

```

DELIMITER $$

CREATE PROCEDURE sp_generer_emploi_du_temps(
    IN p_pole_id INT,
    IN p_numero_semaine INT,
    IN p_date_debut DATE,
    IN p_date_fin DATE,
    OUT p_edt_id INT,
    OUT p_nb_seances_generees INT
)
BEGIN
    DECLARE v_affectation_id INT;
    DECLARE v_groupe_id INT;
    DECLARE v_formateur_id INT;
    DECLARE v_module_id INT;
    DECLARE v_heures_semaine FLOAT;
    DECLARE v_done INT DEFAULT FALSE;
    DECLARE v_seance_id INT;

    -- Curseur pour parcourir toutes les affectations actives du pôle
    DECLARE cur_affectations CURSOR FOR
        SELECT
            am.id as affectation_id,
            am.groupe_id,
            am.formateur_id,
            am.module_id,
            COALESCE(rh.heures_semaine, 0) as heures_semaine
        FROM AffectationModule am
        JOIN Groupe g ON am.groupe_id = g.id
        JOIN Filiere f ON g.filiere_id = f.id
        LEFT JOIN RepartitionHebdomadaire rh ON am.id = rh.affectation_module_id
            AND rh.numero_semaine = p_numero_semaine
        WHERE f.pole_id = p_pole_id
        AND am.statut = 'en_cours'
        AND am.masse_horaire_restant > 0;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET v_done = TRUE;

```

```
DECLARE EXIT HANDLER FOR SQLEXCEPTION
BEGIN
    ROLLBACK;
    SET p_edt_id = -1;
    SET p_nb_seances_generees = 0;
END;

START TRANSACTION;

-- Créer l'emploi du temps en statut brouillon
INSERT INTO EmploiDuTemps (
    pole_id,
    numero_semaine,
    date_debut,
    date_fin,
    statut,
    version
) VALUES (
    p_pole_id,
    p_numero_semaine,
    p_date_debut,
    p_date_fin,
    'brouillon',
    '1.0'
);

SET p_edt_id = LAST_INSERT_ID();
SET p_nb_seances_generees = 0;

-- Parcourir toutes les affectations
OPEN cur_affectations;
read_loop: LOOP
    FETCH cur_affectations INTO
        v_affectation_id,
        v_groupe_id,
        v_formateur_id,
        v_module_id,
```

```

v_heures_semaine;

IF v_done THEN
    LEAVE read_loop;
END IF;

IF v_heures_semaine > 0 THEN
    CALL sp_creer_seance(
        p_edt_id,
        v_groupe_id,
        v_affectation_id,
        'lundi',
        '08:30:00',
        '11:00:00',
        NULL, -- salle à déterminer
        'presentiel',
        'cours',
        v_seance_id
    );

    IF v_seance_id > 0 THEN
        SET p_nb_seances_generees = p_nb_seances_generees + 1;
    END IF;
END IF;
END LOOP;
CLOSE cur_affectations;

CALL sp_detecter_conflits_emploi_temps(p_edt_id, @nb_conflits);

COMMIT;
END$$

DELIMITER ;

```