

Livrable 3

Rapport de BD

VACQUIER Hugo, BIRET-TOSCANO Esteban, FERNANDEZ
Mickael
6 avril 2023

Sommaire :

I-	Nouvelle vue :.....	3
II-	Nouvelle procédure stockée :.....	4
III-	Exécution des tests :.....	8

I- Nouvelle vue :

Voici le résultat de l'interrogation de l'ancienne vue :

```
SELECT * FROM VCARTONS_PAR_ARBITRE;
```

PRENOMARBITRE	NOMARBITRE	CARTON	NUMMATCH	TEMPS	PRENOMJOUEUR	NOMJOUEUR
---------------	------------	--------	----------	-------	--------------	-----------

Ben	O'Keeffe	Carton Jaune	2	66	Luke	Cowan-Dickie
-----	----------	--------------	---	----	------	--------------

```
CREATE OR REPLACE VIEW VCARTONS_PAR_ARBITREV2 (PRENOMARBITRE, NOMARBITRE, CARTON, NUMMATCH, TEMPS, PRENOMJOUEUR, NOMJOUEUR) AS
```

```
SELECT A.prea, A.noma, S.evt, S.nm, S.tps, J.prej, J.nomj
```

```
FROM ArbitreV2 A, SurvenirV2 S, JoueurV2 J, MatchV2 M
```

```
WHERE M.na = A.na
```

```
AND M.nm = S.nm
```

```
AND S.nj = J.nj
```

```
AND evt LIKE 'Carton%'
```

```
ORDER BY 1, 2, 4, 5;
```

Voici maintenant le code de la nouvelle vue :

```
CREATE OR REPLACE VIEW VCARTONS_PAR_ARBITREV2 (PRENOMARBITRE, NOMARBITRE, CARTON,
NUMMATCH, TEMPS, PRENOMJOUEUR, NOMJOUEUR) AS
```

```
SELECT A.prea, A.noma, S.evt, S.nm, S.tps, J.prej, J.nomj
```

```
FROM ArbitreV2 A, SurvenirV2 S, JoueurV2 J, MatchV2 M
```

```
WHERE M.na = A.na
```

View VCARTONS_PAR_ARBITREV2 créé(e).

```
AND M.nm = S.nm
```

```
AND S.nj = J.nj
```

```
AND evt LIKE 'Carton%'
```

```
ORDER BY 1, 2, 4, 5;
```

```
SELECT * FROM VCARTONS_PAR_ARBITREV2;
```

PRENOMARBITRE	NOMARBITRE	CARTON	NUMMATCH	TEMPS	PRENOMJOUEUR	NOMJOUEUR
Ben	O'Keeffe	Carton Jaune	2	66	Luke	Cowan-Dickie
Jaco	Peyper	Carton Jaune	1	49	Josh	Adams
Mathieu	Raynal	Carton Rouge	12	2	Charlie	Ewels
Nic	Berry	Carton Jaune	4	68	Finn	Russell
Nika	Amashukeli	Carton Rouge	9	19	Hame	Faiva
Wayne	Barnes	Carton Jaune	14	77	Ben	White

II- Nouvelle procédure stockée :

```
ALTER TABLE MATCHV2
ADD (
    CONSTRAINT FK_MATCHV2_EQUIPE1 FOREIGN KEY (ne1) REFERENCES EQUIPEV2 (ne),
    CONSTRAINT FK_MATCHV2_EQUIPE2 FOREIGN KEY (ne2) REFERENCES EQUIPEV2 (ne)
);
```

(Rajout des clés étrangères des équipes, oubliées lors de la création)

Voici le code de la nouvelle procédure stockée :

```
create or replace PROCEDURE AjouterMatchV2
(
    p_jrn MatchV2.jrn%TYPE,
    -- p_dtm MatchV2.dtm%TYPE,
    p_stade MatchV2.stade%TYPE,
    p_na MatchV2.na%TYPE,
    p_ne1 MatchV2.ne1%TYPE,
    p_ne2 MatchV2.ne2%TYPE
    --p_scr1 MatchV2.scr1%TYPE,
    --p_scr2 MatchV2.scr2%TYPE
)
IS
    fk_erreur EXCEPTION;
    PRAGMA EXCEPTION_INIT(fk_erreur, -2291);
    ck_erreur EXCEPTION;
    PRAGMA EXCEPTION_INIT(ck_erreur, -2290);
BEGIN
    INSERT INTO MatchV2 (nm, jrn, stade, na, ne1, ne2, scr1, scr2)
    VALUES (seq_matchv2.NEXTVAL, p_jrn, p_stade, p_na, p_ne1, p_ne2, 20, 58);
    COMMIT;
EXCEPTION
    WHEN ck_erreur THEN
        IF (SQLERRM LIKE '%CK_MATCHV2_JRN%') THEN
            RAISE_APPLICATION_ERROR(-20001,'ERREUR : Journée invalide !');
        ELSIF (SQLERRM LIKE '%NN_MATCHV2_NE1%') THEN
            RAISE_APPLICATION_ERROR(-20002,'ERREUR : Equipe #1 manquante !');
        ELSIF (SQLERRM LIKE '%NN_MATCHV2_NE2%') THEN
            RAISE_APPLICATION_ERROR(-20003,'ERREUR : Equipe #2 manquante !');
        ELSIF (SQLERRM LIKE '%CK_MATCHV2_NE1_NE2%') THEN
            RAISE_APPLICATION_ERROR(-20004,'ERREUR : Equipe ne peut pas jouer contre
elle-même !');
        ELSIF (SQLERRM LIKE '%CK_MATCHV2_SCR1%') THEN
            RAISE_APPLICATION_ERROR(-20005,'ERREUR : Le score 1 doit être strictement
positif !');
        ELSE RAISE_APPLICATION_ERROR(-20006,'ERREUR : Le score 2 doit être strictement positif !');
        END IF;
    WHEN fk_erreur THEN
```

```

        IF (SQLERRM LIKE '%FK_MATCHV2_EQUIPE1%') THEN
            RAISE_APPLICATION_ERROR(-20007,'ERREUR : Equipe #1 inconnue !');
        ELSIF (SQLERRM LIKE '%FK_MATCHV2_EQUIPE2%') THEN
            RAISE_APPLICATION_ERROR(-20008,'ERREUR : Equipe #2 inconnue !');
        ELSIF (SQLERRM LIKE '%FK_MATCHV2_ARBITREV2%') THEN
            RAISE_APPLICATION_ERROR(-20009,'ERREUR : Numero d`arbitre invalide !');
        ELSE RAISE_APPLICATION_ERROR(-20010,'ERREUR : Stade inconnu !');
        END IF;
    WHEN DUP_VAL_ON_INDEX THEN
        IF (SQLERRM LIKE '%UK_MATCHV2_NE1_NE2%') THEN
            RAISE_APPLICATION_ERROR(-20011,'ERREUR : Equipes se rencontrent déjà !');
        END IF;
END;

```

```

create or replace PROCEDURE AjouterMatchV2

```

```

(
    p_jrn    MatchV2.jrn%TYPE,
    -- p_dtm    MatchV2.dtm%TYPE,
    p_stade  MatchV2.stade%TYPE,
    p_na     MatchV2.na%TYPE,
    p_ne1    MatchV2.ne1%TYPE,
    p_ne2    MatchV2.ne2%TYPE
    --p_scr1    MatchV2.scr1%TYPE,
    --p_scr2    MatchV2.scr2%TYPE
)
IS
    fk_erreur EXCEPTION;
    PRAGMA EXCEPTION_INIT(fk_erreur, -2291);
    ck_erreur EXCEPTION;
    PRAGMA EXCEPTION_INIT(ck_erreur, -2290);
BEGIN
    INSERT INTO MatchV2 (nm, jrn, stade, na, nel, ne2, scr1, scr2)
    VALUES (seq_matchv2.NEXTVAL, p_jrn, p_stade, p_na, p_ne1, p_ne2, 20, 58);
    COMMIT;
EXCEPTION
    WHEN ck_erreur THEN
        IF (SQLERRM LIKE '%CK_MATCHV2_JRN%') THEN
            RAISE_APPLICATION_ERROR(-20001,'ERREUR : Journée invalide !');
        ELSIF (SQLERRM LIKE '%NN_MATCHV2_NE1%') THEN
            RAISE_APPLICATION_ERROR(-20002,'ERREUR : Equipe #1 manquante !');
        ELSIF (SQLERRM LIKE '%NN_MATCHV2_NE2%') THEN
            RAISE_APPLICATION_ERROR(-20003,'ERREUR : Equipe #2 manquante !');
        ELSIF (SQLERRM LIKE '%CK_MATCHV2_NE1_NE2%') THEN
            RAISE_APPLICATION_ERROR(-20004,'ERREUR : Equipe ne peut pas jouer contre elle-même !');
        ELSIF (SQLERRM LIKE '%CK_MATCHV2_SCR1%') THEN

```

```

        RAISE_APPLICATION_ERROR(-20005,'ERREUR : Le score 1 doit être strictement positif !');
    ELSE RAISE_APPLICATION_ERROR(-20006,'ERREUR : Le score 2 doit être strictement positif !');
    END IF;
WHEN fk_erreur THEN
    IF (SQLERRM LIKE '%FK_MATCHV2_EQUIPE1%') THEN
        RAISE_APPLICATION_ERROR(-20007,'ERREUR : Equipe #1 inconnue !');
    ELSIF (SQLERRM LIKE '%FK_MATCHV2_EQUIPE2%') THEN
        RAISE_APPLICATION_ERROR(-20008,'ERREUR : Equipe #2 inconnue !');
    ELSIF (SQLERRM LIKE '%FK_MATCHV2_ARBITREV2%') THEN
        RAISE_APPLICATION_ERROR(-20009,'ERREUR : Numero d'arbitre invalide !');
    ELSE RAISE_APPLICATION_ERROR(-20010,'ERREUR : Stade inconnu !');
    END IF;
WHEN DUP_VAL_ON_INDEX THEN
    IF (SQLERRM LIKE '%UK_MATCHV2_NE1_NE2%') THEN
        RAISE_APPLICATION_ERROR(-20011,'ERREUR : Equipes se rencontrent déjà !');
    END IF;
END;

```

Elément Procedure AJOUTERMATCHV2 compilé

Testons quelques cas d'erreurs :

Journée invalide :

```
EXEC AJOUTERMATCHV2 (9, 'Aviva Stadium', 4, 'BLA', 'WAL');
```

```

Rapport d'erreur -
ORA-20001: ERREUR : Journée invalide !

```

Équipe 2 inconnue :

```
EXEC AJOUTERMATCHV2 (4, 'Aviva Stadium', 4, 'BLA', 'POL');
```

```

Rapport d'erreur -
ORA-20008: ERREUR : Equipe #2 inconnue !

```

Arbitre inconnu :

```
EXEC AJOUTERMATCHV2 (4, 'Aviva Stadium', 40, 'BLA', 'FRA');
```

```

Rapport d'erreur -
ORA-20009: ERREUR : Numero d'arbitre invalide !

```

Équipes qui se rencontrent déjà dans un autre match :

```
EXEC AJOUTERMATCHV2 (4, 'Aviva Stadium', 4, 'ITA', 'SCO');
```

Rapport d'erreur -

ORA-20011: ERREUR : Equipes se rencontrent déjà !

Enfin, une insertion d'un match valide :

```
EXEC AJOUTERMATCHV2 (5, 'Aviva Stadium', 4, 'BLA', 'SCO');
```

Procédure PL/SQL terminée.

NM	JRN	DTM	STADE	NA	NE1	NE2	SCR1	SCR2
1	1	05/02/22	Aviva Stadium	1	IRL	WAL	29	7
2	1	05/02/22	Murrayfield Stadium	2	SCO	ENG	20	17
3	1	06/02/22	Stade de France	3	FRA	ITA	37	10
4	2	12/02/22	Millennium Stadium	4	WAL	SCO	20	17
5	2	12/02/22	Stade de France	5	FRA	IRL	30	24
6	2	13/02/22	Stade olympique	6	ITA	ENG	0	33
7	3	26/02/22	Murrayfield Stadium	7	SCO	FRA	17	36
8	3	26/02/22	Stade de Twickenham	3	ENG	WAL	23	19
9	3	27/02/22	Aviva Stadium	8	IRL	ITA	57	6
10	4	11/03/22	Millennium Stadium	9	WAL	FRA	9	13
11	4	12/03/22	Stade olympique	10	ITA	SCO	22	33
12	4	12/03/22	Stade de Twickenham	11	ENG	IRL	15	32
13	5	19/03/22	Millennium Stadium	12	WAL	ITA	21	22
14	5	19/03/22	Aviva Stadium	13	IRL	SCO	26	5
15	5	19/03/22	Stade de France	1	FRA	ENG	25	13
16	5	(null)	Aviva Stadium	4	BLA	SCO	20	58

III- Exécution des tests :

Désormais, les tests n°2 et n°3 passent, mais pas le n°1 (il attaque directement une table, alors que les 2 autres sont sur une vue et une procédure).