

Université / Département d'Informatique — Master 1 IA/RSD

Corrigé Type — Examen de Rattrapage

Bases de Données Avancées (BDDA)

Partie 1 : QCM (10 questions = 10 pts)

Barème : Réponse correcte +1 pt — réponse fausse ou absente 0 pt (pas de point négatif)

1. Dans un trigger BEFORE INSERT ... FOR EACH ROW, on modifie NEW.cout_credits puis on exécute RETURN NEW. Que se passe-t-il ?

- A) La modification est ignorée car l'insertion a déjà eu lieu
- B) Une erreur est levée car NEW est en lecture seule
- C) La valeur modifiée est bien enregistrée dans la table
- D) Une récursion infinie du trigger se produit

2. Une table possède plusieurs triggers BEFORE INSERT. Dans quel ordre PostgreSQL les exécute-t-il ?

- A) Dans l'ordre de leur création
- B) Par ordre alphabétique du nom des triggers
- C) Dans un ordre non déterministe
- D) Tous simultanément, en parallèle

3. Dans une fonction trigger, on exécute : SELECT credits_recharge INTO v_credits FROM conducteur WHERE id = NEW.id_conducteur; Si aucune ligne ne correspond à la condition, que se passe-t-il ?

- A) v_credits prend la valeur NULL, sans qu'aucune erreur ne soit levée
- B) Une exception « aucune donnée trouvée » est levée automatiquement
- C) v_credits prend la valeur 0 par défaut
- D) La fonction s'interrompt immédiatement et retourne NULL

4. Dans un trigger AFTER UPDATE, l'instruction NEW.solde_credits := NEW.solde_credits - 10 :

- A) met à jour le solde dans la base de données
- B) lève une exception de type « lecture seule »
- C) annule automatiquement la mise à jour
- D) n'a aucun effet sur les données déjà enregistrées

5. Comment exécuter une fonction sans récupérer sa valeur de retour à l'intérieur d'un trigger ?

- A) CALL ma_fonction();
- B) PERFORM ma_fonction();
- C) RUN ma_fonction();
- D) RETURN ma_fonction();

6. Quelle instruction permet d'afficher un message d'information dans un trigger SANS interrompre l'opération en cours ?

- A) RAISE NOTICE
- B) RAISE EXCEPTION
- C) RAISE ERROR
- D) RETURN FALSE

7. Pour qu'un trigger BEFORE DELETE autorise la suppression de la ligne, la fonction doit retourner :

- A) NEW
- B) NULL
- C) OLD
- D) TRUE

8. Dans un trigger BEFORE INSERT, que provoque l'instruction RETURN NULL ; ?

- A) La ligne est insérée avec des valeurs NULL
- B) Une exception est levée automatiquement
- C) La ligne est insérée normalement, sans modification
- D) L'insertion de la ligne est annulée

9. À l'intérieur d'un trigger défini sur la table T, on exécute un UPDATE sur cette même table T. Que peut-il se produire ?

- A) Une erreur, car c'est strictement interdit
- B) La mise à jour est ignorée silencieusement
- C) Le trigger courant est automatiquement désactivé
- D) Le déclenchement récursif d'autres triggers sur T (risque de boucle)

10. Comment désactiver temporairement TOUS les triggers d'une table sans les supprimer ?

- A) DROP TRIGGER ALL ON table;
- B) ALTER TABLE table DISABLE TRIGGER ALL;
- C) SET TRIGGERS OFF ON table;
- D) ALTER TRIGGER ALL ON table DISABLE;

Partie 2 : Étude de cas — Réseau de bornes de recharge (10 pts)

Les réponses attendues sont surlignées en jaune (0,5 pt par vide).

SCHÉMA DE LA BASE DE DONNÉES

```
CREATE TABLE conducteur (  
  id INTEGER PRIMARY KEY,  
  email VARCHAR(64) UNIQUE NOT NULL,  
  credits_recharge INTEGER DEFAULT 500 CHECK (credits_recharge >= 0),  
  total_sessions INTEGER DEFAULT 0 CHECK (total_sessions >= 0),  
  total_energie_kwh BIGINT DEFAULT 0 CHECK (total_energie_kwh >= 0),  
  type_abonnement VARCHAR(20) DEFAULT 'BASIQUE'  
  CHECK (type_abonnement IN ('BASIQUE', 'PREMIUM', 'FLOTTE'))  
);  
CREATE TABLE session_recharge (  
  id INTEGER PRIMARY KEY,  
  id_conducteur INTEGER NOT NULL REFERENCES conducteur(id),  
  code_borne VARCHAR(32) NOT NULL,  
  energie_kwh INTEGER NOT NULL CHECK (energie_kwh > 0),  
  cout_credits INTEGER NOT NULL CHECK (cout_credits > 0),  
  statut VARCHAR(32) DEFAULT 'TERMINEE'  
  CHECK (statut IN ('TERMINEE', 'ECHEC', 'ARCHIVE'))  
);
```

TRIGGER 1 — Valider une session de recharge et mettre à jour les statistiques

```
CREATE OR REPLACE FUNCTION fn_valider_recharge()  
RETURNS TRIGGER AS $$  
DECLARE  
  v_credits INTEGER;  
BEGIN  
  -- Q1. Vérifiez que l'énergie de la session est bien comprise entre 1  
  --   et 100 kWh ; sinon, l'exception ci-dessous doit être levée.  
  IF (NEW.energie_kwh < 1) OR (NEW.energie_kwh > 100) THEN  
    RAISE EXCEPTION 'Energie invalide (min 1 kWh, max 100 kWh)';  
  END IF;  
  
  -- Q2. Récupérez le solde de crédits du conducteur concerné par la  
  --   session et placez-le dans la variable v_credits.  
  SELECT credits_recharge INTO v_credits  
  FROM conducteur  
  WHERE id = NEW.id_conducteur;
```

```
-- Q3. Vérifiez que ce solde suffit à couvrir le coût de la session.
IF v_credits < (NEW.cout_credits) THEN
    RAISE EXCEPTION 'Credits insuffisants : % disponible(s), % requis',
        v_credits, (NEW.cout_credits);
END IF;

-- Q4. Mettez à jour le conducteur : retirez le coût du solde, ajoutez 1
-- au nombre de sessions, ajoutez l'énergie au total consommé, et
-- ne modifiez que le conducteur concerné (clause WHERE).
UPDATE conducteur
SET credits_recharge = credits_recharge - (NEW.cout_credits),
    total_sessions = (total_sessions) + 1,
    total_energie_kwh = (total_energie_kwh) + (NEW.energie_kwh)
WHERE (id = NEW.id_conducteur);

-- Q5. Retournez la ligne qui permet de valider l'insertion.
RETURN (NEW);
END; $$ LANGUAGE plpgsql;
-- Déclaration du trigger (déjà fournie) : il s'exécute avant chaque
-- insertion sur la table session_recharge.
CREATE TRIGGER trg_valider_recharge
BEFORE INSERT ON session_recharge
FOR EACH ROW
EXECUTE FUNCTION fn_valider_recharge();
```

TRIGGER 2 — Protéger certains comptes contre la suppression

```
CREATE OR REPLACE FUNCTION fn_proteger_flotte()
RETURNS TRIGGER AS $$
BEGIN
    -- Q6. Empêchez la suppression lorsque le compte est de type 'FLOTTE'
    -- ET qu'il a consommé plus de 100 000 kWh.
    IF (OLD.type_abonnement = (FLOTTE)
        AND (OLD.total_energie_kwh) > 100000 THEN
        RAISE EXCEPTION 'Suppression impossible : compte FLOTTE avec % kWh',
            (OLD.total_energie_kwh);
    END IF;
    -- Q7. Sinon, autorisez la suppression en retournant la ligne adéquate.
    RETURN (OLD);
END;
$$ LANGUAGE plpgsql;
-- Q8. Choisissez le moment d'exécution qui permet d'EMPÊCHER la
-- suppression (le trigger agit ligne par ligne sur conducteur).
CREATE TRIGGER trg_proteger_flotte
(BEFORE) DELETE ON conducteur
FOR EACH ROW
EXECUTE FUNCTION fn_proteger_flotte();
```