

SQL DANS UN LANGAGE DE PROG

SEANCE N°4

UNIDDEV

5)

```
ALTER TABLE Travailler DROP CONSTRAINT PK_TRAVAILLER;
```

```
ALTER TABLE TRAVAILLER ADD CONSTRAINT pk_Travailler  
PRIMARY KEY (CODESALARIE, DATETRAVAIL);
```

OU

```
ALTER TABLE TRAVAILLER ADD CONSTRAINT UNIQUE (CODESALARIE,  
DATETRAVAIL);
```

6)

```
ALTER TABLE EQUIPES ADD CONSTRAINT fk_Equipe_ChefAffecte  
FOREIGN KEY (CODEEQUIPE, CODESALARIECHEF) REFERENCES  
ETREAFFECTE(CODEEQUIPE, CODESALARIE);
```

7)

```
CREATE OR REPLACE PROCEDURE AjouterJourneeTravail  
(p_codeSalarie Travailler.codeSalarie%TYPE, p_codeProjet  
Travailler.codeProjet%TYPE, p_dateTravail  
Travailler.dateTravail%TYPE) IS
```

```
BEGIN
```

```
    INSERT INTO TRAVAILLER(codeSalarie, codeProjet,  
dateTravail) VALUES (p_codeSalarie, p_codeProjet,  
p_dateTravail);
```

```
    UPDATE SALARIES
```

```
    SET NBTOTALJOURNEESTRAVAIL = NBTOTALJOURNEESTRAVAIL+1
```

```
    WHERE CODESALARIE=p_codeSalarie;
```

```
END;
```

8)

```
CREATE OR REPLACE PROCEDURE AffecterSalarieEquipe  
(p_codeSalarie EtreAffecte.codeSalarie%TYPE, p_codeEquipe  
EtreAffecte.codeEquipe%TYPE) IS
```

```
    v_nbEquipe NUMBER;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO v_nbEquipe
```

```
    FROM ETREAFFECTE
```

```
    WHERE CODESALARIE = p_codeSalarie;
```

```
    IF v_nbEquipe < 3 THEN
```

```
        INSERT INTO ETREAFFECTE(CODESALARIE, CODEEQUIPE)  
VALUES(p_codeSalarie, p_codeEquipe);
```

```
    ELSE
```

```
        RAISE_APPLICATION_ERROR(-2001, 'Le salarié est  
déjà affecté à au moins 3 équipes');
```

```
    end if;
```

```
END;
```

9)

```
CREATE OR REPLACE PROCEDURE  
AjouterProjetExterne(p_codeProjet  
PROJETSEXTERNES.CODEPROJET%TYPE, p_nomProjet  
PROJETSEXTERNES.nomProjet%TYPE, p_villeProjet  
PROJETSEXTERNES.VILLEPROJET%TYPE, p_clientProjet  
PROJETSEXTERNES.CLIENTPROJET%TYPE, p_codeEquipe  
PROJETSEXTERNES.CODEEQUIPE%TYPE) IS
```

```

        v_numberProjetExterne PROJETSEXTERNES.CODEPROJET%TYPE;
BEGIN
    SELECT CODEPROJET INTO v_numberProjetExterne
    FROM PROJETSINTERNES
    WHERE CODEPROJET=p_codeProjet;

```

```

    INSERT INTO PROJETSEXTERNES(CODEPROJET, NOMPROJET,
VILLEPROJET, CLIENTPROJET, CODEEQUIPE) VALUES
(p_codeProjet, p_nomProjet, p_villeProjet, p_clientProjet,
p_codeEquipe);

```

```

    EXCEPTION WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20002, 'Le codeProjet
existe déjà dans les projets internes');

```

```

END;

```

```

/

```

```

SHOW ERRORS

```

```

CREATE OR REPLACE PROCEDURE
AjouterProjetInterne(p_codeProjet
PROJETSINTERNES.CODEPROJET%TYPE, p_nomProjet
PROJETSINTERNES.nomProjet%TYPE, p_serviceProjet
PROJETSINTERNES.SERVICEPROJET%TYPE, p_codeSalarieResp
PROJETSINTERNES.CODESALARIERESPONSABLE%TYPE) IS

```

```

        v_codeProjetExterne PROJETSEXTERNES.CODEPROJET%TYPE;
BEGIN
    SELECT CODEPROJET INTO v_codeProjetExterne
    FROM PROJETSEXTERNES
    WHERE CODEPROJET=p_codeProjet;

```

```

        INSERT INTO PROJETSINTERNES(CODEPROJET, NOMPROJET,
SERVICEPROJET, CODESALARIERESPONSABLE) VALUES
(p_codeProjet, p_nomProjet, p_serviceProjet,
p_codeSalarieResp);

```

```

        EXCEPTION WHEN NO_DATA_FOUND THEN

            RAISE_APPLICATION_ERROR(-20002, 'Le codeProjet
existe déjà dans les projets externes');

END;
/
SHOW ERRORS

```

10)

```

CREATE OR REPLACE PROCEDURE
AjouterJourneeTravail(p_codeSalarie IN
TRAVAILLER.CODESALARIE%TYPE, p_codeProjet IN
TRAVAILLER.CODEPROJET%TYPE, p_dateTravail IN
TRAVAILLER.DATETRAVAIL%TYPE) IS
v_nb NUMBER;
BEGIN

    SELECT COUNT(*) INTO v_nb FROM TRAVAILLER t JOIN
PROJETSEXTERNES P on P.CODEPROJET = t.CODEPROJET JOIN
ETREAFFECTE e ON p.CODEEQUIPE=e.CODEEQUIPE WHERE
P.CODEPROJET=p_codeProjet AND e.CODESALARIE=p_codeSalarie;

```

```

    IF v_nb = 0 THEN

        RAISE_APPLICATION_ERROR(-20004, 'Le salarié n''est
pas affecté au projet ' || p_codeProjet);

    ELSE

```

```

        INSERT INTO TRAVAILLER VALUES (p_codeSalarie,
p_codeProjet, p_dateTravail);

        UPDATE SALARIES SET NBTOTALJOURNEESTRAVAIL =
NBTOTALJOURNEESTRAVAIL+1 WHERE CODESALARIE=p_codeSalarie;

    end if;

END;

```

11°)

```

CREATE OR REPLACE TRIGGER AFT_INS_TRAVAILLER
AFTER INSERT ON TRAVAILLER
FOR EACH ROW
BEGIN

    UPDATE SALARIES SET NBTOTALJOURNEESTRAVAIL =
NBTOTALJOURNEESTRAVAIL + 1 WHERE
CODESALARIE=:NEW.CODESALARIE;

END;

```

12°)

```

CREATE OR REPLACE TRIGGER AFT_INS_ETAFF
BEFORE INSERT ON ETREAFPECTE
FOR EACH ROW
DECLARE

    v_nb NUMBER;

BEGIN

    SELECT COUNT(*) INTO v_nb
    FROM ETREAFPECTE
    WHERE CODESALARIE=:NEW.CODESALARIE;

```

```

        IF v_nb > 3 THEN
            RAISE_APPLICATION_ERROR(-20012, 'Le salarié est
déjà affecté à au moins 3 equipe');
        end if;
    end;

```

13°)

```

CREATE OR REPLACE TRIGGER tr_jour_travail
AFTER INSERT OR DELETE OR UPDATE OF codeSalarie ON
Travailler
FOR EACH ROW
BEGIN
    IF (INSERTING) THEN
        UPDATE Salaries SET nbTotalJourneesTravail =
nbTotalJourneesTravail + 1 WHERE codeSalarie
= :OLD.codeSalarie;
    END IF;
    IF (DELETING) THEN
        UPDATE Salaries SET nbTotalJourneesTravail =
nbTotalJourneesTravail - 1 WHERE codeSalarie
= :OLD.codeSalarie;
    END IF;
    IF (UPDATING) THEN
        UPDATE Salaries SET nbTotalJourneesTravail =
nbTotalJourneesTravail + 1 WHERE codeSalarie
= :NEW.codeSalarie;
        UPDATE Salaries SET nbTotalJourneesTravail =
nbTotalJourneesTravail - 1 WHERE codeSalarie
= :OLD.codeSalarie;
    END IF;
END;
/
SHOW ERRORS

```

14°)

```

CREATE OR REPLACE TRIGGER BF_INS_PRJT_INT
BEFORE INSERT ON PROJETSINTERNES
FOR EACH ROW

```

```

DECLARE

    v_existeProjetExterne NUMBER;

BEGIN

    SELECT COUNT(*) INTO v_existeProjetExterne
    FROM PROJETSEXTERNES
    WHERE CODEPROJET= :NEW.CODEPROJET;

```

```

    IF v_existeProjetExterne =1 THEN

        RAISE_APPLICATION_ERROR(-20001, 'Le projet est
déjà un projet externe');

    end if;

END;

```

15°)

```

CREATE OR REPLACE TRIGGER BF_INS_TRAV
BEFORE INSERT ON TRAVAILLER
FOR EACH ROW
DECLARE

    v_bool NUMBER;

    v_codeEquipe ETREAFFECTE.CODEEQUIPE%TYPE;

BEGIN

    SELECT CODEEQUIPE INTO v_codeEquipe
    FROM PROJETSEXTERNES
    WHERE PROJETSEXTERNES.CODEPROJET=:NEW.CODEPROJET;

    SELECT COUNT(*) INTO v_bool

```

```

FROM ETREAFFECTE

WHERE ETREAFFECTE.CODESALARIE=:NEW.CODESALARIE AND
ETREAFFECTE.CODEEQUIPE=v_codeEquipe;

```

```

IF v_bool = 1 THEN

    RAISE_APPLICATION_ERROR(-2015, 'Le salarié n''est
pas affectue dans lequipe');

end if;

end;

```

16°)

```

CREATE OR REPLACE VIEW affectations (codeSalarie,
nomSalarie, prenomSalarie, codeEquipe, nomEquipe) AS
SELECT codeSalarie, nomSalarie, prenomSalarie, codeEquipe,
nomEquipe FROM SALARIES S JOIN EQUIPES E on S.CODESALARIE
= E.CODESALARIECHEF;

```

17°)

```

CREATE OR REPLACE TRIGGER tr_affectations
INSTEAD OF INSERT ON AFFECTATIONS
FOR EACH ROW
DECLARE

    v_existeSalarie NUMBER;
    v_existeEquipe NUMBER;

BEGIN

    SELECT COUNT (*) INTO v_existeSalarie FROM SALARIES
    WHERE CODESALARIE= :NEW.CODESALARIE;

    SELECT COUNT (*) INTO v_existeEquipe FROM EQUIPES

```



```
WHERE CODEEQUIPE= :NEW.CODEEQUIPE;

IF (v_existeEquipe=0) THEN

    INSERT INTO EQUIPES (CODEEQUIPE, NOMEQUIPE,
CODESALARIECHEF) VALUES ( :NEW.CODEEQUIPE, :NEW.NOMEQUIPE,
NULL);

end if;

IF (v_existeSalarie=0) THEN

    INSERT INTO SALARIES (CODESALARIE, NOMSALARIE,
PRENOMSALARIE, NBTOTALJOURNEESTRAVAIL) VALUES
( :NEW.CODESALARIE, :NEW.NOMSALARIE, :NEW.PRENOMSALARIE,
0);

end if;

end;
```