Institut Galilée

UNIVERSITÉ PARIS 13 | USPC Université Sorbonne Paris Cité

Rapport du projet base de données

Nom	Prénom	numéro étudiant
➢ RAHMANI	Sylia	11707524
➢ CHALAL	Abdllatif	11705693

Table des matières

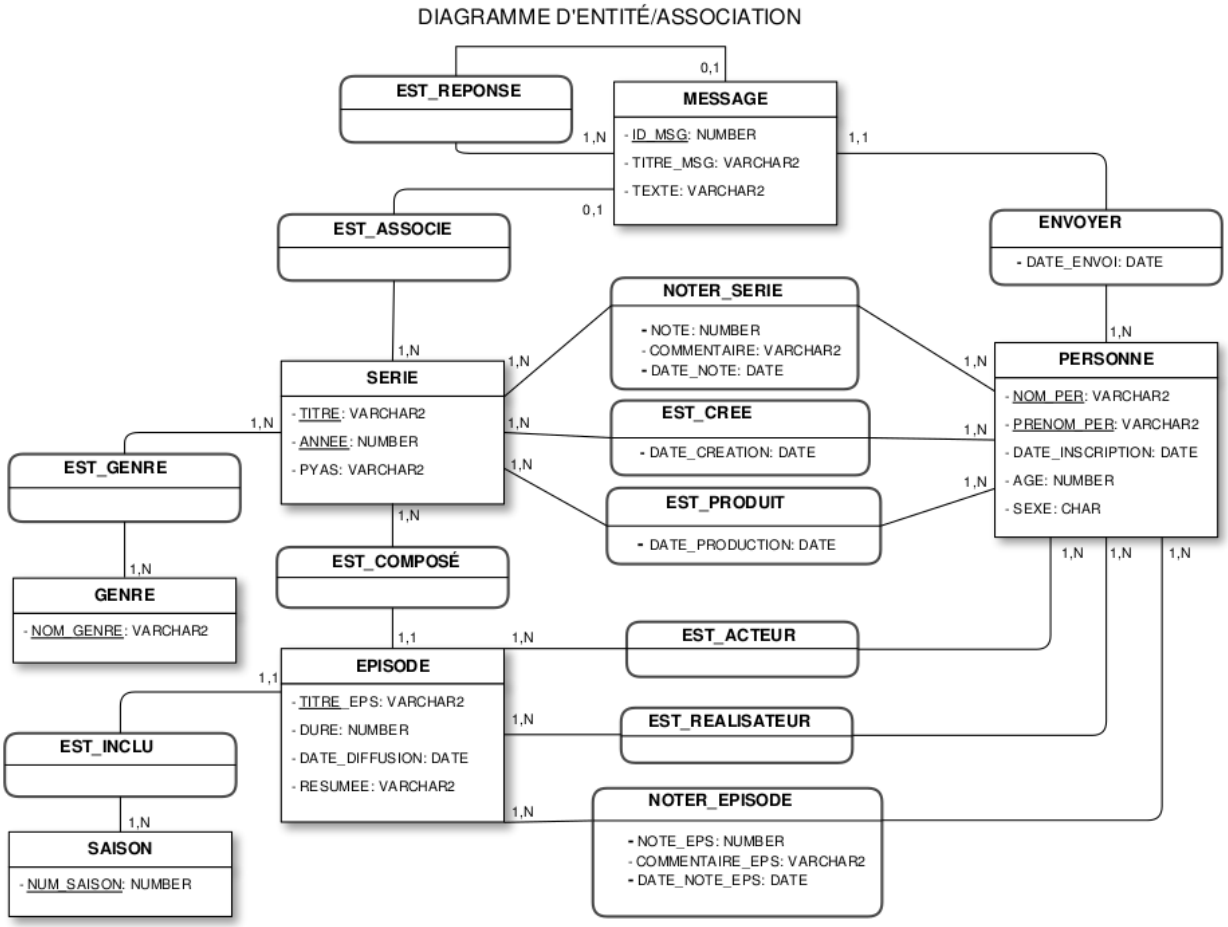
Introduction

Le présent rapport développe le projet d'implémentation en Sql d'un site internet donnant des informations détaillés sur des séries télévisées.

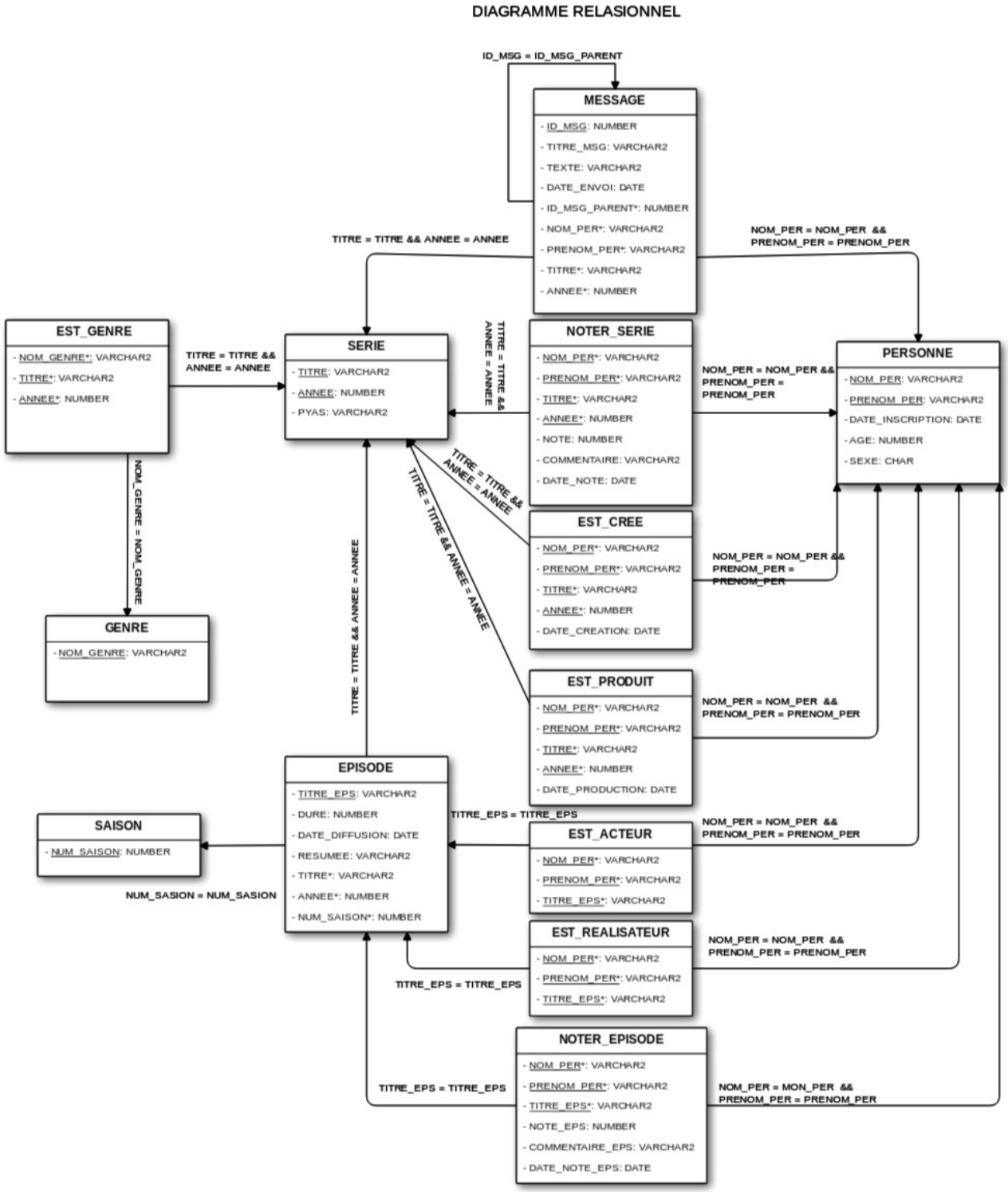
l'objectif est de modéliser le problème sous la forme d'une base de données, en mettant en pratiques toutes les connaissances théorique vue dans le coure de bases de données c'est à dire la modélisation du problème , création et remplissage des tables et enfin l'interrogation de la base de données crée.

Notre réflexion se déploie en quatre étapes majeures; il convient d'établir dans un premier temps une conception du problème sous la forme d'un modèle entités/association et établir le modèle relationnel correspondant, suivie de la création des tables avec toutes les contraintes pertinentes, puis le remplissage des tables crées en respectant toutes les contraintes posées, enfin utilisé la base de donnée crée.

Le modèle entités association pour modéliser le problème



Le modèle relationnel :



Explication

Après une longue période de réflexion nous sommes parvenus à modéliser le problème en 13 tables différentes :

une table « SERIE » dont on va définir chaque série par son titre, année et son pays d'origine.

Une table « EPISODE » dont va décrire chaque épisode de la série par un titre propre à chaque épisode,une duré,la date de diffusion, le résumé de l'épisode,année de sortie et le numéro de la saison dont elle fait partie.

Une table « SAISON »dont on trouve le numéro de saison dons la quelle les épisode sont inclus.

Chaque série a un genre particulier (comédie, action …) pour lister tous les types existant on a créer une table dédié nommé « GENRE » .

la table « EST_GENRE » est pour but d'associé chaque série avec son genre correspond.

Notre site contient forcément des personnes (ils peuvent être des utilisateurs, créateurs,acteurs…) qui sont décrit d'une manière asse globale dans la table « PERSONNE » ,dont on y trouve leurs noms, prénoms, date d'inscription au site, âge, et leurs sexe.

Chaque utilisateur a le droits de noter une EPISODE ou une SERIE, et pour modéliser ce là, on a crée deux tables « NOTER_EPISODE » et « NOTER_SERIE » ou on sauvegarde le nom et le prénom de la personne qui a attribuer la note, le titre de l'épisode noter (respectivement de la série noter), la note attribuer, le commentaire laisser par l'utilisateur ainsi que la date où la note a été attribuer.

Le site contient un forum de discussion où les utilisateurs peuvent s'échanger des messages, la table « MESSAGE » a étè crée dans ce but. Elle contient l'identifiant du message, son titre, le texte du message, la date de son envoie, l'identifiant du messages parent, le nom et prénom de la personne à qui on répond, le titre et l'année de la série invoqué dans la discussion.

Notre site contient aussi les noms de tous les personnes qui ont participer à la création et à la production de chaque série. les créateurs et les producteurs sont décrit respectivement dans la table « EST_CREER » et« EST_PRODUIT » par leurs nom, prénom, titre et l'année de la série qui ont créer (ou produit) ainsi que la date de créateur (production) de la série.

Les acteurs et les réalisateur de chaque épisode sont décrit respectivement dans les tables « EST_ACTEUR » et « EST_REALISATEUR » par leur nom, prénom, et le titre de l'épisode dans la quelle ils ont joué (respectivement réaliser).

Création des tables avec toutes les contraintes piétinante avec explication de chaque choix

```
DROP TABLE SERIE CASCADE CONSTRAINT;
CREATE TABLE SERIE
(
 ANNEE        NUMBER(4)          CONSTRAINT NN_SERIE_ANNEE NOT NULL,
 TITRE        VARCHAR(50)        CONSTRAINT NN_SERIE_TITRE NOT NULL,
 PAYS         VARCHAR(50)        CONSTRAINT CK_SERIE_PAYS CHECK( RPAD(PAYS,1)=
                                 UPPER(RPAD(PAYS,1))),
 CONSTRAINT  PK_SERIE   PRIMARY KEY  (ANNEE,TITRE)
);
```

⇒ l'attribut année et titre ne doivent être nulle car tout les deux sont composants de la clé primaire ;

la combinaison (annee,titre) est considérer comme clé primaire car il est non nulle et unique.

Le nom de pays doit commencer par une majuscule par convention.

```
DROP TABLE PERSONNE CASCADE CONSTRAINT;
CREATE TABLE PERSONNE
(
 NOM_PER              VARCHAR(10)      CONSTRAINT NN_PERSONNE_NOM NOT NULL,
 PRENOM_PER           VARCHAR(20)      CONSTRAINT NN_PERSONNE_PRENOM NOT NULL,
 DATE_INSCRIPTION     DATE ,
 AGE NUMBER(3)     CONSTRAINT CK_PERSONNE_AGE CHECK(AGE > 0),
 SEXE CHAR(4)        CONSTRAINT CK_PERSONNE_SEX CHECK (SEXE  IN ('F','M')),
 CONSTRAINT PK_PERSONNE PRIMARY KEY (NOM_PER,PRENOM_PER)
);
```

⇒ nom, prénom de la personne doivent être nulle car ils sont deux composants de la clé primaire.

l'âge ne doit pas être négative.

Le sexe est soit une femme ou un homme et rien d'autre.

Le couple (NOM_PER,PRENOM_PER) définit la clé primaire de la classe car il est non nulle et unique.

```
DROP TABLE NOTER_SERIE CASCADE CONSTRAINT;

CREATE TABLE NOTER_SERIE

(

 DATE_NOTE          DATE,

 NOTE               NUMBER(4)      CONSTRAINT CK_NOTER_SER_NOT CHECK ( NOTE
                    BETWEEN 0 AND 10 ),

 COMMENTAIRE        VARCHAR(50),

 ANNEE              NUMBER(4)      CONSTRAINT NN_NOTER_SERIE_ANNEE NOT NULL,

 TITRE              VARCHAR(50)    CONSTRAINT NN_NOTER_SERIE_TITRE NOT NULL,

 NOM_PER            VARCHAR(10)    CONSTRAINT NN_NOTER_SERIE_NOM NOT NULL,

 PRENOM_PER         VARCHAR(20)    CONSTRAINT NN_NOTER_SERIE_PRENOM NOT NULL,

 CONSTRAINT PK_NOTER_SERIE PRIMARY KEY (ANNEE,TITRE,NOM_PER,PRENOM_PER),

 CONSTRAINT FK_NOTER_SERIE_SERIE FOREIGN KEY (ANNEE,TITRE) REFERENCES
SERIE(ANNEE,TITRE) ,

 CONSTRAINT FK_NOTER_SERIE_PERSO FOREIGN KEY (NOM_PER,PRENOM_PER)
REFERENCES  PERSONNE(NOM_PER,PRENOM_PER)

);
```

⇒ les série sont notées sur 10, pour ce là les notes doivent être comprises entre 0 et 10.

l'année, titre,nom et le prénom de la personne ne doivent être nulle (composantes de la clés primaire), et comme la combinaison (ANNEE,TITRE,NOM_PER,PRENOM_PER) est unique, elle définit la clés primaire de la classe.

Le couple (NOM_PER,PRENOM_PER) est exactement le nom et le prénom de la personne définit dans la table PERSONNE, dans cette table il définit une référence vers la table PERSONNE, ils doivent être non nulle (si non on auras une référence indéfini).

Le couple (ANNEE,TITRE) fait référence à l'attribut année et titre dans la table SERIE,ils doivent être non nulle (si non on auras une référence indéfini).

```
DROP TABLE EST_CREE CASCADE CONSTRAINT;

CREATE TABLE EST_CREE

(

 ANNEE              NUMBER(4)          CONSTRAINT NN_EST_CREE_ANNEE NOT NULL,

 TITRE              VARCHAR(50)        CONSTRAINT NN_EST_CREE_TITRE NOT NULL,

 NOM_PER            VARCHAR(10)        CONSTRAINT NN_EST_CREE_NOM NOT NULL,

 PRENOM_PER         VARCHAR(20)        CONSTRAINT NN_EST_CREE_PRENOM NOT NULL,

 DATE_CREATION      DATE ,

 CONSTRAINT PK_EST_CREE PRIMARY KEY (ANNEE,TITRE,NOM_PER,PRENOM_PER),
```

```
 CONSTRAINT FK_EST_CREE_SERIE FOREIGN KEY (ANNEE,TITRE) REFERENCES
SERIE(ANNEE,TITRE) ,

 CONSTRAINT FK_EST_CREE_PERSO FOREIGN KEY (NOM_PER,PRENOM_PER) REFERENCES
PERSONNE(NOM_PER,PRENOM_PER)

);
```

⇒ l'année, le titre, le nom et prénom du créateur de la série doivent être connu et non nulle.

Le quadruplets (ANNEE,TITRE,NOM_PER,PRENOM_PER) définit la clé primaire de la classe (combinaison unique et non nulle).

Le couple (NOM_PER,PRENOM_PER) est une référence ver la classe PERSONNE, ils doivent être non nulle (si non on auras une référence indéfini).

Le couple (ANNEE,TITRE) est une référence ver la classe SERIE ,ils doivent être non nulle (si non on auras une référence indéfini) .

```
DROP TABLE EST_PRODUITCASCADE CONSTRAINT;

CREATE TABLE EST_PRODUIT

(

 ANNEE                 NUMBER(4)     CONSTRAINT NN_EST_PRODUIT_ANNEE NOT NULL,

 TITRE                 VARCHAR(50)   CONSTRAINT NN_EST_PRODUIT_TITRE NOT NULL,

 NOM_PER               VARCHAR(10)   CONSTRAINT NN_EST_PRODUIT_NOM NOT NULL,

 PRENOM_PER            VARCHAR(20)   CONSTRAINT NN_EST_PRODUIT_PRENOM NOT NULL,

 DATE_PRODUCTION  DATE ,

 CONSTRAINT PK_EST_PRODUIT     PRIMARY KEY (ANNEE,TITRE,NOM_PER,PRENOM_PER),

 CONSTRAINT FK_EST_PRODUIT_SERIE    FOREIGN KEY (ANNEE,TITRE) REFERENCES
SERIE(ANNEE,TITRE) ,

 CONSTRAINT FK_EST_PRODUIT_PERSO FOREIGN KEY (NOM_PER,PRENOM_PER)
REFERENCES  PERSONNE(NOM_PER,PRENOM_PER)

);
```

⇒ l'année, le titre, le nom et prénom du producteur de la série doivent être connu et non nulle.

La combinaison (ANNEE,TITRE,NOM_PER,PRENOM_PER) définit la clé primaire de la classe (combinaison unique et non nulle).

Le couple (NOM_PER,PRENOM_PER) est une référence ver la classe PERSONNE,ils doivent être non nulle (si non on auras une référence indéfini).

Le couple (ANNEE,TITRE) est une référence ver la classe SERIE,ils doivent être non nulle (si non on auras une référence indéfini).

```
DROP TABLE EPISODE CASCADE CONSTRAINT;

CREATE TABLE EPISODE
```

```
(
  TITRE_EPS          VARCHAR(50)        CONSTRAINT NN_EPISODE_TITRE_EPS NOT NULL,
  DURE               NUMBER             CONSTRAINT NN_SERIE_DURE NOT NULL,
  DATE_DIFFUSION     DATE               CONSTRAINT NN_EPISODE_DATE_DIFF NOT NULL,
  RESUMEE            VARCHAR(200) ,
  ANNEE              NUMBER(4)          CONSTRAINT NN_EPISODE_ANNEE NOT NULL,
  TITRE              VARCHAR(50)        CONSTRAINT NN_EPISONDE_TITRE NOT NULL,
  NUM_SAISON         NUMBER(5)          CONSTRAINT NN_EPISODE_NUM_SAISON NOT
                                        NULL,
  CONSTRAINT         PK_EPISODE PRIMARY KEY (TITRE_EPS),
  CONSTRAINT         FK_EPISODE_SERIE FOREIGN KEY (ANNEE,TITRE) REFERENCES
                     SERIE(ANNEE,TITRE),
  CONSTRAINT         FK_EPISODE_SAISON FOREIGN KEY  (NUM_SAISON) REFERENCES
                     SAISON(NUM_SAISON)
);
```

⇒ le titre, la duré, num_saison et la date de diffusion de chaque épisode ainsi que le titre et l'année de la série dont elle fait partie doivent être connue et non nulle.

Comme le titre de chaque épisode est unique et non nulle, il définit la clé primaire de la classe EPISODE.

Le couple (ANNEE,TITRE) est une référence ver la classe SERIE, ils doivent être non nulle (si non on auras une référence indéfini).

L'attribut (NUM_SAISON) fait référence ver la table SAISON,ils doivent être non nulle (si non on auras une référence indéfini).

```
DROP TABLE GENRE CASCADE CONSTRAINT;
CREATE TABLE GENRE (
  NOM_GENRE          VARCHAR(50)        CONSTRAINT PK_GENRE_NOM PRIMARY KEY
);
```

⇒ l'unique attribue nom du genre doit être non nulle, et il définit la clés primaire de la classe GENRE.

```
DROP TABLE SAISON CASCADE CONSTRAINT;
CREATE TABLE SAISON (
  NUM_SAISON         NUMBER(5)    CONSTRAINT PK_SAISON_NUM PRIMARY KEY
);
```

⇒ l'unique attribue numéro de saison doit être non nulle et unique, il définit la clés primaire de la classe SAISON.

```
DROP TABLE EST_GENRE CASCADE CONSTRAINT;
```

```
CREATE TABLE EST_GENRE (
  NOM_GENRE          VARCHAR(50),
  ANNEE              NUMBER(4) ,
  TITRE              VARCHAR(50) ,
  CONSTRAINT PK_EST_GENRE PRIMARY KEY(NOM_GENRE,ANNEE,TITRE),
  CONSTRAINT FK_EST_GENRE_SERIE FOREIGN KEY (ANNEE,TITRE) REFERENCES SERIE,
  CONSTRAINT FK_EST_GENRE_GENRE FOREIGN KEY (NOM_GENRE) REFERENCES GENRE
);
```

⇒ la clé primaire de cette table est la combinaison unique et non nulle (NOM_GENRE,ANNEE,TITRE).

Le couple (ANNEE,TITRE) est une référence ver la classe SERIE.

(NOM_GENRE) est une référence ver la classe GENRE.

```
DROP TABLE NOTER_EPISODE CASCADE CONSTRAINT;
CREATE TABLE NOTER_EPISODE
(
  DATE_NOTE_EPS      DATE,
  NOTE_EPS           NUMBER(4)      CONSTRAINT CK_NOTER_EPI_NOT CHECK
                                    ( NOTE_EPS BETWEEN 0 AND 10),
  COMMENTAIRE_EPS    VARCHAR(500),
  TITRE_EPS          VARCHAR(50)    CONSTRAINT NN_NOTER_EPISODE_TITRE
                                    NOT NULL,
  NOM_PER            VARCHAR(10)    CONSTRAINT NN_NOTER_EPISODE_NOM NOT
                                    NULL,
  PRENOM_PER         VARCHAR(20)    CONSTRAINT NN_NOTER_EPISODE_PRENOM NOT
                                    NULL,
  CONSTRAINT PK_NOTER_EPISODE PRIMARY KEY (TITRE_EPS,NOM_PER,PRENOM_PER),
  CONSTRAINT FK_NOTER_EPISODE_SERIE FOREIGN KEY (TITRE_EPS) REFERENCES
EPISODE(TITRE_EPS) ,
  CONSTRAINT FK_NOTER_EPISODE_PERSO FOREIGN KEY (NOM_PER,PRENOM_PER)
REFERENCES  PERSONNE(NOM_PER,PRENOM_PER)
);
```

⇒chaque épisode est noter sur 10, sa note doit être compris entre 0 et 10.

le nom, prénom de la personne qui a noter l'épisode ainsi que le titre de l'épisode noté doit être non nulle.

(TITRE_EPS,NOM_PER,PRENOM_PER) ce triplets est unique et non nulle, il est idéale pour être la clé primaire sa table.

L’attribut (TITRE_EPS) fait référence ver la table EPISODE,ils doivent être non nulle (si non on auras une référence indéfini).

L’attribut (NOM_PER,PRENOM_PER) fait référence ver la table PERSONNE,ils doivent être non nulle (si non on auras une référence indéfini).

```
DROP TABLE EST_ACTEUR CASCADE CONSTRAINT;

CREATE TABLE EST_ACTEUR

(

 TITRE_EPS          VARCHAR(50)        CONSTRAINT NN_ACTEUR_TITRE_EPS NOT NULL,

 NOM_PER            VARCHAR(10)        CONSTRAINT NN_ACTEUR_NOM NOT NULL,

 PRENOM_PER         VARCHAR(20)        CONSTRAINT NN_ACTEUR_PRENOM NOT NULL,

 CONSTRAINT PK_ACTEUR PRIMARY KEY (TITRE_EPS,NOM_PER,PRENOM_PER),

 CONSTRAINT FK_ACTEUR_EPISODE FOREIGN KEY (TITRE_EPS) REFERENCES EPISODE,

 CONSTRAINT FK_ACTEUR_PERSO FOREIGN KEY (NOM_PER,PRENOM_PER) REFERENCES
PERSONNE(NOM_PER,PRENOM_PER)

);
```

⇒ le nom, prénom de l’acteur ainsi que le titre de l’épisode qui a tourner doit être non nulle.

(TITRE_EPS,NOM_PER,PRENOM_PER) ce triplets est unique et non nulle, c’est la clé primaire de la table EST_ACTEUR.

L’attribut (TITRE_EPS) fait référence ver la table EPISODE,ils doivent être non nulle (si non on auras une référence indéfini).

L’attribut (NOM_PER,PRENOM_PER) fait référence ver la table PERSONNE,ils doivent être non nulle (si non on auras une référence indéfini).

```
DROP TABLE EST_REALISATEUR CASCADE CONSTRAINT;

CREATE TABLE EST_REALISATEUR

(

 TITRE_EPS    VARCHAR(50)          CONSTRAINT NN_REALISATEUR_TITRE_EPS NOT NULL,

 NOM_PER      VARCHAR(10)          CONSTRAINT NN_REALISATEUR_NOM NOT NULL,

 PRENOM_PER VARCHAR(20)        CONSTRAINT NN_REALISATEUR_PRENOM NOT NULL,

 CONSTRAINT PK_REALISATEUR PRIMARY KEY (TITRE_EPS,NOM_PER,PRENOM_PER),

 CONSTRAINT FK_REALISATEUR_EPISODE FOREIGN KEY (TITRE_EPS) REFERENCES
EPISODE,

 CONSTRAINT FK_REALISATEUR_PERSO FOREIGN KEY (NOM_PER,PRENOM_PER)
REFERENCES PERSONNE(NOM_PER,PRENOM_PER)

);
```

⇒ le nom, prénom du réalisateur ainsi que le titre de l'épisode qui a réaliser doit être non nulle.

(TITRE_EPS,NOM_PER,PRENOM_PER) ce triplets est unique et non nulle, c'est la clé primaire de la table EST_REALISATEUR.

L'attribut (TITRE_EPS) fait référence ver la table EPISODE,ils doivent être non nulle (si non on auras une référence indéfini).

L'attribut (NOM_PER,PRENOM_PER) fait référence ver la table PERSONNE,ils doivent être non nulle (si non on auras une référence indéfini).

```
DROP TABLE MESSAGES  CASCADE CONSTRAINT;
CREATE TABLE MESSAGES(
   ID_MSG                  NUMBER(5)     CONSTRAINT PK_MESSAGES PRIMARY KEY,
   ID_MSG_PARENT           NUMBER(5),
   ANNEE                   NUMBER(4)     CONSTRAINT NN_MESSAGES_ANNEE NOT NULL,
   TITRE                   VARCHAR(50)   CONSTRAINT NN_NMESSAGES_TITRE NOT NULL,
   NOM_PER                 VARCHAR(10)   CONSTRAINT NN_MESSAGES_NOM NOT NULL,
   PRENOM_PER              VARCHAR(20)   CONSTRAINT NN_MESSAGES_PRENOM NOT NULL,
   TEXTE                   VARCHAR2(1000),
   TITRE_MSG               VARCHAR2(100),
   DATE_MESSAGE_ENVOYE   DATE          CONSTRAINT NN_MESSAGES_DATE_MSG_ENVOY
                                       NOT NULL,
   CONSTRAINT FK_MESSAGES_MESSAGES FOREIGN KEY (ID_MSG_PARENT) REFERENCES
MESSAGES(ID_MSG),
   CONSTRAINT FK_MESSAGES_SERIE FOREIGN KEY (ANNEE,TITRE) REFERENCES
SERIE(ANNEE,TITRE) ,
   CONSTRAINT FK_MESSAGES_PERSO FOREIGN KEY (NOM_PER,PRENOM_PER)
REFERENCES  PERSONNE(NOM_PER,PRENOM_PER)
);
```

⇒l'attribue id_msg est un identifiant unique à chaque message, c'est la clé primaire de la table MESSAGES.

le nom, prénom de la personne qui a envoyer le message, la date d'envoi du message, ainsi que le titre de la série invoqué dans son message doit être non nulle.

L'attribut (NOM_PER,PRENOM_PER) fait référence ver la table PERSONNE,ils doivent être non nulle (si non on auras une référence indéfini).

Le couple (ANNEE,TITRE) est une référence ver la classe SERIE.

l'attribue (ID_MSG_PARENT) est une référence réflexive ver la table MESSAGES, il désigne l'attribue (ID_MSG).

Insertion des données :

Voir le fichier insertion.sql

Interrogation de la base de données :

Dans cette partie du rapport les requêtes seront en couleur bleu et les affichage en gris.

--1

```
SELECT TITRE FROM SERIE ;

TITRE

--------------------------------------------------------------------------------

Histoires fantastiques

Les Tiny Toons

Haute Tension

Malcolm

The Big Bang Theory

Breaking Bad

Rick et Morty

La casa de papel
```

--2

```
SELECT DISTINCT COUNT(COUNT(*)) AS NB_PAYS_DE_LA_BASE FROM SERIE GROUP BY
PAYS;

NB_PAYS_DE_LA_BASE

------------------

                   3
```

--3

```
SELECT TITRE FROM SERIE WHERE PAYS = 'Japon' ORDER BY TITRE ;

TITRE

--------------------------------------------------------------------------------

Les Tiny Toons

Rick et Morty
```

--4

```
SELECT PAYS, COUNT(*)  AS NB_SERIE_PAR_PAYS FROM SERIE GROUP BY PAYS;
```

PAYS NB_SERIE_PAR_PAYS

Etats-Unis 5

Japon 2

Espagne 1

--5

SELECT COUNT(*) NB_SERIE_ENTRE_2001_ET_2015 FROM SERIE WHERE ANNEE BETWEEN 2001 AND 2015;

NB_SERIE_ENTRE_2001_ET_2015

3

--6

(SELECT TITRE FROM EST_GENRE WHERE NOM_GENRE = 'Comedie') INTERSECT (SELECT TITRE FROM EST_GENRE WHERE NOM_GENRE='Science-fiction');

TITRE

Rick et Morty

--7

SELECT TITRE FROM EST_PRODUIT WHERE PRENOM_PER ='Spielberg' ORDER BY DATE_PRODUCTION DESC;

TITRE

Haute Tension

Les Tiny Toons

Histoires fantastiques

--8

SELECT DISTINCT TITRE, MAX(NUM_SAISON) AS NBR_SAISON FROM SERIE JOIN EPISODE USING(TITRE) WHERE PAYS='Etats-Unis' GROUP BY TITRE ORDER BY NBR_SAISON DESC;

TITRE	NBR_SAISON
Breaking Bad	3
Malcolm	2
The Big Bang Theory	1

--9

```
SELECT TITRE , COUNT(TITRE_EPS) AS MAX_NBR_EPISODES FROM EPISODE GROUP BY
TITRE HAVING COUNT(TITRE_EPS) >= ALL (SELECT COUNT(TITRE_EPS) FROM EPISODE
GROUP BY TITRE);
```

TITRE	MAX_NBR_EPISODES
Breaking Bad	33

--10

```
SELECT SEXE FROM NOTER_SERIE JOIN PERSONNE USING(PRENOM_PER) WHERE TITRE='The
Big Bang Theory' GROUP BY SEXE HAVING AVG(NOTE) >=
        ALL(SELECT AVG(NOTE) FROM NOTER_SERIE JOIN PERSONNE USING(PRENOM_PER)
WHERE TITRE='The Big Bang Theory' GROUP BY SEXE);
```

SEXE
F

--11

```
SELECT TITRE,AVG(NOTE) FROM NOTER_SERIE GROUP BY TITRE HAVING AVG(NOTE) < 5
ORDER BY 2;
```

TITRE	AVG(NOTE)
Rick et Morty	2,5
Histoires fantastiques	4,5

--12

```
SELECT TITRE,COMMENTAIRE FROM NOTER_SERIE A WHERE  A.NOTE IN (SELECT
MAX(NOTE) FROM NOTER_SERIE B WHERE A.TITRE = B.TITRE  GROUP BY TITRE);
```

TITRE	COMMENTAIRE
Histoires fantastiques	Nul
Les Tiny Toons	Nul
Haute Tension	y a pas grand chose dans cette série
Malcolm	y a pas mieux que cette série
La casa de papel	j en ai jamais regardé une séries comme celle là (:^:)

--13

```
SELECT TITRE,AVG(NOTE_EPS) FROM NOTER_EPISODE NATURAL JOIN EPISODE GROUP BY TITRE HAVING AVG(NOTE_EPS) > 8;
```

TITRE	AVG(NOTE_EPS)
Malcolm	8,6
The Big Bang Theory	8,64285714

--14

```
SELECT TITRE,COUNT(*)/(SELECT COUNT(*) FROM EPISODE B WHERE A.TITRE = B.TITRE) AS MOYENNE_JOUER FROM EST_ACTEUR NATURAL JOIN EPISODE A  WHERE NOM_PER='Bryan' AND PRENOM_PER='Cranston' GROUP BY TITRE;
```

TITRE	MOYENNE_JOUER
Malcolm	0 ,8
Breaking Bad	1

--15

```
SELECT NOM_PER ,PRENOM_PER FROM EST_REALISATEUR NATURAL JOIN EPISODE INTERSECT SELECT NOM_PER,PRENOM_PER FROM EST_ACTEUR NATURAL JOIN EPISODE  ;
```

NOM_PER	PRENOM_PER
Giancarlo	Esposito

--16

```
SELECT PRENOM_PER,NOM_PER,TITRE FROM EST_ACTEUR JOIN EPISODE A
USING(TITRE_EPS) GROUP BY PRENOM_PER,NOM_PER,TITRE HAVING COUNT(*) >=
ALL(SELECT 0.8*COUNT(*) FROM EPISODE WHERE TITRE = A.TITRE GROUP BY TITRE);
```

PRENOM_PER	NOM_PER	TITRE
Paul	Aaron	Breaking Bad
Cuoco	Kaley	The Big Bang Theory
Corberó	ÿrsula	La casa de papel
Esposito	Giancarlo	Breaking Bad
Parsons	Jim	The Big Bang Theory
Helberg	Simon	The Big Bang Theory
Lambert	Betsy	Breaking Bad
Flores	Alba	La casa de papel
Cranston	Bryan	Malcolm
Ituño	Itziar	La casa de papel
Norris	Dean	Breaking Bad
Galecki	Johnny	The Big Bang Theory
Nayyar	Kunal	The Big Bang Theory
Morte	Álvaro	La casa de papel
Cranston	Bryan	Breaking Bad
Gunn	Anna	Breaking Bad
Esposito	Giancarlo	The Big Bang Theory

-- 17

```
SELECT DISTINCT NOM_PER,PRENOM_PER FROM EST_ACTEUR A WHERE NOT EXISTS(
   SELECT * FROM EPISODE E WHERE E.TITRE ='Breaking Bad' AND NOT EXISTS (SELECT *
FROM EST_ACTEUR B WHERE E.TITRE_EPS = B.TITRE_EPS AND A.NOM_PER = B.NOM_PER
AND A.PRENOM_PER = B.PRENOM_PER)
);
```

NOM_PER	PRENOM_PER
Bryan	Cranston
Giancarlo	Esposito
Dean	Norris

Anna	Gunn
Aaron	Paul
Betsy	Lambert

--18

```
SELECT NOM_PER,PRENOM_PER,TITRE AS SERIE_NOTEE FROM NOTER_SERIE;
```

NOM_PER	PRENOM_PER	SERIE_NOTEE
Chalal	Abdellatif	Histoires fantastiques
Rahmani	Sylia	Histoires fantastiques
Chalal	Abdellatif	Les Tiny Toons
Rahmani	Sylia	Les Tiny Toons
Chalal	Abdellatif	Haute Tension
Rahmani	Sylia	Haute Tension
Chalal	Abdellatif	Malcolm
Rahmani	Sylia	Malcolm
Bouchefa	Mahmoud	The Big Bang Theory
Camelia	Bedji	The Big Bang Theory
Chalal	Abdellatif	The Big Bang Theory
Grace	Deny	The Big Bang Theory
Katy	Perry	The Big Bang Theory
Rahmani	Sylia	The Big Bang Theory
Zellag	Lyes	The Big Bang Theory
Bouchefa	Mahmoud	Breaking Bad
Chalal	Abdellatif	Breaking Bad
Rahmani	Sylia	Breaking Bad
Zellag	Lyes	Breaking Bad
Bouchefa	Mahmoud	Rick et Morty
Camelia	Bedji	Rick et Morty
Chalal	Abdellatif	Rick et Morty
Rahmani	Sylia	Rick et Morty
Chalal	Abdellatif	La casa de papel

--19

```
SELECT LEVEL,TITRE FROM MESSAGES START WITH ID_MSG_PARENT IS NULL CONNECT BY ID_MSG_PARENT = PRIOR ID_MSG;
```

LEVEL	TITRE
1	Breaking Bad
2	Breaking Bad
2	Breaking Bad
2	Breaking Bad
1	La casa de papel
2	La casa de papel
2	La casa de papel

-- 20

```
SELECT ID_MSG_PARENT AS ID_MSG_INITIAL,COUNT(ID_MSG)/(SELECT COUNT(ID_MSG) FROM MESSAGES WHERE ID_MSG_PARENT IS NOT NULL) AS MOYENNE_REPONSE

FROM MESSAGES WHERE ID_MSG_PARENT IN (SELECT ID_MSG FROM MESSAGES WHERE TITRE_MSG='Azrod95' AND ID_MSG_PARENT IS NULL) GROUP BY ID_MSG_PARENT;
```

ID_MSG_INITIAL	MOYENNE_REPONSE
1	0,6
2	0,4

conclusion

Ce projet nous a permis de consolider nos connaissances et mieux se familiariser avec le langage SQL. De part sa dimension pédagogique, il nous a donné beaucoup de liberté dans le codage et dans la conception.

d’un point de vue, gestion de projet, nous avons confirmé que la communication est importante lorsqu’on travaille en groupe.l’organisation des réunions d’échange et de mis au point sont importantes pour le cadrage de la méthodologie et l’avancement de la réflexion.

Il est également important de souligner la pertinence du temps accordé à la partie conception du projet (modèle entité/association et le modèle relationnel) pour assurer un déroulé fluide de programmation et de saisie du code.

d’un point de vue, réponse à notre cahier de charge, ce projet semble être une bon réussite. Tous les objectifs principaux sont attends.