c_grp.c_gp_eta_cla%TYPE;

l_g_flh_sel sp_g_tri_ins.g_flh_sel%TYPE;

```

l_g_fr_reg_for      g_for.g_fr_reg_for%TYPE;
l_g_ea_cod_ges      g_tri_ins.g_ea_cod_ges%TYPE;
l_c_ja_cod          c_jur_adm.c_ja_cod%TYPE;
l_c_tj_cod          c_jur_adm.c_tj_cod%TYPE;

```

```

l_s_prod            NUMBER;
l_six_voe          NUMBER;

```

```

UNIQUE_CONSTRAINT EXCEPTION;

```

```

PRAGMA EXCEPTION_INIT (UNIQUE_CONSTRAINT, -00001);

```

```

--classement aléatoire sur voeu 1 groupé relatif

```

```

CURSOR classement_aleatoire_efe IS

```

```

-- on traite d'abord les candidats AEFE s'il y en a

```

```

SELECT c.g_cn_cod,

```

```

        a_ve_ord_vg_rel,          -- Ordre du voeu avec voeux groupés relatifs
licence

```

```

        a_ve_ord_aff,            -- Ordre du voeu avec voeux groupé relatif
licence et tous les autres voeux

```

```

        a_vg_ord,                -- Ordre du sous-voeu dans le voeu
groupé

```

```

        DBMS_RANDOM.value(1, 999999),

```

```

        i.l_ep_cod

```

```

FROM g_can c, l_ins l, a_rec r, a_voe v

```

```

WHERE l.g_ti_cod=o.g_ti_cod

```

```

AND g_gf_cod=o.c_gp_cod

```

```

AND l.g_cn_cod=c.g_cn_cod

```

```

AND c.g_ic_cod > 0
AND NVL(g_cn_flg_aefe, 0)=1          -- Bac EFE
AND i_ep_cod IN (2, 3)              -- Pointés recu (complet ou incomplet)
AND i.i_is_val=1
-- non encore classé
AND NOT EXISTS (SELECT 1 FROM c_can_grp
                WHERE i.g_cn_cod=g_cn_cod
                AND i.g_gf_cod=c_gp_cod
                AND i_ip_cod IN (4, 5))          -- Permet de récupérer les AC

AND i.g_ti_cod=r.g_ti_cod
AND c.g_cn_cod=v.g_cn_cod
AND r.g_ta_cod=v.g_ta_cod

UNION

-- les candidats EFE qui n'ont au final pas classé la formation dans leur liste ordonnée. Ils sont classé,
mais en dernier.

SELECT c.g_cn_cod,
       0,
       0,
       0,
       DBMS_RANDOM.value(1, 999999),
       i.i_ep_cod
FROM g_can c, i_ins i, a_rec r
WHERE i.g_ti_cod=o.g_ti_cod
AND g_gf_cod=o.c_gp_cod
AND i.g_cn_cod=c.g_cn_cod
AND c.g_ic_cod > 0
AND NVL(g_cn_flg_aefe, 0)=1          -- Bac EFE
AND i_ep_cod IN (2, 3)              -- Pointés recu (complet ou incomplet)

```

```

AND i.i_is_val=1
-- non encore classé
AND NOT EXISTS (SELECT 1 FROM c_can_grp
                WHERE l.g_cn_cod=g_cn_cod
                AND i.g_gf_cod=c_gp_cod
                AND i_lp_cod IN (4, 5))           -- Permet de récupérer les AC
AND l.g_ti_cod=r.g_ti_cod
AND NOT EXISTS (SELECT 1 FROM a_voe v WHERE c.g_cn_cod=v.g_cn_cod AND
r.g_ta_cod=v.g_ta_cod)
ORDER BY 2, 3, 4, 5;

CURSOR class_aleatoire_autres_cddts IS
-- les candidats non classés par la requête ci-dessus : les autre bac que EEE
SELECT c.g_cn_cod,
       DECODE(l_six_voe, 1, six_voeu_L1(c.g_cn_cod, g_aa_cod_bac_int, g_cn_flg_int_aca,
o_g_tg_cod), 0),
       .. a_ve_ord_vg_rel,                       -- Ordre du voeu avec voeux groupés relatifs
licence
       .. a_ve_ord_aff,                          -- Ordre du voeu avec voeux groupé relatif
licence et tous les autres voeux
       .. a_vg_ord,                              -- Ordre du sous-voeu dans le voeu
groupé
       DBMS_RANDOM.value(1, 999999),
       i.i_ep_cod,
       i.i_is_dip_val                            -- Pour ceux-ci on prend en plus en compte la validité
du diplôme
FROM g_can c, i_ins l, a_rec r, a_voe v
WHERE l.g_ti_cod=o_g_ti_cod
AND l.g_gf_cod=o_c_gp_cod
AND i_ep_cod IN (2, 3)                          -- Pointés recu (complet ou incomplet)

```

```
AND i.g_cn_cod=c.g_cn_cod
```

```
--TODO2016 => Traiter les groupes néo-rééo ensemble différemment (voir  
correction_classements_neo-reo.sql dans exploit/admissions/simulation/pb ponctuels)
```

```
AND c.g_ic_cod > 0
```

```
AND i.i_is_val=1
```

```
--non encore classé
```

```
AND NOT EXISTS (SELECT 1 FROM c_can_grp
```

```
WHERE i.g_cn_cod=g_cn_cod
```

```
AND i.g_gf_cod=c_gp_cod
```

```
AND i_ip_cod IN (4, 5)) -- Permet de récupérer les AC
```

```
AND i.g_ti_cod=r.g_ti_cod
```

```
AND c.g_cn_cod=v.g_cn_cod
```

```
AND r.g_ta_cod=v.g_ta_cod
```

```
UNION
```

```
-- les candidats qui n'ont au final pas classé la formation dans leur liste ordonnée. Ils sont classé, mais  
en dernier.
```

```
SELECT c.g_cn_cod,
```

```
0,
```

```
0,
```

```
0,
```

```
0,
```

```
DBMS_RANDOM.value(1, 999999),
```

```
i.i_ep_cod,
```

```
i.i_is_dip_val -- Pour ceux-ci on prend en plus en compte la validité  
du diplôme
```

```
FROM g_can c, i_ins i, a_rec r
```

```
WHERE i.g_ti_cod=o.g_ti_cod
```

```
AND i.g_gf_cod=o.c_gp_cod
```

```
AND i_ep_cod IN (2, 3) -- Pointés reçu (complet ou incomplet)
```

```
AND i.g_cn_cod=c.g_cn_cod
AND c.g_ic_cod > 0
AND i.i_ls_val=1
-- non encore classé
AND NOT EXISTS (SELECT 1 FROM c_can_grp
                WHERE i.g_cn_cod=g_cn_cod
                AND i.g_gf_cod=c_gp_cod
                AND i_lp_cod IN (4, 5))           -- Permet de récupérer les AC
AND i.g_ti_cod=r.g_ti_cod
AND NOT EXISTS (SELECT 1 FROM a_voe v WHERE c.g_cn_cod=v.g_cn_cod AND
r.g_ta_cod=v.g_ta_cod)
ORDER BY 2 desc, 3, 4, 5, 6;
```

BEGIN

-- par défaut, on est pas en prod

IS_prod:=0;

-- On vérifie que si on force un classement, on n'est pas en base de prod

X:='01';

BEGIN

SELECT DISTINCT 1 INTO dummy

FROM all_catalog

WHERE OWNER IN ('XXXXXX');

-- on est en prod

IS_prod:=1;


```

-- on ne laisse passer qu'en indic = 10
IF NVL(indic, 0) NOT IN (10)
THEN mess_aff:='On ne peut forcer un classement sur la base d'exploitation.';
      ROLLBACK;
      RETURN 1;
END IF;

EXCEPTION WHEN NO_DATA_FOUND THEN NULL;

END;

mess_aff:= 'Problème d'accès aux données, veuillez vous reconnecter ultérieurement.';

-- On vérifie si le groupe est issu d'une formation de type IDF 2, 3, 5 ou 6 et s'il concerné par
des néo d'IDF
-- alors, on utilisera les six voeux dans le classement sur ordre des voeux
BEGIN
      X:='02';
      SELECT 1
      INTO l_six_voe
      FROM g_tri_ins ti
      WHERE g_ti_cod=o_g_ti_cod
      AND NVL(g_ti_flg_rec_idf, 0) IN (2, 3, 5, 6)
      AND o_g_tg_cod IN (21, 25, 26, 41, 45, 46);

EXCEPTION
      WHEN NO_DATA_FOUND
      THEN l_six_voe:=0;          -- pour les autres groupes, on n'utilise pas les 6 voeux

```

```

END;

X:='03';

-- on vérifie que le classement ne soit pas déjà passé (pas de candidats classés dans
c_can_grp)

BEGIN

    -- Si le groupe est non sélectif, aucun candidat ne doit avoir été traité

    SELECT DISTINCT 1

    INTO dummy

    FROM c_can_grp cg, c_grp g

    WHERE g.c_gp_cod=o_c_gp_cod

    AND g.c_gp_cod=cg.c_gp_cod

    AND NVL(c_gp_flg_sel, 0)=0

    UNION

    -- Si le groupe est sélectif ou à pré-requis, on peut avoir des candidats NC ou AC

    SELECT DISTINCT 1

    FROM c_can_grp cg, c_grp g

    WHERE g.c_gp_cod=o_c_gp_cod

    AND g.c_gp_cod=cg.c_gp_cod

    AND NVL(c_gp_flg_sel, 0) IN (1, 2)

    AND i_ip_cod NOT IN (4, 6);

    mess_aff:='Un classement a déjà été saisi pour le groupe de cette formation : '

                ||o_g_ea_cod_||', '||o_g_ti_cod||', '||o_c_gp_cod;

    ROLLBACK;

    RETURN 1;

EXCEPTION WHEN NO_DATA_FOUND THEN NULL;    -- ok

```

```

END;

-- c'est ok, on va générer. On commence par récupérer des infos en base

BEGIN

    X:='04';

    SELECT g_tg_cod,                                c_gp_flg_sel,
           g_ea_cod_ges,                            ja.c_ja_cod,
           c_tj_cod,
           NVL(g_ti_flh_sel, g_fr_flg_sel),        c_gp_eta_cla
    INTO   l_g_tg_cod,
    l_c_gp_flg_sel,
           l_g_ea_cod_ges,                          l_c_ja_cod,
           l_c_tj_cod,
           l_g_flh_sel,
    l_c_gp_eta_cla
    FROM g_for fr, g_tri_ins ti, c_jur_adm ja, c_grp gp
    WHERE ti.g_ti_cod=o_g_ti_cod
    AND   ti.g_fr_cod_ins=fr.g_fr_cod
    AND   ti.g_ti_cod=ja.g_ti_cod
    AND   ja.c_ja_cod=gp.c_ja_cod
    AND   gp.c_gp_cod=o_c_gp_cod;

EXCEPTION

WHEN NO_DATA_FOUND

THEN mess_aff:='Erreur de traitement, la ligne groupe n'existe pas : c_gp_cod : '
           ||o_c_gp_cod;

    ROLLBACK;

    RETURN 1;

END;

```

```

-- on vérifie les conditions de traitement du groupe

IF IS_prod=0          -- Base de test
    OR                -- Ou
    -- en prod pour les classements formation non sélectives ou les AEFE
    (IS_prod=1 AND indic=10 AND l_g_flh_sel=0)

THEN NULL;  -- on laisse passer. Dans tous les autre cas, c'est une erreur.

ELSE  mess_aff:='On ne peut traiter ce type de classement aléatoire dans ces conditions :
||

                                'ls_prod : '||IS_prod||', indic : '||Indic||'Flag Sel : '||l_c_gp_flg_sel;

    ROLLBACK;

    RETURN 1;

END IF;

-- on vérifie l'état de pointage des dossiers si on est en prod, on est obligé d'accepter
-- des dossiers non reçus, pour les vérifs de diplômes

X:='05';

SELECT COUNT(*) INTO dummy
FROM i_ins i
WHERE g_ti_cod=o_g_ti_cod
AND g_gf_cod=o_c_gp_cod
AND l_js_val=1
AND l_ep_cod NOT IN (0, 2, 3, 7);

IF dummy > 0
THEN  mess_aff:='Pb, des dossiers ne sont pas pointés : étab : '

```

```

||o_g_ea_cod_ins||', for: '||o_g_ti_cod||', grp: '||o_c_gp_cod;

ROLLBACK;

RETURN 1;

END IF;

i:=1;

X:='61';

FOR c_rec IN classement_aleatoire_efe

LOOP BEGIN

INSERT INTO c_can_grp (

g_cn_cod, c_gp_cod,

i_ip_cod, c_cg_ran)

VALUES (

c_rec.g_cn_cod, o_c_gp_cod,

5, i);

EXCEPTION -- Si le candidat est déjà indiqué à classer, on met à jour le i_ip_cod et
le rang sur la ligne existante

WHEN UNIQUE_CONSTRAINT

THEN X:='07';

UPDATE c_can_grp

SET i_ip_cod=5,

c_cg_ran=i

WHERE g_cn_cod=c_rec.g_cn_cod

AND c_gp_cod=o_c_gp_cod

AND i_ip_cod=6;

```

```

IF SQL%ROWCOUNT!=1
THEN mess_err:='pk_generation_classement.gen_class_alea_V1_relatif_grp
X:(' || X || ')'
||'Erreur traitement d'un candidat AC pour l'étab '
|| o_g_ea_cod_ins||' et la formation '||o_g_ti_cod||':
' ||o_c_gp_cod||', le candidat '||c_rec.g_cn_cod
||' et le groupe : '||o_c_gp_cod||', rg : '||i;
ROLLBACK;
RETURN -1;
END IF;

END;

i:=i+1;

END LOOP;

X:='08';

FOR c_rec IN class_aleatoire_autres_cddts
LOOP -- diplôme non validé => non classé
IF c_rec.i_is_dip_val=1
THEN BEGIN
INSERT INTO c_can_grp (
g_cn_cod,
c_gp_cod,
l_lp_cod,
c_cg_ran)
VALUES (
c_rec.g_cn_cod,
o_c_gp_cod,

```

```

                                4,
NULL);

EXCEPTION -- Si le candidat est déjà non classé, on ne met à jour
WHEN UNIQUE_CONSTRAINT
THEN NULL;
END;

ELSE
BEGIN
    X:='09';

    INSERT INTO c_can_grp (
                                g_cn_cod,
c_gp_cod,
                                l_ip_cod,
c_cg_ran)
    VALUES (
                                c_rec.g_cn_cod,                                o_c_gp_cod,
                                5,                                i);

EXCEPTION -- Si le candidat est déjà à classer, on ne met à jour
WHEN UNIQUE_CONSTRAINT
THEN X:='10';

    UPDATE c_can_grp
    SET    l_ip_cod=5,
           c_cg_ran=i
    WHERE g_cn_cod=c_rec.g_cn_cod
    AND   c_gp_cod=o_c_gp_cod
    AND   i_ip_cod=6;

    IF SQL%ROWCOUNT=1

```

```

THEN
mess_err:='pk_generation_classement.gen_class_alea_V1_relatif_grp X : ( ' || X || ' )'

||'Erreur traitement d'un candidat AC pour l'\'etab '
|| o_g_ea_cod_ins||' et la formation
'||o_g_ti_cod||': '||o_c_gp_cod||', le candidat '||c_rec.g_cn_cod
||' et le groupe : '||o_c_gp_cod||', rg : '||i;

ROLLBACK;

RETURN -1;

END IF;

END;

i:=i+1;

END IF;

END LOOP;

```

-- le classement est marqué terminé

X:='11';

retour:=pk_new_classement_commun.MAJ_etat_classement(

```

o_g_ea_cod_ges,          o_g_ea_cod_ins,
o_g_ti_cod,
o_c_gp_cod,             l_c_ja_cod,          l_c_tj_cod,
2,                      5,
login,                  type_login,
mode_dev,
confirm,                saio,
nip,
0,                      indic,
mess_err,               mess_aff);

```

IF retour!=0


```
THEN ROLLBACK;
```

```
RETURN retour;
```

```
END IF;
```

```
-- On vérifie que le classement soit valide. la Trace est mise par cette PS
```

```
X:='12';
```

```
retour:=pk_new_classement_commun.valid_classement_def(
```

```
o_g_ti_cod, l_g_ea_cod_ges, o_g_ea_cod_ins,  
o_c_gp_cod, l_c_ja_cod, l_c_tj_cod,  
5,  
mode_dev, login, type_login,  
nlp, confirm, saio,  
0, indic,  
mess_err, mess_aff);
```

```
IF retour!=0
```

```
THEN ROLLBACK;
```

```
RETURN retour;
```

```
END IF;
```

```
X:='13';
```

```
retour:=pk_new_classement_commun.valid_classement_formation(
```

```
o_g_ea_cod_ins, l_g_ea_cod_ges,  
o_g_ti_cod, 5,
```

```

        login,
        mode_dev,
        type_login,
        confirm,
        nip,
        saio,
        0,
        indic,
        mess_err,
        mess_aff);

```

```

IF retour!=0
THEN ROLLBACK;
    RETURN retour;
END IF;

```

```

-- on indique que le classement est fait de manière automatique

```

```

-- et on gère le cas particulier des AC/NC eta_cla passe de 3 à 4

```

```

X:='14';

```

```

UPDATE c_grp SET      c_gp_flg_cla_oto=1,
                    c_gp_eta_cla=DECODE(l_c_gp_eta_cla, 3, 4, c_gp_eta_cla)

```

```

WHERE c_gp_cod=o_c_gp_cod

```

```

AND c_gp_eta_cla=2;

```

```

COMMIT;

```

```

RETURN 0;

```

```

EXCEPTION

```

```

WHEN OTHERS

```

```

THEN mess_err:='pk_generation_classement.gen_class_alea_V1_relatif_grp X: (' || X || ') '

```

```

        ||'Erreur ORACLE ' ||TO_CHAR(sqlcode)||' '||sqlerrm||' pour l"étab '

```

```
|| o_g_ea_cod_ins||' et la formation '||o_g_ti_cod||': '||o_c_gp_cod;
```

```
ROLLBACK;
```

```
RETURN -9;
```

```
END gen_class_alea_V1_relatif_grp;
```