

Contexte du Royaume des Bases de Données

Bienvenue à **SQLdria**, royaume où magie et données se mêlent. Les héros y défient des dragons pour augmenter leur expérience. Mais attention, certains dragons sont si puissants qu'ils peuvent réduire un héros en cendres numériques.

Deux tables existent :

- heros(id_heros INT, nom VARCHAR(50), niveau_experience INT)
- dragons(id_dragon INT, nom VARCHAR(50), niveau_puissance INT)

Objectif de l'épreuve

Créer une procédure stockée affronter_dragon avec :

- IN id_heros_combat INT
- IN id_dragon_combat INT
- OUT resultat VARCHAR(255)

Règles du combat légendaire

1. Si `niveau_experience > niveau_puissance`, victoire.
 - Le héros gagne `niveau_puissance / 2` points.
 - Si l'écart est supérieur à 20, le héros gagne un bonus de 5 points.
2. Si le dragon est plus fort ou égal :
 - Le héros perd 10 points d'expérience.
 - Si après le combat, `niveau_experience < 0`, il est remis à 0.
3. Le message `resultat` doit suivre le modèle :
 - Victoire de `<nom_heros>` contre `<nom_dragon>`
 - Défaite de `<nom_heros>` face à `<nom_dragon>`
4. Le niveau d'expérience ne peut pas dépasser 100.
5. Si un identifiant n'existe pas, la procédure doit lever une erreur claire.

Travail demandé

1. Créez les tables `heros` et `dragons`, insérez 3 entrées chacune.
2. Créez la procédure `affronter_dragon` :
 - Utiliser `DECLARE` pour les variables locales.
 - Utiliser `SELECT INTO` pour extraire les informations.
 - Utiliser des conditions `IF`, des tests d'existence.
 - Utiliser `SIGNAL SQLSTATE` pour lever des erreurs en cas d'identifiant inexistant.
3. Appel de test :

```
CALL affronter_dragon(2, 3, @resultat);
SELECT @resultat;
```
4. Bonus : gérez un journal de combat dans une table `journal_combats(id, heros, dragon, resultat, date_combat)`.

Exemple de déclaration simplifiée

(À ne pas recopier, à refaire seul en version complète)

```
DELIMITER $$
```

```
CREATE PROCEDURE affronter_dragon(
  IN id_heros_combat INT,
  IN id_dragon_combat INT,
  OUT resultat VARCHAR(255)
)
BEGIN
  -- Déclarations ...
  -- Vérifications ...
  -- Logique du combat ...
  -- Construction du message ...
  -- Mises à jour ...
END$$
```

```
DELIMITER ;
```

Recommandations du Haut Conseil

- Utilisez la clause `DECLARE EXIT HANDLER` pour intercepter les erreurs.
- Limitez les niveaux entre 0 et 100 avec `IF ... THEN`.

- Évitez les requêtes imbriquées inutiles, soyez clairs et efficaces.

Astuce du vieux sorcier : Si vous utilisez `SELECT INTO`, assurez-vous que la requête retourne exactement une ligne, sinon une erreur sera déclenchée.