ENICARTHGE

DÉCEMBRE 2022

RAPPORT PROJET SGBD

Gestion de Specatcles

PRÉSENTÉ PAR

KCHAOU Amira GA

SOMMAIRE

1. CREATION DES TABLES

2. EXTRACTION ET INSERTION DES DONNEES

3. GESTION DES UTILISATEURS ET PRIVILEGES

4. CAS D'UTILISATION « GESTION DE SPECTACLES »

1. CREATION DES TABLES

projetSpectacles

Feuille de calcul | Query Builder

```
CREATE TABLE Lieu
    (idLieu INTEGER PRIMARY KEY,
    NomLieu VARCHAR2 (30) NOT NULL,
    Adresse VARCHAR2 (100) NOT NULL,
    Ville VARCHAR2 (100) NOT NULL,
    capacite NUMBER NOT NULL,
    CHECK (capacite BETWEEN 100 and 2000)
    );
```

Sortie de script | Tâche terminée en 0,08 secondes

```
Table LIEU créé(e).
```

projetSpectacles | LIEU

Feuille de calcul | Query Builder

```
CREATE TABLE Artiste
    (idArt INTEGER PRIMARY KEY,
    NomArt VARCHAR2 (30) NOT NULL,
    PrenomArt VARCHAR2 (30) NOT NULL,
    specialite VARCHAR2 (10) NOT NULL,
    CONSTRAINT ck_spec CHECK (specialite IN('danseur','chanteur','magicien','humouriste','acteur'
    ,'imitateur','musicien'))
    );
```

Sortie de script | Tâche terminée en 0,031 secondes

```
Table LIEU créé(e).

Table ARTISTE créé(e).
```

Activer Windows
Accédez aux paramètres pour activer Windows.

projetSpectacles | LIEU | 0,021 secondes

Feuille de calcul | Query Builder

```
CREATE TABLE SPECTACLE
    (idSpec INTEGER PRIMARY KEY,
    Titre VARCHAR2 (40) NOT NULL,
    dateS DATE NOT NULL ,
    h_debut NUMBER(4,2) NOT NULL,
    dureeS NUMBER(4,2) NOT NULL,
    nbrSpectateur INTEGER NOT NULL,
    idLieu INTEGER,
    CONSTRAINT chk_spect_durees CHECK (dureeS BETWEEN 1 AND 3),
    CONSTRAINT FK_spect_Lieux FOREIGN KEY(idLieu)REFERENCES Lieu(idLieu)
    );
```

-Trigger pour vérifier que la date du Spectacle et supérieure ou égale à la date actuelle

```
CREATE OR REPLACE TRIGGER trg_check_dateS
  BEFORE INSERT OR UPDATE ON SPECTACLE
  FOR EACH ROW
BEGIN

  IF( :new.dateS <= SYSDATE)
  THEN
    RAISE_APPLICATION_ERROR( -20001,
          'Invalid date Spectacle: dateS must be greater than the current date - value = ' ||
          to_char( :new.dateS, 'YYYY-MM-DD HH24:MI:SS' ) );
  END IF;
END;
```

--> Compilation du Trigger:

```
Elément Trigger TRG_CHECK_DATES compilé
```

-Trigger pour vérifier que le nombre de spectateurs est inférieur ou égale capacité du lieu

```
CREATE OR REPLACE TRIGGER trg_check_nbrSpec
  BEFORE INSERT OR UPDATE ON SPECTACLE
  FOR EACH ROW
DECLARE
cp lieu.capacite%type;
id number;
BEGIN
  id:=:new.idLieu;
  select l.capacite into cp from lieu l where l.idLieu=id ;


  IF( :new.nbrSpectateur > cp)
  THEN
    RAISE_APPLICATION_ERROR(20100,'capacité < nbr de spectateurs');
   END IF;

END;
```

--> Compilation du Trigger:

```
Elément Trigger TRG_CHECK_NBRSPEC compilé
```

```
CREATE TABLE Rubrique
    (
     idRub INTEGER PRIMARY KEY,
     idSpec INTEGER NOT NULL,
     idArt INTEGER NOT NULL,
     dateRub DATE NOT NULL ,
     H_debutR NUMBER(4,2) NOT NULL,
     dureeRub NUMBER(4,2) NOT NULL,
     typeRub VARCHAR2(10),
     CONSTRAINT fk_rub_spect FOREIGN KEY(idSpec) REFERENCES SPECTACLE(idSpec) ON DELETE CASCADE,
     CONSTRAINT fk_rub_art FOREIGN KEY(idArt)  REFERENCES Artiste(idArt) ON DELETE CASCADE ,
     CONSTRAINT ck_typeR check( typeRub IN('comedie','theatre','dance','imitation','magie','musique','chant'))
     );

CREATE OR REPLACE TRIGGER trg_check_dateR_dureeR
  BEFORE INSERT OR UPDATE ON Rubrique
```

```
Table RUBRIQUE créé(e).
```

-Trigger pour vérifier que le nombre de rubriques ne dépasse pas 3 dans un spectacle , que la date d'une rubrique est la meme que l'id du spectacle choisi , que l'heure de la rubrique correspond à celle du spectacle et que la durée ne dépasse pas celle du spectacle.

```
CREATE OR REPLACE TRIGGER trg_check_dateR_dureeR
  BEFORE INSERT OR UPDATE ON Rubrique
  FOR EACH ROW
DECLARE
Tdate spectacle.dateS%type;
Tduree spectacle.dureeS%type;
Theure spectacle.h_debut%type;
nbrRub number;
BEGIN
  select count(idSpec) into nbrRub from Rubrique
  where idSpec= :new.idSpec;
  IF (nbrRub =3)
  THEN RAISE_APPLICATION_ERROR(-20105,'nbr de rubrique atteint');
  END IF;
  select dateS,dureeS,h_debut into Tdate,Tduree,Theure from spectacle
  where spectacle.idSpec=:new.idSpec;
  IF( :new.dateRub <> Tdate)
  THEN
    RAISE_APPLICATION_ERROR(-20100,'la date de la rubrique diff de celle du spectacle');
  END IF;
```

```
  IF( :new.H_debutR < Theure)
THEN
  RAISE_APPLICATION_ERROR(-20101,'l heure de la rubrique >  celle du spectacle');
END IF;
    IF( :new.dureeRub > Tduree)
THEN
  RAISE_APPLICATION_ERROR(-20102,'la duree de la rubrique > de celle du spectacle');
END IF;
END;
```

--> Compilation du Trigger:

```
Elément Trigger TRG_CHECK_DATER_DUREER compilé
```

```
CREATE TABLE BILLET
    (idBillet INTEGER PRIMARY KEY,
    categorie VARCHAR2(10),
    prix NUMBER(5,2) NOT NULL,
    idspec INTEGER NOT NULL ,
    Vendu VARCHAR(3) NOT NULL,

CONSTRAINT chk_billet_PRIX CHECK(prix BETWEEN 10 AND 300),
CONSTRAINT fk_billet_spec FOREIGN KEY (idspec)REFERENCES spectacle,
CONSTRAINT chk_billet_vendu CHECK(vendu IN ('Oui','Non')),
CONSTRAINT chk_categorie CHECK(categorie IN('gold','silver','normal'))
);
```

Sortie de script

Tâche terminée en 0,039 secondes

```
Table BILLET créé(e).
```

```
CREATE TABLE CLIENT
      ( idClt INTEGER PRIMARY KEY,
       nomClt VARCHAR(20),
       prenomClt VARCHAR(20),
       tel VARCHAR(8),
       email VARCHAR(50) NOT NULL,
       motP VARCHAR(20) NOT NULL,
       CONSTRAINT chk_valEmail CHECK( email like  '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$'),
       CONSTRAINT chk_valMot  CHECK (tel like '[0-9]{8}')
);
```

Sortie de script

Tâche terminée en 0,037 secondes

```
Table CLIENT créé(e).
```

Activer Windows
Accédez aux paramètres pour activer Windc

2. EXTRACTION ET INSERTION DES DONNEES

-Creation des séquences :

```
CREATE SEQUENCE seq_client START WITH 1;
CREATE SEQUENCE seq_artiste START WITH 1;
CREATE SEQUENCE seq_billet START WITH 1;
CREATE SEQUENCE seq_lieu START WITH 1;
CREATE SEQUENCE seq_rubrique START WITH 1;
CREATE SEQUENCE seq_spectacle START WITH 1;
```

Sortie de script | Tâche terminée en 0,066 secondes

```
Sequence SEQ_ARTISTE créé(e).

Sequence SEQ_BILLET créé(e).

Sequence SEQ_LIEU créé(e).

Sequence SEQ_RUBRIQUE créé(e).

Sequence SEQ_SPECTACLE créé(e).
```

-Ajout des données :

- table client ::

```
INSERT INTO client values(seq_client.NEXTVAL,'Dridi','Ali','24313180','ali@gamil.com','123456');
```

Sortie de script | Tâche terminée en 0,032 secondes

```
1 ligne inséré.
```

--> verification du num tel ::

```
INSERT INTO client values(seq_client.NEXTVAL,'Dridi','Ali','4313180','ali@gamil.com','123456');
```

Sortie de script | Tâche terminée en 0,024 secondes

```
Erreur commençant à la ligne: 141 de la commande -
INSERT INTO client values(seq_client.NEXTVAL,'Dridi','Ali','4313180','ali@gamil.com','123456')
Rapport d'erreur -
ORA-02290: check constraint (PROJETSPECTACLES.CHK_VALTEL) violated
```

--> verification d' email ::

```
INSERT INTO client values(seq_client.NEXTVAL,'Dridi','Ali','24313180','aligamil.com','123456');
```

Sortie de script | Tâche terminée en 0,028 secondes

```
Erreur commençant à la ligne: 141 de la commande -
INSERT INTO client values(seq_client.NEXTVAL,'Dridi','Ali','24313180','aligamil.com','123456')
Rapport d'erreur -
ORA-02290: check constraint (PROJETSPECTACLES.CHK_VALEMAIL) violated
```

- table lieu ::

```
'carthage',866);
INSERT INTO lieu values(seq_lieu.NEXTVAL,'lagora','Rue 1 la marsa','la marsa',603);
```

Résultat de requête | Sortie de script | Tâche terminée en 0,017 secondes

```
1 ligne inséré.
```

--> verification de la capacité::

```
INSERT INTO lieu values(seq_lieu.NEXTVAL,'lagora','Rue 1 la marsa','la marsa',3000);
```

Résultat de requête | Sortie de script | Tâche terminée en 0,026 secondes

```
1 ligne inséré.


Erreur commençant à la ligne: 147 de la commande -
INSERT INTO lieu values(seq_lieu.NEXTVAL,'lagora','Rue 1 la marsa','la marsa',3000)
Rapport d'erreur -
ORA-02290: check constraint (PROJETSPECTACLES.SYS_C007358) violated
```

- table artiste ::

```
INSERT INTO artiste values (seq_artiste.NEXTVAL,'belkadhi','najib','acteur');
INSERT INTO artiste values (seq_artiste.NEXTVAL,'Attia','sarah','chanteur');
```

Sortie de script | Tâche terminée en 0,028 secondes

```
1 ligne inséré.


1 ligne inséré.
```

- table Spectacle ::

```
INSERT INTO spectacle values (seq_spectacle.NEXTVAL,'yyy','21/12/2022',16.00,3,400,1);
```

Résultat de requête | Sortie de script | Tâche terminée en 0,019 secondes

```
1 ligne inséré.
```

--> verification du trigger que la date du spectacle est sup a la date actuelle ::

```
INSERT INTO spectacle values (seq_spectacle.NEXTVAL,'yyy','01/12/2022',16.00,3,400,1);
```

Résultat de requête | Sortie de script | Tâche terminée en 0,038 secondes

```
Erreur commençant à la ligne: 155 de la commande -
INSERT INTO spectacle values (seq_spectacle.NEXTVAL,'yyy','01/12/2022',16.00,3,400,1)
Rapport d'erreur -
ORA-20001: Invalid date Spectacle: dateS must be greater than the current date - value = 2022-12-01 00:00:00
ORA-06512: at "PROJETSPECTACLES.TRG_CHECK_DATES", line 6
ORA-04088: error during execution of trigger 'PROJETSPECTACLES.TRG_CHECK_DATES'
```

--> verification du trigger que le nbr de spectateur est inf à la capacité du lieu ::

```
INSERT INTO spectacle values (seq_spectacle.NEXTVAL,'zzz','26/12/2022',14.00,3,900,2);
```

Résultat de requête | Sortie de script | Tâche terminée en 0,022 secondes

```
Erreur commençant à la ligne: 156 de la commande -
INSERT INTO spectacle values (seq_spectacle.NEXTVAL,'zzz','26/12/2022',14.00,3,900,2)
Rapport d'erreur -
ORA-21000: error number argument to raise_application_error of 20100 is out of range
ORA-06512: at "PROJETSPECTACLES.TRG_CHECK_NBRSPEC", line 11
ORA-04088: error during execution of trigger 'PROJETSPECTACLES.TRG_CHECK_NBRSPEC'
```

- table Rubrique::

```
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,19,1,'21/12/2022',16.00,1,'theatre');
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,19,1,'21/12/2022',16.00,1,'musique');
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,19,1,'21/12/2022',16.00,1,'dance');
```

Sortie de script | Résultat de requête

Tâche terminée en 0,032 secondes

```
1 ligne inséré.

1 ligne inséré.

1 ligne inséré.
```

--> verification du trigger qu'un specatcle ne depasse pas 3 rubriques ::

```
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,19,1,'21/12/2022',16.00,1,'magie');
```

Sortie de script | Résultat de requête

Tâche terminée en 0,032 secondes

```
1 ligne inséré.


Erreur commençant à la ligne: 167 de la commande -
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,19,1,'21/12/2022',16.00,1,'magie')
Rapport d'erreur -
ORA-20105: nbr de rubrique atteint
ORA-06512: at "PROJETSPECTACLES.TRG_CHECK_DATER_DUREER", line 12
ORA-04088: error during execution of trigger 'PROJETSPECTACLES.TRG_CHECK_DATER_DUREER'
```

Sortie de script | Résultat de requête

SQL | Toutes les lignes extraites : 7 en 0,001 secondes

	IDSPEC	TITRE	DATES	H_DEBUT	DUREES	NBRSPECTATEUR	IDLIEU
1	7	xxx	24/12/22	16	3	900	1
2	15	xxx	24/12/22	16	3	300	1
3	19	yyy	21/12/22	16	3	400	1
4	20	yyy	21/12/22	16	3	400	1
5	23	yyy	24/12/22	16	3	200	1
6	24	zzz	26/12/22	14	3	300	2
7	41	aaa	27/12/22	14	2	400	2

--> verification du trigger de la date de la rubrique ::

```
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,41,1,'21/12/2022',14.00,1,'magie');
```

Sortie de script | Résultat de requête

Tâche terminée en 0,025 secondes

```
ORA-04088: error during execution of trigger 'PROJETSPECTACLES.TRG_CHECK_DATER_DUREER'


Erreur commençant à la ligne: 169 de la commande -
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,41,1,'21/12/2022',14.00,1,'magie')
Rapport d'erreur -
ORA-20100: la date de la rubrique diff de celle du spectacle
ORA-06512: at "PROJETSPECTACLES.TRG_CHECK_DATER_DUREER", line 21
ORA-04088: error during execution of trigger 'PROJETSPECTACLES.TRG_CHECK_DATER_DUREER'
```

--> verification du trigger de l'heure de la rubrique ::

```
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,41,1,'27/12/2022',12.00,1,'magie');
```

Sortie de script | Résultat de requête

Tâche terminée en 0,028 secondes

```
ORA-04088: error during execution of trigger 'PROJETSPECTACLES.TRG_CHECK_DATER_DUREER'


Erreur commençant à la ligne: 169 de la commande -
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,41,1,'27/12/2022',12.00,1,'magie')
Rapport d'erreur -
ORA-20101: l heure de la rubrique >  celle du spectacle
ORA-06512: at "PROJETSPECTACLES.TRG_CHECK_DATER_DUREER", line 25
ORA-04088: error during execution of trigger 'PROJETSPECTACLES.TRG_CHECK_DATER_DUREER'
```

--> verification du trigger de la durée de la rubrique ::

```
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,41,1,'27/12/2022',14.00,5,'magie');
```

Sortie de script | Résultat de requête

Tâche terminée en 0,022 secondes

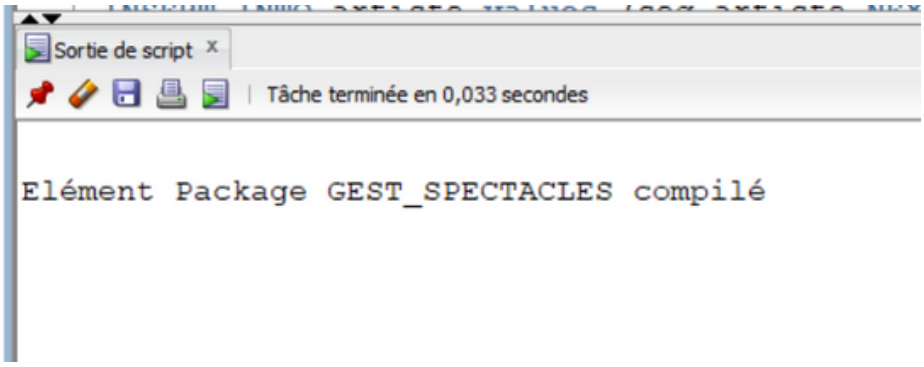
```
ORA-04088: error during execution of trigger 'PROJETSPECTACLES.TRG_CHECK_DATER_DUREER'


Erreur commençant à la ligne: 169 de la commande -
INSERT INTO rubrique values(seq_rubrique.NEXTVAL,41,1,'27/12/2022',14.00,5,'magie')
Rapport d'erreur -
ORA-20102: la duree de la rubrique > de celle du spectacle
ORA-06512: at "PROJETSPECTACLES.TRG_CHECK_DATER_DUREER", line 29
ORA-04088: error during execution of trigger 'PROJETSPECTACLES.TRG_CHECK_DATER_DUREER'
```

4. CAS D'UTILISATION « GESTION DE SPECTACLES »

-Création du package gestion du specatacle : :

```
CREATE OR REPLACE PACKAGE gest_Spectacles IS
procedure ajoutSpectacle(idSpec spectacle.idSpec%type,
titre spectacle.titre%type,dateS spectacle.dateS%type,
h_debut spectacle.h_debut%type,dureeS  spectacle.dureeS%type,nbrSpec spectacle.nbrSpectateur%type,
idLieu spectacle.idLieu%type);
procedure annulerSpectacle(idSp spectacle.idSpec%type);
procedure modifierSpectacle(idSp spectacle.idSpec%type,
titreSp spectacle.titre%type,dateSp spectacle.dateS%type,
h_debutSp spectacle.h_debut%type,dureeSp  spectacle.dureeS%type,
nbrSp spectacle.nbrSpectateur%type, idLieuSp spectacle.idLieu%type);
function chercherSpectacle(idSp spectacle.idSpec%type,titre spectacle.titre%type)
RETURN spectacle%ROWTYPE ;
procedure ajoutRubrique(idRub rubrique.idRub%type,idSpec rubrique.idSpec%type,idArt rubrique.idArt%type,
dateRub rubrique.dateRub%type,
h_debutR rubrique.h_debutR%type,dureeRub rubrique.dureeRub%type,typeRub rubrique.typeRub%type);
procedure modifierRubrique(idRb rubrique.idRub%type,
idArtR rubrique.idArt%type,h_debutRb rubrique.h_debutR%type,
dureeRb rubrique.dureeRub%type);
function chercherRubrique(idSpec rubrique.idSpec%type,nomArt artiste.nomArt%type)
RETURN rubrique%ROWTYPE;
procedure supprimerRubrique(idRub rubrique.idRub%type);
END gest_Spectacles;
```



-Création du body package gestion du specatacle : :

- Ajouter spectacle::

```
CREATE OR REPLACE PACKAGE BODY gest_Spectacles IS

procedure ajoutSpectacle(idSpec spectacle.idSpec%type,
titre spectacle.titre%type,dateS spectacle.dateS%type,
h_debut spectacle.h_debut%type,dureeS  spectacle.dureeS%type,nbrSpec spectacle.nbrSpectateur%type,
idLieu spectacle.idLieu%type) IS
errAjout EXCEPTION;
nbrSp number;
BEGIN
SELECT count(*) into nbrSp from Spectacle S
where S.idLieu=idLieu and S.h_debut=h_debut and S.dateS=dateS;
IF (nbrSp <> 0) THEN raise errAjout;
END IF;
INSERT INTO SPECTACLE VALUES (idSpec ,titre ,dateS,h_debut ,dureeS  ,nbrSpec ,idLieu );
COMMIT;
EXCEPTION
WHEN errAjout THEN
DBMS_OUTPUT.PUT_LINE('lieu pas disponible');
END ajoutSpectacle;
```

```
Elément Procedure AJOUTSPECTACLE compilé
```

- Annuler spectacle::

```
procedure annulerSpectacle(idSp spectacle.idSpec%type) IS
errAnn EXCEPTION;
vidSpec spectacle.idSpec%type;
BEGIN
SELECT idSpec into vidSpec from SPECTACLE ;
UPDATE SPECTACLE SET dateS=NULL
WHERE idSpec=idSp;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('spectacle inexistant');
END annulerSpectacle;
```

```
Elément Procedure ANNULERSPECTACLE compilé
```

- Modifier spectacle::

```
 procedure modifierSpectacle(idSp spectacle.idSpec%type,
titreSp spectacle.titre%type,dateSp spectacle.dateS%type,
h_debutSp spectacle.h_debut%type,dureeSp  spectacle.dureeS%type,
nbrSp spectacle.nbrSpectateur%type, idLieuSp spectacle.idLieu%type) IS
errModif EXCEPTION ;
vidSpec spectacle.idSpec%type;
vdateSp SPECTACLE.dateS%type;
heureSp SPECTACLE.h_debut%type;
nbrSpc NUMBER;
BEGIN
SELECT idSpec into vidSpec from SPECTACLE WHERE idSpec=idSp;
IF(titreSp!=null) then
update spectacle  set titre=titreSp where idSpec=idSp;
commit;
END IF;
IF(dateSp!=null) then
update spectacle  set dateS=dateSp where idSpec=idSp;
commit;
END IF;
IF(h_debutSp!=null) then
update spectacle  set h_debut=h_debutSp where idSpec=idSp;
commit;
END IF;
```

```
IF(dureeSp!=null) then
update spectacle  set dureeS=dureeSp where idSpec=idSp;
commit;
END IF;
IF(nbrSp!=null) then
update spectacle  set nbrSpectateur=nbrSp where idSpec=idSp;
commit;
END IF;
IF(idLieuSp!=null) then
select dateS,h_debut into vdateSp,heureSp from spectacle where idSpec=idSp;
SELECT count(*) into nbrSpc from Spectacle S
where S.idLieu=idLieu and S.h_debut=heureSp and S.dateS=dateSp;
IF (nbrSpc <> 0) THEN raise errModif;
ElSE
update spectacle  set idLieu=idLieuSp where idSpec=idSp;
commit;
END IF;
END IF;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('spectacle inexistant');
WHEN errModif THEN
DBMS_OUTPUT.PUT_LINE('lieu pas disponible');
END modifierSpectacle;
```

```
Sortie de script
Tâche terminée en 0,038 secondes

Elément Procedure MODIFIERSPECTACLE compilé
```

- Chercher spectacle::

```
function chercherSpectacle(idSp spectacle.idSpec%type,
titre spectacle.titre%type) RETURN spectacle%ROWTYPE IS
vSpec spectacle%ROWTYPE;
BEGIN
select * into vSpec from spectacle s where s.idSpec=idSp or s.titre=titre;
return vSpec;
END chercherSpectacle;
```

```
Elément Function CHERCHERSPECTACLE compilé
```

- Ajouter Rubrique::

```
procedure ajoutRubrique(idRub rubrique.idRub%type,idSpec rubrique.idSpec%type,idArt rubrique.idArt%type,
dateRub rubrique.dateRub%type,
h_debutR rubrique.h_debutR%type,dureeRub rubrique.dureeRub%type,typeRub rubrique.typeRub%type)IS
BEGIN
INSERT INTO rubrique VALUES (idRub ,idSpec ,idArt,dateRub ,h_debutR ,dureeRub ,typeRub );
END ajoutRubrique;
```

```
Elément Procedure AJOUTRUBRIQUE compilé
```

- Modifier Rubrique::

```
procedure modifierRubrique(idRb rubrique.idRub%type,
idArtR rubrique.idArt%type,h_debutRb rubrique.h_debutR%type,
dureeRb rubrique.dureeRub%type) IS
vidRub rubrique.idRub%type;
BEGIN
SELECT idRub into vidRub from RUBRIQUE r WHERE r.idRUB=idRb;
if(idArtR!=null) then
update rubrique  set idArt=idArtR where idRub=idRb;
commit;
end if;
if(h_debutRb!=null) then
update rubrique  set h_debutR=h_debutRb where idRub=idRb;
commit;
end if;
if(dureeRb!=null) then
update rubrique  set dureeRub=dureeRb where idRub=idRb;
commit;
end if;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('rubrique inexistant');
END modifierRubrique;
```

```
Elément Procedure MODIFIERRUBRIQUE compilé
```

- Supprimer Rubrique::

```
procedure supprimerRubrique(idRub rubrique.idRub%type) IS
BEGIN
delete from rubrique r where r.idRub=idRub;
END supprimerRubrique;
```

```
Elément Procedure SUPPRIMERRUBRIQUE compilé
```

- chercher Rubrique::

```
function chercherRubrique(idSpec rubrique.idSpec%type,nomArt artiste.nomArt%type)
RETURN rubrique%ROWTYPE IS
vRub rubrique%ROWTYPE;
idAR number:=NULL;
BEGIN
if (nomArt <> NULL ) THEN
SELECT idArt into idAR from Artiste  A
where A.nomArt=nomArt;
END IF;
select * into vRub from rubrique R where R.idSpec=idSpec or R.idArt=idArt;
return vRub;
END chercherRubrique;

END gest_Spectacles;
```

```
Elément Function CHERCHERRUBRIQUE compilé
```

--> Compilation du body package ::

```
Elément Package Body GEST_SPECTACLES compilé
```

3. GESTION DES UTILISATEURS et privileges

-Creation des users : :

```
create user AdminBD IDENTIFIED BY 123 ;
create user PlanificateurEvt IDENTIFIED BY 123;
create user GestCommandesBillets IDENTIFIED BY 123;
```

Script Output

Task completed in 0.036 seconds

```
User ADMINBD created.

User PLANIFICATEUREVT created.

User GESTCOMMANDESBILLETS created.
```

-Creation du package gestion des utilisateurs: :

```
CREATE OR REPLACE PACKAGE gest_utilisateurs IS
procedure ajout_utilisateur(nom VARCHAR(20), role VARCHAR(20));
procedure modifier_utilisateur (nom VARCHAR(20),$role VARCHAR(20));
PROCEDURE chercher_utilisateur (nom all_users.username%TYPE );
PROCEDURE supprimer_utilisateur(username VARCHAR);
END gest_utilisateurs;
```

Script Output

Task completed in 0.044 seconds

```
Package GEST_UTILISATEURS compiled
```

- Supprimer utilisateur ::

```
CREATE OR REPLACE PACKAGE BODY gest_utilisateurs IS
PROCEDURE supprimer_utilisateur(username VARCHAR) IS
    Vusername   VARCHAR2 (30);
    errsup exception;
BEGIN
        Vusername := 'DROP USER ' || username ;
        EXECUTE IMMEDIATE (Vusername);
        DBMS_OUTPUT.put_line( Vusername );
    commit ;
END supprimer_utilisateur;
```

- chercher utilisateur ::

```
PROCEDURE chercher_utilisateur (nom all_users.username%TYPE ) IS
    nb NUMBER;
    erreurt exception ;
    BEGIN
    SELECT count(username) into nb
    FROM ALL_USERS
    where username=nom;
    If (nb=0) then raise erreurt;
    else
    dbms_output.put_line('utilisateur existe');
    end if;
    exception
    when erreurt then  dbms_output.put_line('utilisateur introuvable');
    END chercher_utilisateur ;
```

- modifier utilisateur ::

```
procedure modifier_utilisateur (nom VARCHAR(20),$role VARCHAR(20))
  is
  erreurut exception;
  nb number;
  begin
  SELECT count(username) into nb
  FROM ALL_USERS
      where username=nom;
  If nb=0 then raise erreurut;
  else
  grant role to nom;
  end if;
  exception
  when erreurut then  dbms_output.put_line('utilisateur introuvable');
  end modifier_utilisateur ;
```

- ajouter utilisateur ::

```
procedure ajout_utilisateur(nom VARCHAR(20), role VARCHAR(20))
is
errorrole exception;
erroruser exception;
begin
if upper(role) in ('ADMINBD','PLANIFICATEUREVT','GESTCOMMANDESBILLETS') then
SELECT count(username) into nb
    FROM ALL_USERS
        where username=nom;
if nb=0 then
create user nom
grant role to nom;
else raise erroruser;
end if;
else
raise errorrole;
enf if ;
exception
when errorrole then dbms_output.put_line('saisir un role convenable:ADMINBD,PLANIFICATEUREVT,GESTCOMMANDESBILLETS');
when erroruser then dbms_output.put_line('utilisateur existe deja')
end ;
 end gest_utilisateurs;
```

- execution chercher:

```
DECLARE
BEGIN
gest_utilisateurs.chercher_utilisateur('AAAAA');
END;
```

Script Output

Task completed in 0.03 seconds

```
Package Body GEST_UTILISATEURS compiled

utilisateur existe


PL/SQL procedure successfully completed.
```

- execution supprimer

```
DECLARE
BEGIN
gest_utilisateurs.supprimer_utilisateur('AAAAA');
END;

-------------------------------------
```

Script Output

Task completed in 0.05 seconds

```
DROP USER AAAAA


PL/SQL procedure successfully completed.
```

- role gestion_utilisateur

```
create role Gestion_Utilisateurs;
grant execute on gest_utilisateurs to Gestion_Utilisateurs;
grant Gestion_Utilisateurs to AdminBD;
```

Script Output

Task completed in 0.053 seconds

```
Role GESTION_UTILISATEURS created.


Grant succeeded.


Grant succeeded.
```

- role gestion_uspectacles

```
create role Gestion_Spectacles;
grant execute on gest_Spectacles to Gestion_Spectacles;
grant Gestion_Spectacles to PlanificateurEvt;
```

```
Role GESTION_SPECTACLES created.


Grant succeeded.


Grant succeeded.
```