

TP3 SQL - Procédure stockée

I - Prendre la base de données en main

Question 1 :

```
SELECT COUNT(id) AS nombre_departements, lieu
FROM departement
GROUP BY lieu;
```

Question 2 :

```
SELECT commentaire
FROM evaluation
WHERE id_employe = 84;
```

Question 3 :

```
SELECT ROUND(AVG(salaire), 2) AS salaire_moyen
FROM employe;
```

Question 4 :

```
SELECT e.nom, e.prenom, d.nom AS departement
FROM employe e
JOIN departement d ON e.id_departement = d.id;
```

Question 5 :

```
SELECT e.nom, e.prenom, c.statut AS congé
FROM employe e
JOIN congé c ON e.id = c.id_employe
WHERE c.statut = 'approuvé';
```

Question 6 :

```
SELECT d.nom AS departement, COUNT(e.id) AS nombre_employes
FROM employe e
JOIN departement d ON e.id_departement = d.id
GROUP BY d.nom;
```

Question 7 :

```

SELECT d.nom AS departement, COUNT(c.id) AS conges_approuves
FROM congé c
JOIN employe e ON c.id_employe = e.id
JOIN departement d ON e.id_departement = d.id
WHERE c.statut = 'approuvé'
GROUP BY d.nom;

```

Question 8 :

```

SELECT e.nom, e.prenom
FROM employe e
JOIN congé c ON e.id = c.id_employe
WHERE c.date_debut >= CURDATE() - INTERVAL 6 MONTH;

```

Question 9 :

```

SELECT d.nom AS departement, ROUND(AVG(eva.score), 2) AS note_moyenne
FROM employe e
JOIN evaluation eva ON e.id = eva.id_employe
JOIN departement d ON e.id_departement = d.id
GROUP BY d.nom;

```

Question 10 :

```

WITH note_moyenne_entreprise AS (
    SELECT AVG(score) AS moyenne_entreprise
    FROM evaluation
)
SELECT e.nom, e.prenom, d.nom AS departement
FROM employe e
JOIN evaluation eva ON e.id = eva.id_employe
JOIN departement d ON e.id_departement = d.id
WHERE eva.score > (SELECT moyenne_entreprise FROM note_moyenne_entreprise);

```

II - Procédures sans paramètres

Question 1 :

```

DELIMITER //
CREATE PROCEDURE ListeEmploye()
BEGIN
    SELECT e.nom, e.prenom, d.nom AS departement
    FROM Employe e
    JOIN Departement d ON e.id_departement = d.id;
END //

```

```
DELIMITER ;
```

Question 2 :

```
DELIMITER //
CREATE PROCEDURE employesAvecCongesApprouves()
BEGIN
SELECT e.nom, e.prenom, c.date_debut, c.date_fin
FROM employe e
JOIN conge c ON e.employe_id = c.employe_id
WHERE c.statut = 'approuvé';
END//
DELIMITER ;
```

Question 3 :

```
DELIMITER //
CREATE PROCEDURE ListeDepartement()
BEGIN
SELECT departement_id, COUNT(*) AS nombre_employes
FROM employe e
GROUP BY departement.id;
END //
DELIMITER ;
```

Question 4 :

```
DELIMITER //
CREATE PROCEDURE EvaluationEmploye()
BEGIN
SELECT e.nom, e.prenom, evaluation.note
FROM employe e
JOIN evaluation ON e.id = evaluation.id_employe
WHERE evaluation.note < 5
ORDER BY evaluation.note ASC;
END //
DELIMITER ;
```

Question 5 :

```
DELIMITER //
CREATE PROCEDURE CongeCumule()
BEGIN
SELECT employees.nom, employees.prenom, SUM(DATEDIFF(conges.date_fin,
conges.date_debut)) AS jours_total
FROM employees
JOIN conges ON employees.employe_id = conges.employe_id
GROUP BY employees.employe_id
```

```
HAVING jours_total > 20;  
END //  
DELIMITER ;
```

III - Procédures avec paramètres

Question 1 :

```
DELIMITER //  
CREATE PROCEDURE salaire_total_departement(IN dep_id INT)  
BEGIN  
    SELECT d.nom AS departement, SUM(e.salaire) AS salaire_total  
    FROM employe e  
    JOIN departement d ON e.id_departement = d.id  
    WHERE d.id = dep_id;  
END //  
DELIMITER ;
```

Question 2 :

```
DELIMITER //  
CREATE PROCEDURE congésApprouvésEmploye(IN employe_id INT)  
BEGIN  
    SELECT congés.date_debut, congés.date_fin  
    FROM congés  
    WHERE congés.employe_id = employe_id AND congés.statut = 'Approuvé';  
END //  
DELIMITER ;
```

Question 3 :

```
DELIMITER //  
CREATE PROCEDURE mettreAJourDepartement(IN employe_id INT, IN nouveau_dept_id  
INT)  
BEGIN  
    UPDATE employes  
    SET departement_id = nouveau_dept_id  
    WHERE employe_id = employe_id;  
END //  
DELIMITER ;
```

Question 4 :

```
DELIMITER //  
CREATE PROCEDURE supprimerEmploye()
```

```

BEGIN
    DELETE FROM conges WHERE employe_id = employe_id;
    DELETE FROM evaluations WHERE employe_id = employe_id;
    DELETE FROM employes WHERE employe_id = employe_id;
END//
DELIMITER ;

```

Question 5 :

```

DELIMITER //
CREATE PROCEDURE mise_a_jour_salaire_employe(IN emp_id INT, IN new_salaire
DECIMAL(10,2))
BEGIN
    UPDATE employe
    SET salaire = new_salaire
    WHERE id = emp_id;
END//
DELIMITER ;

```

Question 6 :

```

DELIMITER //
CREATE PROCEDURE employes_departement_score_min(IN dep_id INT, IN min_score
INT)
BEGIN
    SELECT e.nom, e.prenom, eva.score
    FROM employe e
    JOIN evaluation eva ON e.id = eva.id_employe
    WHERE e.id_departement = dep_id AND eva.score >= min_score;
END//
DELIMITER ;

```

Question 7 :

```

DELIMITER //
CREATE PROCEDURE mise_a_jour_commentaire_evaluation(IN eval_id INT, IN
new_commentaire TEXT)
BEGIN
    UPDATE evaluation
    SET commentaire = new_commentaire
    WHERE id = eval_id;
END//
DELIMITER ;

```

Question 8 :

```

DELIMITER //

```

```

CREATE PROCEDURE mise_a_jour_statut_conge(IN conge_id INT, IN new_statut
VARCHAR(20))
BEGIN
    UPDATE conge
    SET statut = new_statut
    WHERE id = conge_id;
END//
DELIMITER ;

```

Question 9 :

```

DELIMITER //
CREATE PROCEDURE employes_sans_conge(IN start_date DATE, IN end_date DATE)
BEGIN
    SELECT e.nom, e.prenom
    FROM employe e
    WHERE e.id NOT IN (
        SELECT DISTINCT id_employe
        FROM conge
        WHERE date_debut BETWEEN start_date AND end_date
    );
END//
DELIMITER ;

```

Question 10 :

```

DELIMITER //
CREATE PROCEDURE jours_conge_employe(IN emp_id INT, IN start_date DATE, IN
end_date DATE)
BEGIN
    SELECT e.nom, e.prenom, SUM(DATEDIFF(c.date_fin, c.date_debut)) AS
jours_conge,
        CONCAT(start_date, ' à ', end_date) AS periode
    FROM employe e
    JOIN conge c ON e.id = c.id_employe
    WHERE e.id = emp_id AND c.date_debut BETWEEN start_date AND end_date;
END//
DELIMITER ;

```

