
COMPTE RENDU

TP3 – SQL ET PROG

Christopher Marie-Angélique / Groupe 2B

Table des matières

Introduction	2
Mise en place d'une procédure PL/SQL	2

Introduction

Ce troisième TP de SQL et Programmation traite des procédure PL/SQL.

Nous sommes chargés dans ce TP de développer au complet une procédure avec la mise en place d'exception pour gérer les éventuels cas d'erreur.

Mise en place d'une procédure PL/SQL

1. Créez la procédure PL/SQL AjouterMatch répondant aux spécifications cidessous

```
create or replace PROCEDURE AJOUTERMATCH (  
  p_jrn Match.jrn%TYPE,  
  p_dtm Match.dtm%TYPE,  
  p_stade Match.stade%TYPE,  
  p_ville Match.ville%TYPE,  
  p_prea Match.prea%TYPE,  
  p_noma Match.noma%TYPE,  
  p_ne1 Match.ne1%TYPE,  
  p_ne2 Match.ne2%TYPE) IS  
  
  fk_error Exception;  
  PRAGMA Exception_init(fk_error, -2291);  
  
  ck_error Exception;  
  PRAGMA Exception_init(ck_error, -2290);  
  
BEGIN  
  
  INSERT INTO Match(nm, jrn, dtm, stade, ville, prea, noma, ne1, ne2)  
  VALUES(SEQ_MATCH.nextval, p_jrn,p_dtm, p_stade, p_ville, p_prea, p_noma, p_ne1, p_ne2);  
  COMMIT;  
  
  DBMS_OUTPUT.PUT_LINE('Match inséré.');  
EXCEPTION  
  
WHEN fk_error THEN  
  if (SQLERRM LIKE '%FK_MATCH_EQUIPE1%') then  
    DBMS_OUTPUT.PUT_LINE('ERREUR : L''équipe n°1 n''existe pas.');  end if;  
  
  if (SQLERRM LIKE '%FK_MATCH_EQUIPE2%') then  
    DBMS_OUTPUT.PUT_LINE('ERREUR : L''équipe n°2 n''existe pas.');  end if;  
  
  if (SQLERRM LIKE '%FK_MATCH_ARBITRE%') then  
    DBMS_OUTPUT.PUT_LINE('ERREUR : L''arbitre spécifié n''existe pas.');  end if;
```

```

WHEN ck_error THEN
    if (SQLERRM LIKE '%CK_MATCH_JRN%') then
        DBMS_OUTPUT.PUT_LINE('ERREUR : Numéro de journée invalide.');
```

```

    end if;

    if (SQLERRM LIKE '%CK_MATCH_NE1_NE2%') then
        DBMS_OUTPUT.PUT_LINE('ERREUR : Une équipe ne peut pas se rencontrer elle-même.');
```

```

    end if;

    if (SQLERRM LIKE '%NN_MATCH_NE1%') then
        DBMS_OUTPUT.PUT_LINE('ERREUR : Equipe 1 manquante.');
```

```

    end if;

    if (SQLERRM LIKE '%NN_MATCH_NE2%') then
        DBMS_OUTPUT.PUT_LINE('ERREUR : Equipe 2 manquante.');
```

```

    end if;

WHEN DUP_VAL_ON_INDEX THEN
    DBMS_OUTPUT.PUT_LINE('ERREUR : Ces deux équipes se sont déjà rencontrées.');
```

```

END;
```

2. Testez la procédure PL/SQL en vérifiant

- Cas nominal :

```

SET SERVEROUTPUT ON;
EXEC ajoutermatch(1, '01/01/01', 'Stade de France', 'Paris', 'Sean', 'Escassut', 'ENG', 'SCO');
```

Procédure PL/SQL terminée.

- Numéro de journée invalide :

```

SET SERVEROUTPUT ON;
EXEC ajoutermatch(7, '01/01/01', 'Stade de France', 'Paris', 'Sean', 'Escassut', 'ENG', 'SCO');
```

ERREUR : Numéro de journée invalide.

Procédure PL/SQL terminée.

- Arbitre inconnu :

```

SET SERVEROUTPUT ON;
EXEC ajoutermatch(1, '01/01/01', 'Stade de France', 'Paris', 'John', 'Cena', 'ENG', 'SCO');
```

ERREUR : L'arbitre spécifié n'existe pas.

Procédure PL/SQL terminée.

- Equipe 1 manquante :

```
SET SERVEROUTPUT ON;  
EXEC ajoutermatch(1, '01/01/01', 'Stade de France', 'Paris', 'Sean', 'Escassut', NULL, 'SCO');
```

```
ERREUR : Equipe 1 manquante.  
  
Procédure PL/SQL terminée.
```

- Equipe 2 manquante :

```
SET SERVEROUTPUT ON;  
EXEC ajoutermatch(1, '01/01/01', 'Stade de France', 'Paris', 'Sean',  
'Escassut', 'SCO', NULL);
```

```
ERREUR : Equipe 2 manquante.  
  
Procédure PL/SQL terminée.
```

- Equipe 1 inconnue :

```
SET SERVEROUTPUT ON;  
EXEC ajoutermatch(1, '01/01/01', 'Stade de France', 'Paris', 'Sean', 'Escassut', 'TUV', 'SCO');
```

- Une équipe joue contre elle-même :

```
SET SERVEROUTPUT ON;  
EXEC ajoutermatch(1, '01/01/01', 'Stade de France', 'Paris', 'Sean', 'Escassut', 'ENG', 'ENG');
```

```
ERREUR : Une équipe ne peut pas se rencontrer elle-même.  
  
Procédure PL/SQL terminée.
```