

Université / Département d'Informatique

Master 1 — IA / RSD

Évaluation SQL (2/2025) — Triggers Avancés

Durée : 45 Minutes | Documents non autorisés | Répondre sur le sujet

Nom :	Prénom :	Spécialité :	Groupe :	Note :
		<input type="checkbox"/> IA		/20
		<input type="checkbox"/> RSD		

Schéma de la base de données (Suivi médical) :

```

CREATE TABLE Patient (
  id_patient  INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
  nom         VARCHAR(50) NOT NULL,
  prenom      VARCHAR(50) NOT NULL,
  date_nais   DATE NOT NULL,
  sexe        CHAR(1) CHECK (sexe IN ('M', 'F')),
  telephone   VARCHAR(15)
);

CREATE TABLE Mesure (
  id_mesure   INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
  id_patient  INTEGER REFERENCES Patient(id_patient),
  date_mesure TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

  poids       DECIMAL(5,2),      -- en kg
  taille      DECIMAL(3,2),      -- en mètres (ex: 1.75)
  imc         DECIMAL(4,2),      -- calculé automatiquement
  categorie_imc VARCHAR(20),     -- Maigreur, Normal, Surpoids, Obésité

  tension_systolique INTEGER,    -- en mmHg
  tension_diastolique INTEGER,   -- en mmHg

  alerte      BOOLEAN DEFAULT FALSE,
  message_alerte TEXT
);

```

Rappels : IMC = poids / taille²

IMC : Maigreur < 18.5 | Normal : 18.5-24.9 | Surpoids : 25-29.9 | Obésité ≥ 30

Tension normale : Systolique (90-140 mmHg) | Diastolique (60-90 mmHg)

Partie 1 : QCM (10 questions × 1 pt = 10 pts)

Cochez la bonne réponse pour chaque question.

1. Quel bloc délimite le corps d'une fonction trigger en PL/pgSQL ?

- ☒ A) BEGIN ... END; avec RETURNS TRIGGER obligatoire
- ☐ B) START ... STOP; avec RETURNS TRIGGER
- ☐ C) BEGIN ... END; avec RETURNS void
- ☐ D) DO ... DONE; avec RETURNS TRIGGER

2. Quel opérateur permet d'affecter une valeur à une variable dans un trigger ?

- ☐ A) = (comme dans SELECT INTO)
- ☒ B) := (affectation PL/pgSQL)
- ☐ C) Les deux := et = sont équivalents
- ☐ D) == (double égal)

3. Comment récupérer le poids d'un patient dans une variable v_poids ?

- ✓ ☒ A) `SELECT poids INTO v_poids FROM Mesure WHERE id_patient = 1;`
- ☐ B) `SET v_poids = (SELECT poids FROM Mesure);`
- ☐ C) `v_poids := SELECT poids FROM Mesure;`
- ☐ D) `SELECT poids TO v_poids FROM Mesure;`

4. Quand un trigger BEFORE INSERT est-il exécuté ?

- ✓ ☒ A) Avant l'insertion, permettant de modifier NEW
- ☐ B) Avant l'insertion, mais NEW est en lecture seule
- ☐ C) Après la validation des contraintes, avant l'écriture
- ☐ D) Avant l'insertion, mais après le calcul de l'IDENTITY

5. Dans un trigger BEFORE UPDATE, que représente NEW.poids ?

- ✓ ☒ A) La nouvelle valeur qui sera enregistrée (modifiable)
- ☐ B) La nouvelle valeur qui sera enregistrée (lecture seule)
- ☐ C) L'ancienne valeur avant modification
- ☐ D) La différence entre l'ancienne et la nouvelle valeur

6. Que doit retourner une fonction trigger pour valider l'opération ?

- ✓ ☒ A) NEW pour INSERT/UPDATE, OLD pour DELETE
- ☐ B) TRUE pour valider, FALSE pour annuler
- ☐ C) NEW pour toutes les opérations
- ☐ D) 1 pour valider, 0 pour annuler

7. Quelle clause déclenche un trigger pour chaque ligne affectée ?

- ✓ ☒ A) FOR EACH ROW (accès à NEW/OLD)
- ☐ B) FOR EACH STATEMENT (accès à NEW/OLD)
- ☐ C) FOR ALL ROWS (accès à NEW/OLD)
- ☐ D) FOR EACH RECORD (accès à NEW/OLD)

8. Comment annuler une insertion depuis un trigger BEFORE INSERT ?

- ☐ A) RETURN NULL; annule silencieusement
- ✓ ☒ B) RAISE EXCEPTION annule avec erreur, RETURN NULL continue
- ☐ C) RETURN FALSE; annule l'opération
- ☐ D) ABORT; annule la transaction

9. Dans un trigger AFTER DELETE, quelle variable contient les données ?

- ✓ ☒ A) OLD (NEW est NULL)
- ☐ B) OLD et NEW contiennent les mêmes données
- ☐ C) DELETED (variable spéciale)
- ☐ D) NEW (OLD est NULL)

10. Dans un trigger AFTER INSERT, peut-on modifier NEW.imc ?

- ☐ A) Non, les données sont déjà enregistrées
- ☐ B) Oui, avec RETURN NEW modifié
- ☐ C) Oui, mais nécessite un UPDATE supplémentaire
- ✓ ☒ D) Non, NEW n'existe pas dans AFTER INSERT

Partie 2 : Exercices pratiques (10 pts)

Exercice 1 (5 pts) — Calcul automatique de l'IMC et détection d'alertes

Complétez le trigger qui, lors de l'insertion ou modification d'une mesure :

- Calcule automatiquement l'IMC (poids / taille²)
- Détermine la catégorie (Maigreur, Normal, Surpoids, Obésité)
- Refuse la mesure si le poids ou la taille sont invalides (≤ 0)
- Active l'alerte avec un message si l'IMC indique une obésité (≥ 30)

```
CREATE OR REPLACE FUNCTION categorie_imc
RETURNS TRIGGER AS $$
BEGIN
    -- Validation des données
    IF NEW.poids <= 0 OR NEW.taille <= 0 THEN
        RAISE EXCEPTION 'Le poids et la taille doivent être positifs';
    END IF;

    -- Calcul de l'IMC : poids / taille²
    NEW.imc := ROUND(NEW.poids / (NEW.taille * NEW.taille), 2);

    -- Détermination de la catégorie
    IF NEW.imc < 18.5 THEN
        NEW.categorie_imc := 'Maigreur';
    ELSIF NEW.imc < 25 THEN
        NEW.categorie_imc := 'normal';
    ELSIF NEW.imc < 30 THEN
        NEW.categorie_imc := 'Surpoids';
    ELSE
        NEW.categorie_imc := 'obésité';
        NEW.alerte := TRUE;
        NEW.message_alerte := 'IMC critique : ' || NEW.imc;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER tg_calculer_imc
BEFORE INSERT OR UPDATE ON Mesure
FOR EACH ROW
EXECUTE FUNCTION categorie_imc();
```

Exercice 2 (5 pts) — Contrôle de la tension artérielle

Complétez le trigger qui vérifie la tension et génère une alerte si elle est anormale :

- Tension systolique normale : entre 90 et 140 mmHg
- Tension diastolique normale : entre 60 et 90 mmHg
- Si hors limites : activer l'alerte et ajouter le message approprié

```
CREATE OR REPLACE FUNCTION verifier_tension()
RETURNS TRIGGER AS $$
DECLARE
    v_message TEXT := '';
BEGIN
    -- Vérifier seulement si les tensions sont renseignées
    IF NEW.tension_systolique IS NOT NULL THEN
        -- Vérifier la tension systolique (normale: 90-140)
        IF NEW.tension_systolique < 90 OR
           NEW.tension_systolique > 140 THEN
            v_message := 'Systolique anormale: ' ||
                        NEW.tension_systolique || ' mmHg';
        END IF;
        -- Vérifier la tension diastolique (normale: 60-90)
        IF NEW.tension_diastolique < 60 OR
           NEW.tension_diastolique > 90 THEN
            IF v_message <> '' THEN
                v_message := v_message || ' | ';
            END IF;
            v_message := v_message || 'Diastolique anormale: ' ||
                        NEW.tension_diastolique || ' mmHg';
        END IF;
        -- Si une anomalie est détectée
        IF v_message <> '' THEN
            NEW.alerte := True;
            IF NEW.message_alerte IS NOT NULL THEN
                NEW.message_alerte := NEW.message_alerte || ' | ' || v_message;
            ELSE
                NEW.message_alerte := v_message;
            END IF;
        END IF;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER tg_verifier_tension
BEFORE INSERT OR UPDATE ON patient
FOR EACH ROW
EXECUTE FUNCTION verifier_tension();
```