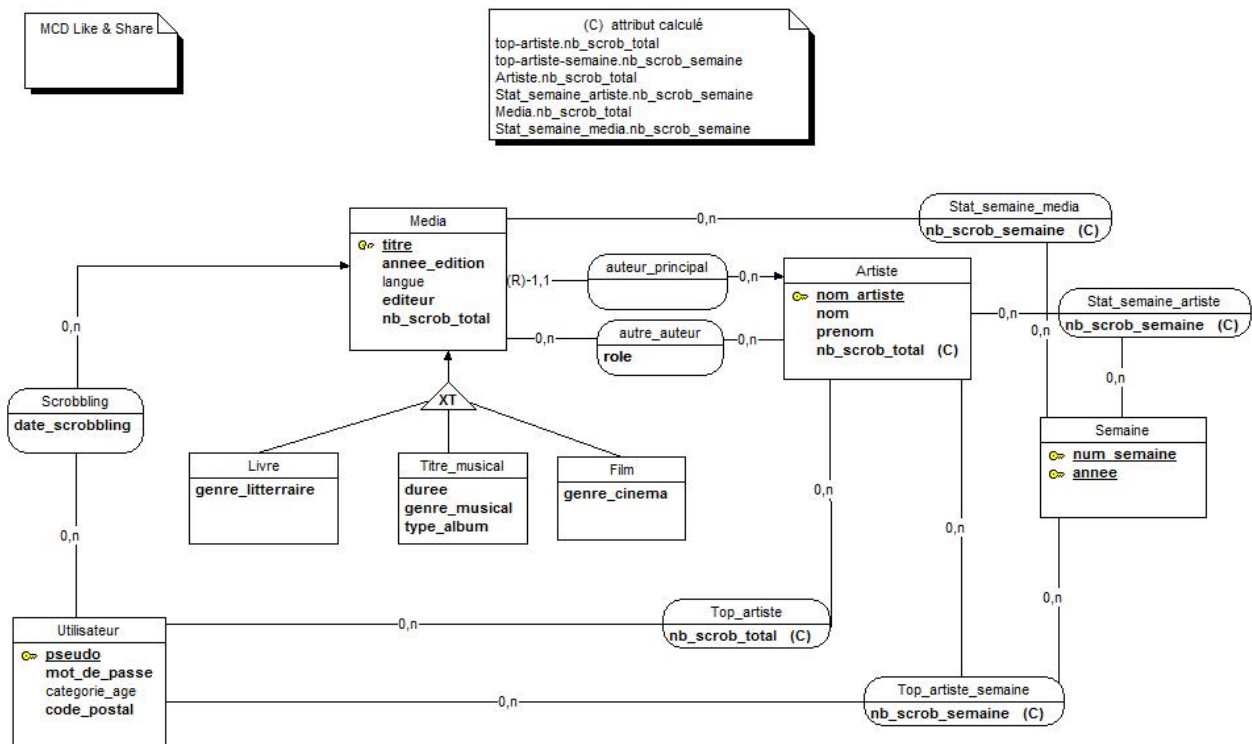


## Projet BDD2 Like & Share<sup>1</sup>

### Éléments fournis à la fin de l'analyse & conception

#### I) Version avec sauvegarde des statistiques hebdomadaires

#### Modèle conceptuel entité-association



<sup>1</sup> "On aime, on partage"

**Modèle logique de données en dehors des attributs calculés (définis comme des vues)**

(clés primaires en gras, étrangères en italique)

UTILISATEUR ( **PSEUDO**, MOT\_DE\_PASSE, CATEGORIE\_AGE, CODE\_POSTAL )

ARTISTE ( **NOM\_ARTISTE**, *NOM*, *PRENOM* )

MEDIA ( ***NOM\_ARTISTE***, **TITRE**, ANNEE\_EDITION, LANGUE, EDITEUR )

LIVRE ( ***NOM\_ARTISTE***, **TITRE**, GENRE\_LITTERAIRE )

FILM ( ***NOM\_ARTISTE***, **TITRE**, GENRE\_CINEMA )

TITRE\_MUSICAL ( ***NOM\_ARTISTE***, **TITRE**, DUREE, GENRE\_MUSICAL, TYPE\_ALBUM )

AUTRE\_AUTEUR ( ***NOM\_ARTISTE***, **TITRE**, ***NOM\_ARTISTE\_1***, *ROLE\_AA* )

SCROBBLING ( ***NOM\_ARTISTE***, **TITRE**, **PSEUDO**, **DATE\_SCROBBLING** )

SEMAINE ( **NUM\_SEMAINE**, **ANNEE** )

**Les vues permettant de calculer les attributs manquants**

CREATE OR REPLACE VIEW **V\_SCROB\_ARTISTE** AS

(SELECT NOM\_ARTISTE, count(\*) as *NB\_SCROB\_TOTAL*

FROM SCROBBLING

GROUP BY NOM\_ARTISTE);

CREATE OR REPLACE VIEW **V\_SCROB\_ARTISTE\_SEMAINE** AS

(SELECT NOM\_ARTISTE, NUM\_SEMAINE, ANNEE, count(\*) as *NB\_SCROB\_SEMAINE*

FROM SCROBBLING, SEMAINE

WHERE to\_char(DATE\_SCROBBLING, 'YYYY') = ANNEE AND

```
to_number(to_char(DATE_SCROBBLING,'WW')) = NUM_SEMAINE  
GROUP BY NOM_ARTISTE, NUM_SEMAINE, ANNEE);
```

CREATE OR REPLACE VIEW **V\_SCROB\_MEDIA** AS

```
(SELECT  NOM_ARTISTE, TITRE, count(*) as  NB_SCROB_TOTAL  
FROM SCROBBLING  
GROUP BY NOM_ARTISTE, TITRE);
```

CREATE OR REPLACE VIEW **V\_SCROB\_MEDIA\_SEMAINE** AS

```
(SELECT  NOM_ARTISTE, TITRE, NUM_SEMAINE, ANNEE , count(*) as  NB_SCROB_SEMAINE  
FROM SCROBBLING , SEMAINE  
WHERE  to_char(DATE_SCROBBLING,'YYYY') = ANNEE AND  
to_number(to_char(DATE_SCROBBLING,'WW')) = NUM_SEMAINE  
GROUP BY NOM_ARTISTE, TITRE, NUM_SEMAINE, ANNEE);
```

CREATE OR REPLACE VIEW **V\_STAT\_ARTISTE** AS

```
(SELECT  PSEUDO, NOM_ARTISTE, count(*) as  NB_SCROB_TOTAL  
FROM SCROBBLING  
GROUP BY PSEUDO , NOM_ARTISTE);
```

CREATE OR REPLACE VIEW **V\_STAT\_ARTISTE\_SEMAINE** AS

```
(SELECT  PSEUDO, NOM_ARTISTE, NUM_SEMAINE, ANNEE , count(*) as  NB_SCROB_SEMAINE  
FROM SCROBBLING , SEMAINE  
WHERE  to_char(DATE_SCROBBLING,'YYYY') = ANNEE AND  
to_number(to_char(DATE_SCROBBLING,'WW')) = NUM_SEMAINE  
GROUP BY PSEUDO , NOM_ARTISTE, NUM_SEMAINE, ANNEE);
```

CREATE OR REPLACE VIEW **V\_STAT\_MEDIA** AS

```
(SELECT PSEUDO, NOM_ARTISTE, TITRE, count(*) as NB_SCROB_TOTAL  
FROM SCROBBLING  
GROUP BY PSEUDO, NOM_ARTISTE, TITRE);
```

CREATE OR REPLACE VIEW **V\_STAT\_MEDIA\_SEMAINE** AS

```
(SELECT PSEUDO, NOM_ARTISTE, TITRE, NUM_SEMAINE, ANNEE, count(*) as  
NB_SCROB_SEMAINE  
FROM SCROBBLING, SEMAINE  
WHERE to_char(DATE_SCROBBLING, 'YYYY') = ANNEE AND  
to_number(to_char(DATE_SCROBBLING, 'WW')) = NUM_SEMAINE  
GROUP BY PSEUDO, NOM_ARTISTE, TITRE, NUM_SEMAINE, ANNEE);
```

### Comment utiliser les vues dans le projet Like & share ?

- Soit on mémorise le contenu de chaque vue dans une table qu'on "rafraichit" chaque dimanche  
Exemple :

```
CREATE TABLE STAT_SEMAINE_MEDIA  
as (SELECT * FROM V_STAT_SEMAINE_MEDIA )
```

- Il est aussi possible de créer des **vues matérialisées** pour lesquelles on impose une date de rafraîchissement périodique<sup>2</sup> (tous les dimanche à 20h).  
Exemple :

CREATE **MATERIALIZED VIEW** **V\_STAT\_SEMAINE\_MEDIA**

**REFRESH FAST NEXT NEXT\_DAY(TRUNC(SYSDATE), 'SUNDAY') + 20/24**

```
AS (SELECT NOM_ARTISTE, TITRE, NUM_SEMAINE, ANNEE, count(*) as NB_SCROB_SEMAINE  
FROM SCROBBLING, SEMAINE  
WHERE year(DATE_SCROBBLING) = ANNEE AND  
NUM_SEMAINE= to_number(to_char(DATE_SCROBBLING, 'WW'))  
GROUP BY NOM_ARTISTE, TITRE, NUM_SEMAINE, ANNEE);
```

---

<sup>2</sup> - On peut aussi indiquer la date du premier rafraichissement.

## Modèle physique de données (SGBD Oracle)

complété avec les définitions des vues ci-dessus

```
-- -----  
--      Génération d'une base de données pour  
--      Oracle Version >= 9i  
--      (24/4/2017 13:54:10)  
-- -----
```

```
--      Projet : Like & Share  
--      Auteur : Smail-Tabbone Malika  
--      Date de dernière modification : 24/4/2017 13:53:52  
-- -----
```

DROP TABLE LIVRE CASCADE CONSTRAINTS;

DROP TABLE UTILISATEUR CASCADE CONSTRAINTS;

DROP TABLE ARTISTE CASCADE CONSTRAINTS;

DROP TABLE MEDIA CASCADE CONSTRAINTS;

DROP TABLE SEMAINE CASCADE CONSTRAINTS;

DROP TABLE FILM CASCADE CONSTRAINTS;

DROP TABLE TITRE\_MUSICAL CASCADE CONSTRAINTS;

DROP TABLE AUTRE\_AUTEUR CASCADE CONSTRAINTS;

DROP TABLE SCROBBLING CASCADE CONSTRAINTS;

```
-- -----  
--      TABLE : UTILISATEUR  
-- -----
```

CREATE TABLE UTILISATEUR

```
(  
  PSEUDO VARCHAR(100) ,  
  MOT_DE_PASSE VARCHAR(100) NOT NULL,  
  CATEGORIE_AGE VARCHAR(100) NULL CHECK (CATEGORIE_AGE IN ('0-20', '21-30','31-101')),  
  CODE_POSTAL VARCHAR(5) NOT NULL  
, CONSTRAINT PK_UTILISATEUR PRIMARY KEY (PSEUDO)
```

```
);
```

```
-----  
--      TABLE : ARTISTE  
-----
```

```
CREATE TABLE ARTISTE  
(  
    NOM_ARTISTE VARCHAR(100) ,  
    NOM VARCHAR(100) NOT NULL,  
    PRENOM VARCHAR(100) NOT NULL  
, CONSTRAINT PK_ARTISTE_LIKE PRIMARY KEY (NOM_ARTISTE)  
);
```

```
-----  
--      TABLE : MEDIA  
-----
```

```
CREATE TABLE MEDIA  
(  
    NOM_ARTISTE VARCHAR(100) ,  
    TITRE VARCHAR(100),  
    ANNEE_EDITION NUMBER(4) DEFAULT to_char(sysdate,'YYYY') NOT NULL,  
    LANGUE VARCHAR(100) NULL CHECK (LANGUE IN ('fr', 'angl', 'autre')),  
    EDITEUR VARCHAR(100) NOT NULL  
, CONSTRAINT PK_MEDIA PRIMARY KEY (NOM_ARTISTE, TITRE)  
);
```

```
-----  
--      TABLE : LIVRE  
-----
```

```
CREATE TABLE LIVRE  
(  
    NOM_ARTISTE VARCHAR(100) ,  
    TITRE VARCHAR(100) ,  
    GENRE_LITTERAIRE VARCHAR(100) NULL CHECK (GENRE_LITTERAIRE IN ('roman', 'essai', 'nouvelle'  
, 'autre')),  
    CONSTRAINT PK_LIVRE_LIKE PRIMARY KEY (NOM_ARTISTE, TITRE)  
);
```

```
-----  
--      TABLE : FILM  
-----
```

```
CREATE TABLE FILM  
(
```

```

    NOM_ARTISTE VARCHAR(100) ,
    TITRE VARCHAR(100) ,
    GENRE_CINEMA VARCHAR(100) NULL CHECK (GENRE_CINEMA IN ('comedie', 'SF', 'horreur', 'autre'))
,
    CONSTRAINT PK_FILM_LIKE PRIMARY KEY (NOM_ARTISTE, TITRE)
);

```

```

-----
--      TABLE : TITRE_MUSICAL
-----

```

```

CREATE TABLE TITRE_MUSICAL
(
    NOM_ARTISTE VARCHAR(100) ,
    TITRE VARCHAR(100),
    DUREE NUMBER(5,2) NOT NULL,
    GENRE_MUSICAL VARCHAR(100) NULL CHECK (GENRE_MUSICAL IN ('rock', 'rap', 'pop', 'autre')),
    TYPE_ALBUM VARCHAR(100) NOT NULL CHECK (TYPE_ALBUM IN ('single', 'full', 'mini')),
    CONSTRAINT PK_TITRE_MUSICAL PRIMARY KEY (NOM_ARTISTE, TITRE)
);

```

```

-----
--      TABLE : SEMAINE (penser à ajouter une semaine chaque dimanche)
-----

```

```

CREATE TABLE SEMAINE
(
    NUM_SEMAINE NUMBER(2) CHECK (NUM_SEMAINE >0 AND NUM_SEMAINE <53),
    ANNEE NUMBER(4) ,
    CONSTRAINT PK_SEMAINE PRIMARY KEY (NUM_SEMAINE, ANNEE)
);

```

```

-----
--      TABLE : AUTRE_AUTEUR
-----

```

```

CREATE TABLE AUTRE_AUTEUR
(
    NOM_ARTISTE VARCHAR(100) ,
    TITRE VARCHAR(100) ,
    NOM_ARTISTE_1 VARCHAR(100),
    ROLE_AA VARCHAR(100) NOT NULL CHECK (ROLE_AA IN ('compositeur', 'producteur', 'acteur', 'autre')) ,
    CONSTRAINT PK_AUTRE_AUTEUR PRIMARY KEY (NOM_ARTISTE, TITRE, NOM_ARTISTE_1)
);

```

-- TABLE : SCROBBLING

---

```
CREATE TABLE SCROBBLING
(
  NOM_ARTISTE VARCHAR(100) ,
  TITRE VARCHAR(100) ,
  PSEUDO VARCHAR(100) ,
  DATE_SCROBBLING VARCHAR(100) DEFAULT sysdate,
  CONSTRAINT PK_SCROBBLING PRIMARY KEY (NOM_ARTISTE, TITRE, PSEUDO, DATE_SCROBBLING)
);
```

---

-- CREATION DES REFERENCES DE TABLE

---

```
ALTER TABLE LIVRE ADD (
  CONSTRAINT FK_LIVRE_MEDIA
    FOREIGN KEY (NOM_ARTISTE, TITRE)
      REFERENCES MEDIA (NOM_ARTISTE, TITRE)) ;
```

```
ALTER TABLE MEDIA ADD (
  CONSTRAINT FK_MEDIA_ARTISTE
    FOREIGN KEY (NOM_ARTISTE)
      REFERENCES ARTISTE (NOM_ARTISTE) ON DELETE CASCADE) ;
```

```
ALTER TABLE FILM ADD (
  CONSTRAINT FK_FILM_MEDIA
    FOREIGN KEY (NOM_ARTISTE, TITRE)
      REFERENCES MEDIA (NOM_ARTISTE, TITRE) ) ;
```

```
ALTER TABLE TITRE_MUSICAL ADD (
  CONSTRAINT FK_TITRE_MUSICAL_MEDIA
    FOREIGN KEY (NOM_ARTISTE, TITRE)
      REFERENCES MEDIA (NOM_ARTISTE, TITRE)) ;
```

```
ALTER TABLE AUTRE_AUTEUR ADD (
  CONSTRAINT FK_AUTRE_AUTEUR_MEDIA
    FOREIGN KEY (NOM_ARTISTE, TITRE)
      REFERENCES MEDIA (NOM_ARTISTE, TITRE)) ;
```

```
ALTER TABLE AUTRE_AUTEUR ADD (
  CONSTRAINT FK_AUTRE_AUTEUR_ARTISTE
    FOREIGN KEY (NOM_ARTISTE_1)
      REFERENCES ARTISTE (NOM_ARTISTE)) ;
```



```
ALTER TABLE SCROBBLING ADD (  
    CONSTRAINT FK_SCROBBLING_MEDIA  
    FOREIGN KEY (NOM_ARTISTE, TITRE)  
    REFERENCES MEDIA (NOM_ARTISTE, TITRE)) ;
```

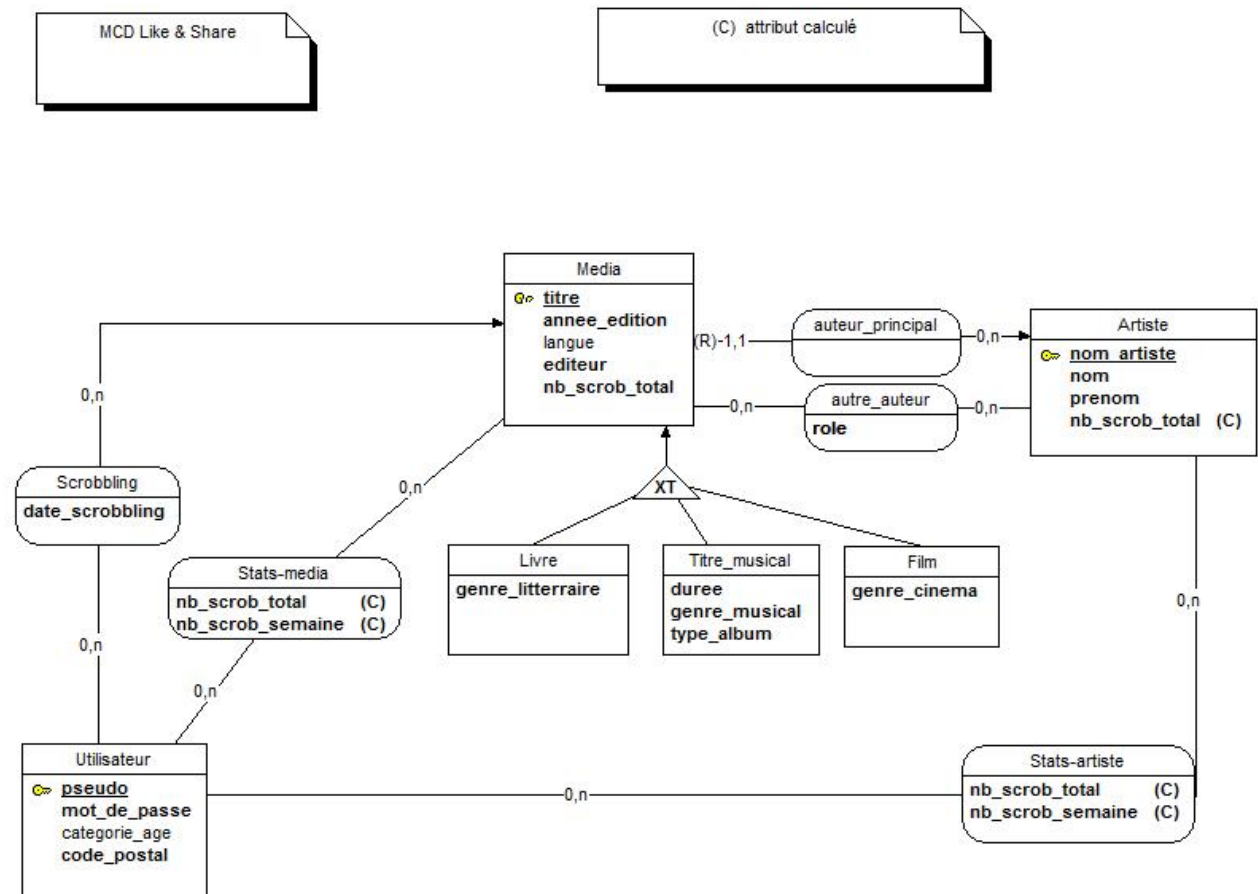
```
ALTER TABLE SCROBBLING ADD (  
    CONSTRAINT FK_SCROBBLING_UTILISATEUR  
    FOREIGN KEY (PSEUDO)  
    REFERENCES UTILISATEUR (PSEUDO)) ;
```

```
-----  
--          FIN modèle physique  
-----
```

```
-----  
--          Definition des vues (voir ci-dessus la definition des vues)  
-----
```

## II) Version sans sauvegarde des statistiques hebdomadaires

### Modèle conceptuel entité-association



**Modèle logique de données en dehors des attributs calculés (définis comme des vues)**

(clés primaires en gras, étrangères en italique)

UTILISATEUR ( **PSEUDO**, MOT\_DE\_PASSE, CATEGORIE\_AGE, CODE\_POSTAL )

ARTISTE ( **NOM\_ARTISTE**, NOM, PRENOM)

MEDIA ( **NOM\_ARTISTE**, **TITRE**, ANNEE\_EDITION, LANGUE, EDITEUR)

LIVRE ( **NOM\_ARTISTE**, **TITRE**, GENRE\_LITTERAIRE)

FILM ( **NOM\_ARTISTE**, **TITRE**, GENRE\_CINEMA)

TITRE\_MUSICAL ( **NOM\_ARTISTE**, **TITRE**, DUREE, GENRE\_MUSICAL, TYPE\_ALBUM)

AUTRE\_AUTEUR ( **NOM\_ARTISTE**, **TITRE**, **NOM\_ARTISTE\_1**, ROLE\_AA)

SCROBBLING ( **NOM\_ARTISTE**, **TITRE**, **PSEUDO**, **DATE\_SCROBBLING**)

**Les vues permettant de calculer les attributs manquants**

CREATE OR REPLACE VIEW **V\_SCROB\_ARTISTE** AS

(SELECT NOM\_ARTISTE, count(\*) as NB\_SCROB\_TOTAL

FROM SCROBBLING

GROUP BY NOM\_ARTISTE);

CREATE OR REPLACE VIEW **V\_SCROB\_ARTISTE\_SEMAINE** AS

(SELECT NOM\_ARTISTE, count(\*) as NB\_SCROB\_SEMAINE

FROM SCROBBLING

WHERE to\_char(DATE\_SCROBBLING,'YYYY') = to\_char(sysdate,'YYYY') AND

```
to_number(to_char(DATE_SCROBBLING,'WW')) = to_number(to_char(sysdate,'WW'))  
  
GROUP BY NOM_ARTISTE;
```

CREATE OR REPLACE VIEW **V\_SCROB\_MEDIA** AS

```
(SELECT  NOM_ARTISTE, TITRE, count(*) as  NB_SCROB_TOTAL  
  
FROM  SCROBBLING  
  
GROUP BY NOM_ARTISTE, TITRE);
```

CREATE OR REPLACE VIEW **V\_SCROB\_MEDIA\_SEMAINE** AS

```
(SELECT  NOM_ARTISTE, TITRE, count(*) as  NB_SCROB_SEMAINE  
  
FROM  SCROBBLING  
  
WHERE  to_char(DATE_SCROBBLING,'YYYY') = to_char(sysdate,'YYYY') AND  
  
to_number(to_char(DATE_SCROBBLING,'WW')) = to_number(to_char(sysdate,'WW'))  
  
GROUP BY NOM_ARTISTE, TITRE;
```

CREATE OR REPLACE VIEW **V\_STAT\_ARTISTE** AS

```
(SELECT  PSEUDO, NOM_ARTISTE, count(*) as  NB_SCROB_TOTAL  
  
FROM  SCROBBLING  
  
GROUP BY PSEUDO , NOM_ARTISTE);
```

CREATE OR REPLACE VIEW **V\_STAT\_ARTISTE\_SEMAINE** AS

```
(SELECT  PSEUDO, NOM_ARTISTE, count(*) as  NB_SCROB_SEMAINE  
  
FROM  SCROBBLING  
  
WHERE  to_char(DATE_SCROBBLING,'YYYY') = to_char(sysdate,'YYYY') AND  
  
to_number(to_char(DATE_SCROBBLING,'WW')) = to_number(to_char(sysdate,'WW'))  
  
GROUP BY PSEUDO , NOM_ARTISTE;
```

CREATE OR REPLACE VIEW **V\_STAT\_MEDIA** AS

```
(SELECT PSEUDO, NOM_ARTISTE, TITRE, count(*) as NB_SCROB_TOTAL  
FROM SCROBBLING  
GROUP BY PSEUDO, NOM_ARTISTE, TITRE );
```

CREATE OR REPLACE VIEW **V\_STAT\_MEDIA\_SEMAINE** AS

```
(SELECT PSEUDO, NOM_ARTISTE, TITRE, count(*) as NB_SCROB_SEMAINE  
FROM SCROBBLING  
WHERE to_char(DATE_SCROBBLING,'YYYY') = to_char(sysdate,'YYYY') AND  
to_number(to_char(DATE_SCROBBLING,'WW')) = to_number(to_char(sysdate,'WW'))  
GROUP BY PSEUDO, NOM_ARTISTE, TITRE);
```

**Modèle physique de données (SGBD Oracle)**

**complété avec les définitions des vues ci-dessus**

DROP TABLE LIVRE CASCADE CONSTRAINTS;

DROP TABLE UTILISATEUR CASCADE CONSTRAINTS;

DROP TABLE ARTISTE CASCADE CONSTRAINTS;

DROP TABLE MEDIA CASCADE CONSTRAINTS;

DROP TABLE FILM CASCADE CONSTRAINTS;

DROP TABLE TITRE\_MUSICAL CASCADE CONSTRAINTS;

DROP TABLE AUTRE\_AUTEUR CASCADE CONSTRAINTS;

DROP TABLE SCROBBLING CASCADE CONSTRAINTS;

```
-----  
--      TABLE : UTILISATEUR  
-----
```

```
CREATE TABLE UTILISATEUR  
(  
    PSEUDO VARCHAR(100) ,  
    MOT_DE_PASSE VARCHAR(100) NOT NULL,  
    CATEGORIE_AGE VARCHAR(100) NULL CHECK (CATEGORIE_AGE IN ('0-20', '21-30','31-101')),  
    CODE_POSTAL VARCHAR(5) NOT NULL  
, CONSTRAINT PK_UTILISATEUR PRIMARY KEY (PSEUDO)  
);
```

```
-----  
--      TABLE : ARTISTE  
-----
```

```
CREATE TABLE ARTISTE  
(  
    NOM_ARTISTE VARCHAR(100) ,  
    NOM VARCHAR(100) NOT NULL,  
    PRENOM VARCHAR(100) NOT NULL  
, CONSTRAINT PK_ARTISTE_LIKE PRIMARY KEY (NOM_ARTISTE)  
);
```

```
-----  
--      TABLE : MEDIA  
-----
```

```
CREATE TABLE MEDIA  
(  
    NOM_ARTISTE VARCHAR(100) ,  
    TITRE VARCHAR(100),  
    ANNEE_EDITION NUMBER(4) DEFAULT to_char(sysdate,'YYYY') NOT NULL,  
    LANGUE VARCHAR(100) NULL CHECK (LANGUE IN ('fr', 'angl', 'autre')),  
    EDETEUR VARCHAR(100) NOT NULL  
, CONSTRAINT PK_MEDIA PRIMARY KEY (NOM_ARTISTE, TITRE)  
);
```

```
-----  
--      TABLE : LIVRE  
-----
```

```
CREATE TABLE LIVRE  
(  
    NOM_ARTISTE VARCHAR(100) ,
```

```

TITRE VARCHAR(100) ,
GENRE_LITTERAIRE VARCHAR(100) NULL CHECK (GENRE_LITTERAIRE IN ('roman', 'essai', 'nouvelle'
,'autre')),
CONSTRAINT PK_LIVRE_LIKE PRIMARY KEY (NOM_ARTISTE, TITRE)
);

```

```

-----
--  TABLE : FILM
-----

```

```

CREATE TABLE FILM
(
  NOM_ARTISTE VARCHAR(100) ,
  TITRE VARCHAR(100) ,
  GENRE_CINEMA VARCHAR(100) NULL CHECK (GENRE_CINEMA IN ('comedie', 'SF', 'horreur', 'autre'))
,
  CONSTRAINT PK_FILM_LIKE PRIMARY KEY (NOM_ARTISTE, TITRE)
);

```

```

-----
--  TABLE : TITRE_MUSICAL
-----

```

```

CREATE TABLE TITRE_MUSICAL
(
  NOM_ARTISTE VARCHAR(100) ,
  TITRE VARCHAR(100),
  DUREE NUMBER(5,2) NOT NULL,
  GENRE_MUSICAL VARCHAR(100) NULL CHECK (GENRE_MUSICAL IN ('rock', 'rap', 'pop', 'autre')),
  TYPE_ALBUM VARCHAR(100) NOT NULL CHECK (TYPE_ALBUM IN ('single', 'full', 'mini')),
  CONSTRAINT PK_TITRE_MUSICAL PRIMARY KEY (NOM_ARTISTE, TITRE)
);

```

```

-----
--  TABLE : AUTRE_AUTEUR
-----

```

```

CREATE TABLE AUTRE_AUTEUR
(
  NOM_ARTISTE VARCHAR(100) ,
  TITRE VARCHAR(100) ,
  NOM_ARTISTE_1 VARCHAR(100),
  ROLE_AA VARCHAR(100) NOT NULL CHECK (ROLE_AA IN ('compositeur', 'producteur', 'acteur'
,'autre')),
  CONSTRAINT PK_AUTRE_AUTEUR PRIMARY KEY (NOM_ARTISTE, TITRE, NOM_ARTISTE_1)
);

```

```
-----  
--  TABLE : SCROBBLING  
-----
```

```
CREATE TABLE SCROBBLING  
(  
  NOM_ARTISTE VARCHAR(100) ,  
  TITRE VARCHAR(100) ,  
  PSEUDO VARCHAR(100) ,  
  DATE_SCROBBLING VARCHAR(100) DEFAULT sysdate,  
  CONSTRAINT PK_SCROBBLING PRIMARY KEY (NOM_ARTISTE, TITRE, PSEUDO, DATE_SCROBBLING)  
);
```

```
-----  
--  CREATION DES REFERENCES DE TABLE  
-----
```

```
ALTER TABLE LIVRE ADD (  
  CONSTRAINT FK_LIVRE_MEDIA  
    FOREIGN KEY (NOM_ARTISTE, TITRE)  
      REFERENCES MEDIA (NOM_ARTISTE, TITRE)) ;
```

```
ALTER TABLE MEDIA ADD (  
  CONSTRAINT FK_MEDIA_ARTISTE  
    FOREIGN KEY (NOM_ARTISTE)  
      REFERENCES ARTISTE (NOM_ARTISTE) ON DELETE CASCADE) ;
```

```
ALTER TABLE FILM ADD (  
  CONSTRAINT FK_FILM_MEDIA  
    FOREIGN KEY (NOM_ARTISTE, TITRE)  
      REFERENCES MEDIA (NOM_ARTISTE, TITRE) ) ;
```

```
ALTER TABLE TITRE_MUSICAL ADD (  
  CONSTRAINT FK_TITRE_MUSICAL_MEDIA  
    FOREIGN KEY (NOM_ARTISTE, TITRE)  
      REFERENCES MEDIA (NOM_ARTISTE, TITRE)) ;
```

```
ALTER TABLE AUTRE_AUTEUR ADD (  
  CONSTRAINT FK_AUTRE_AUTEUR_MEDIA  
    FOREIGN KEY (NOM_ARTISTE, TITRE)  
      REFERENCES MEDIA (NOM_ARTISTE, TITRE)) ;
```

```
ALTER TABLE AUTRE_AUTEUR ADD (  
  CONSTRAINT FK_AUTRE_AUTEUR_ARTISTE  
    FOREIGN KEY (NOM_ARTISTE_1)
```



```
REFERENCES ARTISTE (NOM_ARTISTE)) ;
```

```
ALTER TABLE SCROBBLING ADD (  
  CONSTRAINT FK_SCROBBLING_MEDIA  
    FOREIGN KEY (NOM_ARTISTE, TITRE)  
      REFERENCES MEDIA (NOM_ARTISTE, TITRE)) ;
```

```
ALTER TABLE SCROBBLING ADD (  
  CONSTRAINT FK_SCROBBLING_UTILISATEUR  
    FOREIGN KEY (PSEUDO)  
      REFERENCES UTILISATEUR (PSEUDO)) ;
```

```
-----  
--      FIN modèle physique  
-----
```

```
-----  
--      Definition des vues (voir ci-dessus la definition des vues)
```